Universität Bremen

Extraction of information from the dynamical activities of neural networks

David Rotermund

September 2007

Extraction of information from the dynamical activities of neural networks

Vom Fachbereich für Physik und Elektrotechnik

der Universität Bremen

zur Erlangung des akademischen Grades eines

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

von

Dipl. Phys. David Rotermund

aus Delmenhorst

1. Gutachter: Prof. Dr. rer. nat. Klaus Pawelzik

2. Gutachter: Prof. Dr. rer. nat. Andreas Kreiter

Eingereicht am: 11. September 2007

Datum des Kolloquiums: 29. November 2007

Abstract

Interacting with our dynamic environment requires to process huge amounts of sensory data in short time. This incoming stream of information is combined with internal states (e.g. memories or intentions) and results in actions. The fundamental mechanisms behind this fast information processing are still not understood. Even how information is stored in, and transmitted with sequences of action potentials is still under heavy debate. This thesis provides novel ideas to accomplish fast information processing, to understand adaptive coding strategies, and to perform unsupervised online learning of non-stationary representations.
In its first, genuinely theoretical part (chapter 3 - Information Processing Spike by Spike) this thesis develops a new concept in the field of fast information processing with single action potentials. The framework is based on stochastic generative models using Poissonian spike trains as input. It is capable of realizing arbitrary input-output functions, updating an internal representation with each incoming spike, for performing computations as fast as possible.
Leaving those purely theoretical considerations behind, the second part of this thesis (chapter 4 - Selective Visual Attention in V4/V1) investigates principles of adaptive neural coding in real data, focusing on the question how an internal cortical state, evoked by selective visual attention, modifies information processing in the brain. In collaboration with monkey neuro-physiologists we studied the influence of attention on the discriminability of visual stimuli through their neuronal correlates recorded as epidural field potentials.
The final part in this thesis (chapter 5 - Stabilizing Decoding Against Non-Stationaries) takes us towards a medical application for extracting internal brain states from neuronal activities. For controlling prosthetic devices with brain signals, reliable algorithms for estimating the intended actions of a person are required. A method was designed which allows to stabilise the estimator of a neuro-prosthesis against disruptions from non-stationarities in the characteristics of coding the intended actions, and from changes in their representations in the measured neuronal correlates.

Taken together, this thesis presented three new contributions:
A theoretical method of processing information spike by spike in a fast and efficient fashion. This study also showed that it is sufficient to use neurons, generating Poissonian spike trains, for performing fast and efficient information processing (Ernst et al., 2007b).

A new mechanism, produced through selective visual attention, was revealed that renders information about different visual stimuli, represented in γ-band oscillatory activity of neuronal populations, more distinct for an external observer and probably for the brain itself. It also showed that internal states of the brain can alter the neuronal activity pattern in a complex manner and it demonstrated that the power of the γ-band contains significant information about visually perceived shapes (Rotermund et al., 2007a).

A method for neuronal-prostheses capable of protecting estimators of intended actions (like arm movements) against non-stationaries, for the cost of an extra error signal describing the mismatch between the intended and executed action (Rotermund et al., 2006a).

Contents

Chapter 1

Introduction

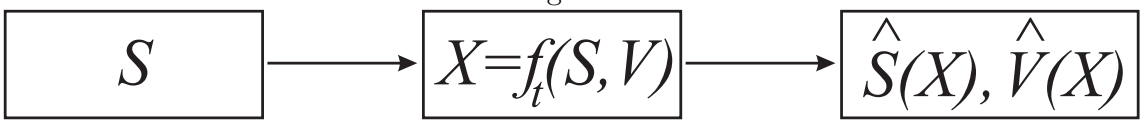
Standing in the kitchen while cutting vegetables, observing cooking pots, and telephoning in parallel is a normal scene in our daily lives. In this busy situation a glass filled with water is moved accidentally over the edge of the table and falls toward the floor. Before hitting the ground, the glass is caught by a fast arm movement. This everyday situation, in which even sophisticated robots would fail, demonstrates the enormous flexibility and reliability of the human brain in processing high amounts of information within very short periods of time.

Let us examine this example in more detail for understanding why there is actually no technical complement that could compete with the computational capabilities of the brain: A typical kitchen comprises plenty and partially occluded objects, but we are capable to interact with and within this environment under largely varying conditions. The lighting may be bright sunlight or dim candlelight, or we may observe the scene from different points of view, but still we are able to execute our task. The perception of our surrounding is filtered in such a way that we are able to focus only on tools and parts of the environment necessary for completing our task. Our nervous system integrates the incoming multiple streams of information into one coherent percept, like combining the visual image of our hand cutting the vegetables with the haptic feedback from the knife in our hand, in order to avoid cutting ourselves. In this situation, the glass sliding over the table's edge draws the attention to a totally different and subjectively more important problem, requiring our brain to focus its resources quickly onto the new situation. This takes place in less than a second in an extremely efficient manner. How the brain performs the necessary invariant object recognition, integrates the perceived multimodal information, and is able to switch between different tasks is largely not understood, nor do we know how to implement all its capabilities into a machine analogon.

For understanding these functions of the brain one has to investigate the fundamental principles of information processing based on large networks of nerve cells with vast interconnections, subjected to the typical physiological constraints revealed by empir-

ical studies.
A prerequisite to identify these principles is to understand the 'language' used by nerve cells for exchanging information. Since the importance of electric current for the central nervous system (CNS) was discovered by Galvani 200 years ago, an unbelievable amount of detailed data was collected, but our understanding of how objects, internal states, or properties of sensory input are represented in the patterns of neuronal activities is still far from complete. For inferring coding principles from measured data, information extraction methods are necessary. Thus, besides a well-educated guess about the typical features of the data which carry the maximum amount of information, the development of algorithms and tools for extracting this information from observed neuronal activities has become an important part of brain research. These methods can in addition be used to quantify how information coding or the neuronal representation changes in dependence on the actual situation, task, or during learning. This can hereby further advance our understanding of brain functions.

In my thesis I will analyse different aspects of information coding and information processing in the brain. These investigations cover the range from theoretical studies up to the development of algorithms for real applications. Before focusing on the different parts of my thesis, I will explain the general framework for these studies which can be condensed into the following scheme:



Information S enters the brain, e.g. through sensors on the retina for visual stimuli or through hair cells for auditory stimuli. The incoming information is then transcoded into a neuronal response X via the encoding function $f_t(.)$.

The function $f_t(.)$ is not necessarily stationary over time. For several reasons, like e.g. adaptation or learning processes in the brain, quite substantial changes in $f_t(.)$ can occur. But this is not the only factor which could change the mapping of S onto the neuronal response X: Internal brain states, which are denoted by V in this framework, can substantially alter the coding. This includes conditions of attention, memories, or other non-observable (hidden) variables of the system.

As the last component in this framework, $\hat{S}(X)$ interprets or performs a computation on the neuronal responses X. This computation can be implemented in the CNS itself or being performed by an external observer. For example, $\hat{S}(X)$ could be a higher brain area inferring the presence of a contour from activities X in primary visual cortex or an estimator attempting to reconstruct the visual information S, respectively. In higher brain areas, X often barely depends on S: such a typical situation arises in the context of 'reading the mind' using brain prosthesis, where the intention V (which is a hidden variable to us) of a handicapped person $\hat{V}(X)$ needs to be estimated.

Of course, noise may compromise each one of these processing and transmission steps, essentially making inference a hard task for both the brain and the researcher.

An astonishing feature of the mammalian brain is the rapidness of handling large amounts of information. Experiments showed that a complex task like detecting whether an animal is present or absent in a picture can be executed within 150 ms (Thorpe et al., 1996) and that a skilled ball game player (e.g. table tennis or cricket) needs only a few hundreds of milliseconds for extracting and processing all the necessary information from a perceived ball's trajectory (Land and McLeod, 2000). These experiments and other experiences from daily life, like catching a falling glass, suggest that information has to be transmitted and processed rapidly by our brain. It has been suggested (Thorpe et al., 2001) that rapid information processing can be explained by rank-order coding, but it is largely unclear how this idea might work under realistic assumptions on neuronal noise and in situations where stimuli are not flashed, but changing in a continuous fashion. In my thesis, I will investigate an alternative coding scheme where rapid processing and transmission of information can be implemented by biological means, assuming the information is stored only in the (relative) number of action-potentials received as sensory evidence.

The aspect of fast information transmission corresponds in the presented framework to finding the optimal combination of $\mathsf{f}(\mathsf{S})$ and $\hat{\mathsf{S}}(\mathsf{X})$, under selected biological boundary conditions defined through bounds on S and X. In a precursory work focusing on information transmission only, it was possible to show that optimal coding strategies of storing information into firing rates of neurons are very sensitive to the present noise level. Analytical considerations revealed phase transitions between families of optimal coding strategies for different noise levels (Bethge et al., 2003a; Bethge et al., 2003b; Bethge et al., 2002a). However, fast information processing is a second, even more important aspect of the whole problem, which is at the core of our investigations in this thesis: Assuming that the brain builds a probabilistic representation from its input, we design a novel information processing algorithm based on the (relative) number of single action-potentials received from the input neurons. The algorithm allows to extract common structures, like sets of basis functions, from presented input distributions. Also it is possible to use it for pattern recognition or to perform calculations, like e.g. Boolean functions. We will see that the resulting information processing method is very fast and extracts efficiently, in comparison with other benchmark methods, information from the incoming action-potentials.

A key challenge in brain research is to understand how information about objects is encoded into neuronal activities. Over the years, empirical studies collected valuable information and established several concepts in understanding the neuronal representation in the early visual system (Carandini et al., 2005). However, the major part of this work concentrates only on single, localized aspects of objects, like the orientation of an edge within a small region in visual space. The neuronal representation of whole objects or spatially extended stimuli is largely unknown, partly because reliable multi-electrode recordings became available only during the past 10 years, and partly because encoding quickly becomes non-linear when stimuli extend beyond the classical receptive fields. Although very specialised neuronal population were found that encode complex object like familiar persons (Quiroga et al., 2005), it seems that the generic

case is that objects are represented distributed over large neuronal populations or even over multiple brain areas. This fact provokes the question how different attributes of one single objects are 'glued' together into a coherent percept (often loosely termed as the 'binding problem'). It has been suggested that this act of composition can be provided by shared oscillatory behaviour of neuronal populations (Fries et al., 2007). Furthermore, it seems that the representation of objects can be enhanced or suppressed by adapting the coding to actual demands. Visual selective attention, which for example allows us to select behaviourally relevant parts of a visual scenery, falls under this category. In this thesis it is studied how different visual objects are represented in collective oscillatory neuronal activity patterns and how selective visual attention can alter these patterns. In a study applying data analysis methods to electro-physiological data from animal experiments, we will evaluate different schemes of coding object information, and compare them to principles discussed in the literature.

Analysing neuronal activity patterns to uncover principles of information coding used by the nervous system are not solely performed for academical reasons. One concrete application is found current medical research: the development of brain prostheses. The idea of such a neuro-prosthetic device is to help handicapped or paralysed people to regain autonomy. This could be accomplished by inferring an intended action, like an arm movement, from the patient's brain activity, which is subsequently used by a mechanical device for executing this action. Many researchers are working in this field of functional neuro-prosthetics, and their results (Wolpaw et al., 2002; Wolpaw et al., 2000; Kuebler et al., 2001; Curran and Stokes, 2003; Hochberg et al., 2006) suggest that such a medical device may be available within the next years. In a brain prosthesis, hidden states V, representing intended actions like arm movements, are extracted from measured neuronal activities X by an estimator $\hat{\mathsf{V}}(\mathsf{X})$. While the performance of the developed estimators attained over the years a reasonable level, harvesting the necessary amount of data from the CNS over a long period of time with high spatial and temporal resolution still remains a challenge (Pawelzik et al., 2006b).

The used estimators $\hat{\mathsf{V}}(\mathsf{X})$ depend heavily on correct knowledge about the coding scheme $\mathsf{f}_t(\mathsf{V})$ for reconstruction meaningful information from the observed neuronal responses. Wrong information about the coding scheme leads to bad reconstruction performances. Learning $\mathsf{f}_{t=0}(\mathsf{V})$ at one point in time may not be sufficient, because it may be subject to ongoing changes due to non-stationaries inside the brain or at the interface between brain and machine. For long-term medical applications it will be important to find strategies of counteracting these non-stationaries, allowing to stabilise the control of neuro-prostheses. In this thesis a new on-line adaptation strategy is presented which is able to counteract these perturbations. The novel idea behind this stabilisation is to measure an additional neuronal signal related to the actual perceived error between the user's intention, and the action executed by the prosthetic device. Applying this adaptation strategy has the potential to increase the stability of such prosthetic devices over time, at the cost of requiring to record a second signal source.

This thesis will be split into different parts, addressing the three main topics outlined

above. In summary, these parts will focus on

- Chapter 3 (Information Processing Spike by Spike):
  A theoretical study is presented where information about underlying signals S is reconstructed as fast as possible from neuronal responses X, where X is caused by the signals S and includes noise generated by a Poissonian process / multinomial process. The model is based on the assumption that the brain uses a probabilistic representation of a combination of causes which generated the received input. For this reconstruction procedure the estimator $\hat{\mathsf{S}}(\mathsf{X})$ has to learn $\mathsf{f}(\mathsf{S}, \mathsf{V})$. A suitable algorithm will be developed from first principles, which updates the probabilistic representation with each single spike received. Properties of this algorithm will be analysed, with focus on how fast and precise this reconstruction or information processing based on this input can be done.

- Chapter 4 (Selective Visual Attention in V4/V1):
  In this part, a thorough data analysis is presented, with the ultimate goal to learn more about how objects are represented in oscillatory signals from neuronal populations. This includes to investigate how selective visual attention enhances the information content of the signal. The signals S are in this case visual stimuli (the outline of different shapes) presented to monkeys in an experiment. The neuronal responses X are recorded from the visual cortex of the animals, while they are required to perceive S. X is generated under different states of hidden variables V that can be identified with the two different conditions of attention. The stimuli S will be reconstructed from X. Furthermore, it will be analysed how the state of the hidden variable V changes $\mathsf{f}(\mathsf{S}, \mathsf{V})$.

- Chapter 5 (Stabilizing Decoding Against Non-stationaries):
  The final part of my thesis will develop an approach towards a system for protecting data extraction algorithms of functional neuro-prosthetic devices against non-stationaries. The signals V are intended actions for controlling functional neuro-prostheses. These intended actions (e.g. arm movements) are decoded from the neuronal responses X. For decoding V, $\mathsf{f}_t(\mathsf{S}, \mathsf{V})$ has to be learned by the estimator. Due to changes of $\mathsf{f}_t(\mathsf{S}, \mathsf{V})$ over time, the knowledge of the estimator about $\mathsf{f}_t(\mathsf{S}, \mathsf{V})$ has to be adapted for following these changes. A strategy for updating this information, based on additional neuronal responses describing the actual performance of the prosthetic device, is developed.

Chapter 2

Theoretical and Biological Background

2.1 Encoding information into sequences of action potentials

The brain is one of the most complex systems in nature. It was estimated that the human brain comprises about 10^{12} cells and that each of these cells typically receives signals from hundreds up to thousands of other nerve cells, amounting to approximately 10^{14} to 10^{15} connections (through synapses) in total (Hubel, 1989). Along these connections, information is mainly transmitted by action potentials. An action potential is generated by an exchange of ions (e.g. potassium and sodium ions) between the inside and outside of the nerve cell that travels along the membrane. Since different action potentials (also called 'spikes') emitted from one neuron are nearly indistinguishable it is believed that their shape carries no information.

There exist several hypotheses how information can be carried by a sequence of spikes ('spike train'). For a short introduction of these hypotheses the detailed shape of the action potentials will be ignored and the spikes will be substituted by Dirac δ functions. Now we can describe a sequence of spikes by

$$s(t) = \sum_i \delta(t - t_i),$$

where i denotes the i-th spike at time t_i.

In 1926 Adrian (Maass and Bishop, 2000) showed that the activity of muscles and the mean number of spikes in a given time interval are correlated. Since that time, the possibility of encoding information in the mean activities of neurons, called 'rate' coding (Dayan and Abbott, 2001) was investigated intensively. For obtaining the rate

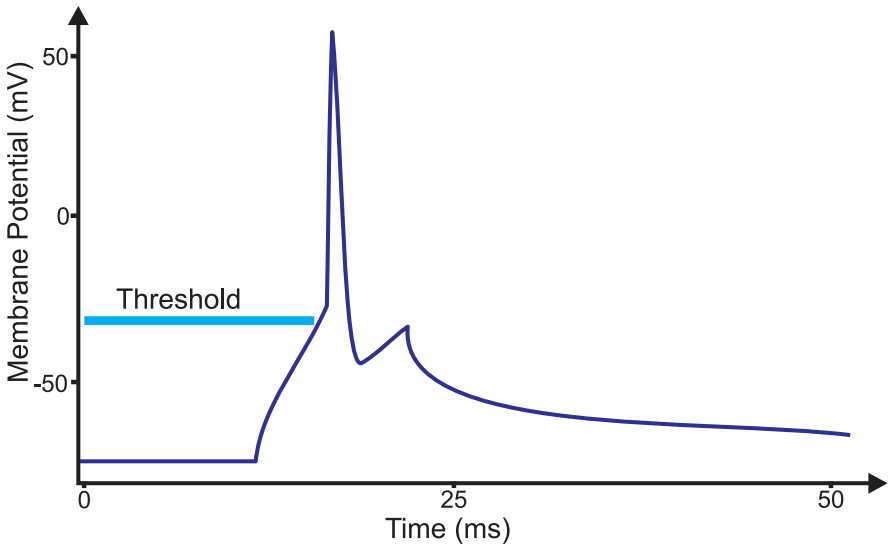


Figure 2.1: Schematic view of an action potential. After the membrane potential of the nerve cell was risen above a threshold, an action potential is released. Different types of ions start to flow from the inside of the cell through membrane channels to the outside of the cell, and visa versa. Through the differences in the dynamics of membrane channels for different types of ions, the exchange of ions creates a characteristic rise and fall of the electric potential at the membrane. This excitation travels along the membrane into the axon and dendritic compartment. Ion pumps in the membrane restore the concentration differences of ions between the inside and the outside of the cell like it was before the initialisation of the action potential. (The figure was adapted from wikipedia.org)

r from a given spike train $s(t)$ one computes:

$$r = \frac{n}{T} = \frac{1}{T}\int_{t_0}^{t_1} s(\tau)\,d\tau.$$

n represents the number of spikes in a time interval with length T, starting at t_0 and ending at t_1.

Experimental investigations of spike trains from cortical neurons reveal a high variability of neuronal responses even when repeatedly using the same stimuli (Tomko and Crapper, 1974; Tolhurst et al., 1983; Snowden et al., 1992; Burns and Webb, 1976; Britten et al., 1993). This observation suggests to describe neuronal responses as a stochastic process. Experimentally it has been found (Shadlen and Newsome, 1998) that the inter-spike-interval distribution often approximately follows an exponential probability distribution, if a neuron fires with a constant rate over a period of time. Such distributions may arise from Poissonian point processes, whose spike count distributions are fully determined by the trial averaged rate $< r >$ and a time window

T

$$p(k \mid r \cdot T) = \frac{1}{k!} (< r > \cdot T)^k e^{-<r> \cdot T}. \tag{2.1}$$

Drawing samples from this probability distribution will yield spike counts k with a mean value of $< r > \cdot T$ and a variance of $< r > \cdot T$.

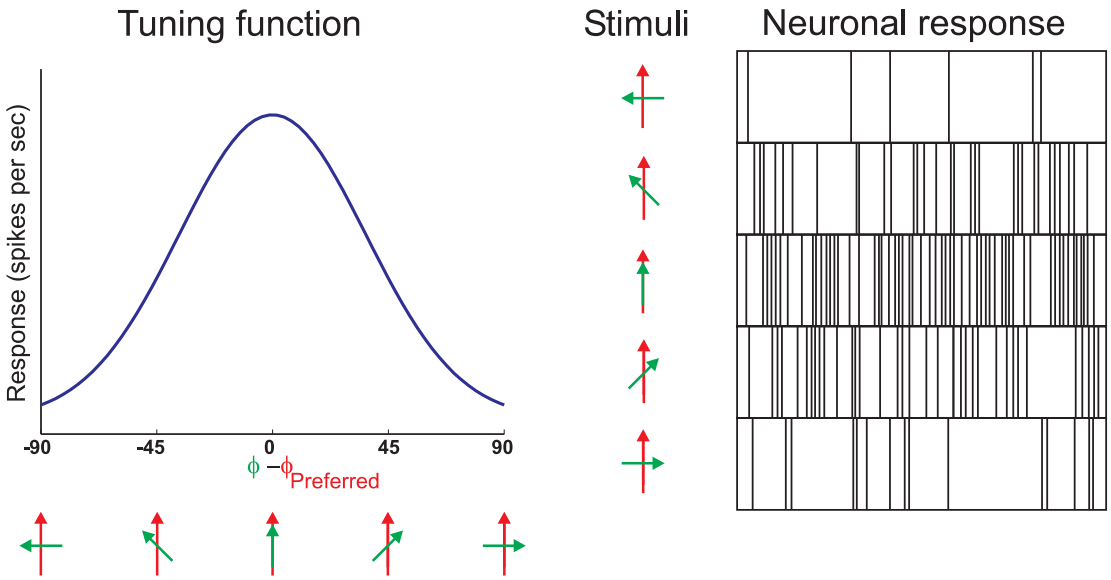


Figure 2.2: Sketch of a typical tuning curve for a visual cortical region. A full field grating with orientation ϕ is presented on a screen and the resulting neuronal spike activity is measured. On the left hand side, the tuning function for the differences between the orientation of the bar and the preferred orientation (orientation with the maximal neuronal response) of the neuron is displayed. On the right hand side, exemplary spike trains evoked by the presented stimuli are shown.

Typically, the activity of a neuron depends on its input, which may vary depending on the stimulus being presented. We can describe this dependency by allowing the averaged firing rate $< r >$ to be a function of the stimulus x. The averaged rate $< r(x) >$ is called neuronal response tuning curve (or short 'tuning function'). for an example see Fig. 2.2. The tuning curve in Fig. 2.2 is a function depending solely on the one-dimensional variable $\phi - \phi_{Preferred}$. In the brain, neuronal responses may depend on many features of a stimulus. Often tuning functions capture only single aspects of these selectivities. Since neurons are vastly connected to other neurons, it is possible that a sound detected by hailcells in the ear can later influence the activity of neurons in the visual cortex. Thus tuning curves calculated from measured data are only a projection of the complete tuning function to the tested stimulus space. To make the situation even more complicated: Stimulus $x(t)$ may change over time, responses may depend on the history of activities (e.g. fatigue and adaptation effects), and responses may be correlated to the activity of neighbouring neurons (e.g. synchronisation effects).

Nevertheless, the concept of tuning functions is often used and in many cases very helpfull.

Another important concept for sensory neurons are 'receptive fields'. The receptive field (RF) of a neuron describes the regions in stimulus space where the neuron reacts to a stimuli (e.g. regions on the retina). For neurons processing visual sensory information, this would e.g. be the spatial position of a stimulus in the visual field.

The firing rate captures only the information of how many spikes occur inside a given interval of time. Information regarding the timing of the action potentials is ignored, but it is known that fluctuating input or stimuli with a rich temporal structure can generate spike trains with a precision of milliseconds (Mainen and Sejnowski, 1995; Buracas et al., 1998; Azouz and Gray, 2000). Motivated by this evidence alternative coding hypotheses were proposed, e.g. 'timing code' and 'rank order code'.

The idea of a time code is to store information in the precise timing of spikes relative to a reference event ('latency'). For example, one could imagine that the measured time in milliseconds between the occurrence of a spike and a reference event represents a numerical value. The amount of information transported by one spike is then determined by the precision of the latency (see Fig. 2.3 as an example). Theoretically, if a period of time could be measured with infinite precision and a neuron could produce spikes with infinite precision, one spike would carry an infinite amount of information.

This coding scheme has the drawback that the decoding mechanism requires the precise latency of the spike, implying that the decoding neuron needs access to the incoming spike as well as to reference events (Thorpe et al., 2001). Nevertheless, experimental evidence for a time code was found. Experiments on human volunteers showed that response latencies are used as a code in somatosensory tasks (Johansson and Birznieks, 2004). A paradigm was used where the subject had to estimate the distance between two mechanical stimulations on the skin. In the visual system of blowflies a latency coding for motion-sensitive neurons was found (Warzecha and Egelhaaf, 2000), where the latency decreases with increasing contrast and temporal frequency of a moving pattern. Additionally latency coding was found in rat barrel cortex (Petersen et al., 2002).

So far, we have discussed coding schemes only for single neurons. Using more than one neuron for coding allows for the use of 'population codes', which allow to store additional information by using combinatorics. The 'rank order code' (Thorpe et al., 2001; van Rullen and Thorpe, 2002) is one of these combinatorial coding strategies that encodes information into the order of incoming action potentials (see Fig. 2.3). For decoding, the sequence in which the neurons were active is taken and has to be compared to a dictionary. This dictionary allows to decode rank order sequences by selecting the corresponding values for the observed sequence of activities. The bandwidth of this coding strategy is mainly defined by the necessary time interval between to spikes for preventing unwanted permutations due to noise. A neurophysiologically plausible implementation of rank order coding (van Rullen and Thorpe, 2002) seems to

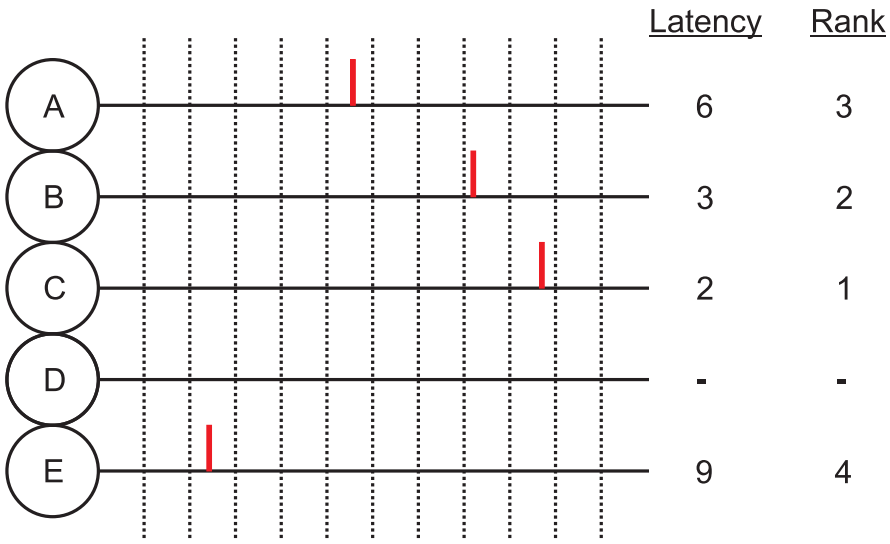


Figure 2.3: Example of encoding information into the rank order and the timing of action potentials. The dotted vertical lines represent the units of measuring latencies due to precision. (The figure was adapted from (Thorpe et al., 2001))

be more simple than a realisation of a timing coding with biological plausible means. Rank order coding requires a reference event, otherwise the decoding neuron wouldn't be able to determine when a new sequence of spikes starts. For the visual system, it was suggested that saccadic (van Rullen and Thorpe, 2002) or micro-saccadic (Martinez-Conde et al., 2000) eye movements can be used as such reference events for rank order and time coding.

From the observation that (human) brains can react very fast to visual stimuli (e.g. detecting whether an animal is shown on a screen requires less than 150 ms (Thorpe et al., 1996)), it was concluded (Thorpe et al., 2001) that rate coding is too slow for such operations because of the time required for counting spikes. Thus, it has been suggested that something like a rank order code should be used. In chapter 3 of this thesis it will be demonstated that even with information processing based on rate coding, complex calculations can be performed with few single spikes.

Other hypotheses were proposed (e.g. using complex spike patterns (Warland et al., 1997) and codes using synchrony), which we will not discuss here.

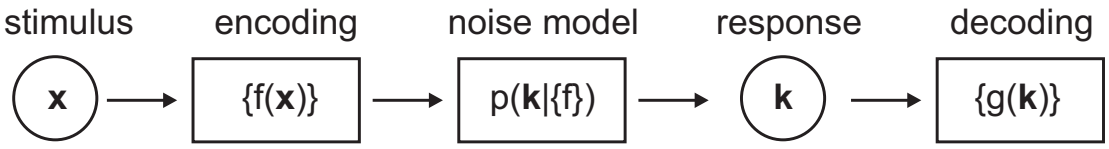


Figure 2.4: Schematic diagram of encoding and decoding information with action potentials. The stimulus $\mathbf{x}$ is translated via a set of tuning functions $f(\mathbf{x})$ and noise model $p(\mathbf{k} \mid f)$ (e.g. Poisson process) into a temporal and spatial pattern of action potentials $\mathbf{k}$.

2.2 Reconstructing information from sequences of action potentials

The last section was concerned about hypotheses how information can be stored in spike trains. Fig. 2.4 shows a schematic illustration of a stimulus $\mathbf{x}$ that is coded in a neuronal response $\mathbf{k}$. This response $\mathbf{k}$ contains more or less information about the stimulus $\mathbf{x}$. It is interesting for several purposes (e.g. functions in neuro-prosthetics) to extract the stored information from $\mathbf{k}$, but it is important to understand that the neuronal response may not allow to reconstruct $\mathbf{x}$ without any inaccuracies. One reason can be that the amount of transmitted information is bounded by constraints, like e.g. limited number of neurons, limitations on the available time for information transmission, limitations on available energy, and neuronal activity of neurons. Taken together the capacity of the 'channel' (Shannon, 1948), the information has to pass, may be limited.

Source After transmission Reconstruction

Figure 2.5: Example of transmitting information through a limited and unreliable channel. The pixel values $I(x, y) \in [0, \ldots, 1]$ of the original image (left) were transmitted through channel with Poissonian noise (with a mean value of $I(x, y) \cdot 5$). The output of the channel is shown in the middle image. Using a 'Minimum Mean Squared Error-Estimator' (see section 2.2.3 ; with flat prior) creates the right image as reconstruction from the noisy data. (The original picture was adapted from wikipedia.org)

Additional problems can arise from the fact that the channel may transmit the information unreliable. An example is shown in Fig. 2.5. Loosely speaking, the limited

capacity of the channel can be understood as a reduction on the number of transmitted symbols (e.g. sending a message by using only the numbers 1,2,3 and 4) per time unit. The unreliability of the channel may generate misinterpretations resulting from the received symbols. It is possible to reduce the unreliability by introducing redundancy or error correction schemes into the used code. For extracting information from neuron-to-neuron communication it is necessary to take unreliabilities (e.g. by synapses) into account. Many decoding strategies use methods from stochastics for dealing with these problems. Before explaining these methods in more detail, a short introduction into the nomenclature of stochastics will be given.

2.2.1 Probabilities

Expectation value and variance

Assuming that a buttered toast may have the probability ρ of falling on the side with butter and the probability $1-\rho$ of falling on the other side, we can use the binomial distribution (MacKay, 2003)

$$p(r \mid \rho, N) = \begin{pmatrix} N \\ r \end{pmatrix} \rho^r (1-\rho)^{N-r} \tag{2.2}$$

for calculating the probability p that the buttered side will hit the ground r times out of N times tossing the toast. $\begin{pmatrix} N \\ r \end{pmatrix} = \frac{N!}{(N-r)!r!}$ are called binomial coefficients.

The mean value $E[r]$ and the variance $Var[r]$ for the binomial distribution are defined by

$$E[r] = \sum_{r=0}^{N} p(r \mid \rho, N) r = N \cdot \rho$$

and

$$\begin{aligned} Var[r] &= E\left[(r - E[r])^2\right] \\ &= E\left[r^2\right] - (E[r])^2 = \sum_{r=0}^{N} p(r \mid \rho, N) r^2 - (E[r])^2 \\ &= N\rho(1-\rho)\,. \end{aligned}$$

More general, the expectation value $E[f(\mathbf{r})]$ and variance $Var[f(\mathbf{r})]$ of a function $f(\mathbf{r})$ with a probability function $p(\mathbf{r})$ with $\mathbf{r}$ as a multi-dimensional continuous variable are

defined by

$$\begin{aligned} E[f(\mathbf{r})] &= \int f(\mathbf{r})p(\mathbf{r})d\mathbf{r} && (2.3)\\ Var[f(\mathbf{r})] &= \int (f(\mathbf{r}) - E[f(\mathbf{r})])^2 p(\mathbf{r})d\mathbf{r} && (2.4)\\ 1 &= \int p(\mathbf{r})d\mathbf{r} && (2.5) \end{aligned}$$

or with $\mathbf{r}$ as a multi-dimensional discrete variable by

$$\begin{aligned} E[f(\mathbf{r})] &= \sum_{\mathbf{r}} f(\mathbf{r})p(\mathbf{r}) && (2.6)\\ Var[f(\mathbf{r})] &= \sum_{\mathbf{r}} (f(\mathbf{r}) - E[f(\mathbf{r})])^2 p(\mathbf{r}) && (2.7)\\ 1 &= \sum_{\mathbf{r}} p(\mathbf{r})\,. && (2.8) \end{aligned}$$

Eq.(2.5) and Eq.(2.8) represent the normalisation equations, which all probability distributions must fullfill.

Furthermore, if two random variables r_1 and r_2 are statistically independent then the following relation can be used:

$$\begin{aligned} E[r_1 + r_2] &= E[r_1] + E[r_2]\\ Var[r_1 + r_2] &= Var[r_1] + Var[r_2] \end{aligned}$$

Bayes Theorem

A probability distribution for an ensemble of random variables r and s, is termed joint probability distribution $p(r, s)$. A joint probability can be reduced ('marginalised') to a marginal probability distribution (MacKay, 2003), by averaging over one or more variables e.g.

$$p(r) = \sum_{s} p(r, s)\,.$$

Also it is possible to use the 'product rule' (or also called 'chain rule') for expressing a joint probability with a conditional probability distribution

$$p(r \mid s)p(s) = p(r, s)\,.$$

As a combination of these procedures we gain the following rules :

$$\begin{aligned} p(r, s) &= p(r \mid s)p(s) = p(s \mid r)p(r)\\ p(r) &= \sum_{s} p(r, s) = \sum_{s} p(r \mid s)p(s) \end{aligned}$$

Another derivation of the product rule is the important Bayes' theorem:

$$p(r \mid s) = \frac{p(s \mid r)p(r)}{p(s)} \tag{2.9}$$

Bayesian inference

Using the Bayesian theorem, it is possible to calculate the conditional probability of unobserved variables (e.g. parameters of the distribution λ) given the observed variables (e.g. measured data D). This computation is called Bayesian inference.

$$p(\lambda \mid D) = \frac{p(D \mid \lambda) \cdot p(\lambda)}{p(D)}$$

The specific probabilities in the Bayes theorem are often termed likelihood of the parameters for $p(D \mid \lambda)$, prior $p(\lambda)$, evidence (or marginal likelihood) for $p(D)$ and posterior $p(\lambda \mid D)$ (MacKay, 2003). We can assign interpretations to these probabilities, which will show more clearly what they represent:

- Posterior $p(\lambda \mid D)$
  The posterior is a conditional probability and allows to judge how probable a hypotheses (e.g. a set of parameters) are given a set of (observed) data.

- Prior $p(\lambda)$
  The prior allows to introduce knowledge or hypotheses about parameters. A simple and frequently used choice is the 'uniform prior'. It allows to constrain a parameter on an interval (e.g. $[\alpha, \beta]$):

  $$p(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{for } \alpha < x < \beta \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

  For most problems, the prior will depend on assumptions and thus it will always be subjective.

- Likelihood $p(D \mid \lambda)$
  The conditional probabilities $p(D \mid \lambda)$ characterise how likely it is that this set of data will be generated by a random process with these parameters. This conditional probability can also be interpreted as the likelihood $\mathcal{L}(\lambda \mid D) = p(D \mid \lambda)$. $\mathcal{L}$ is a function of the parameters for a given set of data. It is not normalised over parameter space.

- Marginal likelihood $p(D)$
  The marginal likelihood describes the probability that this set of data D will be generated by a given random process. It can be calculated e.g. from the likelihood and prior by

  $$p(D) = \sum_{\lambda} p(D \mid \lambda)p(\lambda)\,.$$

2.2.2 Information measures and loss functions

After the overview regarding probabilities and using them for calculations, it is also important how to measure distances or similarities between probability distributions or to quantify the richness of the structure of a distributions.

A loss function (or utility function) can be demonstrated by the following example (Bernardo, 1979): Let us assume that an experiment is performed in which a random variable x is observed. Through this experiment we want to learn more about the function $\Psi = \Psi(\theta)$. After the experiment, we can express all the knowledge gathered about Ψ by the experiment in $p(\Psi \mid x)$. As our hypothesis about the distribution of Ψ (the 'unknown' true value) we use $p^\dagger(\Psi)$. If $p^\dagger(\Psi)$ represents Ψ best, then the expectation value of the utility will be at its optimum. This expectation value of the utility can be calculated by

$$D(p^\dagger(\Psi) \parallel \Psi) = \int d(p^\dagger(\Psi) \parallel \Psi) p(\Psi \mid x) d\Psi \,,$$

where d is the utility function. It was shown (Bernardo, 1979) that the log loss

$$d(p^\dagger(\Psi) \parallel \Psi) = A \log\left(p^\dagger(\Psi)\right) + B(\Psi) \,,$$

where A is a constant and $B(\Psi)$ an arbitrary function of Ψ, has good properties as utility function (it is the only smooth, proper and local loss function).

Information-theoretic entropy

The information-theoretic entropy $H(X)$ is an information measure, based on the log - loss. It is often called 'Shannon-Entropy' as reference to Shannon's article from 1948 (Shannon, 1948) but it is said (Smith, 2001) that the entropy was already applied by Pauli in 1933 (Pauli, 1933) to problems of statistical mechanics. The information-theoretic entropy is defined by

$$H(X) = -\sum_x p(x) \log p(x) \,. \tag{2.11}$$

For understanding properties of the entropy, let us assume an example where a random process generates two observable classes, e.g. drawing oranges and apples from a bag. If the bag is filled with the same number of elements for both classes then it is maximally uncertain which class will be drawn next. For this situation the entropy is at its maximum. Increasing the number of elements for one class and/or decreasing the number of elements for the other class will reduce the uncertainty regarding which class will be drawn next. In this case the value for the entropy will decrease. In the extreme, where the probability for one class goes to 1 and for the other class goes to 0, there is no uncertainty left and the entropy approaches 0.

Kullback - Leibler divergence

Derivated from Eq.(2.11) for the entropy, the cross entropy $H(X_P, X_Q)$ for two probability distributions $p(x)$ and $q(x)$ can be defined by

$$H(X_P, X_Q) = -\sum_x p(x) \log q(x) .$$

The cross entropy is not a symmetric measure with respect to both distributions, which means that exchanging the two probability distributions will result in a different value of H. The cross entropy is part of the frequently used Kullback - Leibler divergence (introduced by Solomon Kullback and Richard Leibler in 1951 (Kullback and Leibler, 1951)), which shares the same asymmetry:

$$\begin{aligned} D_{KL}(X_P \parallel X_Q) &= \sum_x p(x) \log \frac{p(x)}{q(x)} \\ &= \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x) \\ &= -H(X) + H(X_P, X_Q) \end{aligned} \tag{2.12}$$

In the case where $p(x) = q(x)$ (for all x), the sum $-\sum_x p(x) \log q(x)$ matches the entropy and thus the value for the Kullback - Leibler divergence will be 0. For all other combinations of $p(x)$ and $q(x)$, the Kullback - Leibler divergence will be positive. We can interpret the Kullback - Leibler divergence as the necessary amount of extra information when a codebook based on $p(x)$ is rewritten by using $q(x)$ instead.

Mutual information

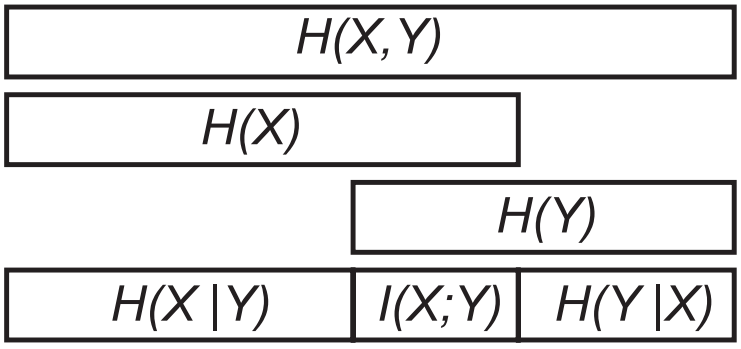


Figure 2.6: The relationship between joint entropy $H(X, Y)$, (marginal) entropies $H(X)$ and $H(Y)$, conditional entropies $H(X \mid Y)$ and $H(Y \mid X)$ and mutual information $I(X; Y)$. (Figure taken from (MacKay, 2003).)

Given a random process that generates two random variables (X and Y), it may be interesting to determine how much information one variable carries about the other

variable. Or in other words, if we have a bag full of tomatoes and apples (which both can have the colour green or red) we can ask how much information about the colour we obtain if we draw a tomato out of the bag. Obviously, the answer depends on the probabilities $p(\text{green} \mid \text{apple})$ and $p(\text{green} \mid \text{tomato})$.
The entropy for both random variables together is called joint entropy. In Fig. 2.6 it is shown how the joint entropy

$$\begin{aligned} H(X,Y) &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\,\log\left(p(x,y)\right) \\ &= H(X \mid Y) + H(Y) = H(Y \mid X) + H(X) \\ &= H(X) + H(Y) - I(X;Y) \end{aligned}$$

can be broken down into the (marginal) entropy Eq.(2.11) , the conditional entropy

$$H(X \mid Y) = -\sum_{x}\sum_{y} p(x \mid y)\,\log p(x \mid y)\,,$$

and the mutual information or 'transinformation'

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\,\log\frac{p(x,y)}{p(x)\cdot p(y)}\,. \tag{2.13}$$

In our example, the mutual information Eq.(2.13) measures the common information about X (tomato or apple) that is shared by Y (green or red). In the case, if and only if X and Y are independent random variables, which means that both random variables are not sharing information, then $p(x,y) = p(x)\cdot p(y)$ and the transinformation becomes zero. In contrast to the Kullback - Leibler divergence, the mutual information is symmetric in X and Y. Leaving the tomatoes and apples behind, the mutual information can also be used as tool for analysing neuronal codes, e.g. (Eckhorn and Poeppel, 1974). For example, with this measure we can evaluate how much information a neuronal response shares with a stimulus and compare different hypotheses of how stimuli may be coded by neuronal responses.

Mean squared error

A descriptive measure is the mean squared error ('MSE'). It quantifies the distance between two functions by the expectation value calculated over the squared differences of their components

$$\begin{aligned} \mathrm{MSE}(x) &= E\left[(f(x) - g(y))^2 \mid x\right] \\ &= \sum_{y} p(y \mid x)\,(f(x) - g(y))^2\,. \end{aligned} \tag{2.14}$$

Let us assume, that a response k was generated by a noise model $p(k \mid x)$ based on the quantity x, which is not directly observable. For reconstructing x out of k, an

estimator $\hat{x}$ is used. These assumptions allow us to decompose the MSE into 'bias' and variance (see section 2.2.1) by the so called 'bias - variance - decomposition':

$$\begin{aligned}
E\left[(x-\hat{x}(k))^2 \mid x\right] &= \sum_k p(k \mid x)\,(x-\hat{x}(k))^2 \\
&= \sum_k p(k \mid x)\,(x - E[\hat{x}(k) \mid x] + E[\hat{x}(k) \mid x] - \hat{x}(k))^2 \\
&= \sum_k p(k \mid x)\left((x - E[\hat{x}(k) \mid x])^2 + (E[\hat{x}(k) \mid x] - \hat{x}(k))^2\right) \\
&\quad +2\,(x - E[\hat{x}(k) \mid x]) \cdot \underbrace{\left(E[\hat{x}(k) \mid x] - \sum_k p(k \mid x)\hat{x}(k)\right)}_{E[\hat{x}(k)|x]-E[\hat{x}(k)|x]=0} \\
&= \underbrace{(x - E[\hat{x}(k) \mid x])^2}_{\mathrm{Bias}_{[\hat{x}(k)|x]^2}} + \underbrace{\sum_k p(k \mid x)\,(E[\hat{x}(k) \mid x] - \hat{x}(k))^2}_{\mathrm{Var}_{[\hat{x}(k)|x]}} \\
&= \mathrm{Bias}[\hat{x}(k) \mid x]^2 + \mathrm{Var}[\hat{x}(k) \mid x] \qquad (2.15)
\end{aligned}$$

The bias quantifies how far the expectation value of the estimator differs from the unobservable quantity x. Eq.(2.15) shows that the mean squared error is composed by variance and bias. Furthermore, it shows that an unbiased estimator (bias equals zero) can still have a mean squared error larger than zero. In general, it can be a problem to reduce one of these quantities independently from each other (the so called bias - variance tradeoff, e.g. (Geman et al., 1991)).

Fisher Information and Cramer - Rao bound

Using the Fisher information $\mathcal{I}(x)$ (Fisher, 1922) for e.g. exploring theoretically the properties of estimators and tuning functions is popular in the field of population coding (Pouget et al., 2003; Wilke and Eurich, 2002).

$$\mathcal{I}(x) = E\left[\left(\frac{\partial}{\partial x}\log\left(p(k \mid x)\right)\right)^2 \middle| x\right] \qquad (2.16)$$

The reason for this popularity is the link between the Fisher information and the Cramer - Rao bound (Cramer, 1946; Rao, 1946). The Cramer - Rao bound provides an asymptotic lower limit for any estimator $\hat{x}(k)$ as a function of two quantities: the bias of the estimator and the Fisher information $\mathcal{I}(x)$ (Pouget et al., 2003).

$$\mathrm{Var}[\hat{x}(k) \mid x] \geq \frac{\left(\frac{\partial}{\partial x}E[\hat{x}(k) \mid x]\right)^2}{\mathcal{I}(x)} = \frac{\left(1 + \frac{\partial}{\partial x}\mathrm{Bias}[\hat{x}(k) \mid x]\right)^2}{\mathcal{I}(x)}. \qquad (2.17)$$

It is not guaranteed that an estimator exists which can reach this bound. For population coding, the Fisher information (for a multi-dimension version of the Fisher

information see (Blahut, 1987; Lehmann and Casella, 1999)) can be used as a measure that represents the expectation how strongly two recorded activities for two slightly different stimuli will differ (Pouget et al., 2003) or in other words: The Fisher information quantifies the expected curvature of the likelihood. For any unbiased estimator, the Cramer-Rao bound is solely a function of the Fisher information

$$\mathrm{Var}\left[\hat{x}(k) \mid x\right] \geq \frac{1}{J(x)}\,. \tag{2.18}$$

It is known that the MS estimator (Lehmann and Casella, 1999) (section 2.2.3) and maximum likelihood estimator (Seung and Sompolinsky, 1993) (section 2.2.3) are asymptotically efficient. Efficient means that an unbiased estimator saturates the Cramer - Rao bound. Thus, for an increasing number of observations the MSE asymptotically approaches $\frac{1}{J(x)}$ for all x. It also should be noted that there exist pitfalls in the use of the Fisher Information. For example, we demonstrated that the use of Fisher information for characterizing the precision of a code for a given decoding time T strongly depends on the particular coding scheme. We have shown that the optimal width of a population of Gaussian tuning curves depends on the available decoding time, while Fisher-optimal codes are always independent of T (Bethge et al., 2002c).

2.2.3 Propability based estimators

Minimum mean squared error estimator

A function that maps the neuronal response back to the underlying stimuli using given knowledge (see Fig. 2.4) is called estimator. The minimum mean squared error (MMSE) estimator is based on the posterior risk and thus on the Bayes theorem. Sometimes it is also called Bayes estimator, but in fact it is just one member of the whole family of Bayes estimators, which are defined by

$$\hat{x}_{\mathrm{Bayes}}(\mathbf{k}) \;=\; \underset{\hat{x}(\mathbf{k})}{\mathrm{argmin}}\; r_{\mathrm{PL}}\left(x, \hat{x}(\mathbf{k}) \mid \mathbf{k}\right) \tag{2.19}$$

with

$$r_{\mathrm{PL}}\left(x, \hat{x}(\mathbf{k}) \mid \mathbf{k}\right) \;=\; \int r_{\mathrm{L}}\left(x, \hat{x}(\mathbf{k}) \mid \mathbf{x}\right) p(\mathbf{x} \mid \mathbf{k})\, d\mathbf{x}\,. \tag{2.20}$$

As a simple consequence, the Bayes estimator also minimizes the averaged risk (also called Bayes risk) (Bethge, 2003). In the general case of Bayes estimators, the loss function r_L needs not to be the MSE but, as the name suggests, for the MMSE the loss function is defined by the mean squared error and so Eq.(2.19) and Eq.(2.20) are given by

$$\hat{x}_{\mathrm{Bayes}}(\mathbf{k}) \;=\; \underset{\hat{x}(\mathbf{k})}{\mathrm{argmin}}\; E_{\mathrm{Posterior}}\left[(x - \hat{x}(\mathbf{k}))^2 \mid \mathbf{k}\right]\,. \tag{2.21}$$

with

$$E_{\text{Posterior}}\left[(x-\hat{x}(\mathbf{k}))^2 \mid \mathbf{k}\right] = \int E\left[(x-\hat{x}(\mathbf{k}))^2 \mid x\right] p(x \mid \mathbf{k})\, dx\,. \tag{2.22}$$

The Bayes risk is defined for the MSE through

$$\begin{aligned}\chi^2 &= E\left[(x-\hat{x}(\mathbf{k}))^2\right] \\ &= E\left[E\left[(x-\hat{x}(\mathbf{k}))^2 \mid x\right]\right] = E\left[E\left[(x-\hat{x}(\mathbf{k}))^2 \mid \mathbf{k}\right]\right]\,.\end{aligned} \tag{2.23}$$

The best Bayes estimator under this loss function can be formulated by

$$\hat{x}(\mathbf{k})_{MSE} = E[x \mid \mathbf{k}] = \int x p(x \mid \mathbf{k}) dx \tag{2.24}$$

which allows to rewrite χ^2 as

$$\chi^2 = E\left[(x-\hat{x}_{MSE}(\mathbf{k}))^2\right] \tag{2.25}$$

$$= E\left[x^2\right] - E\left[\hat{x}_{MSE}(\mathbf{k})^2\right]\,. \tag{2.26}$$

It is often difficult to compute $\hat{x}(\mathbf{k})_{MSE}$ but sometimes it is possible to find solutions in closed form. In the following, I will present an example where this is possible and the result will be needed later in section 5. The MMSE estimator for the function

$$f_{lin}(x) = N \cdot ((f_{max} - f_{min})\, x + f_{min}) \tag{2.27}$$

will be calculated. The tuning function Eq.(2.27) describes the mean firing rate of a population, composed of N neurons, which are coding the continuous value $x \in [0,1]$ with the same linear mapping for all neurons. The dynamic range of each single neuron lies between f_{min} and f_{max}. The channel that will be used in section 5 for transmitting this information is subject to Poisson noise (see Eq.(2.1)) and delivers as neuronal response the spike count K (containing the number of action potentials from all neurons of the population) for a time window T.

The simplicity of this tuning function in combination with the Poissonian noise model allows to compute the optimal Bayes estimator using the expression

$$\begin{aligned}\hat{f}(K) &= \frac{\int_0^1 p(K \mid T f_{lin}(x))\, x\, dx}{\int_0^1 p(K \mid T f_{lin}(x))\, dx} \\ &= \frac{\int_0^1 x \cdot ((f_{max}-f_{min})\, x + f_{min})^K\, e^{-N((f_{max}-f_{min})x+f_{min})T} dx}{\int_0^1 ((f_{max}-f_{min})\, x + f_{min})^K\, e^{-N((f_{max}-f_{min})x+f_{min})T} dx}\,.\end{aligned} \tag{2.28}$$

Evaluation of the integrals yields the final expression

$$\hat{f}(K) = \frac{\Gamma(2+K, F_{min}, F_{max})}{(F_{max}-F_{min})\ \Gamma(1+K, F_{min}, F_{max})} - \frac{F_{min}}{(F_{max}-F_{min})} \tag{2.29}$$

with $F_{min} = NTf_{min}$, $F_{max} = NTf_{max}$ and $\Gamma(k, a, b)$ denoting the incomplete Gamma function

$$\Gamma(k, a, b) = \Gamma_{a,b}(k) = \int_a^b x^{k-1} e^{-x} dx \,. \tag{2.30}$$

It should be noted that this estimator was derived from a uniform prior distribution of values x in the interval $[0, 1]$ Eq.(2.10). Any deviation of the real $p(x)$ from this assumption leads to a bias in estimation.

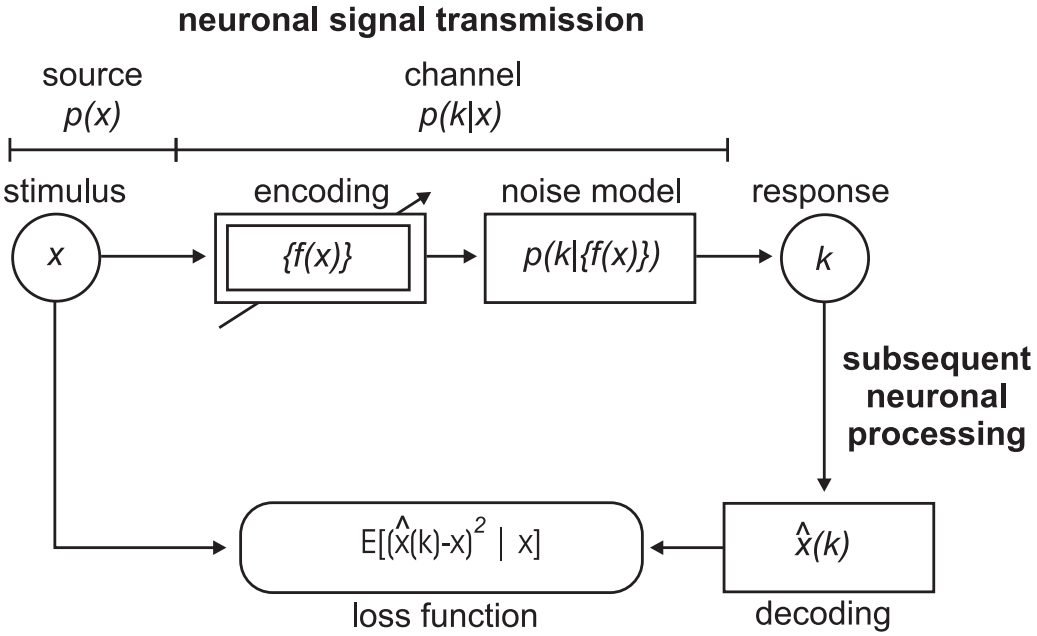


Figure 2.7: General population coding scheme. The statistical properties of the source for the stimulus x are described by $p(x)$. The stimulus will be encoded by a set of tuning functions $\{f(x)\}$ into a mean firing rate and then coded by a noise model $p(x \mid \{f(x)\})$ into a neuronal response $\mathbf{k}$. This neuronal response is decoded by the estimator $\hat{x}(\mathbf{k})$ to reconstruct of the stimulus x. The estimate $\hat{x}$ and the real stimulus x are compared with respect to a given loss function, here the mean squared error.(The figure was adapted from (Bethge, 2003))

For theoretical analyses of coding strategies, the MMSE estimator and the MSE can be used e.g. to find optimal tuning functions. Schematically such an optimisation task is shown in Fig. 2.7. For a given set of noise model, stimulus statistics, type of estimator, and loss function, the task is to find the optimal tuning function which minimises the loss function under the given constrains. An example is shown in section A.2.1.

Linear minimum mean squared error estimator

The MMSE estimator has no functional limitations for mapping neuronal responses onto estimates but there are reasons (like e.g. biological constraints, limited compu-

tational power and the possibility to obtain analytical solutions in closed form) which make it necessary to restrict the estimator to the class of linear functions (Salinas and Abbott, 1994). This optimal linear Bayes estimator (OLE) is defined by the equation

$$\hat{\mathbf{x}}(\mathbf{k}) = \sum_{i=1}^{N} k_i \mathbf{A}_i + \mathbf{B}\,,$$

with the parameter vectors $\{\boldsymbol{A}_i\}_{i=1,\ldots,N}$ and $\mathbf{B}$ and $\mathbf{k} = \{k_1, \ldots, k_N\}$ denoting the signal (e.g. the spike count vector of the neurons).

The optimal choice of parameters for an OLE depends on the error function, which in this case is defined by

$$\chi^2 \;=\; \sum_{\mathbf{k}} \int_{\mathbf{x}} \rho(\mathbf{x})\, p(\mathbf{k} \mid \mathbf{x}, \mathbf{P}) \left(\mathbf{x} - \left(\sum_{i=1}^{N} k_i \mathbf{A}_i + \mathbf{B} \right) \right)^2 , \tag{2.31}$$

with $\mathbf{P}$ being the set of parameters describing the properties of the system.

The optimal $\mathbf{A}_i$ and $\mathbf{B}$ can be computed from χ^2 using the calculus of variations as exemplified in (Salinas and Abbott, 1994), which yields the following function based on Eq.(2.31)

$$\hat{\mathbf{x}}(\mathbf{k}) = \sum_{j=1}^{N} (k_j - M_j)\, \mathbf{D}_j + \mathbf{Z}\,, \tag{2.32}$$

using the following abbreviations

$$\begin{aligned} \mathbf{D}_j = \sum_i (\mathbf{L}_i - M_i \mathbf{Z}) \left[\mathbf{R}^{-1}\right]_{i,j}\,, \qquad & \mathbf{R} = \{Q_{i,j} - M_i M_j\}_{i,j=1,\ldots,N}\,, \\ M_i = \int_{\mathbf{x}} \rho(\mathbf{x}) g_i(\mathbf{x})\,, \qquad & \mathbf{L}_i = \int_{\mathbf{x}} \rho(\mathbf{x}) g_i(\mathbf{x}) \mathbf{x}\,, \\ \mathbf{Z} = \int_{\mathbf{x}} \rho(\mathbf{x}) \mathbf{x}\,, \qquad & Q_{i,j} = \sum_{\mathbf{k}} \int_{\mathbf{x}} \rho(\mathbf{x}) p(\mathbf{k} \mid \mathbf{x}, \mathbf{P})) k_i k_j\,, \\ \text{and} \qquad & g_i(\mathbf{x}) = \sum_{\mathbf{k}} p(\mathbf{k} \mid \mathbf{x}, \mathbf{P}) k_i\,. \end{aligned} \tag{2.33}$$

In section 5 we will make extensive use of the OLE.

Maximum likelihood estimator

Another method of constructing estimators is the maximum likelihood method (MLE). The idea behind this type of estimator was developed by Fisher (Fisher, 1922) in the years between 1912 and 1922 (Aldrich, 1997). The MLE is commonly used and a

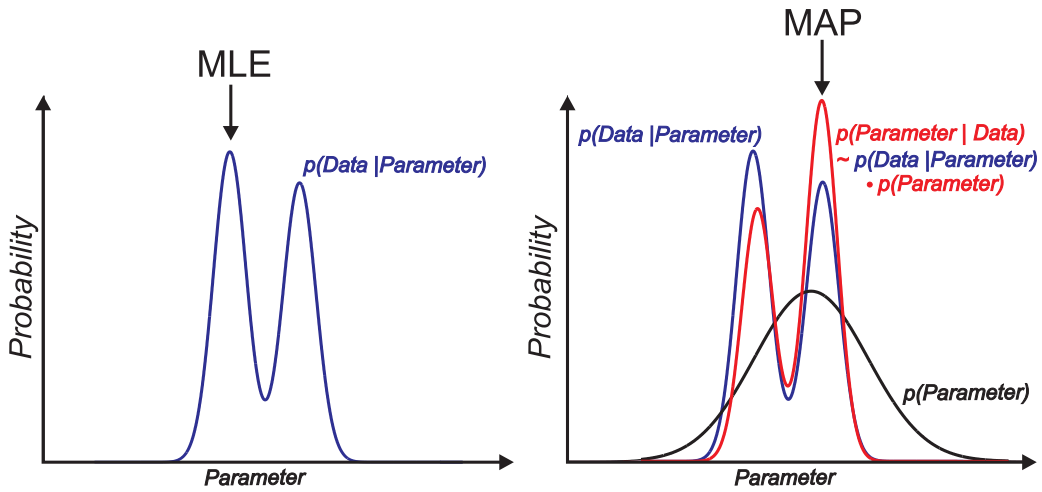


Figure 2.8: Comparison between the maximum likelihood estimator and the maximum a posteriori estimator. The figure on the left hand side shows the likelihood $\mathcal{L}(\text{Parameter} \mid \text{Data}) = \mathrm{p}(\text{Data} \mid \text{Parameter})$. The MLE picks the maximum of the likelihood function with respect to the parameter. The other figure shows in comparison the posterior probability distribution (shown in red colour). The additional prior information about the parameter (shown in black) was combined with the likelihood (shown in red colour), using the Bayes theorem, to obtain the posterior probability distribution. The parameter now selected by the MAP differs from that selected by the MLE.

lot of studies in the field of neuroscience are based on this method, e.g. (Seung and Sompolinsky, 1993; Deneve et al., 1999; Schulzke, 2006).

As explained in the previous part about Bayes inference, the likelihood is defined as a function of parameters and a fixed set of data

$$\begin{aligned} \mathcal{L}(\text{Parameters} \mid \text{Data}) &= \mathrm{p}(\text{Data} \mid \text{Parameters}) \\ \mathcal{L}(\mathbf{x} \mid \mathbf{k}) &= \mathrm{p}(\mathbf{k} \mid \mathbf{x})\,. \end{aligned}$$

The maximum likelihood estimator simply chooses the $\mathbf{x}$ which maximises the likelihood function,

$$\hat{\mathbf{x}}_{\mathrm{MLE}}(\mathbf{k}) = \underset{\hat{\mathbf{x}}(\mathbf{k})}{\operatorname{argmax}}\, \mathcal{L}(\hat{\mathbf{x}}(\mathbf{k}) \mid \mathbf{k})\,. \tag{2.34}$$

For an example see Fig. 2.8 (left).

Maximum a posteriori estimator

The maximum a posteriori estimator (MAP; see Fig. 2.8) is closely related to the MLE. Instead of maximising the likelihood function $\mathcal{L}(\mathbf{x} \mid \mathbf{k})$, the MAP maximizes on

the posterior distribution

$$\hat{x}_{MAP}(k) = \underset{\hat{x}(k)}{\operatorname{argmax}}\, p_{posterior}(\hat{x}(k) \mid k) \tag{2.35}$$

with

$$p_{posterior}(x \mid k) = \frac{p(k \mid x)p_{prior}(x)}{\sum_x p(k \mid x)p_{prior}(x)}\,. \tag{2.36}$$

Since the denominator does not depend on the parameter, we can also write

$$\hat{x}_{MAP}(k) = \underset{\hat{x}(k)}{\operatorname{argmax}}\, \{p(k \mid \hat{x}(k))p_{prior}(\hat{x}(k))\}\,. \tag{2.37}$$

2.2.4 Discrimination and classification

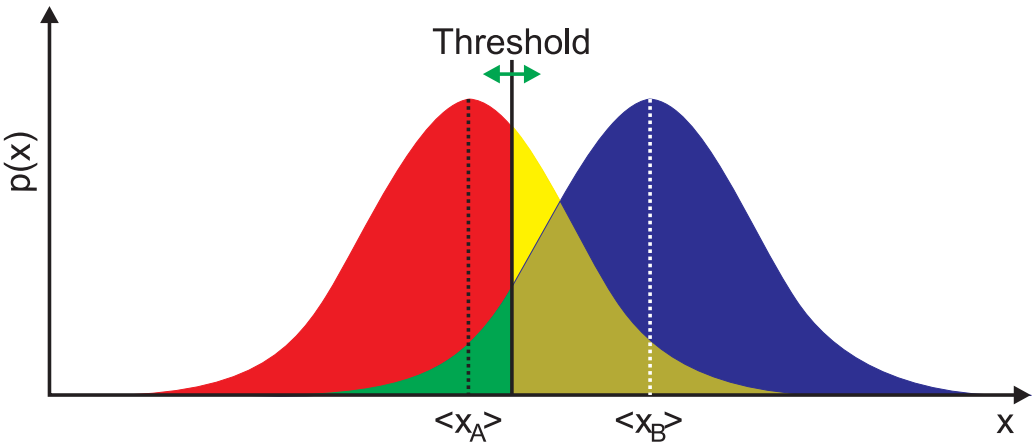


Figure 2.9: Chart illustrating discriminability. The task is to discriminate whether a value x was drawn from the Gaussian probability function with mean value $< x_A >$ or from the other Gaussian curve with mean value $< x_B >$. If a threshold is introduced as discrimination criteria, we can label the different regions: The red region is called 'true negatives' and the blue region is named 'true positives'. The green area is termed 'false negatives' and the last region (yellow) is denoted as 'false positives'. (The figure was adapted from wikipedia.org)

Let us imagine an experiment, where we can observe a neuronal response x (e.g. the firing rate measured in a fixed time window). We know that x represents two different classes but the regarded neuronal responses are drawn from two overlapping probability distributions (see Fig. 2.9). Given a sample x, we want to discriminate whether x was generated by class A or class B. For Gaussian functions with the same variance σ, the discriminability d'

$$d' = \frac{< x_A > - < x_B >}{\sigma} \tag{2.38}$$

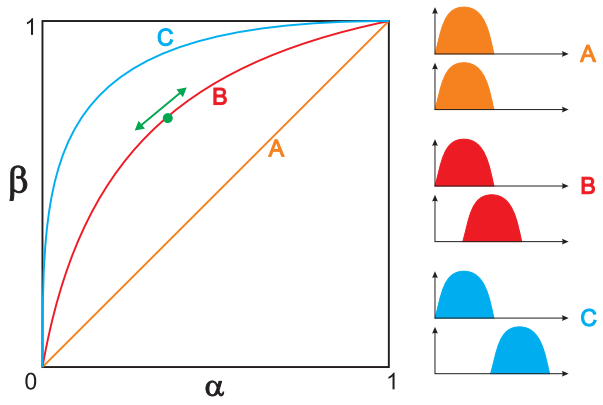


Figure 2.10: Examples for ROC curves. Three examples are shown where the performance of how well samples from two distributions can be distinguished, are different. The corresponding set of distributions are shown on the right hand side. Both distributions for case A are identical and thus not distinguishable. The result is a diagonal in the ROC diagram. The discrimination performance is better for case B and even better for the set of distributions marked with C. A change in threshold in Fig. 2.9 results in a corresponding movement of the green dot on the ROC curve. (The figure was adapted from wikipedia.org)

allows to measure the separability. The difficulty to distinguish between class A and B and thus the discriminability, is related to the degree to which both distributions overlap. The larger the value of d' is, the more separable the distributions are (Dayan and Abbott, 2001).

One way of performing the discrimination is to introduce a threshold ϑ. If $x \geq \vartheta$, we decide that the response was generated by class B and if $x < \vartheta$ then we decide that x was generated by class A. As a consequence of this threshold, we will guess correctly with a probability of $p(x \geq \vartheta \mid \text{Class B})$ when $x \geq \vartheta$ was generated by class B, and we will make a wrong decision with a probability of $p(x \geq \vartheta \mid \text{Class A})$ when the response was created by class A. These probabilities can be computed using the normalisation properties of probabilities

$$\begin{aligned} p(x < \vartheta \mid \text{Class A}) &= 1 - p(x \geq \vartheta \mid \text{Class A}) \\ p(x < \vartheta \mid \text{Class B}) &= 1 - p(x \geq \vartheta \mid \text{Class B})\,. \end{aligned}$$

In signal detection theory x is called test, $\alpha(\vartheta) = p(x \geq \vartheta \mid \text{Class A})$ denoted as the size or false alarm rate of the test and $\beta(\vartheta) = p(x \geq \vartheta \mid \text{Class B})$ is the power or hit rate of the test (Dayan and Abbott, 2001). In the context of discriminability with binary decisions, class B is often symbolised by '+' and class A by '-'. Following this terminology, in Fig. 2.9 different regions are label and marked by colours. The region 'true positives' represent cases where the neuronal responses were generated by the '+' class and were discriminated correctly, the 'true negatives' correspond to the '-' class that have been classified correctly, and 'false positives' as well as 'false negatives' are representing the corresponding errors.

From Fig. 2.9 we realise that the performance of the discrimination is directly related to the threshold ϑ. Selecting the threshold such that $\alpha = 0$ and $\beta = 1$ is in general not possible, so an optimal solution would be to find a ϑ which maximizes β.

ROC

The receiver operating characteristics (ROC) allows to measure the discrimination performance depending on the threshold ϑ. Each point on the ROC curve, with α on the abscissa and β on the ordinate, corresponds to a different threshold ϑ (see Fig. 2.10 and Fig. 2.9). The area under the ROC curve is related to the performance of how well samples from two distributions can be distinguished in a discrimination task (Mason and Graham, 2002). For a two-alternative forced choice task, the area under the ROC curve corresponds directly to the performance.

$$p\,(\text{Correct}) \;=\; \int_0^1 \beta(\alpha)\,d\alpha\,. \tag{2.39}$$

For Gaussian functions, where both curves share a common variance σ^2, the ROC curve depends only on d'

$$p\,(\text{Correct}) = \frac{1}{2}\text{erfc}\left(\frac{-d'}{2}\right)$$

with the complementary error function

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{(-y^2)}\,dy\,.$$

It is common to calculate d' even if the underlying probability distributions are other than Gaussian (Dayan and Abbott, 2001).

Nearest Neighbour Method

A relatively simple, but in many cases successful method of reconstructing 'hidden' information from data is the 'nearest neighbour' method. Given a labeled set (with M members) of N-dimensional data vectors $\mathbf{X}_i = \{X_{i,1}, \ldots X_{i,N}\}$ with a corresponding set of labels L_i for setting the free parameters of the nearest neighbour ('training' the nearest neighbour by memorizing the whole training data $\mathbf{X}_i$ in a 'dictionary') and a test data vector $\mathbf{Y} = \{Y_1, \ldots Y_N\}$ with unknown label U, the distance can be calculated between the test vector and the training data

$$D(\mathbf{X}_i \| \mathbf{Y}) = \sum_{j=1}^{N} \mid X_{i,j} - Y_j \mid^{\alpha}\ .$$

Typically $\alpha = 2$, for an Euclidean distance, or $\alpha = 1$ are used. In a next step the data vector from the training data set with the smallest distance to the test vector is evaluated

$$\eta = \underset{i}{\operatorname{argmin}} \{D(\mathbf{X}_i \| \mathbf{Y})\}.$$

As a result we get $U = L_\eta$ as estimate for the label of the test vector.

This method can be extended by taking more than the nearest neighbour into account. A selected set of labels from the proximity of the smallest distance have then to be reduced to one value by e.g. a weighted sum.

SVM

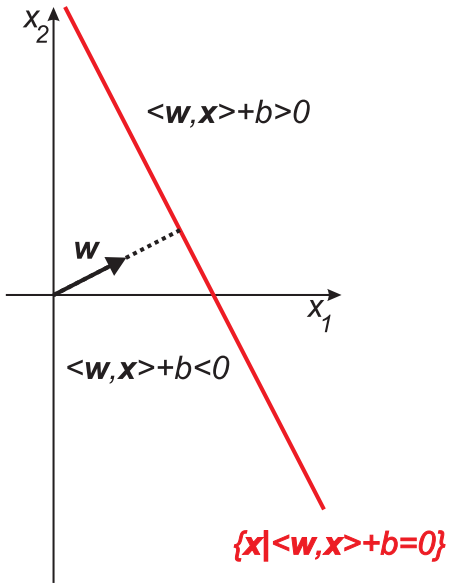


Figure 2.11: Example of a hyperplane. The plane is given by the $\mathbf{x}$ values that fullfill the equation $< \boldsymbol{w}, \mathbf{x} > + b = 0$. The vector $\boldsymbol{w}$ is orthogonal to the hyperplane. All points below the plane are defined by $< \boldsymbol{w}, \mathbf{x} > + b < 0$, and the points above are determined by $< \boldsymbol{w}, \mathbf{x} > + b > 0$.(The figure was adapted from (Schölkopf and Smola, 2001))

In the beginning of section 2.2.4, we introduced the method discriminating two distributions by a threshold. A related strategy is the very popular support vector machine (SVM) method from the field of machine learning (I will follow the text of (Schölkopf and Smola, 2001) for the explanation of SVM's). SVM's are based on the idea of using hyperplanes

$$\{\mathbf{x} | < \boldsymbol{w}, \mathbf{x} > + b = 0\}$$

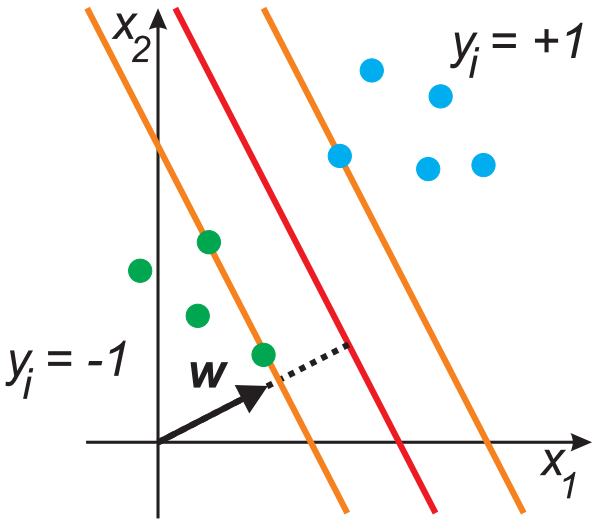


Figure 2.12: A hyperplane separates two classes of data points. The distance between the canonical hyperplane and the nearest point is given by $\frac{1}{\|w\|}$. The margin planes (orange lines) are defined by $\{\mathbf{x} \mid < \boldsymbol{w}, \mathbf{x} > +b = -1\}$ for the margin below the separating hyperplane and $\{\mathbf{x} \mid < \boldsymbol{w}, \mathbf{x} > +b = +1\}$ the upper one. (The figure was adapted from (Schölkopf and Smola, 2001))

for separating multi-dimensional data (see Fig. 2.11). $\boldsymbol{w}$ is a vector orthogonal to the hyperplane and $< \boldsymbol{w}, \mathbf{x} >$ is the dot product (in Euclidean space).

Let us assume that we have a data set for training $\{\mathbf{x}_i, y_i\}_{i=1,\ldots,m}$, where y_i are labels that assign all vectors $\mathbf{x}_i$ to one of two classes $y_i \in \{+1, -1\}$. If it is possible to isolate all data points of one class from all data points of the other class by a hyperplane, then we call the data separable. In this case, the distances d_i between the hyperplane

$$d_i = \left(\left\langle \frac{\boldsymbol{w}}{\| \boldsymbol{w} \|}, \mathbf{x}_i \right\rangle + \frac{b}{\| \boldsymbol{w} \|} \right)$$

multiplied with the label y_i, $z_i = y_i \cdot d_i$ are all positive for the whole data. A positive z_i shows that the regarding data point lies on the correct side of the hyperplane (see Fig. 2.12). If the data points are not separable, then some z_i will be smaller than 0, which signalises that these data points lie beyond the hyperplane.

$$f_{\boldsymbol{w},b}(\mathbf{x}_i) = \operatorname{sgn}\left(< \boldsymbol{w}, \mathbf{x}_i > +b\right) \tag{2.40}$$

can be used for calculating on which side of the hyperplane the data point $\mathbf{x}_i$ lies. If the hyperplane $(\hat{\boldsymbol{w}}, \hat{b})$ fulfils

$$(\hat{\boldsymbol{w}}, \hat{b}) = \underset{\boldsymbol{w},b}{\operatorname{argmin}} \{| < \boldsymbol{w}, \mathbf{x}_i > +b \mid = 1\}_{\forall\, i=1,\ldots,m}$$

for all data points of the training set, then the hyperplane is called canonical and has a distance of $\frac{1}{\|w\|}$ ($\| \mathbf{x} \|$ represents the length of vector $\mathbf{x}$) to the nearest data point

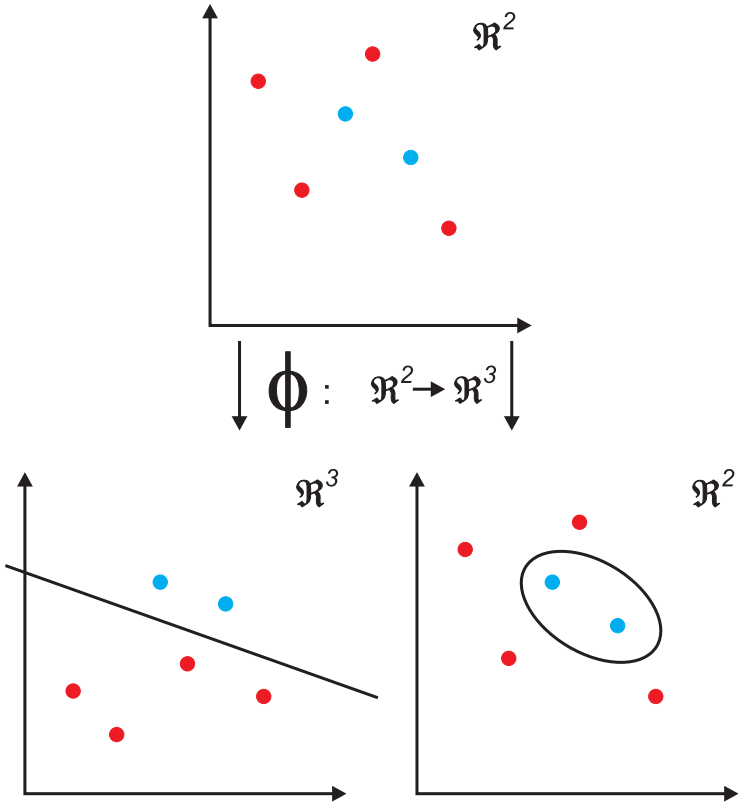


Figure 2.13: Using a linear hyperplane, it is impossible to separate the data points of the two classes (upper figure). Applying a non-linear mapping ϕ allows such a separation in a high dimensional feature space (lower, left figure) of which a suitable projection is shown. This corresponds to a non-linear decision surface in input space (lower, right figure). (The figure was adapted from (Schölkopf and Smola, 2001))

(called 'margin', see Fig. 2.12). $\frac{1}{\|w\|}$ is an important measure for the SVM because it is correlated with the robustness against perturbations (e.g. noise). The larger $\frac{1}{\|w\|}$ is, the better.

The goal is to find the optimal margin hyperplane with the largest $\frac{1}{\|w\|}$. To do this, we have to calculate (for a separable set of data)

$$\{w_{opt}, b_{opt}\} = \min_{w,b} \tau(w)$$

with

$$\tau(w) = \frac{1}{2} \| w \|^2 \tag{2.41}$$

and also subjected to

$$y_i (\langle w, x_i \rangle + b) \geq 1 \qquad \forall\, i = 1, \ldots, m\,. \tag{2.42}$$

Or as an alternative, which can be more convenient, it is possible to optimise the Lagrangian

$$L(\boldsymbol{w}, b, \alpha) = \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 - \sum_{i=1}^{m} \alpha_i \left(y_i \left(k(\boldsymbol{x}_i, \boldsymbol{w}) + b \right) - 1 \right)$$

where $\{\alpha_i\}_{i=1,\ldots,m}$ are Lagrange multipliers. L has to be maximised with respect to α_i and has to be minimised with respect to $\boldsymbol{w}$ and b (the optimum we search for is a saddle point). We are looking for $\boldsymbol{x}_i$ with $\alpha_i > 0$ because these vectors lie exactly on the margin hyperplane. These $\boldsymbol{x}_i$ are called support vectors.

As a result of the Lagrangian optimisation, we have to find the α that minimizes $W(\alpha)$, given by

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2.43}$$

and subject to $\alpha_i > 0 \; \forall \; i = 1, \ldots, m$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$. Using the $\alpha_i > 0$, we then can calculate $\boldsymbol{w}$ by

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i x_i$$

and b with

$$b = y_i - \sum_{i=1}^{m} y_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

for all support vectors. Sometimes it may be helpful not to use this b value and utilise b for adjusting the false positives and false negatives. Using Eq.(2.43) in Eq.(2.40) gives us a reformulated decision function

$$f(\boldsymbol{x}) = \operatorname{sgn} \left(\sum_{i=1}^{m} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b \right) . \tag{2.44}$$

Beginning with the Lagrangian L, the euclidean dot product $< \boldsymbol{x}, \boldsymbol{x}_i >$ is replaced by a distance measurement function $k(\boldsymbol{x}, \boldsymbol{x}_i) =< \phi(\boldsymbol{x}), \phi(\boldsymbol{x}_i) >$ (with ϕ as non-linear mapping from input into feature space), also called 'kernel'. The probability of replacing one positive definite kernel by another positive definite kernel is called 'kernel trick', or in other words: The SVM framework doesn't depend on the euclidean distance measure and can be replaced by other suitable kernels, e.g. the polynomial classifier of degree d

$$k(\boldsymbol{x}, \boldsymbol{x}_i) \;\; = \;\; < \boldsymbol{x}, \boldsymbol{x}_i >^d,$$

or the radial basis function classification with Gaussian kernel

$$k(\mathbf{x}, \mathbf{x}_i) = \exp\left(\frac{-\| \mathbf{x} - \mathbf{x}_i \|^2}{c}\right) \tag{2.45}$$

(with $c > 0$), or the tanh activation function

$$k(\mathbf{x}, \mathbf{x}_i) = \tanh\left(\kappa < \mathbf{x}, \mathbf{x}_i > + \Theta\right) \tag{2.46}$$

with $\kappa > 0$ as gain and Θ as horizontal shift. In Fig. 2.13 an example is shown where the euclidean dot product can not solve the problem, while a non-linear kernel can find a separation between the two classes.

The presented SVM can only discriminate two classes, but it is often necessary to distinguish more than two classes (e.g. section 4). For this problem different extensions for the SVM are available like, e.g.

- One versus the rest
  The data from one class is separated from the rest and a binary classifier is trained on these new sets. Thus for M classes one obtains M binary classifiers. The result of the classification is determined by

  $$\underset{j=1,\ldots,M}{\operatorname{argmax}}\, g^j(\mathbf{x})$$

  with

  $$g^j(x) = \sum_{i=1}^{m} y_i \alpha_i^j k(\mathbf{x}, \mathbf{x}_i) + b^j$$

- Pairwise classification
  For this strategy, the data for any two classes are selected and binary classifiers trained on these sub-data sets, obtaining $\frac{(M-1)M}{2}$ classifiers for M classes. For classification, the class with the most votes is selected as result.

It might be possible that the data can not be separated perfectly or it may be advantageous to avoid the use of this strict separation (because of e.g. outliers or mislabeled data in the data set and to avoid overfitting). This problem can be solved by introducing 'slack variables' $\xi_i > 0 \,\forall\, i = 1, \ldots, m$ and changing Eq.(2.41) by adding a penalty term into

$$\tau(\boldsymbol{w}) = \frac{1}{2} \| \boldsymbol{w} \|^2 + \underbrace{\frac{C}{m} \sum_{i=1}^{m} \xi_i}_{\text{penalty term}}$$

with a constant $C > 0$ and Eq.(2.42) into

$$y_i \left(\langle \boldsymbol{w}, \mathbf{x}_i \rangle + b\right) \geq 1 - \xi_i \qquad \forall\, i = 1, \ldots, m\,.$$

Both equations have to be used for the optimization process and for calculating the α_i and thus $\boldsymbol{w}$ and b. The constant C is controlling the tradeoff between minimization of the training error and minimization of the margin, but it can be a problem to select or to find an optimal C. This type of support vector classifier is called C-SVC. Other methods like the ν-SVC are available, for details see (Schölkopf and Smola, 2001).

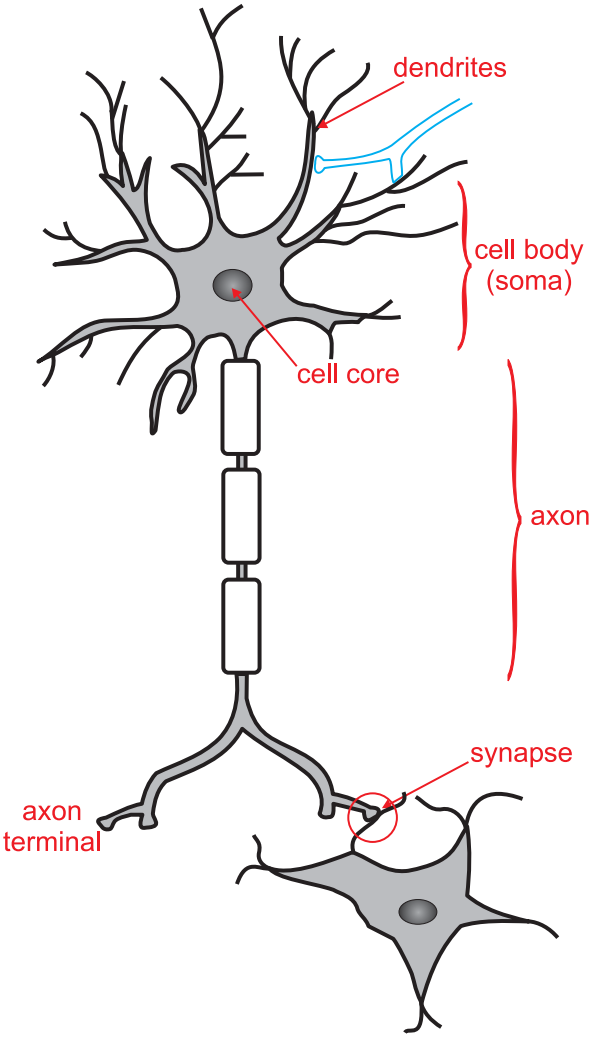


Figure 2.14: Morphological structure of a typical neuron. The central part of the cell is called cell body (soma). A typical nerve cell collects the output of other neurons over a branched dendritic tree. Furthermore, the cell possesses in most cases a signal generative zone (axon hill) and one axon that carries the action potential to other connected cells. The endings of the axon (axon terminal) and the dendrites are coupled by electric or chemical synapses. (Figure adapted from wikipedia.org)

2.3 Modeling of neurons

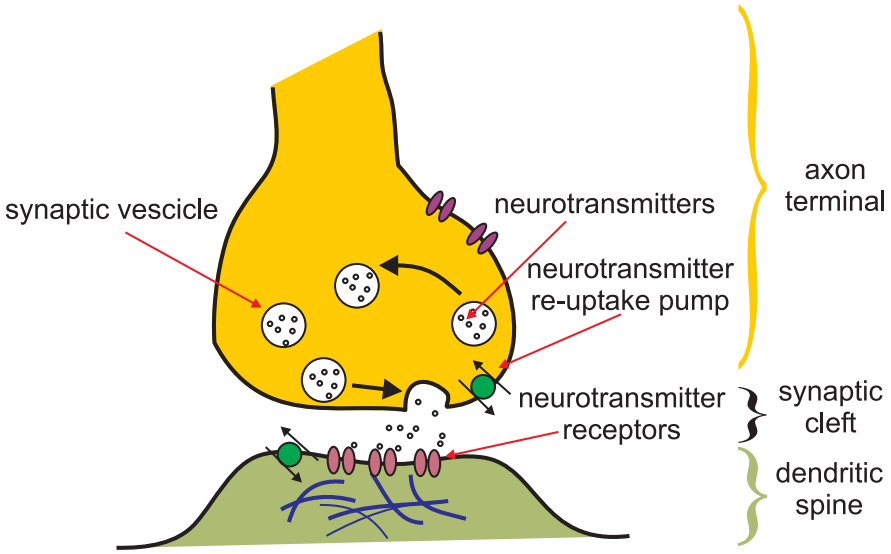


Figure 2.15: Simplified illustration of chemical synapses. An incoming action potential on the pre-synaptic side can create a release of neurotransmitters into the synaptic cleft. For this release of neurotransmitters, a synaptic vesicle filled with neurotransmitter fuses with the membrane of the axon terminal. If the neurotransmitters reach suitable receptors on the dendritic spine then a post-synaptic reaction can be created, representing the incoming action potential. The influence of the received neurotransmitters on the post-synaptic cell can vary from increasing or decreasing the excitability. And finally it may lead to a new action potential. The synaptic cleft is then cleaned from neurotransmitters by 're-uptake' pumps, which recycle the transmitter. (Figure adapted from wikipedia.org)

Real neurons in the (mammalian) brain show an astonishingly high variability in their morphology. The question, how neurons and their connecting elements, the synapses, function is still under heavy research. In Fig. 2.14 and Fig. 2.15 simplified illustrations of both of these basic building blocks in the brain are shown. The detailed structures of real neurons are too complex for most theoretical computational models of neurons and neuronal networks.

For this reason different types of simplified models have been developed to describe only the necessary aspects for treating a specific problem and ignore the rest of physiological knowledge. In the following text we will take a look on some of these models after discussing how to measure neuronal activities.

2.3.1 Measuring neuronal responses

Various methods of recording neuronal activities have been developed. These methods differ in their area of application and in the properties they observe. Some of them are designed for handling many neurons and other can access the detailed processes and dynamics of single cells.

One example is the 'patch clamp' method (Sakmann and Neher, 1984; Edwards et al., 1990). An electrode (glass pipette filled with conducting fluid) is brought into contact with the membrane of the nerve cell. This method even allows to investigate single ion channels on the membrane of neurons. Patch clamp has similarities in its concept to the older voltage clamp method, which uses two electrodes for measuring e.g. voltage and impedance (Cole and Curtis, 1939).

It requires high efforts to bring an electrode into contact with the surface of a neuron without damaging the cell. Often it is sufficient to bring the electrode into the proximity of a neuron. Electrodes are used for this purpose are typically made out of metal and are almost completely covered by isolatory material. For measurements, only the tip of the electrode is used. This allows to record the electric activity of the neuron extracellularly. Depending on the attributes of the electrode and its tip, the electrode shows different recording properties. E.g. electrodes differ in their abilities to measure frequency components of electrical activity. They also can differ in their maximal radius of detecting the activity of cells. In a typical case an electrode records spike activity of more than one neuron. It is often possible to re-assign single spikes to the different generating neurons by an appropriate spike sorting algorithm, e.g. (Lewicki, 1998). Not only single electrodes are used, but also arrays of electrodes.

A frequency range up to 20.000 Hz is recorded and then splitted into a lower frequency component with up to 200 Hz and a higher frequency component beyond 1.000 Hz. The higher frequency range contains the time course of the spikes. In the lower frequency range of such an extracellular electro-physiological recording, the combined electric activities of many neurons from the proximity of the electrode tip can be found. This measured variable is termed local field potential (LFP). If the electrode is shaped as a loop and placed on top of the *dura mater* then the epi-dural field potential can be measured, which bears a lot of similarities with the LFP. Electrodes placed on the scalp recorded for Electroencephalography (EEG) measurements.

Some other popular types of measuring neuronal activities are:

- fMRI (functional magnetic resonance imaging)
  This technology allows to measure the dynamics of the blood flow and its oxygenation. A MRI-tomograph is a large and expensive system. The temporal resolution is not fast enough to resolve single spike events and the spatial resolution does not allow to see single neurons. Furthermore, the dependence of the measured signal on the neuronal activity in the tissue is still under research

(Heeger and Ress, 2002). A correlation between the fMRI signal (BOLD) and the local field potential was found (Logothetis et al., 2001). fMRI is often used to obtain an overview about the activity distribution in the brain emerging for a specific cognitive task.

- optical imaging
  Optical imaging can measure through changes in the properties of scattered and reflected light e.g. the haemoglobin oxygenation level (like fMRI) and other quantities of the cell which are correlated with the scattering and reflection properties of the tissue (Villringer and Chance, 1997). The measurements can be done on a (sub-)millisecond time scale and the spatial resolution can resolve structures up to single synapses (e.g. using two photon microscopy). In addition, special dyes can be applied that allow to access other biochemical variables like e.g. concentrations of molecules and ions.

- MEG (Magnetoencephalography)
  The MEG is based on the idea to detect the magnetic fields created through the neuro-electric activities in the brain by SQUIDs (Superconducting Quantum Interference Devices) and thus gain information about the electric activity in the brain (Hämäläinen et al., 1993). MEGs are even more expensive than fMRI systems and very rare.

2.3.2 Integrate-and-fire neurons

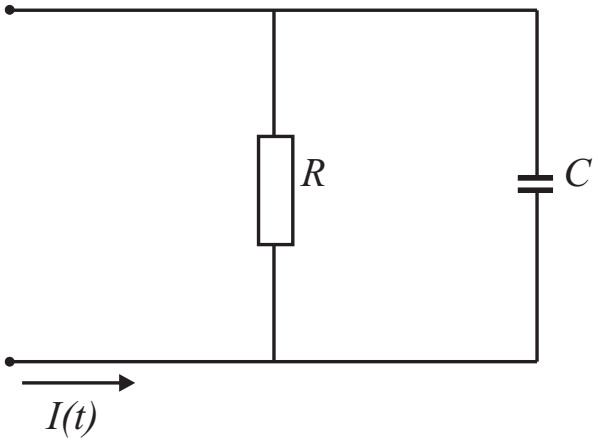


Figure 2.16: Circuit diagram of an integrate-and-fire neuron. The model neuron is composed of a resistor and a capacitor, which are connected in parallel.

The 'integrate-and-fire' (IAF) neuron is one of the most popular neuron models (The description of two other neuron models, the Hodgkin and Huxley model, and McCulloch and Pitts neurons, can be found in section A.1). The IAF neuron represents the dynamics of the membrane potential of a neuron simply by a capacitor and a resistor

(see Fig. 2.16). The capacitor is charged until a specific threshold is reached, at this moment an action potential is generated and the membrane potential is reset. The idea of integrate-and-fire neurons was apparently introduced in the year 1907 by Lapicque (Lapicque, 1907), which was roughly 45 years before the Hodgkin Huxley model was described. An extensive review of the integrate-and-fire model can be found in (Burkitt, 2006a; Burkitt, 2006b). Here we will focus only on the basics of the model and will follow loosely (Burkitt, 2006a). For more details on this topic, please see these reviews.

The dynamics of the membrane potential $v(t)$ is given by

$$C\frac{dv(t)}{dt} = I_{leak}(t) + I_{syn}(t) + I_{inj}(t)\,, \tag{2.47}$$

where C represents the membrane capacity, $I_{leak}(t)$ describes the current generated by passive leakage of the membrane, $I_{syn}(t)$ models the synaptic input, and $I_{inj}(t)$ allows to introduce a current that is externally injected into the cell. This equation of the membrane potential's dynamics does not include the process of generating spikes. When the membrane potential crosses the given threshold V_{th} then a spike is generated and the membrane potential is set to V_{reset}. The time course of the spike has to be defined by extra equations. In many studies action potentials are represented by Dirac delta functions because often the shape of the spikes are considered to be irrelevant for the current problem.

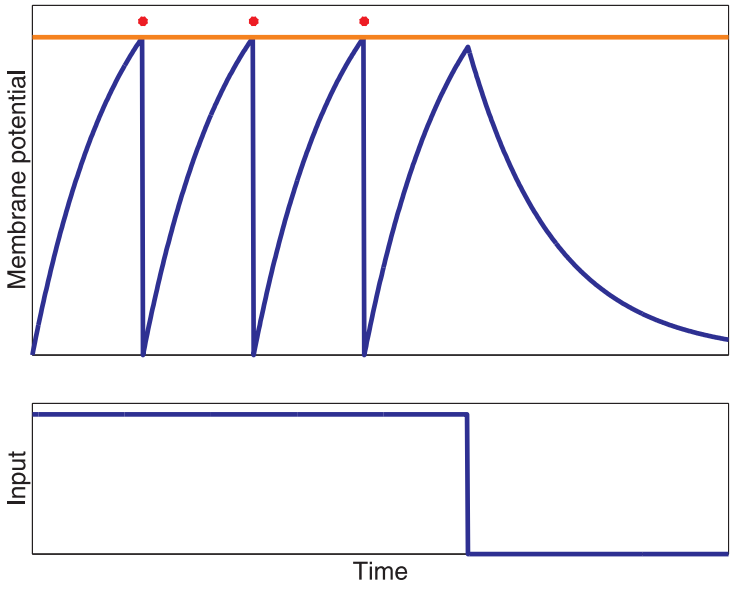


Figure 2.17: Membrane potential of an integrate-and-fire neuron under injection of constant current. After three spikes, which are marked with red dots, the input was turned off (see the lower figure for the time course of the input) and the membrane potential decays exponentially. The threshold of the neuron is represented by an orange line.

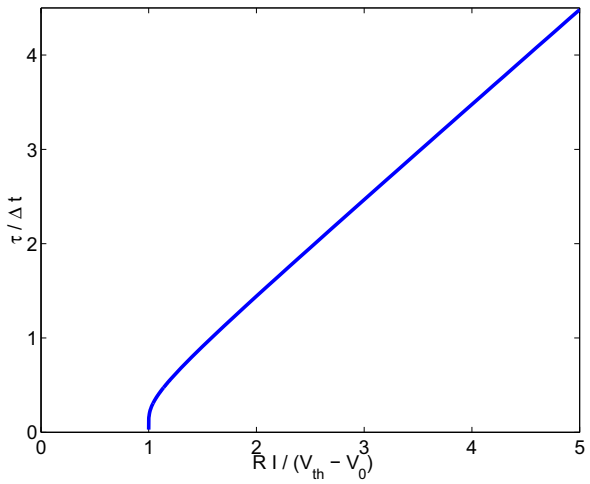


Figure 2.18: Firing rate of an integrate-and-fire neuron with constant input, see Eq.(2.50). The rate is shown in units of the membrane constant τ, and the input $R \cdot I$ in units of $V_{th} - V_0$.

The leak current in Eq.(2.47) can be described by

$$I_{leak}(t) = -\frac{C}{\tau}\left(v(t) - V_0\right) . \tag{2.48}$$

τ is the (passive) membrane time constant, defined by the resistor and the capacitor $\tau = R \cdot C$. V_0 is called resting potential and represents the asymptotic value of the membrane potential when no input from external sources is given to the integrate-and-fire neuron. Without input the relaxation process follows an exponential decay described by the membrane time constant.

If we restrain the input to $I_{injection}(t)$ and set the membrane potential at the initial time t_0 to resting potential ($v(t_0) = V_0$) then Eq.(2.47) together with Eq.(2.48) can be solved to express the membrane potential below threshold by (Burkitt, 2006a; Tuckwell, 1988)

$$v(t) = V_0 + e^{-\frac{t}{\tau}} \int_{t_0}^{t} \frac{1}{C} I_{injection}(t') e^{\frac{t'}{\tau}} \, dt' . \tag{2.49}$$

For constant current $I_{injection}(t) = I$, the Eq.(2.49) can be simplified to

$$v(t) = V_0 + I \cdot R \left(1 - e^{-\frac{t-t_0}{\tau}}\right) .$$

In the case where the threshold is fixed to V_{th} we can calculate the duration $\Delta t = t - t_0$ which is necessary to increase the membrane potential from the resting potential to the threshold:

$$\Delta t = -\tau \log\left(1 - \frac{V_{th} - V_0}{I \cdot R}\right) . \tag{2.50}$$

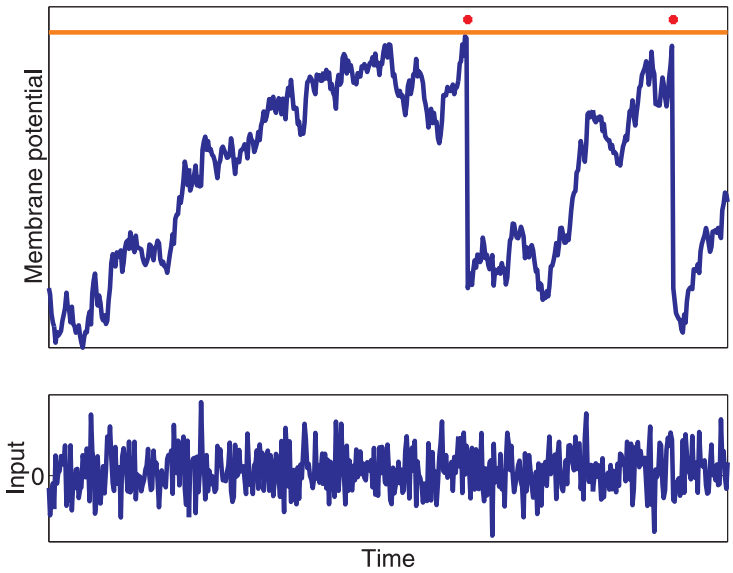


Figure 2.19: Membrane potential of an integrate-and-fire neuron with noisy input. Spikes are marked with red dots and the lower figure shows the time course of the input. The threshold of the neuron is represented by an orange line.

Eq.(2.50) shows that if $I \cdot R < V_{th} - V_0$ then no spike can be generated by the integrate-and-fire neuron because the decay caused by leaking currents is stronger than the increase of the membrane potential by the input current I in such a way that the threshold can never be reached. Assuming that after a spike is emitted, the membrane potential is reset directly to resting potential ($V_{reset} = V_0$), the IAF- neuron fires regularly with a spike-rate $\frac{1}{\Delta t}$ and an interspike interval (interval between two subsequent spikes) of Δt. An additional (absolute) refractory period τ_{ref}, where the neuron does not react to any input, can be included into this consideration and changes the spike-rate to $\frac{1}{\tau_{ref}+\Delta t}$.

We can describe the synaptic contributions to the membrane potential from other neurons in two different ways. Let us assume that the incoming neuronal activities of other neurons are given by

$$\begin{aligned} S_{E,k}(t) &= \sum_{t_i} \delta\left(t - t^i_{E,k}\right) \\ S_{I,k}(t) &= \sum_{t_i} \delta\left(t - t^i_{I,k}\right) . \end{aligned}$$

$S_{E,k}$ represents the excitatory post synaptic spikes from neuron k and $S_{I,k}$ denotes the inhibitory post synaptic input. The excitatory input increases the membrane potential, while the inhibitory input decreases the membrane potential. $t^i_{E,k}$ and $t^i_{I,k}$ are denoting the times where neuron k fires its i-th spike.

One convenient way of quantifying the influence of incoming spikes on the membrane

potential is to model 'current synapses'. Here, the impact of the input is independent of the actual membrane potential and can be described by

$$I_{syn}(t) = C\sum_{k=1}^{N_E} a_{E,k}S_{E,k}(t) + C\sum_{k=1}^{N_I} a_{E,I}S_{E,I}(t)\,, \tag{2.51}$$

where $a_{E,k} > 0$ and $a_{I,k} < 0$ reflect how strongly one spike affects the membrane potential of the integrate-and-fire neuron.

A more biologically realistic approach is to use 'conductance synapses' (Burkitt, 2006a; Tuckwell, 1979)

$$\begin{aligned} I_{syn}(t) &= C\,(V_E - v(t))\sum_{k=1}^{N_E} g_{E,k}S_{E,k}(t) \\ &\quad + C\,(V_I - v(t))\sum_{k=1}^{N_I} g_{I,k}S_{I,k}(t)\,. \end{aligned} \tag{2.52}$$

Comparing with A.1.1, the equations show some similarities to Eq.(2.52) because the Hodgkin-Huxley model is also a conductance-based, one-compartmental model. The crucial difference between these two models is the missing temporal dynamics of the conductances $g_{E,k} > 0$ and $g_{I,k} > 0$ for this model. V_E and V_I represent the reversal potentials for the excitatory and inhibitory synaptic connections ($V_I \leq V_{reset}$) $< V_{th} <$ V_E).

2.4 Learning and using (neuronal) networks

The neurons in the brain are not just transmitting information from one point to another, bu also processing this information. Incoming data from sensory systems is filtered by neuronal processes. This data is evaluated, including learned/ memorized information. The reduction in information starts already in the sensory system itself, where different receptors are only sensitive to a narrow part of the full spectrum of a signal. For example, for humans it is impossible to see infra-red or ultra-violet light, or to hear ultra-sonic sounds like other animals can do. Some of the information reduction steps in the brain are leading to illusionary effects like e.g. change blindness (Simons and Rensink, 2005). As a result of this whole information processing chain, typically a behavioural reaction is generated that interacts with the environment. Nevertheless, these steps of reducing the complexity of the input are very important for decreasing the computational demands of processing the stream of incoming information.

The central nervous system shows a high degree of variation and complexity in its structures, interactions and activity patterns. This makes it interesting to search for fundamental information processing mechanisms that can mimic the processing properties of real nervous systems even under these circumstances. One of the proposed classes are 'artificial neuronal networks' (or just 'neuronal networks').

Neuronal networks are typically composed of inter-connected artificial neurons. Computational rules describing how information is transmitted between the neurons, how input is included, and output is generated are also part of the neuronal network. Different neuronal network models may vary in their degree of biological plausibility and the included biological constraints. Some examples demonstrating the variety of approaches are different types of information representation (e.g. boolean values/ action potentials, real values), the resolution of temporal dynamics (from time steps to continuous time), the size of memory for previous inputs or outputs, deterministic or probabilistic interactions, the complexity of the connection structure (e.g. recurrent or pure feed-forward connections) and information flow, and the complexity of the neuron models (e.g. linear model, leaky integrate and fire model, Hodgkin Huxley model).

An important issue for neuronal networks is learning. If an artificial neuronal network should perform a specific computational function, then typically the structure of connections and other parameters of the model have to be adapted to the problem. One way for finding the best set of connections and parameters, a target or loss function (e.g. mean squared distance between the desired and actual outcome of the model) is necessary for evaluating the performance. During this process it may be helpful to constrain parameters, e.g. the number and strength of connections, during learning. Otherwise it can happen that the network just memorises the training data and will not perform well ('generalize') on other data.

Another possibility is the use of learning mechanisms that are based on the (temporal) correlation of neuronal activity patterns like e.g. the Hebb's rule (Hebb, 1949;

Sejnowski and Tesauro, 1989), long-term potentiation (LTP) (Bliss and Collingridge, 1993) and long-term depression (LTD) (Linden and Connor, 1995), or spike-time dependent plasticity (STDP) (Kistler and van Hemmen, 2000). This type of learning is often referred to as 'neurons that fire together, wire together'. In more detail, given a weight that describes the postsynaptic response to an incoming spike (e.g. the synaptic efficacy), we can store information in the values of these weights or we can adjust the weights such that the performance of a network for a given task is optimized (Gerstner and Kistler, 2002b). These changes can be driven by e.g. the correlated activity of pre- and postsynaptic neurons.

In the following, we will introduce some types of neuronal networks and learning procedures. The overview will start with structurally simple neuronal network models (like perceptrons) will continue with learning strategies. In section A.3 and A.4 a glance on more complex neuronal network models will be given.

2.4.1 Feedforward networks

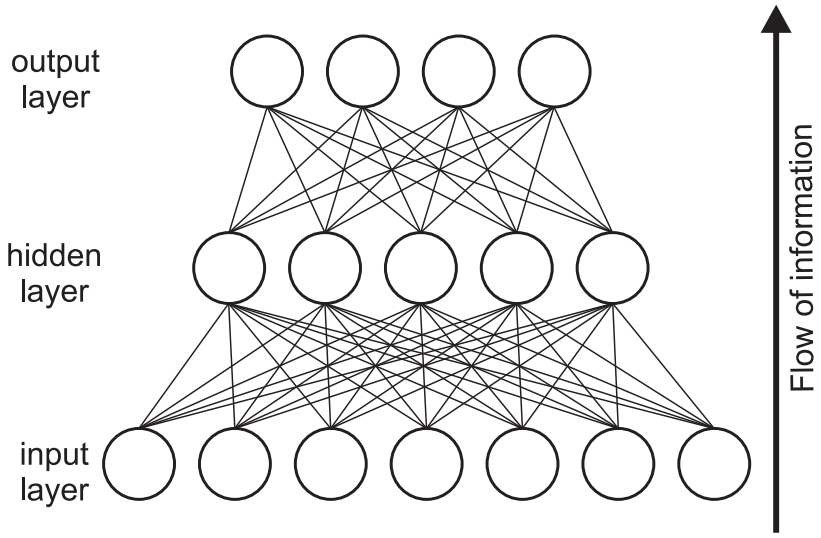


Figure 2.20: Illustration of a feedforward neuronal network with one hidden layer.

In purely feedforward networks, the flow of information proceeds from one layer to the next layer, without propagating backwards to a lower layer, or exchanging information between neurons within the same layer, and also without direct connections beyond the next layer (see Fig. 2.20). Information enters such a neuronal network at the first, the input layer. The last, output, layer of this network represents the results of the computation. Layers between input and output layer are often termed hidden layers, because they have no direct access to the input or output.

Perceptrons

One prominent example relying on this feedforward structure are perceptrons, originally proposed by Rosenblatt (Rosenblatt, 1958). A detailed overview of the perceptron can be found in (Hertz et al., 1991). This summary uses this review as basis. The simple perceptron is composed of only an input and an output layer. The output Ω_i of neuron ('unit') i is described by

$$\Omega_i = g\left(\sum_j w_{i,j} \cdot x_j\right) ,$$

where x_j represents the input from input unit j and $w_{i,j}$ the 'strength' of the connection between input neuron j and output neuron i. The function $g(.)$ is called 'activation function', typical examples are the Heavyside function with variable threshold ϑ_i, the signum function, sigmoid-like functions (including parameters for shift and steepness) or just a linear function.

Simple perceptrons show similarities to support vector machines. The perceptron defines a hyperplane parametrised by the threshold and weights in multi-dimensional Euclidean space that can be used to separate two different classes of input data. A simple perceptron can only define one hyperplane. If a data set with two classes can be separated by a simple perceptron, the problem is called linearly separable (Hertz et al., 1991; Minsky and Papert, 1969).

Assuming a linearly separable problem of this type, the weights and thresholds can be learned by finding support vectors with (non-soft) margin hyperplanes and linear kernel, see section 2.2.4 for details, or (Schölkopf and Smola, 2001; Hertz et al., 1991).

If the activation function $g(.)$ is differentiable and if an error measure is given, the weights can be computed via a gradient descent algorithm (which is also called 'method of steepest descent'). One example for an error measure is the sum-of-squares cost function

$$\begin{aligned} E &= \frac{1}{2}\sum_{i,\mu} (O_i^\mu - \Omega_i^\mu)^2 \\ &= \frac{1}{2}\sum_{i,\mu} \left(O_i^\mu - g\left(\sum_j w_{i,j} \cdot x_j^\mu\right)\right)^2 , \end{aligned} \tag{2.53}$$

where μ identifies the input pattern and O_i^μ defines the desired result for output neuron i and input pattern μ.

The gradient descent starts with an initial set of weights $w_{i,j}^{t=0}$ and moves with each iteration step in direction of the local downhill gradient $-\frac{\partial E}{\partial w_{i,j}}$ until reaching a set of weights that (locally) minimize the error. The updates of the weights are calculated

by

$$\begin{aligned} w_{i,j}^{t+1} &= w_{i,j}^{t} + \Delta w_{i,j} \\ &= w_{i,j}^{t} - \eta \frac{\partial E}{\partial w_{i,j}}, \end{aligned}$$

where η is a learning parameter. A bad choice of η may lead to problems finding a local minimum. Often it is helpful to reduce the learning parameter during the learning process (simulated annealing). For Eq.(2.53) the downhill gradient is given by

$$-\eta \frac{\partial E}{\partial w_{i,j}} = \eta \sum_{\mu} \left(O_i^{\mu} - g\left(\sum_k w_{i,k} \cdot x_k^{\mu} \right) \right) x_j^{\mu} \frac{\partial\, g\left(\sum_k w_{i,k} \cdot x_k^{\mu}\right)}{\partial w_{i,j}}. \tag{2.54}$$

It is important to note that in general a gradient descent method may find only local minima, depending on the initial setting.

Since simple perceptrons can only represent linear separable functions, it is desirable to introduce hidden layers for extending their functional capabilities. This extension caused problems for some time because it was not clear how to train the weights of the multiple layers. The invention of the 'back-propagation' algorithm (Bryson and Ho, 1969; Werbos, 1974; Parker, 1982; Rumelhart et al., 1986a; Rumelhart et al., 1986b) solved this problem. In its core, the back-propagation method is also a gradient descent performed on the weights.

The output neuron i in layer l is defined as follows: For the first hidden layer (the input layer is denoted by a superscript 0) as

$$h_{i,1}^{\mu} = g\left(\sum_j w_{i,j}^{0 \to 1} \cdot x_j^{\mu} \right),$$

for all other hidden layers as

$$h_{i,l}^{\mu} = g\left(\sum_j w_{i,j}^{(l-1) \to l} \cdot h_{j,(l-1)}^{\mu} \right),$$

and for the output layer, which is denoted by index M as

$$\Omega_i^{\mu} = g\left(\sum_j w_{i,j}^{(M-1) \to M} \cdot h_{j,(M-1)}^{\mu} \right).$$

The weights $w_{i,j}^{(l-1) \to l}$ in this definition have been indexed with superscripts for the regarded layer (upper layer (l) and lower layer ($l-1$)).

Using a gradient descent algorithm on the sum-of-squares cost function, the update for the weights is

$$\Delta w_{i,j}^{0 \to 1} = \eta \sum_{\mu} \delta_{i,1}^{\mu} \cdot x_j^{\mu} \tag{2.55}$$

$$\Delta w_{i,j}^{(l-1) \to l} = \eta \sum_{\mu} \delta_{i,l}^{\mu} \cdot h_{j,(l-1)}^{\mu} \tag{2.56}$$

with

$$\begin{aligned}\delta^{\mu}_{j,l} &= \frac{\partial\, h^{\mu}_{j,l}}{\partial\, w^{(l-1)\to l}_{i,j}} \sum_{i} w^{l\to(l+1)}_{i,j} \cdot \delta^{\mu}_{i,(l+1)} \\ \delta^{\mu}_{j,M} &= \frac{\partial\, \Omega^{\mu}_{j}}{\partial\, w^{(M-1)\to M}_{i,j}} \left(O^{\mu}_{j} - \Omega^{\mu}_{j}\right)\end{aligned}$$

While the derivatives $\frac{\partial h}{\partial w}$ and the values of $h^{\mu}_{i,l}$ (and Ω^{μ}_{i}) are propagated with the current to the next layer, the propagating errors $\delta^{\mu}_{i,l}$ are send into the opposite direction, thus giving the algorithm its name. The method can be extended to other differentiable error measures. For more details on back-propagation and variations of this method see (Hertz et al., 1991).

Eq.(2.55) and Eq.(2.56) can be calculated and summed over all input patterns at once, which is then called learning in batch mode, or with only one input pattern, which changes after each iteration step. The latter is called incremental update and has often advantages over the batch mode, e.g. requiring less memory capacity for each learning step.

2.4.2 Bayesian networks

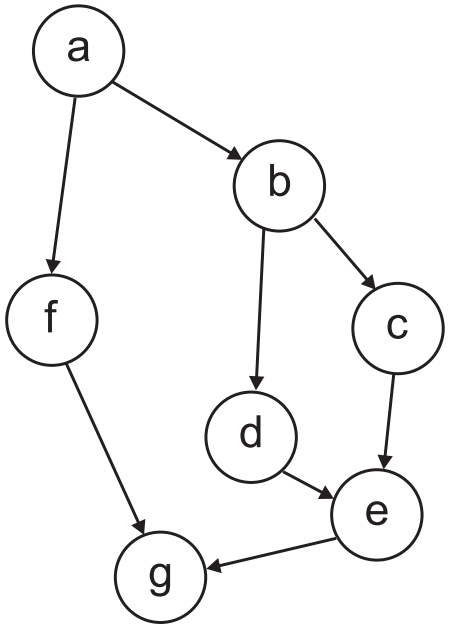


Figure 2.21: Example of a Bayesian network. The network contains the prior and conditional probabilities: $p(a)$, $p(f \mid a)$, $p(b \mid a)$, $p(c \mid b)$, $p(d \mid b)$, $p(e \mid c, d)$, and $p(g \mid f, e)$. (The image was taken from (Zhang and Poole, 1994))

Bayesian networks (also called 'Bayesian belief networks' or 'belief networks') are a popular method to capture interactions between probabilistic variables, like probabilistic expert knowledge (Heckerman et al., 1995). This type of networks can be used to learn about underlying relations hidden in a data set (Pearl, 2000) and can make predictions on the basis of this information. Bayesian networks can handle incomplete data sets where one or several variables are missing, using learned correlations between the variables (Heckerman, 1995). Reviews on this topic can be found in (Heckerman, 1995; Pearl and Russell, 2003; Zhang and Poole, 1994).

A Bayesian network $\aleph$ consists of three basic components. First, a set of stochastic variables V. Second, a set of arcs A that specify how these variables are related to each other. A together with V forms a directed graph, e.g. see Fig. 2.21. If this graph does not contain loops, it is called a directed acyclic graph. An undirected graphical model is called Markov network (Pearl, 1988). The representation of a variable is called 'vertex' and the connections 'edges'. Vertices without incoming edges are called 'sources' and vertices without outgoing edges 'sinks'. $A \rightarrow B$ means that the observation of B depends on A, and that A is a parent of B. The last component of a Bayesian network is a set of conditional probabilities for all variables, given their respective parents. Thus, $p(\nu \mid \pi_\nu)$ are the conditional probabilities of variable ν conditioned by its parents denoted by π_ν. The set of parents π_ν is empty for sources (Zhang and Poole, 1994).

Using this information, it is possible to calculate the prior joint probability

$$P_\aleph(V) = \prod_{\nu \in V} p(\nu \mid \pi_\nu) \,.$$

We can marginalise this joint probability distribution through

$$P_\aleph(X) = \sum_{V-X} P_\aleph(V) \,.$$

The sum has to be calculated over all variables from the set V with the exception of the variables from set X. Using the Bayes theorem we can calculate the posterior probability

$$P_\aleph(X \mid Y) = \frac{P_\aleph(X, Y)}{P_\aleph(Y)} \,.$$

It is often helpful to remove non-involved vertices. This requires to remove all arcs and remove or modify the conditional probabilities that are connected with this vertex. At least, the vertex itself has to be purged (Zhang and Poole, 1994).

Furthermore, it was shown that the calculation of posterior probabilities in this framework can be NP-hard (Cooper, 1990). In many cases using the particular structure of the problem can help to reduce the computational effort. Important for this reduction is to use as much as possible factorizations of joint probabilities. This allows to sum out variables with less computational effort and to create a replacing compound vertex. Using this idea and similar tricks makes it possible to reduce the computation time

needed for solving the problem (Shafer and Shenoy, 1998; Zhang and Poole, 1994). It is also possible to use different types of Monte Carlo simulation methods (e.g. Gibbs sampling) for approximating desired properties (Neal, 1993; Pearl, 1988; Geman and Geman, 1984; Madigan et al., 1995).

Bayesian learning

Several strategies for learning Bayesian networks (Buntine, 1994; Hinton et al., 2006) from complete or incomplete data sets exist. Learning aims at finding conditional probability distributions (Jordan, 1999; Binder et al., 1997) as well as the structure of the network (Friedman, 1998) that explain the data best.

Given a data set consisting of samples generated by an unknown system, we may be interested in predicting other events produced by the same system. If we assume a prior distribution over possible models $p(\mathit{Model})$ and a probability distribution $p(\Theta \mid \mathit{Model})$ regarding the parameters for the models, we can express the prediction of the probability for an event Z through

$$\begin{aligned} p(Z \mid \mathit{Data}) &= \sum_{\mathit{Models}} p(Z \mid \mathit{Model}, \mathit{Data})\, p(\mathit{Model} \mid \mathit{Data}) \\ &= \sum_{\mathit{Models}} p(Z \mid \mathit{Model}, \mathit{Data}) \frac{p(\mathit{Data} \mid \mathit{Model})\, p(\mathit{Model})}{p(\mathit{Data})} \end{aligned}$$

with

$$p(Z \mid \mathit{Model}, \mathit{Data}) = \int p(Z \mid \Theta, \mathit{Model})\, p(\Theta \mid \mathit{Model}, \mathit{Data})\, d\Theta$$

and

$$p(\mathit{Data} \mid \mathit{Model}) = \int p(\mathit{Data} \mid \Theta, \mathit{Model})\, p(\Theta \mid \mathit{Model})\, d\Theta\,.$$

The model likelihood $p(\mathit{Data} \mid \mathit{Model})$ is often the basis for Bayesian model selection (Buntine, 1994). For comparing two Bayesian network models (MacKay, 1995), we can use e.g.

$$\frac{p(\mathit{Model}^{\alpha} \mid \mathit{Data})}{p(\mathit{Model}^{\beta} \mid \mathit{Data})} = \frac{p(\mathit{Model}^{\alpha})\, p(\mathit{Data} \mid \mathit{Model}^{\alpha})}{p(\mathit{Model}^{\beta})\, p(\mathit{Data} \mid \mathit{Model}^{\beta})}\,.$$

Also a very popular score (especially for flat priors $p(\mathit{Model})$) for comparing two models is the so called Bayes-factor

$$\frac{p(\mathit{Data} \mid \mathit{Model}^{\alpha})}{p(\mathit{Data} \mid \mathit{Model}^{\beta})}\,.$$

When it is ineffective (or impossible) to calculate $p(Z \mid \mathit{Data})$, it may be sufficient to use an approximation of $p(Z \mid \mathit{Data})$. This can be done e.g. by evaluating the

maximum *a posteriori* model or a sum over a selection of the models. Using maximum *a posteriori* parameters leads to

$$p\left(Z \mid Model, Data\right) \approx p\left(Z \mid Model, \hat{\Theta}\right) ,$$

because $p\left(\Theta \mid Model, Data\right)$ is approximated by a Dirac delta distribution (Friedman, 1998). $\hat{\Theta}$ is the parameter that maximises

$$p\left(\Theta \mid Model, Data\right) = \frac{p\left(Data \mid \Theta Model\right) p\left(\Theta \mid Model\right)}{p\left(Data \mid Model\right)} .$$

For finding $\hat{\Theta}$, we can use a gradient descent method (see section 2.4.1) or the EM algorithm (see section 2.4.3).

When the structure is given, the process of learning is reduced to optimize conditional probabilities $p(v \mid \pi_v)$. It is important to incorporate the probabilistic nature of these variables by requiring

$$0 \geq p(v \mid \pi_v) \geq 1$$

and

$$\sum_v p(v \mid \pi_v) = 1 .$$

These constraints can be satisfied by projecting the gradient (for finding the optimal conditional probabilities) on a constrained surface (Binder et al., 1997; Lanteri et al., 2002; Lanteri et al., 2001).

2.4.3 Monte Carlo methods and expectation maximisation algorithm

In the previous section about Bayesian belief networks, we encountered the problem of finding a maximum *a posteriori* solution e.g. for learning Bayesian network structures and conditional probabilities. Different strategies are available for calculating an approximation of the solution. One example is the so called 'expectation maximisation' algorithm (in short EM-algorithm). Since this method has a lot in common with the 'Gibbs sampling' method (Buntine, 1994), after introducing Monte Carlo basics we will discuss Gibbs sampling first and continue with the EM-algorithm.

Gibbs sampling is one member of the family of Monte Carlo algorithms, of which a review can be found in (Neal, 1993) and (Walsh, 2002), which we will use as guideline. First, a simple Monte Carlo method, rejection sampling is described.

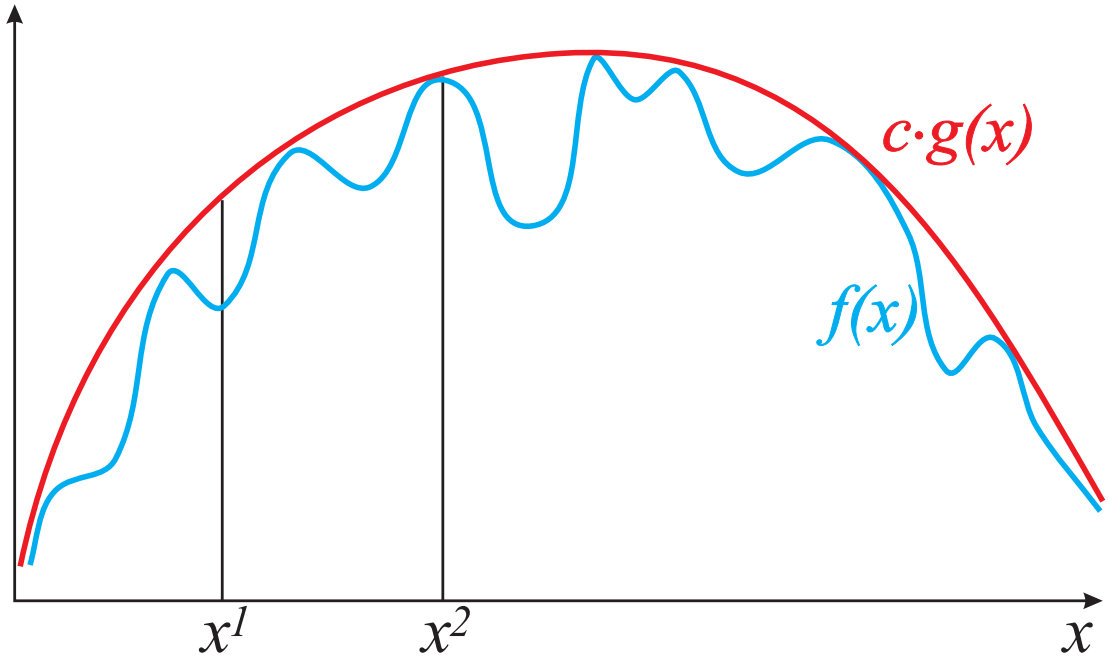


Figure 2.22: Rejection sampling allows to draw samples indirectly from a complex probability distribution $p(x)$. Instead of using $p(x)$ we will use $f(x)$ (has still the same complexity as $p(x)$), which is exactly $p(x)$ without normalisation. In addition, we need an auxiliary function $g(x)$ with $f(x) \leq c \cdot g(x)$. $g(x)$ has to be selected in such a way that it is easy to draw samples from it. For considering the original course of $f(x)$ (and thus $p(x)$) the samples will be rejected with probability $p_{accept}(x^\star) = \frac{f(x^\star)}{c \cdot g(x^\star)}$. The figure shows an example for $g(x)$ and $f(x)$. x^2 is less often rejected than x^1 because at x^2 $f(x)$ is better represented by $cg(x)$. The closer $cg(x)$ approximates $f(x)$, the less samples are rejected.

Rejection and importance sampling

If it is not possible to draw samples from a probability distribution $p(x_1, \cdots x_N)$ itself because e.g. the cumulative distribution function is not known or it is computationally to extensive to draw samples, then rejection sampling can help to solve this problem. This method uses an (more simple) auxiliary function for drawing the samples. For considering the original course of $p(x_1, \cdots x_N)$ each sample is rejected with a probability, which represents the difference between $p(x_1, \cdots x_N)$ and the auxiliary function (see Fig. 2.22).

Typically, Monte Carlo methods are used for evaluating equations from the type

$$< a >= \sum_{x_1, \cdots, x_N} a(x_1, \cdots, x_N) \cdot p(x_1, \cdots x_N)$$

by approximating $< a >$ through

$$< a > \approx \frac{1}{N} \sum_{t=0}^{N-1} a\left(x_1^t, \cdots, x_N^t\right) , \tag{2.57}$$

where $x_1^t, \cdots, x_N^t$ are drawn from the distribution $p\left(x_1, \cdots x_N\right)$.

Given the case that $p\left(x_1, \cdots x_N\right)$ can not be used to generate the necessary samples for Eq.(2.57), we need to use a 'workaround' like 'rejection sampling' (Ripley, 1987; Devroye, 1986). For this algorithm, we need to assume a function $f(\boldsymbol{x})$ that represents $p(\boldsymbol{x})$ by

$$p\left(x_1, \cdots x_N\right) = \frac{f\left(x_1, \cdots x_N\right)}{\sum_{x_1, \cdots, x_N} f\left(x_1, \cdots x_N\right)}$$

and a function $g(\boldsymbol{x})$ with

$$f(\boldsymbol{x}) \leq c \cdot g(\boldsymbol{x}) \ \ \forall \, \boldsymbol{x}$$

where c is a constant. It may not be possible to draw samples $\boldsymbol{x}^t$ from $f(\boldsymbol{x})$ or $p(\boldsymbol{x})$ directly, but it is possible use $g(\boldsymbol{x})$ as a substitute for $f(\boldsymbol{x})$. We draw a $\boldsymbol{x}^\star$ from $g(\boldsymbol{x})$ and this sample will be accepted as part of the sum in Eq.(2.57) with the probability

$$p_{accept}(\boldsymbol{x}^\star) = \frac{f(\boldsymbol{x}^\star)}{c \cdot g(\boldsymbol{x}^\star)} .$$

The effectivity of this method depends on how well $g(\boldsymbol{x})$ approximates $f(\boldsymbol{x})$. This is expressed through the number of sampled $\boldsymbol{x}^\star$ being accepted. For example, in the case where $f(\boldsymbol{x}) = g(\boldsymbol{x})$ with $c = 1$, all $\boldsymbol{x}^\star$ will be accepted.

The same problem of calculating an approximation of $< a >$, can be addressed by importance sampling (Hastings, 1970). Here, the function $g(\boldsymbol{x})$ has to be $g(\boldsymbol{x}) \neq 0$ where $f(\boldsymbol{x}) \neq 0$. Then we can use

$$< a > \approx \frac{\sum_t a\left(\boldsymbol{x}^t\right) \frac{f\left(\boldsymbol{x}^t\right)}{g\left(\boldsymbol{x}^t\right)}}{\sum_t \frac{f\left(\boldsymbol{x}^t\right)}{g\left(\boldsymbol{x}^t\right)}} .$$

It may happen that this approximation is not very accurate for a small number of samples but this method is not rejecting information from any samples.

Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm (Walsh, 2002; Hastings, 1970; Metropolis and Ulam, 1949; Metropolis et al., 1953) is a further development of the rejection sampling method, which uses and generates Markov chains while sampling the probability

distribution. If this Markov chain is 'ergodic' (Norris, 1997) then it is guaranteed that the whole distribution is used for sampling, independently of initial values. This algorithm can be used to sample from $p(\mathbf{x})$ when

$$p(\mathbf{x}) = \frac{f(\mathbf{x})}{c}\,, \tag{2.58}$$

where c is a normalisation constant that is hard to evaluate. While applying the Metropolis algorithm, a Markov chain $\{\mathbf{x}^t\}_{t=0,\cdots,n}$ is generated. Such a Markov chain is based on a Markov process, which is a random process with memory only for the latest state. Expressed in conditional probabilities

$$p\left(\mathbf{x}^{n+1} \mid \mathbf{x}^n, \{\mathbf{x}^t\}_{t=0,\cdots,(n-1)}\right) = p\left(\mathbf{x}^{n+1} \mid \mathbf{x}^n\right)\,,$$

where $\mathbf{x}^{n+1}$ represents the values for the random variables $\mathbf{X}$ in the next step, and $\mathbf{x}^n$ for the actual step. Using this Markov assumption, we can build a Markov chain with values $\{\mathbf{x}^t\}_{t=0,\cdots,n}$ for the regarded random variables.

There are several ways of producing Markov chains, one way to generate a Markov chain is to start with an initial probability distribution $p_0(\mathbf{x})$ and to propagate the distribution via

$$p_{t+1}(\mathbf{x}) = \sum_{\mathbf{x}'} p_t(\mathbf{x}') \cdot \Pi_t(\mathbf{x}', \mathbf{x})$$

(Chapman-Kolomogrov equation), where $\Pi_t(\mathbf{x}', \mathbf{x})$ represents the transition probabilities between state $\mathbf{x}'$ and $\mathbf{x}$. Markov chains can be 'ergodic' in such a way that there exists a stationary state $\psi(\mathbf{x})$ for which

$$\psi(\mathbf{x}) = \sum_{\mathbf{x}'} \psi(\mathbf{x}') \cdot \Pi_t(\mathbf{x}', \mathbf{x})\,.$$

If a Markov chain is ergodic, then after a 'burn-in' phase, it does not depend on the initial samples anymore and it will show no periodic behavior. The use of such a ergodic Markov chain guarantees to sample the whole space with the Monte Carlo algorithm independent of the start values. For more details on ergodic Markov chains see (Neal, 1993; Walsh, 2002).

Back to the Metropolis algorithm, starting with an initial $\mathbf{x}^0$ with $f(\mathbf{x}^0) > 0$ and $\Pi(\mathbf{x}', \mathbf{x})$ (called candidate-generating distribution) for describing the probability for making a transition (or jump) from $\mathbf{x}'$ to $\mathbf{x}$. For this algorithm, it is necessary that $\Pi(\mathbf{x}', \mathbf{x}) = \Pi(\mathbf{x}, \mathbf{x}')$. Beginning with an initial $\mathbf{x}$, we sample a $\mathbf{x}^\star$ using the candidate-generating distribution. This candidate $\mathbf{x}^\star$ will be accepted with the probability

$$\begin{aligned} p_{\text{accept}} &= \min\left(\frac{p(\mathbf{x}^\star)}{p(\mathbf{x}^{t-1})}, 1\right) \\ &= \min\left(\frac{f(\mathbf{x}^\star)}{f(\mathbf{x}^{t-1})}, 1\right)\,. \end{aligned}$$

If the candidate is accepted then it is stored as a new element in the Markov chain. Otherwise new candidates are drawn and tested via p_{accept} until a suitable new element is found. Hastings (Hastings, 1970) extended this procedure to candidate-generating distributions that do not require $\Pi(\mathbf{x}', \mathbf{x}) = \Pi(\mathbf{x}, \mathbf{x}')$. For such distributions p_{accept} is given by

$$p_{accept} = \min\left(\frac{f(\mathbf{x}^\star)\,\Pi(\mathbf{x}^\star, \mathbf{x}^{t-1})}{f(\mathbf{x}^{t-1})\,\Pi(\mathbf{x}^{t-1}, \mathbf{x}^\star)}, 1\right).$$

This method is called Metropolis-Hastings algorithm. While applying this method, especially for building a good $\Pi(\mathbf{x}^\star, \mathbf{x}^{t-1})$, several problems can occur, see (Walsh, 2002; Neal, 1993).

Gibbs sampling

Gibbs sampling (Geman and Geman, 1984; Walsh, 2002; Neal, 1993) is a special case of the Metropolis-Hastings algorithm. The candidate-generating distribution $\Pi(\mathbf{x}, \mathbf{x}')$ is given through univariant conditional probability distributions in which all random variables are fixed values with the exception of one. Let us assume that we have a multi-dimensional probability distribution $p(a, b, c)$, then we can calculate the univariant conditional distribution with the fixed values $b = b^\star$ and $a = a^\star$ by

$$p(a \mid b^\star, c^\star) = \frac{p(a, b^\star, c^\star)}{\sum_a p(a, b^\star, c^\star)}$$

Furthermore, p_{accept} is set to 1 which makes the algorithm accept all the samples (Walsh, 2002).

The whole procedure works as follows: Starting with a set of initial values (a^0, b^0) we perform one 'scan'

$$\begin{aligned} c^i &\leftarrow p(c \mid a = a^{i-1}, b = b^{i-1}) \\ a^i &\leftarrow p(a \mid c = c^i, b = b^{i-1}) \\ b^i &\leftarrow p(b \mid a = a^i, c = c^i) \end{aligned}$$

by sampling from the univariant conditional distributions while successively replacing old values through new ones. At the end of the scan, we obtain a new set $\mathbf{x}^i = (a^i, b^i, c^i)$. For more variables the procedure can be extended accordingly.

If we drop the sets from the burning-in phase and use only the remaining $\mathbf{x}$, we can calculate with this information the expectation value of a function $f(\mathbf{x})$

$$E[f(\mathbf{x})] \approx \frac{1}{m}\sum_{i=1}^{m} f(\mathbf{x}^i)$$

or can compute a marginalized probability distribution

$$p(x) \approx \frac{1}{m}\sum_{i=1}^{m} p(x \mid z = z^i, y = y^i).$$

EM-algorithm

Let us assume that a joint probability distribution $p(\mathbf{x}, \mathbf{y} \mid \theta)$ is parametrized by θ and is depending on a set of random variables $\mathbf{z} = (\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is observable and $\mathbf{y}$ is not. The task is to find the θ, which maximizes the marginalised probability distribution $p(\mathbf{x} \mid \theta)$ with given values for $\mathbf{x}$. Sometimes it is helpfull to use the log-likelihood

$$L(\theta \mid \mathbf{x}) \;=\; \log\left(p\left(\mathbf{x} \mid \theta\right)\right)$$

instead of $p(\mathbf{x} \mid \theta)$(Bilmes, 1997), because products can be replaced by sums. This can help to simplify the (analytically) handling of equations.

For finding a maximum of L we can use the expectation maximisation algorithm (or 'EM'-algorithm) (Dempster et al., 1977; Buntine, 1994; Neal and Hinton, 1999; Bilmes, 1997). This algorithm compromises two steps, which are applied repetitively:

1. The expectation step, which computes the expectation for e.g. $\log(p(\mathbf{x}, \mathbf{y} \mid \theta))$, given the observed values for $\mathbf{x}$ (data), using the probability distribution $p(\mathbf{y} \mid \mathbf{x}, \theta^{i-1})$, based on the set of parameters θ^{i-1} from the last iteration

$$\Pi(\theta, \theta^{i-1}) \;=\; \int_{\mathbf{y}} d\mathbf{y}\, \log\left(p\left(\mathbf{x}, \mathbf{y} \mid \theta\right)\right) \cdot p\left(\mathbf{y} \mid \mathbf{x}, \theta^{i-1}\right) .$$

2. The maximisation step, where the optimal parameter θ is computed for the expectation Π,

$$\theta^{i} \;=\; \underset{\theta}{\operatorname{argmax}}\, \Pi(\theta, \theta^{i-1}) .$$

Each iteration step increases (or conserves) the log-likelihood (instead of the likelihood also the *a posteriori* distribution can be used). Finding a local maximum by this method is guaranteed (Buntine, 1994). If a θ^{i} only increases the expectation with respect to the last set of parameters and is not the optimum, then the method is called generalized EM (Neal and Hinton, 1999). A discussion of other variants of the EM algorithm can be found in (Neal and Hinton, 1999). Furthermore, EM can be understood as a deterministic version of Gibbs sampling (Buntine, 1994; Sahu and Roberts, 1999). As a consequence, the expectation step can be approximated by Monte Carlo methods.

2.4.4 Reinforcement learning

Let us assume that we want to build a machine ('agent') that interacts with its environment. The agent is able to observe different states of the environment and is able to perform different actions that change the agent's environment. Some of the

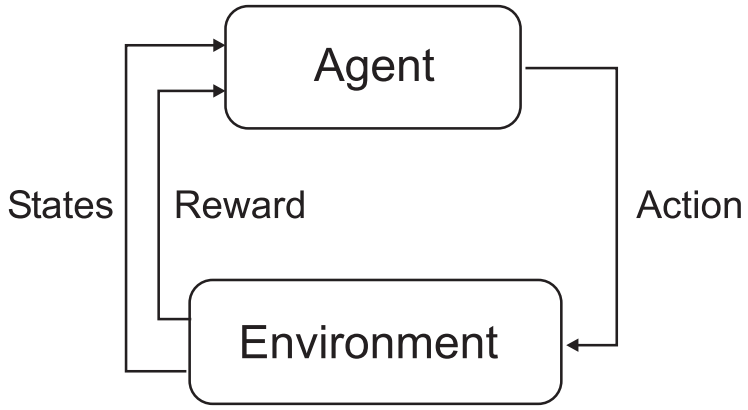


Figure 2.23: Illustration of the interaction and interchange of information between agent and environment in a closed loop situation during reinforcement learning.

observed states have a special meaning for the agent. The agent tries to manipulate the environment in such a way that these states show an optimal configuration. An example for such a setup would be a computer gambling against a human. The computer is instructed to optimize the amount of the gained money (reward). For doing so the computer cam actively choose its own moves and can observe the moves of its opponent. Fig. 2.23 illustrates the situation. The computational problem is how to map situations onto actions in a way that the reward is maximized.

For this purpose reinforcement learning (Hertz et al., 1991; Sutton and Barto, 1998) has been developed, inspired by reward signals found in the brain (see section 5.2.2). In reinforcement learning, the agent learns from its own experiences. This is different from supervised learning, where a supervisor provides knowledge about the situation. Reinforcement learning can include such instructions as information source but reinforcement learning typically depends to a higher degree on evaluative feedback. Supervised learning is often not applicable to situations, which depend on interactions. Reinforcement learning can include planning. Elements of reinforcement learning are (Sutton and Barto, 1998):

- Policies
  A policy can be understood as a mapping from observed states in the external world on to actions. Another name for policy is stimulus-response rule. In simple settings this can be a stationary function or look-up table. In general, the policies may be stochastic and very complex. For example, a policy for an agent can be described by the probability $\pi_t(\mathtt{State}, \mathtt{Action})$ for selecting the next action. Reinforcement learning methods act on these policies and change them, as a result of the agent's experiences, such that e.g. the total amount of accumulated reward is maximized.

- Reward function
  The reward function defines the goal for the reinforcement learning process. It

represents the worth of reaching state values for the agent by a single number, the reward. Maximzing the actual reward or the total accumulated reward is the aim of reinforcement learning. Rewards may be stochastic with respect to the performed action. In addition, the reward may not be distributed directly, but delayed, after the performed action. Actions may also effect all subsequent rewards.

- Value function
  Always selecting the action which yields maximal reward (called 'greedy algorithm') may not maximize the accumulated reward over time (see Fig. 2.24). The value function represents the long-term desirability of a state including the expectations of accessibility and rewards for the following states. Rewards are provided by the environment, the value must be estimated. The value function may include in its evaluation only a given number of future actions.

- Model (about the environment used by the agent)
  A model can be used for estimating how a behavior will interact with the environment. Models are especially interesting for planning, e.g. combinations of action, and for inferring rewards and values of states. It is possible to perform reinforcement learning without having an explicit model.

In reinforcement learning, a trade-off can arise between exploitation behavior, using learned actions to realise rewards, and exploration behavior, searching for better actions yielding more reward in the future.

Many different approaches of implementing learning strategies were proposed (Sutton and Barto, 1998; Wörgötter and Porr, 2005). For example, the 'classical' way of training is dynamical programming. Corresponding algorithms are based on evaluating the optimal policy from a value function for a (perfect) model of the environment. Dynamical programing (DP) algorithms are known to be slow and typically need a high computational effort. Monte Carlo (MC) methods, on the other hand, do not require complete knowledge of the environment. These methods are based only on experiences harvested by sequences of states, actions and rewards from interacting with the surroundings. More popular than dynamical programing is temporal-difference (TD) learning, which combines ideas from dynamical programing and Monte Carlo algorithms (Sutton and Barto, 1998). It is also possible to use expectation maximization algorithms for reinforcement learning (Dayan and Hinton, 1997).

I want to briefly introduce an interesting idea from Xie and Seung training irregular spiking neural networks by reinforcement learning strategies (Xie and Seung, 2004b). In this publication the authors show that it is possible to train the weights of a neuronal network (for solving the XOR-problem) with the reward R and the correlation between the input and output of a network. The model is based on a network of neurons that transmits information by Poissonian spike trains. The instantaneous firing rate for neuron i is then given by

$$\lambda_i(t) = f_i(I_i(t)) \,,$$

with $I_i(t)$ as the total incoming synaptic current from neuron i and $f_i(.)$ as the mean firing frequency of emitted spikes. In turn, the total input $I_i(t)$ is defined by

$$I_i(t) = \sum_j w_{i,j} h_j(t) \,,$$

where $w_{i,j}$ are weights describing the connection structure and $h_j(t)$ are the single synaptic currents. The authors model the time course of the incoming synaptic currents through the differential equation

$$h_i(t) = -\tau_{syn} \frac{d\, h_i(t)}{d\, t} + \sum_a \delta(t - T_i^a) \,.$$

τ_{syn} defines a time constant and T_i^a describes the time stamp for the a-th spike of neuron i.

Using this setup, it is possible to build a learning rule that works on the activity of the neuronal network during a time interval T. The learning rule uses the 'eligibility trace' defined as

$$e_{i,j} = \int_0^T \frac{d\, f_i(I_i(t))}{d\, t} \frac{h_i(t)}{f_i(I_i(t))} \left(\left(\sum_a \delta(t - T_i^a) \right) - f_i(I_i(t)) \right) dt$$

and changes the weights proportionally to the reward R by

$$\Delta w_{i,j} = \eta\, R\, e_{i,j} \,,$$

where η is a positive learning rate. This learning rule uses correlations between the reward and the presynaptic and postsynaptic activities to perform a stochastic gradient ascent on the expected reward. This temporally non-local learning rule can be extended to on-line learning via

$$\frac{d\, w_{i,j}}{d\, t} = \eta\, R(t)\, \bar{e}_{i,j}(t) \,,$$

while $\bar{e}_{i,j}(t)$ is defined by the differential equation

$$\tau_{et} \frac{d\, \bar{e}_{i,j}(t)}{d\, t} + \bar{e}_{i,j}(t) = \frac{d\, f_i(I_i(t))}{d\, t} \frac{h_i(t)}{f_i(I_i(t))} \left(\left(\sum_a \delta(t - T_i^a) \right) - f_i(I_i(t)) \right) \,,$$

with τ_{et} as time constant. It was shown that this type of correlation-based algorithm can be used to train neuronal networks to perform certain computations like solving the XOR problem. While the whole idea is appealing, learning large networks can be very slow and may be not applicable for this reason.

Regarding this thesis, in chapter 5 reinforcement learning will reappear. There, I will show that it is interesting to combine reinforcement learning-like strategies with neuro-prosthetic controls for compensating non-stationaries in the recordings. For this purpose we will use the idea of controlling a random walk in parameter space of the prosthetic's controller with a reward/error signal in a closed loop situation.

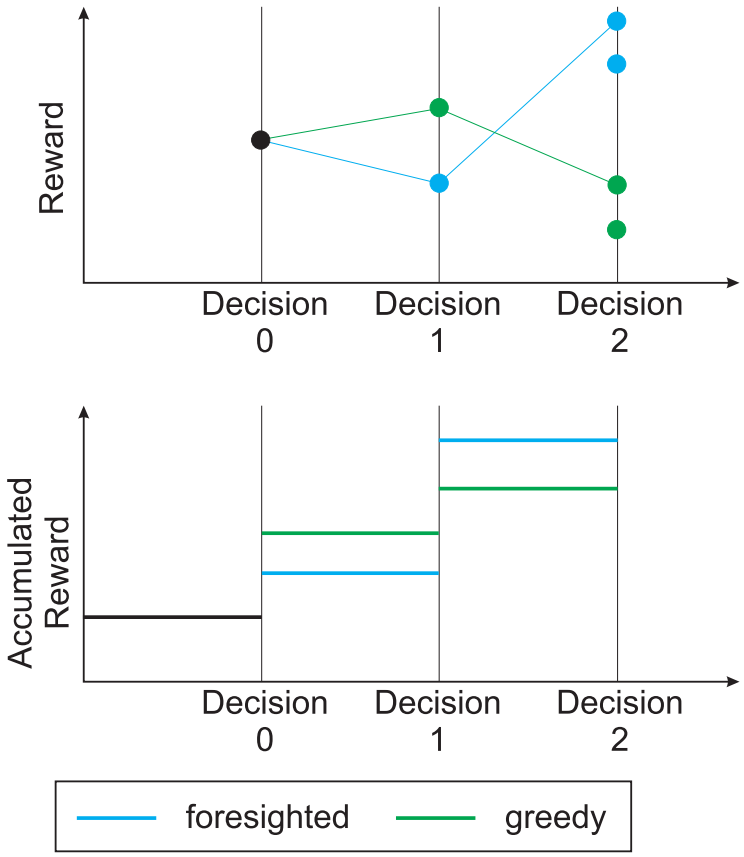


Figure 2.24: Dependency of the accumulated reward on the decision strategy. Taking always the maximal reward (greedy strategy) may not be a good decision. In the long run it may be sometimes better to use non-optimal decisions which may allow a better accumulated reward over longer time intervals. In this example (beginning at decision 0) two types of strategies are available for 'Decision 1'. The green strategy has the higher actual reward and would be selected by a greedy strategy. Selecting the green or the blue strategy, produces two new possible options. Taking these new options at 'Decision 2' into account, then the non-optimal (blue) strategy at 'Decision 1' allows to generate a higher accumulated reward in the long run.

Chapter 3

Information Processing Spike by Spike

3.1 Motivation

Perception of our environment and thus information processing in the brain is known to be fast. The work of Thorpe et al. (Thorpe et al., 1996) revealed that the brain requires only 150 ms for the decision whether a natural scene contains an animal or not. Assuming typical firing frequencies of cortical neurons ($\approx$ 15-45 Hz) and approximating the number of involved processing stages with 10, it was estimated that about one to three spikes are used per processing step and neuron for this task. A different example for the high speed of the mammalian brain was found in contour integration: macaque monkeys were trained to correctly detect aligned edge elements within a field of randomly oriented distractor edges from presentations lasting 30-50 ms before a mask appeared (Mandon and Kreiter, 2005). These experiments demonstrated that even if contour integration is performed locally within the primary visual cortex, this computation can rely only on very few spikes per neuron. In masking experiments it has been shown that grouping processes involved in Gestalt perception are capable of binding features, which were presented for only 20-30 ms (Herzog and Fahle, 2002). In addition to the aim to use low numbers of spikes, these spikes are typically emitted stochastically and show a high degree of variability in their timing (In section 2.1 we already discussed this topic in more detail). In the following, we will investigate the question of how information processing can be accomplished even if only a low number of stochastically emitted spikes are available (Ernst et al., 2007b) .

A deeper examination of this topic requires a hypothesis of how information is coded into a series of emitted action potentials. One approach is to propose a rank-order code (see section 2.1), whose appliance may fail when the spiking process is not strictly deterministic. It also typically requires a defined time interval for information processing. It is doubtful that these requirements for a rank-order code are fullfilled in higher brain

areas. If the use of stochastic spikes for transmitting information is explicitly assumed, then several coding principles allowing for a high velocity of signal transmission have been proposed. Examples include massive population rate codes with and without a balance of excitation and inhibition (Panzeri et al., 1999; Gerstner and Kistler, 2002a; Fourcaud and Brunel, 2002), or codes where the signal is coded into the variance of inputs onto a large population of neurons (Silberberg et al., 2004). In addition, the shape of neuronal tuning curves might be optimized to facilitate fast information coding. Depending on the time available for decoding, either binary tuning curves (for short times) or gradual tuning curves (for long times) were found to be optimal (Bethge et al., 2003a; Bethge et al., 2003b) (see section 2.2.3).

These approaches alone provide no conclusive explanation of fast perception for two reasons: First, the work on population coding largely ignores the problem of processing information with respect to a certain task. Second, it is doubtful if the massive amounts of neurons required for accurate information transmission and information processing are available in the brain. This is the reason why this work will be focused on a plausible neuronal on-line algorithm for small networks, which might explain fast and precise computations despite a high degree of stochasticity in the spiking process.

In section A.4 generative models were introduced as a framework that is well suited for describing probabilistic computation. In technical contexts, generative models have been used successfully, e.g. for the de-convolution of noisy, blurred images (Richardson, 1972; Lucy, 1974). Since information processing in the brain, too, seems to be stochastic in nature, generative models are also promising candidates for modeling neural computation. In many practical applications and – because firing rates are positive – possibly also in the brain, generative models are subject to non-negativity constraints on their parameters and variables. The sub-category of update algorithms for these constrained models was termed non-negative matrix factorization (NNMF) by Lee and Seung (Lee and Seung, 1999). The objective of NNMF is to iteratively factorize 'scenes' (images) $\mathbf{V}_\mu$ into a linear additive superposition $\mathbf{H}_\mu$ of elementary 'features' $\mathbf{W}$ such that $\mathbf{V}_\mu \approx \mathbf{W}\,\mathbf{H}_\mu$. For this well-defined problem, multiplicative update and learning algorithms have been derived and analyzed for different noise models and various further constraints on the parameters $\mathbf{W}$ and internal representation $\mathbf{H}_\mu$ (Lee and Seung, 2000).

Generative models can be seen as algorithmic realizations of a principle stated first by Helmholtz which hypothesizes, that the brain attains internal states that are maximally predictive of the source of sensory inputs. While this idea has often been used in phenomenological models to explain psychophysical and neurobiological evidence, it is tempting to apply it also to neural subsystems in the sense that a network should evolve into a state which is 'the best explanation' of all its inputs. This perspective on neural systems might appear rather philosophical, but it turns out to be a very useful framework for understanding neuronal computation. Generative models also cover deterministic computations by being able to predict the missing part of a typical, but incomplete input pattern from its internal representation (for an example see Fig. 3.1).

In general, non-deterministic case generative models are obviously related to Bayesian estimation. If input signals originate from different sensory modalities, a generative model can perform optimal multimodal integration, and the internal states will then represent optimal Bayesian estimates of underlying features (Deneve et al., 2001) (Fig. 3.1). Recent experimental evidence demonstrates that the brain can indeed adopt a Bayesian strategy when integrating two sources of information; in one example tactile and visual information about the width of a bar is integrated in an optimal manner (Ernst and Banks, 2002), while in another example uncertain information about the position of a virtual hand is evaluated incorporating prior knowledge about a random shift in the hand position (Körding and Wolpert, 2004). In addition, motion processing in area MT has been successfully described by a Bayesian approach, replicating the neurons' firing properties to various moving random dot patterns (Koechlin et al., 1999). Taken together generative models provide a very general and plausible framework for investigations of neural computation. Nevertheless, it is an open question if it is possible to implement a biologically plausible generative network which is able to cope with the high stochasticity of spike trains, while at the same time being as fast as possible in integrating the incoming information into a suitable internal representation of the external world.

In this chapter, a generative network which is designed to solve a typical factorization problem with non-negativity constraints will be investigated. In contrast to previous work using noise-free analog input patterns (Lee and Seung, 1999), the network receives a stochastic sequence of input spikes from a scene $\mathbf{V}_\mu$. A novel algorithm which updates the internal representation using only a single spike per iteration step will be derived. It turns out to be surprisingly simple and efficient, and can be complemented by an online learning algorithm for the model parameters that can be interpreted as a Hebbian rule with weight decay. Furthermore, modifying the basic framework of a generative model allows to implement deterministic computations of arbitrary functions, which will be shown to be closely linked to optimal multimodal integration. The dynamics and performance of this approach is demonstrated by applying it to two illustrative cases: A network trained with Boolean functions demonstrates the learnability and errorless performance for arbitrary complex functions. A small network for handwritten digit recognition demonstrates that less than one stochastic spike per input neuron can suffice to achieve impressing performance in a visual task exceeding the nearest neighbor classifier (explained in section 2.2.4). Furthermore the Boolean parity function is used to demonstrate that hierarchical combinations of the basic model can perform efficient spike based computation in a self-consistant manner. Taken together, the framework provides a novel approach for explaining fast perception with stochastic spikes in a generative network model of the brain.

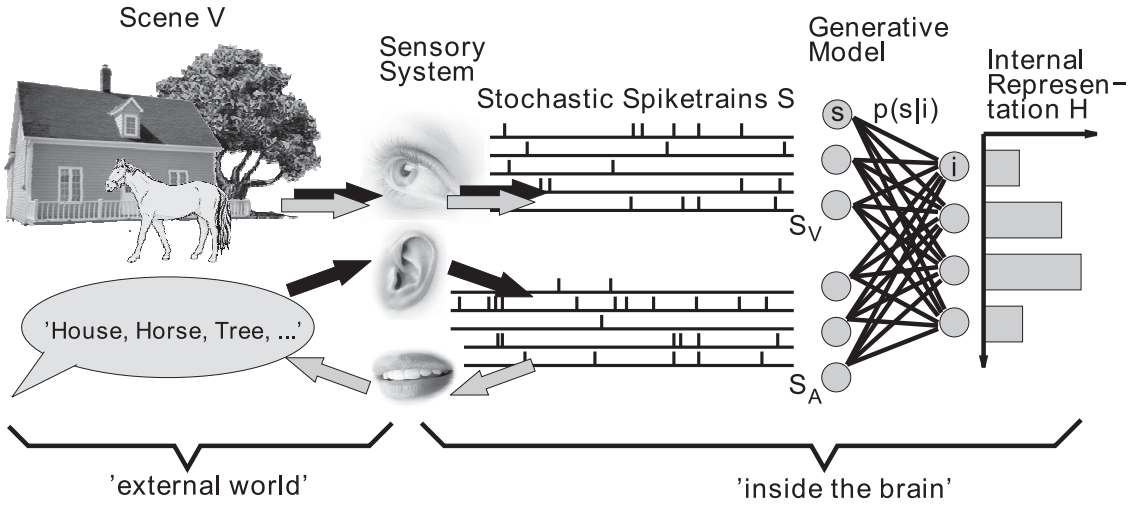


Figure 3.1: Scheme of the general framework investigated in this chapter: A scene $\mathbf{V}$ composed as a superposition of different features or objects is coded into stochastic spiketrains $\mathbf{S}$ by the sensory system. In this example the scene comprises a house, a tree, and a horse together with their spoken labels. With each incoming spike at one input channel $\mathtt{s}$, the model updates the internal variables $\mathtt{h}(\mathtt{i})$ of the hidden nodes $\mathtt{i}$. Normally, the generative network uses the likelihoods $\mathtt{p}(\mathtt{s}|\mathtt{i})$ to evolve its internal representation $\mathbf{H}$ towards 'the best explanation' which maximizes the likelihood of the observation $\mathbf{S}$. In this example, the network seeks the optimal, *combined* explanation of the spikes from the audio-visual input stream (black arrows). An alternative application of a generative model can be used to infer functional dependencies in its input. In every natural environment, certain features have a high likelihood of occurrence because they describe two different aspects of a single object. In this example, the impinging signals can be divided into two parts $\mathbf{S}_\mathcal{V}$ and $\mathbf{S}_\mathcal{A}$ which represent the input and output arguments of a function $\mathbf{S}_\mathcal{A} = \mathtt{f}(\mathbf{S}_\mathcal{V})$. In our example, the function $\mathtt{f}$ describes meaningful combinations of visual and auditory input. Presenting only the 'visual part' $\mathbf{S}_\mathcal{V}$ to an appropriate generative network will lead to an 'explanation' $\mathbf{H}^*$ (grey arrows to the right). This internal representation $\mathbf{H}^*$ in turn can generate a sharp distribution over the function values, centered at the correct 'auditory' counterpart $\mathbf{S}_\mathcal{A}$ to the input $\mathbf{S}_\mathcal{V}$. In this sense, the model computes the function $\mathtt{f}$ mapping the correct spoken label to each of the visual objects (grey arrows to the left).

3.2 A Spike-Based Generative Model

Every natural environment composed of various objects leads to an abundance of spikes which networks in the brain receive from the sensors or other brain regions. A generative network should infer (or 'recognize') typical 'objects' or 'features' which underlie (i.e. generate or 'predict') an observed spike train (Fig. 3.1). Hereby, the brain must combine prior knowledge about typical features and their frequency of occurrence as well as previously observed signals and input from other networks. This prior knowledge has to be acquired during earlier learning phases, where typical correlations in the stream of information from all sources should be extracted in order to increase recognition speed and performance in a later task.

3.2.1 Basic Model

Let us assume a structurally simple realization of a generative model. A part of the network is given by a layer of input nodes indexed by $s = 1, \ldots, S$. These input nodes are connected via weights $w_{s,i}$ to a layer of 'hidden' nodes indexed by $i = 1, \ldots, H$. The hidden nodes hold the model's internal states $h_{\mu,i}$ (Fig. 3.1). The connection weights represent information about the input statistics when typical features i are part of a scene μ. Adopting the framework chosen in (Lee and Seung, 1999), the model is supposed to explain an observed scene $\mathbf{V}_\mu$ as a linear, nonnegative superposition of the features $\mathbf{W}$ via

$$v_{\mu,s} \approx \sum_{i=1}^{H} w_{s,i} h_{\mu,i} \,. \tag{3.1}$$

At a first glance, this approach is limited because of its linearity constraint on the decomposition of an observation. Nevertheless, such a framework has been successfully applied to many problems in natural image processing with analog input patterns (Hoyer, 2004; Olshausen and Field, 1996; Lee and Seung, 1999), and allows to compare the differences between this novel spike-based stochastic model directly to known analog deterministic algorithms. It will be demonstrated that this approach is sufficiently complex to allow for computations of arbitrary deterministic functions with few stochastic neurons in the brain. Furthermore, the mathematical derivation of spike-based information processing can be applied also to non-linear generative networks using similar techniques.

3.2.2 From Poisson to Bernoulli Processes

Because of the high degree of stochasticity of spike events, networks in the brain receive information about a scene μ through stochastic spike trains impinging on the input

nodes s. In this model, we assume that spike trains are generated with rates $\mathbf{R}_\mu \propto \mathbf{V}_\mu$ from independent Poissonian point processes. In this case the number of spikes in each channel counted within a given time interval are a sufficient statistics, i.e. they provide the maximal amount of information about $\mathbf{V}_\mu$ available from spike observation.

For setting up the model, the rate $r_{\mu,s}$ is reformulated in terms of the probability $p_\mu(s)$ to receive the respective *next* spike in channel s. A straightforward calculation yields

$$\tilde{p}_\mu(s) \;=\; r_{\mu,s} / \sum_{s'=1}^{S} r_{\mu,s'} = \nu_{\mu,s} / \sum_{s'=1}^{S} \nu_{\mu,s'} \;. \tag{3.2}$$

The representation of the scene μ through these event probabilities has the convenient properties that it is invariant with respect to the total rate and ignores the true timing of the spikes. It changes the description of the spike statistics by Poissonian point processes to the more tractable Bernoulli processes and identifies each spike event with the index of the channel in which this spike occurred. These properties will strongly simplify the subsequent construction of this model.

Instead of real time, we will now use the running number of spike events, which is denoted here by t. Note that this spike-by-spike clocking implies that real time is on *average* proportional to the mean of $t/\sum_{s=1}^{S} r_{\mu,s}$. Thus global knowledge about the total input rate allows to estimate the real time elapsed during t events, but is not necessary for deriving the following algorithms.

3.2.3 From Deterministic to Probabilistic Decomposition

Similar to the transformation of $\nu_{\mu,s}$ into $\tilde{p}_\mu(s)$, the weights $w_{s,i}$ are proportional to the probability $p(s|i)$ of an input spike in channel s given that the scene μ is composed of the single feature i,

$$p(s|i) \;=\; w_{s,i} \Bigg/ \sum_{s'=1}^{S} w_{s',i} \;. \tag{3.3}$$

Reformulating the input scene $\mathbf{V}_\mu$ in terms of the (normalized) firing probabilities $\tilde{p}_\mu(s)$ as described in the last subsection transforms the factorization problem (Eq.(3.1)) into the expression

$$\tilde{p}_\mu(s) \;\approx\; \frac{\sum_{i=1}^{H} p(s|i) h_{\mu,i}}{\sum_{i=1}^{H} h_{\mu,i} \sum_{s'=1}^{S} p(s'|i)} = \sum_{i=1}^{H} p(s|i) \frac{h_{\mu,i}}{\sum_{i'=1}^{H} h_{\mu,i'}} \;. \tag{3.4}$$

It is obvious that a simple variable transformation $h_\mu(i) = h_{\mu,i} / \sum_{i'=1}^{H} h_{\mu,i'}$ reduces this problem to

$$\tilde{p}_\mu(s) \;\approx\; \sum_{i=1}^{H} p(s|i) h_\mu(i) = p_\mu(s) \;, \tag{3.5}$$

with the non-negativity and normalization constraints

$$\begin{aligned} p(s|i) > 0 \qquad & h_\mu(i) > 0 \\ \sum_{s=1}^{S} p(s|i) = 1 \qquad & \sum_{i=1}^{H} h_\mu(i) = 1 \,. \end{aligned} \tag{3.6}$$

It should be mentioned here that the new variables $h_\mu(i)$ denoting the normalized superposition coefficients have a corresponding stochastic interpretation. To explain the next action potential arriving at one input node s, the generative model assumes a doubly-stochastic process underlying spike generation: first, a specific feature, or cause i is drawn from the probability distribution $h_\mu(i)$. In a second step, a spike is drawn from the corresponding conditional probability distribution $p(s|i)$, which is observed in channel s. While at a first glance this interpretation seems rather academical, it is a natural description of the physical process of how a visual scene composed of linearly superimposed features creates photons impinging on retinal ganglion cells.

While we interpret the conditional probabilities $p(s|i)$ to be related to synaptic weights connecting input neuron s with hidden neuron i, the assumption of a continuous internal state variable to be present in each neuron somewhat goes beyond usual network models. Usually the models use the actual activities for representing the network's state. Since real neurons are spatially extended objects with many internal state variables, we assume that some of them parametrize neuronal excitability rather independently from its actual activity. The characteristic dynamics of internal state variables derived in the next section may then serve as specific predictions of this model.

3.2.4 Estimation and Learning Spike by Spike

Bayesian estimation implies that prior information about the internal state should be multiplied with the conditional probability of observing the actual external input, given the actual internal state. Therefore, generative models with iterative Bayesian algorithms appear to be suitable candidates for a comprehensive, abstract model for function and dynamics of networks in the brain (Mumford, 2002; Lee and Mumford, 2003; Rao, 2004). In the case of perception subjected to time constraints, prior knowledge about the causes underlying the previous sensory input has to be integrated online as efficiently as possible while the sequence of observations of the spikes arrive at the different input channels. As we already discussed, previous models largely neglect this realistic condition by using analog, noise-free input patterns in each iteration step instead of only a single spike.

An update algorithm for the internal states and synaptic weights can be derived by considering stochastic ensembles of exactly T incoming spikes for a scene μ, which can be written as the temporally ordered sequences of input channels at which these spikes

arrive,

$$\mathbf{S}^{T}{}_{\mu} \;=\; \left\{ s_{\mu}^{t} \right\}^{t=1,\ldots,T} . \tag{3.7}$$

We now seek an iterative algorithm which with every spike encountered tends to maximize the likelihood

$$P\left(\left\{ \mathbf{S}^{T}{}_{\mu} \right\}^{\mu} \middle| \, \{ h_{\mu}(i) \}^{\mu,i}, \{ p(s|i) \}^{s,i} \right) \tag{3.8}$$

of observing the ensemble of sequences $\left\{ \mathbf{S}^{T}{}_{\mu} \right\}^{\mu=1,\ldots,M}$ over the space of all internal states $\{ h_{\mu}(i) \}^{\mu,i}$ and model parameters $\{ p(s|i) \}^{s,i}$. With

$$\hat{p}_{\mu}(s) \;=\; 1/T \sum_{t=1}^{T} \delta_{s,s_{\mu}^{t}} \tag{3.9}$$

denoting the relative number of spikes in channel s in one observation sequence μ, this likelihood is given by

$$P \;=\; \prod_{\mu=1}^{M} \left(\frac{T!}{\prod_{s=1}^{S} (T\hat{p}_{\mu}(s))\,!} \right) \prod_{s=1}^{S} \left(p_{\mu}(s)^{T\hat{p}_{\mu}(s)} \right) . \tag{3.10}$$

$p_{\mu}(s)$ is defined as in Eq.(3.5) [1]. The standard approach to this optimization problem is to minimize the negative logarithm of the likelihood P,

$$L = -1/T \log P = C - \sum_{s=1}^{S} \hat{p}_{\mu}(s) \log p_{\mu}(s) . \tag{3.11}$$

under the constraints mentioned in Eq.(3.6). C is a constant which does not depend on the optimization variables. If there is no prior knowledge about the typical features of which scenes μ are composed, the generative network will have to do both, estimate $h_{\mu}(i)$, and learn the likeliest set of features $p(s|i)$ in order to explain the ensemble of observed spike sequences. As soon as suitable $p(s|i)$ have been found and can be held constant, minimizing L turns into a convex optimization problem for the $h_{\mu}(i)$ only.

Unfortunately, serial algorithms updating their representations with each incoming spike from one input pattern can not be deduced directly from known update equations which work on $\mu = 1, \ldots, M$ full observation sequences $\mathbf{S}^{T}{}_{\mu}$ in parallel (Lee and Seung, 1999). The reasons for this will become clear during the following derivations, in which we will seek for a rapid update rule for the $h_{\mu}(i)$'s and $p(s,i)$'s minimizing L, which are based on a single spike observation s_{μ}^{t} in each iteration step t.

[1] Note that $p(s)$, $\hat{p}(s)$, and $\tilde{p}(s)$ all denote different variables. While $\tilde{p}(s)$ describes the real, but unknown input statistics of a scene, $\hat{p}(s)$ denotes a particular stochastic realization of this scene measured as the relative spike count at the input nodes s. The approximation of $\hat{p}(s)$ by a linear superposition of elementary features is then denoted by $p(s)$.

In particular, we will aim at a multiplicative update rule, because in many cases (Lanteri et al., 2002), multiplicative algorithms are known to converge very fast when compared to simple additive gradient methods. Unfortunately, it was not possible to find the fastest algorithm possible by deriving it directly from L and it seems that this can not be done for the general case.

For devising a multiplicative rule by means of a gradient descent, the derivatives $-\nabla L$ of the log-likelihood have to be computed first,

$$-\frac{\partial L}{\partial h_\mu(i)} = \sum_{s=1}^{S} \frac{\hat{p}_\mu(s)}{p_\mu(s)} p(s|i) \tag{3.12}$$

$$-\frac{\partial L}{\partial p(s|i)} = \sum_{\mu=1}^{M} \sum_{s'=1}^{S} \hat{p}_\mu(s') \frac{\delta_{s',s}}{p_\mu(s')} h_\mu(i) = \sum_{\mu=1}^{M} \frac{\hat{p}_\mu(s)}{p_\mu(s)} h_\mu(i) \; . \tag{3.13}$$

For unconstrained problems, $\mathbf{D}\nabla L$ is a valid descent direction if $\mathbf{D}$ is a positive definite matrix (Bertsekas, 1995). With z denoting the iteration step, the gradient descent in its most general form reads

$$\tilde{h}_\mu^{z+1}(i) = h_\mu^z(i) - \gamma_h \left[\mathbf{D}_h^z \nabla_{h_\mu} L^z \right]_i \tag{3.14}$$

$$\tilde{p}^{z+1}(s|i) = p^z(s|i) - \gamma_p \left[\mathbf{D}_p^z \nabla_p L^z \right]_{s\,i} \; . \tag{3.15}$$

In this case, let us choose $\mathbf{D}_h^z$ and $\mathbf{D}_p^z$ as diagonal matrices with entries $[\mathbf{D}_h^z]_{i,i} = h^z(i)$ and $\left[\mathbf{D}_p^z\right]_{s\,i,s\,i} = p^z(s|i)$. $\mathbf{D}_h^z$ and $\mathbf{D}_p^z$ explicitly depend on the update step z. Furthermore, an update constants γ_h ('estimation rate') and γ_p ('learning rate') are introduced. Since the update is always positive, we have to satisfy the normalization constraints only, which is done by a post-update normalization scheme

$$h_\mu^{z+1}(i) = \frac{\tilde{h}_\mu^{z+1}(i)}{\sum_{j=1}^{H} \tilde{h}_\mu^{z+1}(j)} \tag{3.16}$$

$$p^{z+1}(s|i) = \frac{\tilde{p}^{z+1}(s|i)}{\sum_{s'=1}^{S} \tilde{p}^{z+1}(s'|i)} \; . \tag{3.17}$$

Combining update and normalization, and substituting $\epsilon = \gamma_h/(1+\gamma_h)$ and $\gamma = \gamma_p/(1+\gamma_p)$ yields the equations

$$h_\mu^{z+1}(i) = h_\mu^z(i) \left((1-\epsilon) + \epsilon \sum_{s=1}^{S} \frac{\hat{p}_\mu(s)}{p_\mu^z(s)} p(s|i) \right) \tag{3.18}$$

$$p^{z+1}(s|i) = \frac{p^z(s|i) \left((1-\gamma) + \gamma \sum_{\mu=1}^{M} \frac{\hat{p}_\mu(s)}{p_\mu^z(s)} h_\mu(i) \right)}{(1-\gamma) + \gamma \sum_{\mu=1}^{M} h_\mu(i) \sum_{s'=1}^{S} \frac{\hat{p}_\mu(s')}{p_\mu^z(s')} p^z(s'|i)} \; . \tag{3.19}$$

These equations still use the full sequences $\mathbf{S}^T{}_\mu$ for all $\mu = 1, ..., M$ patterns to do one update step from z to $z+1$. If a given collection of T spikes is considered a fixed input

vector, Eq.(3.18) and Eq.(3.19) have the same fixed points as the corresponding NNMF algorithms (Lee and Seung, 2000) which can be derived using estimation-maximization (EM), and for $\epsilon \to 1$ and $\gamma \to 1$ they become equivalent. However, at one instant in real time, only one scene μ is presented and only one or very few spikes are observed. In order to convert Eqs.(3.18) and (3.19) to an on-line update rule, we restrict the algorithms to one pattern μ at a time. Furthermore, let us assume that from this pattern, only one spike observed at time t enters the update equations for h and/or p. This procedure is equivalent to dropping the summations over μ, replacing z by t, and replacing the full pattern $\hat{p}_\mu(s)$ by δ_{s,s^t}. The corresponding update equations ('on-line learning' and 'on-line estimation') resulting from Eqs.(3.18) and (3.19) then read

$$h^{t+1}(i) = h^t(i)\left((1-\epsilon) + \epsilon\frac{p(s^t|i)}{p^t(s^t)}\right) \tag{3.20}$$

$$p^{t+1}(s|i) = p^t(s|i)\left(1 + \frac{\gamma h(i)}{(1-\gamma)p^t(s^t) + \gamma h(i) p^t(s^t|i)}\left(\delta_{s,s^t} - p^t(s^t|i)\right)\right) \tag{3.21}$$

In the following, the algorithm defined by equation Eq.(3.20) will be termed 'spike-by-spike on-line estimation', while Eq.(3.21) will be referred to as 'spike-by-spike on-line learning'.

Eq.(3.20) and Eq.(3.21) are on-line algorithms, in contrast to Eq.(3.18) and Eq.(3.19) and the NNMF learning rules. Their update relies on only one spike, but it is straightforward to construct versions which take into account several spikes for each iteration step (this is simply done by choosing T in the Eq.(3.9) for $\hat{p}_\mu$ equal to the number of spikes which should be considered in one step). Per construction, the on-line versions have finite memory, and also for stationary input patterns, fluctuations of the hidden states will in general remain finite due to the stochastic drive. Therefore both algorithms can at most achieve approximate maximization of the likelihood for the full input spike sequence. In the next section it will be demonstrated by means of simple examples that with suitably chosen update parameters $\epsilon \in [0, 1]$ and $\gamma \in [0, 1]$, this approximate convergence is sufficient to achieve high computational performance.

3.2.5 Simplified algorithm with batch learning

The on-line learning equation can be transformed to reduce computational complexity. In this case, learning of the $p(s|i)$ is assumed to take place on a much larger time scale than the estimation process of the internal state h.

In reality, learning is a slow process extending over the (repeated) presentation of many training scenes μ. Eq.(3.21) hereby imposes a high computational load during on-line learning. This load can be significantly reduced by assuming that on a fast timescale, the hidden representations h^t_μ for each μ, $\mu = 1, \ldots, M$, are computed with the on-line estimation algorithm Eq.(3.20) for T spikes and time steps each. After these computations, now on a much slower timescale, the averaged hidden activities $<$

$h>^{\Delta}_{\mu}= 1/\Delta \sum_{t=T-\Delta+1}^{T} h^t_\mu$ are used in parallel for a single update step of the conditional probabilities $p(s|i)$ according to Eq.(3.19). A suitable multiplicative algorithm for this update has been proposed by Seung et al. (Lee and Seung, 1999) which reads

$$\tilde{p}^z(s|i) = p^z(s|i) \sum_{\mu=1}^{M} \left(\hat{p}_\mu(s) <h>^{\Delta}_{\mu} (i) \Big/ \sum_{i'=1}^{H} p^z(s|i') <h>^{\Delta}_{\mu} (i') \right) \quad (3.22)$$

$$p^{z+1}(s|i) = \tilde{p}^z(s|i) \Big/ \sum_{s'=1}^{S} \tilde{p}^z(s'|i) \ . \quad (3.23)$$

The parameter $z = 1, \ldots, Z$ counts the learning steps and is identical to the update step used in Eqs.(3.18) and (3.19). With M different scenes (stimuli) in total, one update of the $p(s|i)$ is made each TM spikes. Eqs.(3.22) and (3.23) together with Eq.(3.20) define the *Spike-by-Spike batch learning algorithm (short: SbS-batch)*. Note that Eqs.(3.22) and (3.23) can be derived directly from Eq.(3.19) by substituting the average $<h>$ for h, and by choosing the maximum update constant $\gamma \to 1$.

3.3 Results

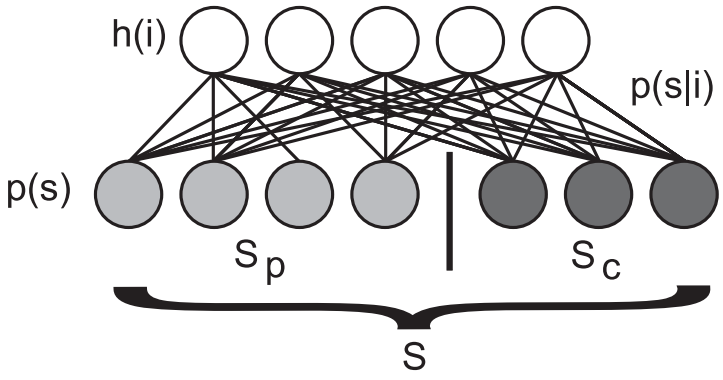


Figure 3.2: Spike-by-spike network for training a classification task. The training patterns with S_p components, together with their correct classification into one of S_c classes, are presented as randomly drawn spike trains to $S = S_p + S_c$ input nodes during learning. The network thereby finds a suitable representation $p(s|i)$ of the input ensemble, and estimates an internal state $h(i)$ for each input pattern according to either the SbS on-line or the SbS batch algorithm.

In this section, it will be demonstrated that the SbS-online and SbS-batch algorithms optimize suitable representations of scene ensembles by predicting the spikes arriving at the input nodes. For learning Boolean functions, the scene ensemble will consist of all input and output bit patterns of the respective function. For learning handwritten

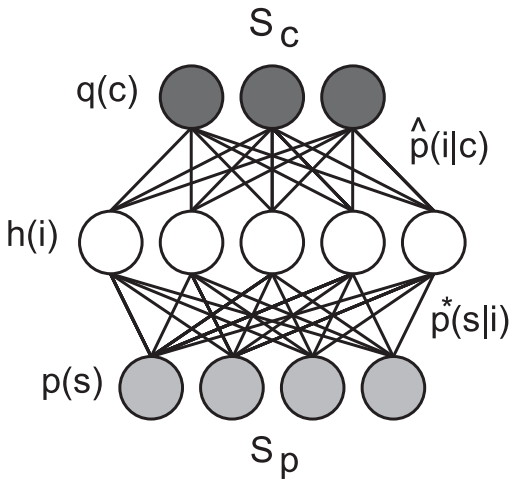


Figure 3.3: Spike-by-Spike network for classification of patterns. The test patterns are presented to the S_p input nodes, and an appropriate internal state $h(i)$ 'explaining' the actual pattern is estimated by the SbS algorithm. From the internal state, a classification $q(c)$ is inferred and compared to the correct classification.

digits, the scene ensemble will be represented by the pixel patterns of the digit images together with their correct classifications from a training data set.

The performance of the learned representation will be evaluated in a subsequent computation or classification task on a test data set. Results will be compared to a simple nearest-neighbor classifier (abbreviated by *NN classifier*, for details see section 2.2.4) and to the NNMF-algorithm by Seung et. al (Lee and Seung, 1999).

At the end of this section, the SbS batch-learning algorithm will be applied to natural stimuli.

3.3.1 A Simple Example

First, let us illustrate characteristic properties of the algorithm for spike-by-spike update of internal states and the resulting coding in the network using a minimal model.

For this purpose we will consider a highly over-complete situation where $S = 2$ input neurons project to $H = 100$ hidden neurons. The optimization problem is therefore under-constrained, and accordingly, the Seung model converges to different internal states $\mathbf{H}$ for different initial conditions. It appears that the algorithm prefers mixtures of almost all available features $\mathbf{W}$ as shown in Fig. 3.4(a).

In contrast, the sparse solution $\mathbf{H}$ found by the on-line update algorithm turns out to be unique for almost all initial conditions (Fig. 3.4(b)). The degree of sparseness is tightly

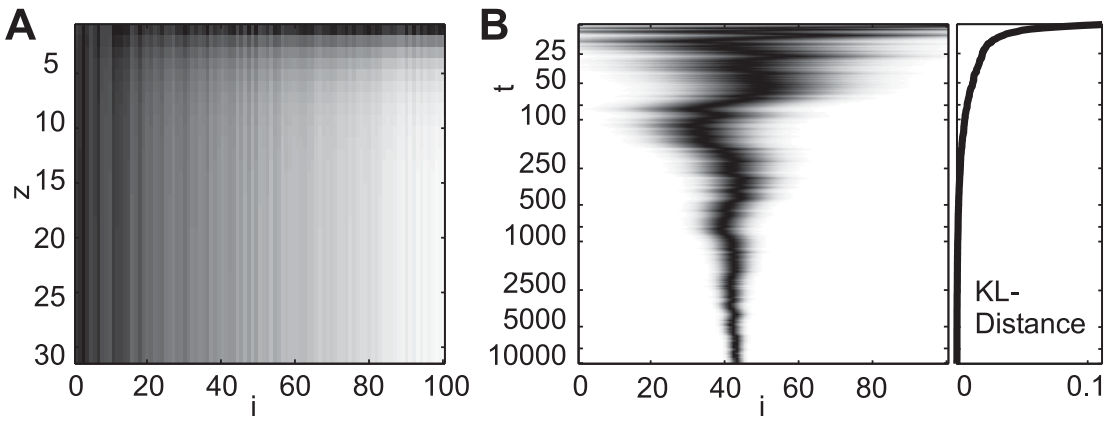


Figure 3.4: (a) Dynamics of internal representation $\mathbf{H}$ in dependence on the iteration step z for the NNMF-model by Seung et al., and (b) for the spike-by-spike network in dependence on the number t of incoming spikes. The networks comprise $S = 2$ input nodes and $H = 100$ hidden nodes, with the weights chosen as $p(1|i) = (i + 1)/H$, and $p(2|i) = 1 - p(1|i)$. The initial state was random, but identical for both models, and the input was given by the vector $\mathbf{V} = \{0.42, 0.58\}$. While the NNMF-model converges to a broad mixture of all feature vectors $p(s|i)$, the SbS-network converges to a sparse state where the internal representation peaks around the feature vector $p(s|42) = \{0.42, 0.58\}$ and accurately reflects the input rate distribution. The update constant ϵ was 0.5. In addition we show in the right plot of (b), that the Kullback-Leibler (KL)-divergence, averaged over 1000 runs with different $\mathbf{V}$, decreases with iteration time, thus demonstrating convergence of our algorithm.

linked to the parameter ϵ: the larger ϵ, the sparser becomes the internal representation. With $\epsilon = 1$, only one hidden state will be active which has the maximum likelihood to explain the input sequence. In the example shown in Fig. 3.4(b), the high value of ϵ causes the representation to concentrate on the feature vector which is in this case the 'most natural' explanation of the input spike statistics.

3.3.2 Pre-Processing, Training, and Classification

The simulations are carried out in two stages which consist of a training and test run, respectively. Before presenting the results on learning and performance for specific computations, let us briefly introduce constructing, pre-processing and partitioning the data into training and test sets.

Training. For learning a classification or computation task, each of the M_{tr} training scenes $\mathbf{V}_\mu^{tr}$ comprises a pattern or image together with its correct classification into one of S_c classes. First, the average is removed from each pattern (or image). Since this transformation leads to negative pattern values, and because firing rates are always

positive, it is necessary to distribute the corresponding values to twice the number of channels. Hereby positive values are directly assigned to the odd channels, while the absolute values of the negative entries are assigned to the even channels. Second, the classification is represented by a vector with S_c entries with only one non-zero entry at the position corresponding to the correct class. Finally, this vector and the processed pattern are weighted and combined to form a single, normalized vector $\mathbf{V}_\mu^{tr}$ from which the input spikes are drawn. The mathematical details of this procedure are described in section B.1 and B.2.

On $\mathbf{V}_\mu^{tr}$, the network now learns a suitable representation including the correct match between patterns and classes in an unsupervised manner. This procedure is schematically explained in Fig. 3.1, and employs the network structure sketched in Fig. 3.2.

Test. For testing the trained generative network, the M_{ts} test scenes $\mathbf{V}_\mu^{ts}$ consist only of the transformed, positive input patterns without the class information (details described in section B.3, see also Fig. 3.3). The correct class c_μ^{ts} is compared to the prediction $\hat{c}_\mu$ being inferred from the internal states of the generative model. This method is described in Fig. 3.1 as inference procedure with partial sensory input and employs the network structure sketched in Fig. 3.3.

3.3.3 Boolean functions

For a proof of principle we used the problem of learning and computing Boolean functions. Realizing arbitrary Boolean functions in networks that receive only stochastic input is particularly nasty. The spike noise needs to be cancelled, because Boolean functions are non-smooth in the sense that flipping of one input bit can result in a complete change of the output values. If a network is capable of performing the required operation, this has interesting consequences for multimodal integration and attention, which will be explained in the section 3.4. In any case, efficient computation of Boolean functions would demonstrate that starting from the well-known XOR problem, in principle any possible computation, can be rapidly performed in small networks with stochastic spikes.

For the simulations, Boolean functions of $N = 5$ input bits mapped to one output bit were selected randomly, leading to $M_{tr} = M_{ts} = 32 = 2^N$ input-output patterns for one specific function if one unit is identified with each input-output relation [2]. Consequently, a number of $H = 2^N$ hidden units should suffice to represent one Boolean function of N bits. In this case, each weight vector $p(s|i)$ comprising the conditional probabilities for a fixed i matches one of the M_{ts} test patterns. Consequently, for

[2]The same set of patterns were used both for training and classification. Note, however, that the realization of the patterns was different in both runs because the patterns are represented by a finite number of input spikes, which are randomly drawn in each repetition of a presentation.

each of the input patterns, only one of the hidden units h should be active (winner-takes-all network). In contrast, it might be possible that less than 2^N hidden nodes in the network are sufficient to perfectly represent a specific Boolean function. In this case, each input pattern can be represented by a mixture of active hidden nodes (soft competition network). As we considered only one output bit, there are only $S_c = 2$ different classes to which an input pattern can be assigned. For details of the simulation, see section B.4.

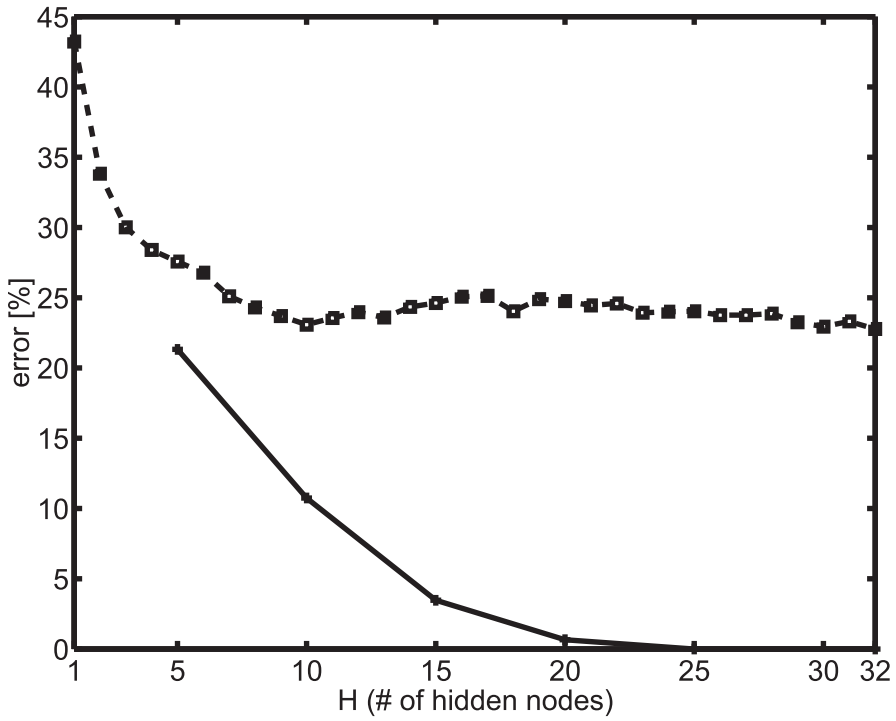


Figure 3.5: Mean classification error for Boolean functions of 5 input and 1 output bits, in dependence on the number of nodes H in the hidden layer of the network. The SbS-online learning (solid line) clearly outperforms the noiseless NNMF-algorithm (dashed-dotted line), which barely approaches 20% error.

Performance of the network. Fig. 3.5 shows the mean classification error of different algorithms in dependence on the number of hidden units H. The Spike-by-Spike algorithms (with on-line learning) performs considerably better than the NNMF-algorithm. The NNMF-algorithm tends to explain the input as a very broad mixture of many basic features as in the other example shown in Fig. 3.4, while the Spike-by-Spike algorithm concentrates on the most predictive features. With SbS the error drops to approximately zero as soon as more than $H = 25$ hidden units are in the network. In these cases, the input patterns are reconstructed using a mixture of two or more features represented by the activated hidden nodes. Fig. 3.6 shows how the classification error decreases with the number of incoming spikes. Here, SbS leads to a classification error which reaches zero after only about three spikes per input node. In comparison,

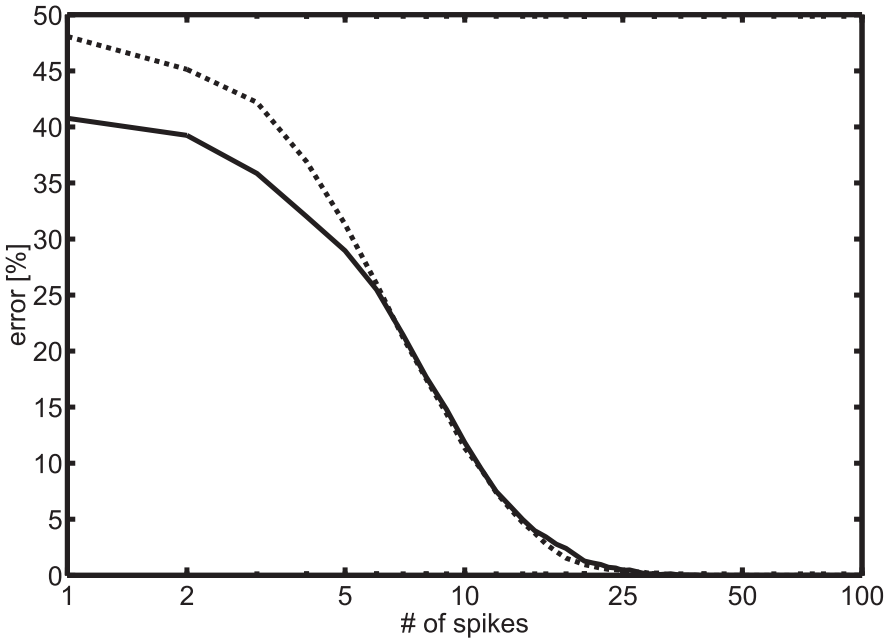


Figure 3.6: Error rate e of SbS on-line algorithm (solid line) for the task introduced in Fig. 3.5, compared to the error of a NN-classifier (dotted line). The SbS on-line algorithm is as fast and as accurate as the NN-classifier which in this case represents an optimal look-up table of function values. Note that a mean of 3 spikes per input neuron is sufficient to achieve perfect classification.

the NN-classifier is also astonishingly fast and closely approaches the error curve of the SbS algorithm. In this specific example, the complexity of the 'codebooks' of the SbS algorithm and NN-classifier are comparable.

3.3.4 Handwritten Digits

As a third example we will select the problem of handwritten digit recognition in order to demonstrate that real-world estimation problems can be solved efficiently and very fast by estimation with single spikes. We will use the US postal service data set which serves as a benchmark for classification algorithms and allows comparison with human performance. For parameters of our simulations, see section B.5.

Performance of the network. Fig. 3.9 shows the classification error of the network versus the mean number of spikes per input node. It turns out that the NN-classifier is remarkably good, reaching 6.1 percent error on the test set after about a mean of one spike per input node. This performance comes, however, at the price of using the full $M_{\mathrm{tr}} = 7291$ training patterns for classification. In contrast, a network with only $H = 500$ nodes optimized on the training set with the batch algorithm with one spike

per input neuron has a superior performance. Using an annealing strategy an increase in classification speed and quality can be obtained by adapting ϵ ('cooling') during learning and estimation of the internal states: the corresponding curve (dotted line in Fig. 3.9) shows that with only $H = 100$ the classification is nearly as fast and good as the NN-classifier with its 7291 stored patterns. Fig. 3.8 shows the complete weight set for the case SbS with $H = 500$ and $\epsilon = 0.1$. About half of the weights consist of templates which, however, become combined with particular feature weights during classification runs and are required to achieve the classification performance shown. With the same parameters, the classification error of the network in dependence of H is shown in Fig. 3.10. Using $H = 25$ hidden nodes is already sufficient to obtain an error which is only twice as high as the error of the NN-classifier using the full set of 7291 pattern templates. In addition, the error curve shows no signatures of overfitting as it decreases monotonically even for very large networks with up to $H = 7290$ nodes.

Robustness against noise. While the algorithm by construction is robust against noise in the spiking process, the question is whether this remains true for other types of noise. To address this issue, the digit patterns were subjected to two types of perturbations: first, an increasing number of vertical or horizontal lines were occluded in the digit patterns (starting with the two center lines), and second, an increasing amount of noise to each pixel in a pattern was added. Fig. 3.11(a) shows the classification performance in dependence on the number of covered lines. Up to a number of 6 lines, classification error stays below 20 percent. Fig. 3.11(b) demonstrates that pixel noise must increase to a considerable amount $\eta = 0.35$ in order to raise the error above 20 percent.

3.3.5 Hierarchical Networks

Let us demonstrate that this basic network can be combined by using the hidden variables $h(i)$ as probabilities for sending out *next* spikes, i.e. we turn them into input neurons of subsequent networks. As a simple example a hierarchical network will be presented which solves the Boolean function of the parity problem which requires the output to be zero whenever an unequal number of input neurons are active and to be one otherwise. While this problem can be solved trivially with $H = 2^N$ hidden neurons, a hierarchical arrangement of simple SbS-networks can drastically reduce the size of the network. This network proves that the approach can be generalized in a self-consistent manner, using the output of one simple module as a meaningful input to another simple module. Therefore, one can conclude that SbS-networks can be combined for iterative computations. Together with the results on Boolean functions, spike-by-spike networks are thus able to compute arbitrary complex functions with any required precision in a finite amount of time.

The parity function network. The binary parity function is 1 if the number of input bits being 1 is an odd number, and 0 otherwise. For an arbitrary number of input bits, the parity function is easily computed by a hierarchical tree of XOR-modules. For the

simulations, $N = 16$ input bits are chosen, thus having $S_p = 2N = 32$ on-off input channels, and two output classification nodes (Fig. 3.12).

The full network consists of 15 spike-by-spike XOR-modules, whose weights were set up manually (i.e. a value of 0.5 was assigned to all connections drawn in Fig. 3.12). In every single module, input/output nodes are shown as white discs, and hidden nodes as gray discs. The hidden node activities $h(i)$ and output variables $\hat{p}(k, n)$ of one XOR-module are updated by Eqs.(3.20) and (B.8) with each incoming spike. Spikes are elicited by either the external input, or by the activities $\hat{p}(k, n)$ of an internal output node n, which is at the same time the input node for the next XOR-module. With probability $p_I = 0.05$, a spike is drawn according to the probability distribution of the actual input bit pattern, and sent to one of the 8 XOR-modules in the input layer. With probability $p_H = \hat{p}(k, n)(1 - p_I)/14$, a spike is drawn from one of the output nodes of the 14 XOR-modules in the lower layers of the hierarchy, and passed on to the next XOR-module.

If the network performs the intended computation perfectly, by construction each pair of input/output nodes should display a well-defined activity for each input pattern. As an example, consider the two output nodes marked by an arrow in Fig. 3.12. If the number of bits of value 1 within the first 8 bits of the input pattern is an even number, the output variable of the left node should take a lower value (ideally 0) than the output variable of the right node (ideally 1), thus computing the parity function for the first 8 input bits. Fig. 3.13 displays the corresponding errors of the network for the output layer and for the input/output nodes in the hidden layers in dependence of the number of spikes arriving in the input layer. All errors go down to zero level, but on different time scales. Clearly, it can be seen that the errors in the lower layers of the hierarchy decrease earlier than errors in the higher layers. This behavior is to be expected from the structure of the network: only when the input from the $m - 1$-th layer is correct, can the output error of the m-th layer decrease to 0.

3.3.6 Steps toward biological plausibility

In this chapter, an algorithm was presented which is capable to perform information processing on the basis of single spikes. It is still unclear, how this algorithm can be implemented by biologically plausible mechanisms. In the following, we will see that spike-by-spike information processing can be a part of a neuronal networks consisting of leaky integrate-and-fire neurons (see Fig. 3.14). In this type of network, the spike-by-spike algorithm can be interpreted as one compartment of a multi-compartment neuron model. It will be shown that this combination allows a self-consistent way of information transmission. The analog h-values are converted into spikes. This allows to construct multi-layer networks where the information is only transported by single spikes from one neuron to another. The simulations depicted in Fig. 3.15 and Fig. 3.16 show that this is possible. Both figures were generated by simulating a network

built with leaky integrated-and-fire neurons, described by the equation:

$$\tau_m \dot{V}_i(t) = -V_i(t) + R_I I_i(t) + R_\eta \eta_i \tag{3.24}$$

η emulates noise on the membrane potential and is drawn from a normal distribution with mean and variance 1. The two resistances were selected with $R_\eta = 0.04$ and $R_I = 0.06$. The time constant of the relaxation of the membrane potential was set to $\tau_m = 18ms$. A spike was be generated if $V_i(t)$ was larger than the threshold $\vartheta = 1.0$. After emitting a spike, the membrane potential was reset to $V_{Reset} = 0$.

The spike-by-spike algorithm is used to model the input current $I_i(t)$. $I_i(t)$ is composed of spikes released at the times T_S. These spikes are represented by excitatory post synaptic potentials (EPSPs), that are modelled by a simple exponential decay with time-constant $\tau_{EPSP} = 2ms$. Before summing up the single EPSPs to $I_i(t)$, the EPSPs are weighted by the $h(i)$-values from the spike-by-spike dynamics such that $I_i(t)$ is described by

$$I_i(t) = \sum_s \exp\left(-\frac{t - T_s}{\tau_{EPSP}}\right) \Theta(t - T_s)\, h^{T_s}(i)\,. \tag{3.25}$$

The input spikes for generating $I_i(t)$ were drawn from a Poisson distribution, using the input values from the actual task.

After a spike is released from one of the integrate-and-fire neurons, it is multiplied by weight values and then accumulated in the layer of output neurons. The output neurons with the highest accumulated value is used as result of the information processing process. Fig. 3.15 shows (with $\epsilon_{XOR} = 0.5$ for the h-dynamics) that it is possible to build functioning XOR-gates with such a network configuration. For Fig. 3.16, the same framework (with $\epsilon_{USPS} = 0.1$) was used to classify handwritten digits from the USPS database. The curve in Fig. 3.16 shows the mean performance of this classification in dependency of processing time and it also shows that this type of information processing scheme works still well. Both simulations were performed with 0.1 ms for each simulation step.

Another aspect of the spike-by-spike algorithm, which makes problems with the biological plausibility, is the denominator in the equation for the h-dynamics. It is an unsolved question how the necessary information about the h-values is exchanged for the update procedure. This may be regulated by extracellular GABA (γ - aminobutyric acid). Another approach is to use populations of neurons for representing the product $h_i \cdot p(s^t \mid i)$ in terms of their spiking activity. For this setup, the h-value dynamics is calculated by

$$h^{t+1}(i) \;=\; h^t(i)\left((1-\epsilon) + \epsilon\Xi\frac{p(s^t|i)}{\sum_j \varpi_j^t}\right)\,. \tag{3.26}$$

$h_i \cdot p(s^t \mid i)$ is used as mean value for a Poissonian process, where ϖ_i^t denotes the noisy representation of $h_i \cdot p(s^t \mid i)$. Using a Poissonian random process made it necessary

to replace the $\sum_i h(i) = 1$ condition by $\sum_i h(i) = \Xi$. This new parameter allows to control the mean number of spikes representing $h_i \cdot p(s^t \mid i)$ and thus to control the precision and noise level of the noisy nominator. Fig. 3.17 shows the performance of classifying handwritten digits from the USPS database in dependency of the Ξ-value. Ξ can be interpreted as parameter for describing the size of the population of Poissonian neurons. With increasing Ξ, the maximal possible performance increases. Furthermore, Fig. 3.17 shows that with only a few spikes representing the nominator for each transmission, the performance approaches the error for the non-noisy h-dynamics.

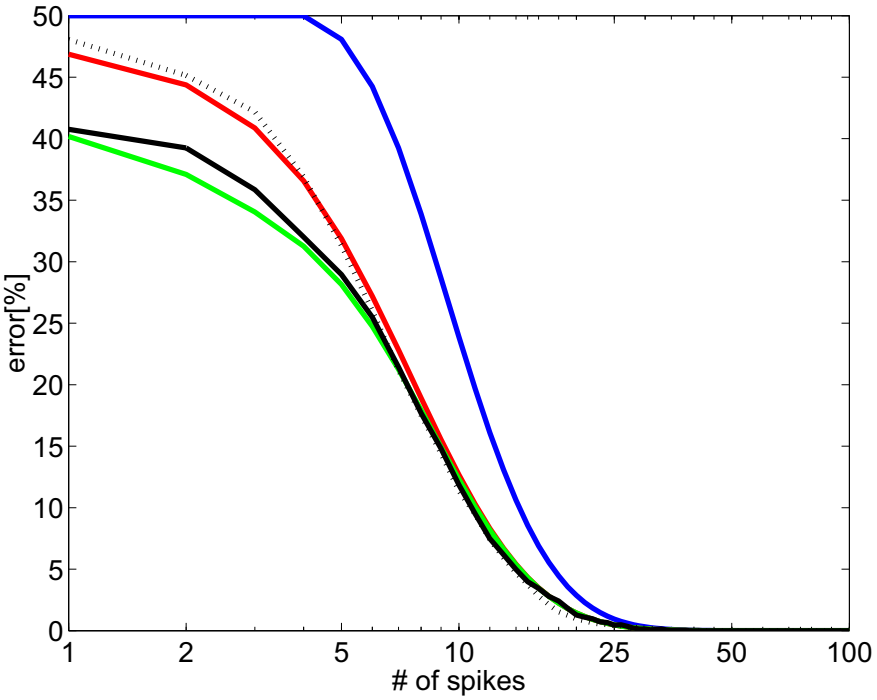


Figure 3.7: Error rate e of SbS on-line algorithm (solid line) and the NN-classifier (dotted line) from Fig. 3.6 in comparison with the performance of three types of estimators. The blue curve represents an estimator which draws randomly an output value as long as all input neurons fired at least once. The red line was generated by an sub-optimal estimator which complements missing input channels randomly as long as all input neurons fired at least once. The used realisation of the NN-classifier finds the nearest memorized sample. If more than one sample with similar distance are found, then one of these samples is chosen randomly. This strategy is similar to the red estimator. The last (green) curve is the error produced by an optimal estimator. This estimator also complements missing input channels as long as all input neurons fired at least once. But it uses the most likely input bit configuration of the actual Boolean function set for the missing bits. Similar can be expected from the SbS on-line algorithm.

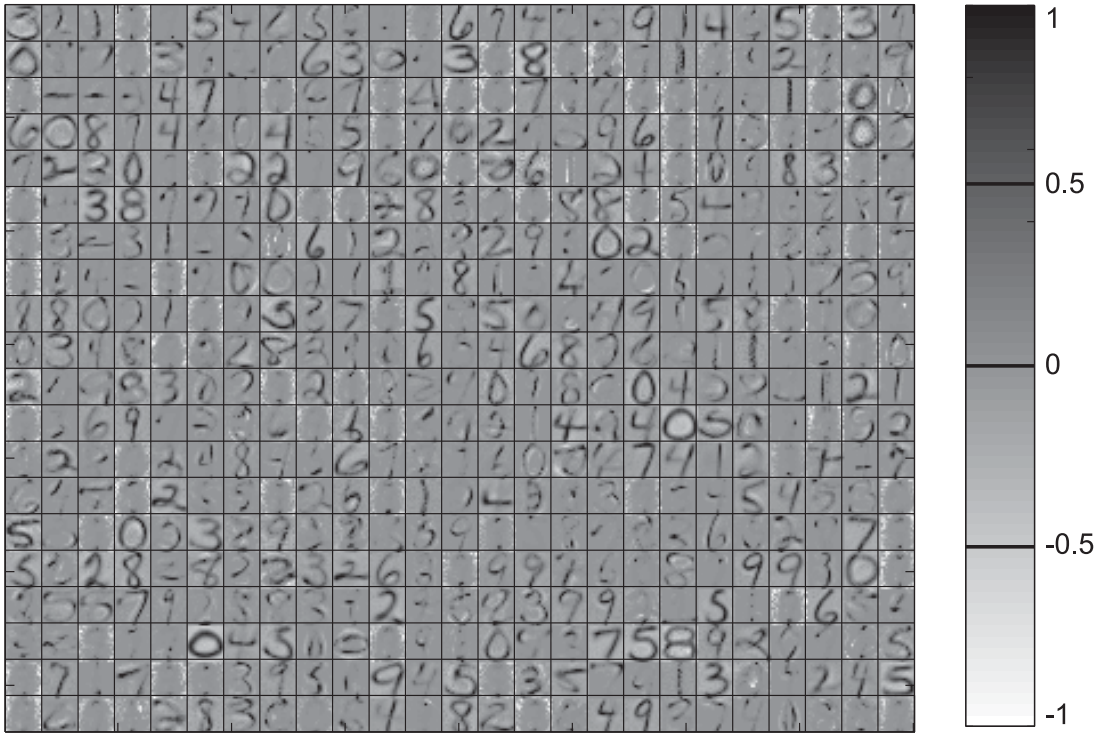


Figure 3.8: Conditional probabilities (weight vectors) $p^*(s|i)$ for the SbS batch learning algorithm using $H = 500$ hidden nodes. Vectors i for even and odd input nodes s are combined and individually re-scaled to a grey value $g_i(s) \in [-1, 1]$, and then displayed in a 16×16 pixel raster. The $g_i(s')$ were computed as $g_i(s') = \tilde{g}_i(s')/\max\{|\tilde{g}_i(s')|\}$ with $\tilde{g}_i(s') = p^*(2s'-1|i) - p^*(2s'|i)$. Parameters for batch-learning were chosen as described in the section B.5.

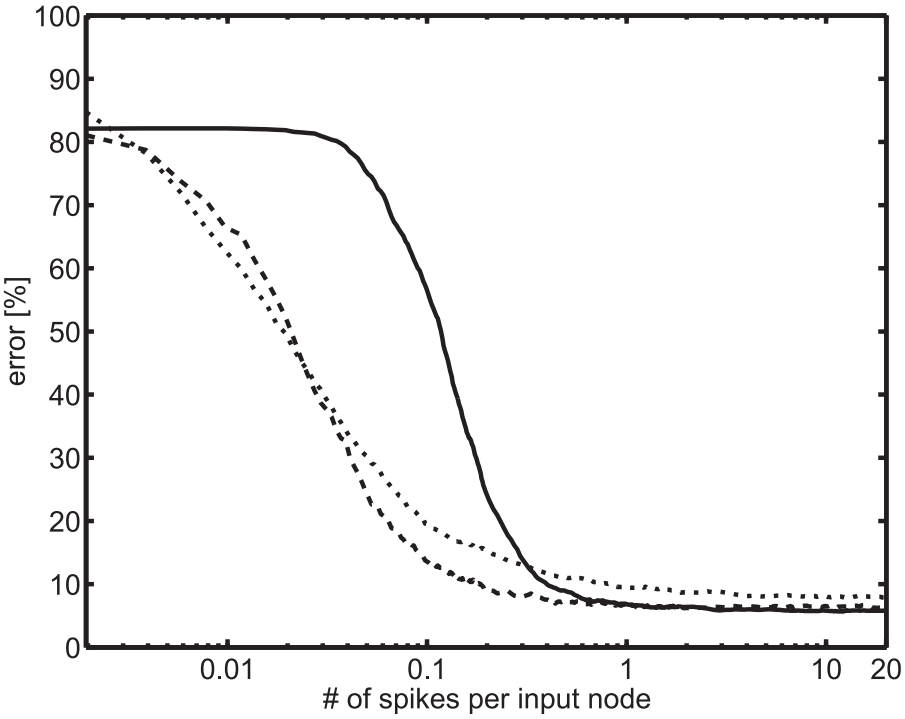


Figure 3.9: Mean classification error e for the USPS data base shown for different numbers of neurons in the hidden layer, in dependence on the number of input spikes (parameters as in Fig. 3.8). The dashed line shows the error of a NN-classifier in comparison. The SbS algorithm with 500 hidden neurons (solid line) is considerably slower, but exceeds the performance of the NN-classifier. If learning and classification is performed with an adaptive estimation constant $\epsilon \propto t^{-1/3}$ (dotted line), the performance with only 100 hidden neurons approaches speed and performance of the NN-classifier. The weights in this example were trained with the batch learning rule.

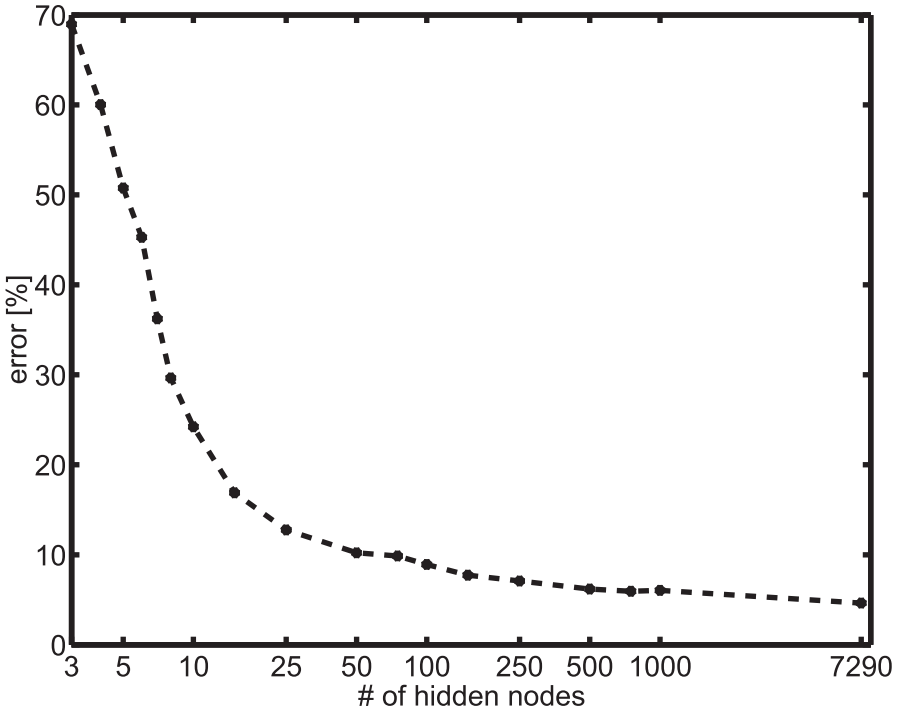


Figure 3.10: Mean classification error e for the USPS data base in dependence of the number of hidden nodes after 10 spikes per input node. The weights are attained via the batch learning rule and the other parameters are as chosen for Figure 3.8.

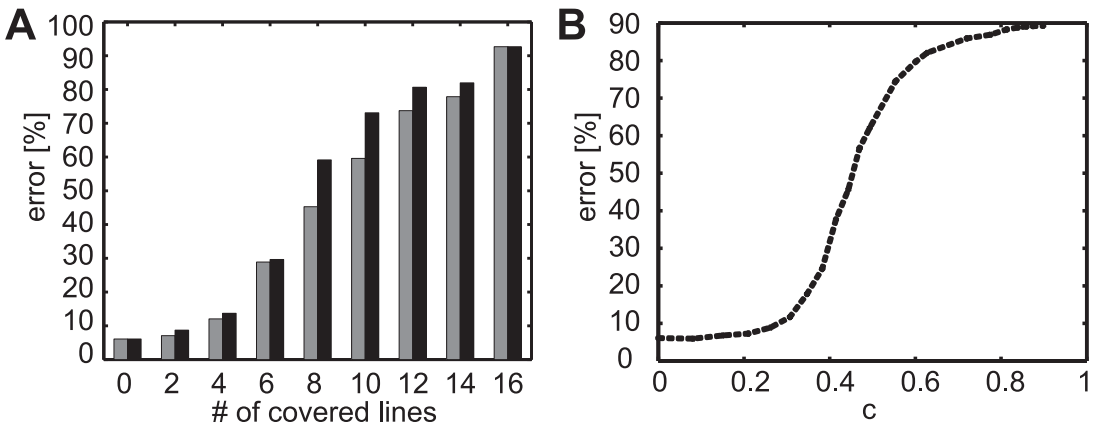


Figure 3.11: Error rate of the SbS algorithm (a) for digit patterns partially occluded by grey bars, and (b) for digit patterns subjected to a varying amount η of pixel noise (weights and parameters as in Fig. 3.8). In (a), the grey bars indicate the error rate under a horizontal occlusion of pixel rows (for details see B.5), while black bars indicate the error under a vertical occlusion of pixel columns in dependence on the number of occluded rows or columns. For a number of less than 4 occluded pixel rows or columns, and for a noise level less than $\eta = 0.25$, the recognition error remains below 10 percent.

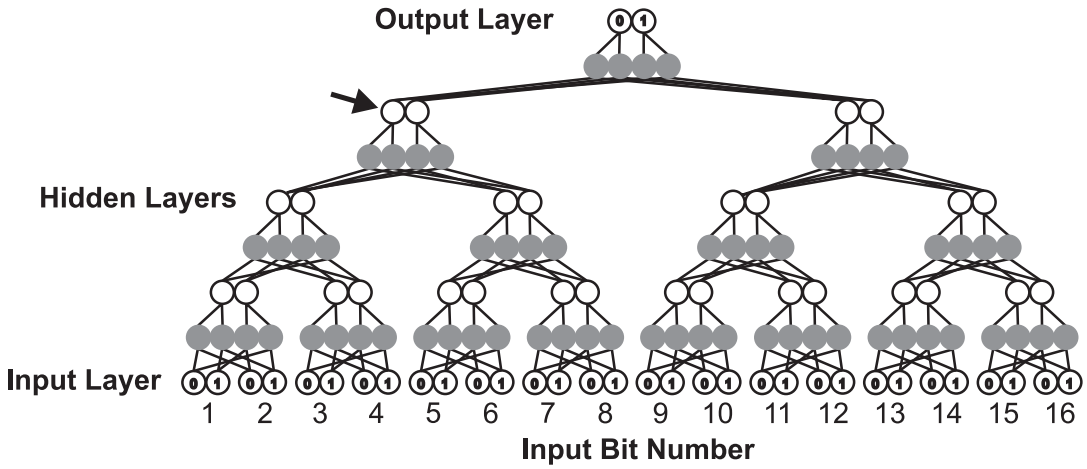


Figure 3.12: Structure of a hierarchical network constructed for solving the parity function with 16 input bits. The network consists of 15 XOR-modules linked together. Grey discs denote the hidden layer nodes of the single modules, while open circles denote the input/output nodes. Connections between the nodes are marked with solid lines (all connections set to a weight of 0.5). The arrow indicates the output nodes which compute the parity sub-function for the first 8 input bits (from left).

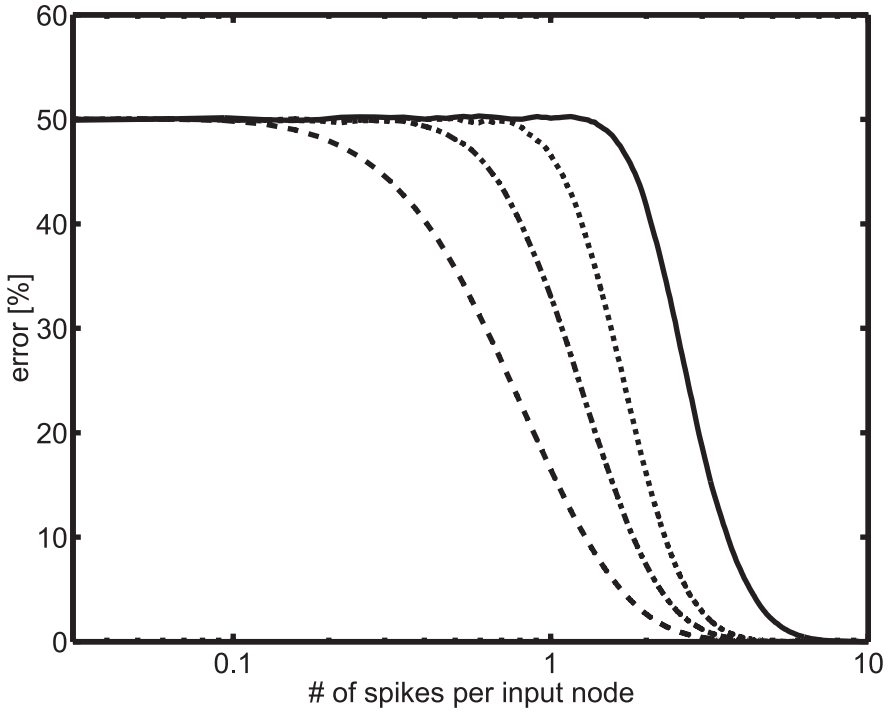


Figure 3.13: Classification error of the hierarchical SbS-network displayed in Fig. 3.12 in dependence on the mean number of spikes per input node (solid line). In a hierarchical network, the accuracy of the output of a higher layer relies on the accuracy of the output of the lower layers. This dependence is demonstrated by the error curves for the first hidden layer (dashed line), the second hidden layer (dashed-dotted line), and the third hidden layer (dotted line). Correspondingly, the curves show that the error in the n-th layer falls below 10 percent after about $1-2$ spikes per input node later than in the $n-1$th layer.

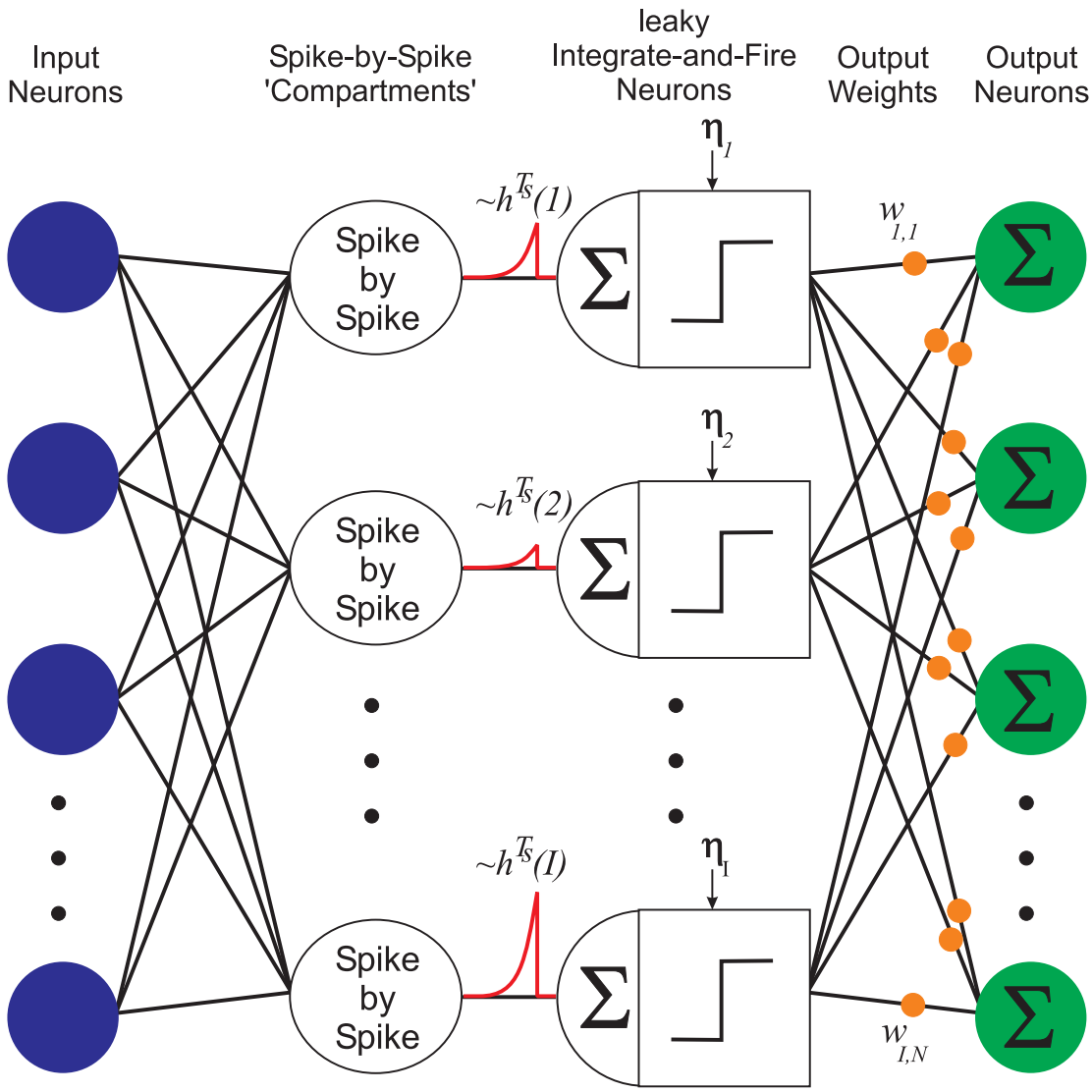


Figure 3.14: An overview how the spike-by-spike algorithm is combined within a network of leaky integrate-and-fire neurons. The input is generated like in the other examples for the spike-by-spike algorithm. The typical spike-by-spike layer (denoted as spike-by-spike 'compartments') uses the incoming spikes to update its h-values via the spike-by-spike h-update rule. In a new next step, the spike-by-spike compartments convert the incoming spike into an excitatory post-synaptic potential. This EPSP decays exponentially over time and its height it proportional to $h^{T_s}(i)$ for the spike-by-spike neuron i, where T_s denotes the spike at time T_s. This EPSP is the input for i-th leaky integrate-and-fire neuron (with additive noise η_i). If the membrane potential of the integrate-and-fire neurons surpasses the threshold, a delta impulse is send to the output neurons. The delta impulse is multiplied with the relative weights and is accumulated by the output neurons. The index of the output neuron with the highest value is selected as result of the computation.

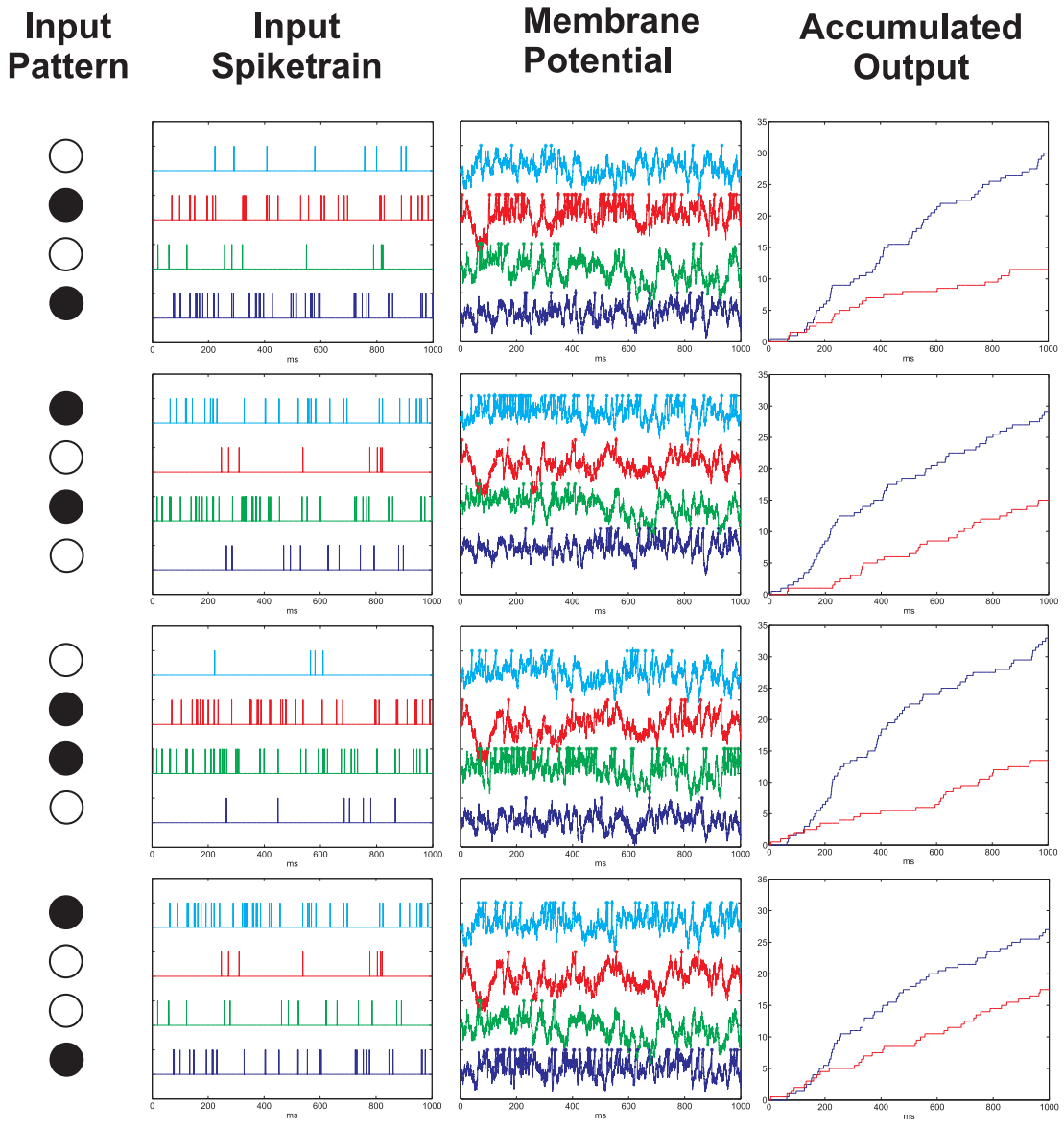


Figure 3.15: Implementation of the XOR-gate (parity function with 2 input bits) through a combination of spike-by-spike update algorithm and leaky integrate-and-fire neurons. One of the four possible input strings (left most column) splitted into on/off center channels are used for generating input spike-trains drawn from Poisson distributions. The input spikes (second column) were convoluted with realistic post-synaptic potentials and then weighted with the h-values from the spike-by-spike algorithm before they were used as input currents (see Eq.(3.25)) for the leaky and noisy integrate-and-fire neurons (see Eq.(3.24)). When one of the membrane potentials $V_i(t)$ hit the threshold ϑ, the membrane potential was reset to V_{Reset}, a spike weighted, and sent to the output layer. A sample time course of the membrane potential is shown in the third column. The last column shows the accumulated output. When the red line lies below the blue line then the output of the network is correct.

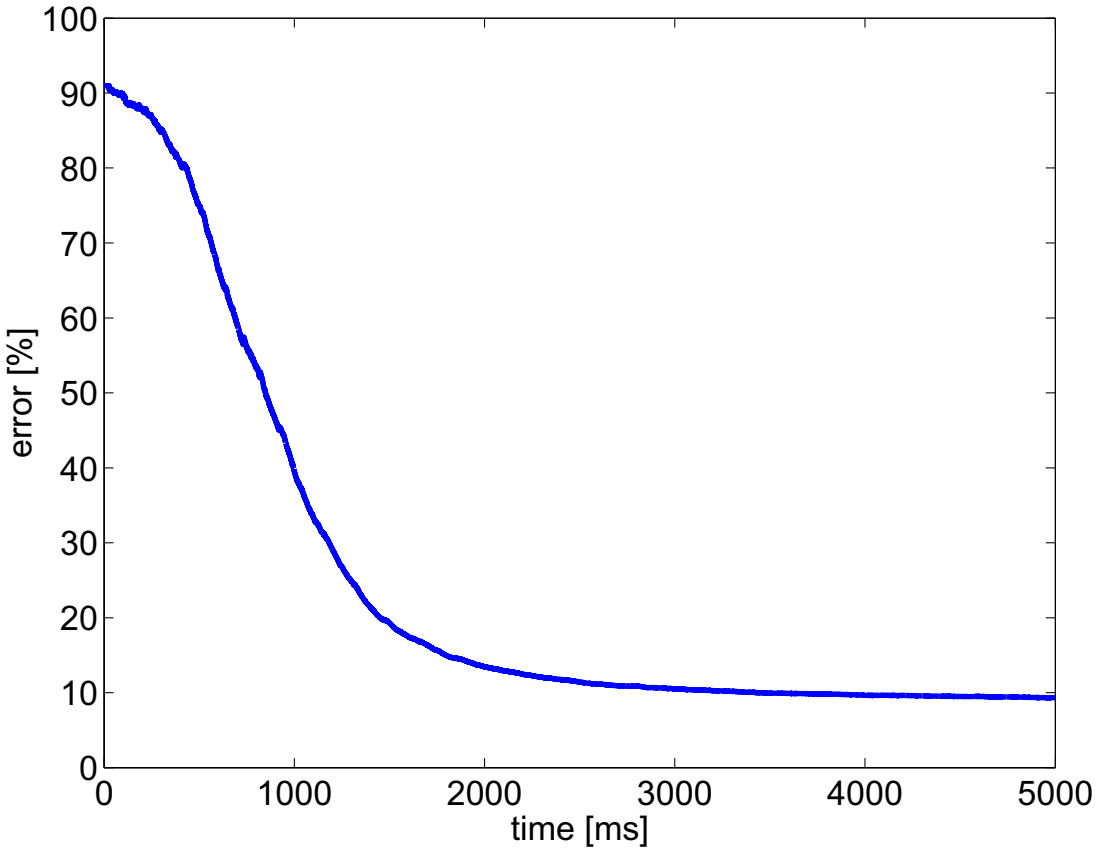


Figure 3.16: Implementation of the recognition of handwritten digits through a combination of the spike-by-spike update algorithm with leaky integrate-and-fire neurons. The setup is similar to Fig. 3.15, using the weights and input structure introduced in Fig. 3.8. This curve shows that the classification of the handwritten digits is also possible within this noisy and biologically more plausible model. The curve shows the error averaged over 25 different initial conditions for the 2007 handwritten digits from the USPS test dataset.

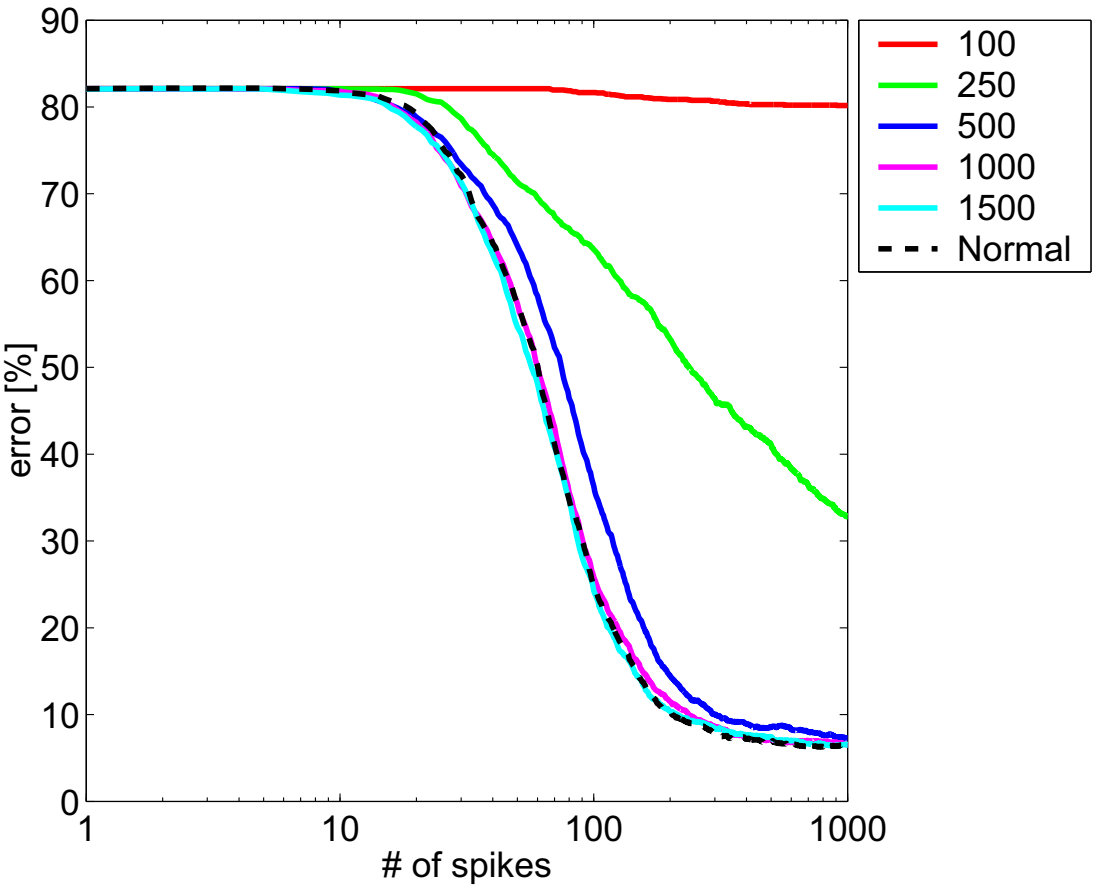


Figure 3.17: Classification performance of the USPS handwritten digits in dependency of the number of input spikes (x-axis) and reliability of the nominator (color code). The nominator was represented through $\sum_i \varpi_i^t$, with ϖ_i^t drawn from a Poisson distribution with mean value $h_i \cdot p(s^t \mid i)$ (see Eq.(3.26)). $\Xi = 100$ (red line) results in ≈ 0.03 mean spikes for each normalisation step. With this low number of spikes, the performance of the algorithm is close to chance level. Using $\Xi = 250$ (green line) increases the mean number of spikes to ≈ 0.35 and improves the recognition performance significantly. With $\Xi = 500$ (blue line), a mean number of ≈ 2.5 spikes is used for the information transmission. The performance with this Ξ value approaches the performance of the case with the un-perturbed information propagation. A further increase to $\Xi = 1000$ (magenta line, mean spike count ≈ 5.5 spikes) and $\Xi = 1500$ (cyan line, mean spike count ≈ 12.8) results only in a small enhancement in performance. The 'normal' algorithm, with the un-perturbed nominator, is represented by the black dashed line.

3.3.7 Artificial and natural images

Seen as a putative model for information processing in the visual cortex, it is interesting to look at the dynamics of spike-by-spike networks driven by natural images. Before using natural images, in an preceding step a spike-by-spike network was fed with artificial pictures composed of superpositions of horizontal, vertical, and diagonal rectangles. The idea behind this analysis is to show that results and weights from a spike-by-spike network make sense. This is important to know before using natural images as inputs. For natural images we do not know what statistical structures the pictures are containing. Thus it is not easy to judge if the algorithm delivers us for natural images meaningful information. For showing that the networks is able to extract such meaningful information, the spike-by-spike algorithm is applied to artificial images were we know all aspects of the structure of the input and also know what the output of the network should be. Fig. 3.18 shows the details and results of this simulation. The spike-by-spike batch learning rule was able to extract the basis functions of which the training data set was composed. Using the emerging weight set, it was possible to select one of those artificial images as input for the network and to use the resulting h-value distribution for reconstructing the input images nearly perfectly. It should be noted that it was not necessary to use on/off channel decomposition like it was applied for the USPS handwritten digits.

Similar to the procedure with the artificial images, 5000 images from the Corel natural image database were taken. From each of these images a 12 pixel x 12 pixel patch was cut out and used as training data. Weights for different ϵ-values were learned by the spike-by-spike batch learning rule. Fig. 3.19 shows four sets of these weights. Depending on the ϵ-value, the structural complexity of the weights differs. For larger ϵ-values spatially structured weights were generated. Smaller ϵ-values lead to weight sets resulting a pixel-based representation.

Using these weight sets for reconstructing images as a mosaique of smaller patches leads to the problem, that the method removes the absolute range of each tile. This is due to the fact that the input pattern is converted into a probability distribution which sums up to the value of 1. As countermeasure, one extra channel is added to the set of input channels. For all patches, the extra pixel value was set to 256, before calculating the probability distribution from the pixels $\in [0, \cdots, 256]$. Using this extra information allows to reconstruct the absolute luminance range for each patch from the distribution of h-values. For Fig. 3.20 and Fig. 3.22 this idea was applied to an image from the Corel picture database (because of the larger number of input channels, the weight sets had to be learned again). Several reconstructions (patch-based) are shown for a gray-colour version of this picture in Fig. 3.20 and for a 24-bit colour version in Fig. 3.22. The used weight sets are displayed in Fig. 3.21 and Fig. 3.23.

Fig. 3.24 shows the quality of the reconstructions in dependency of the number of spikes, using the weights from Fig. 3.22 with 432 hidden neurons. After each of the input neurons fired a few times, the subject of the image can already be recognised.

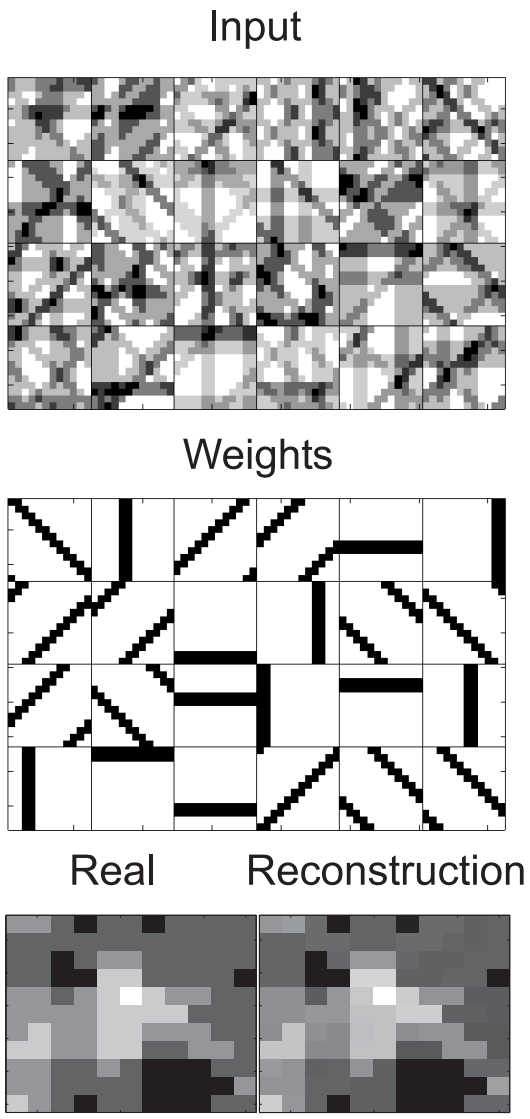


Figure 3.18: Reconstruction of artificial images via spike-by-spike algorithm. A set of 3600 images composed of superpositions of vertical, horizontal, and diagonal bars were used as training data for a spike-by-spike network with 24 hidden neurons using the batch learning algorithm. An example from these input patterns is shown in the top panel. The algorithm was capable, after several repetitions of drawing about 30.000 spikes from each input patch, to extract the generating weights (basis functions, center panel). Using these weights, an input patch (bottom panel, left) was fed into the spike-by-spike network. After iterating 4000 input spikes with the spike-by-spike dynamic, it was possible to reconstruct the input patch almost perfectly from the h-values via $\sum_i p(s \mid i)h_i$ (bottom panel, right). An ϵ-value of 0.1 was used for this simulation.

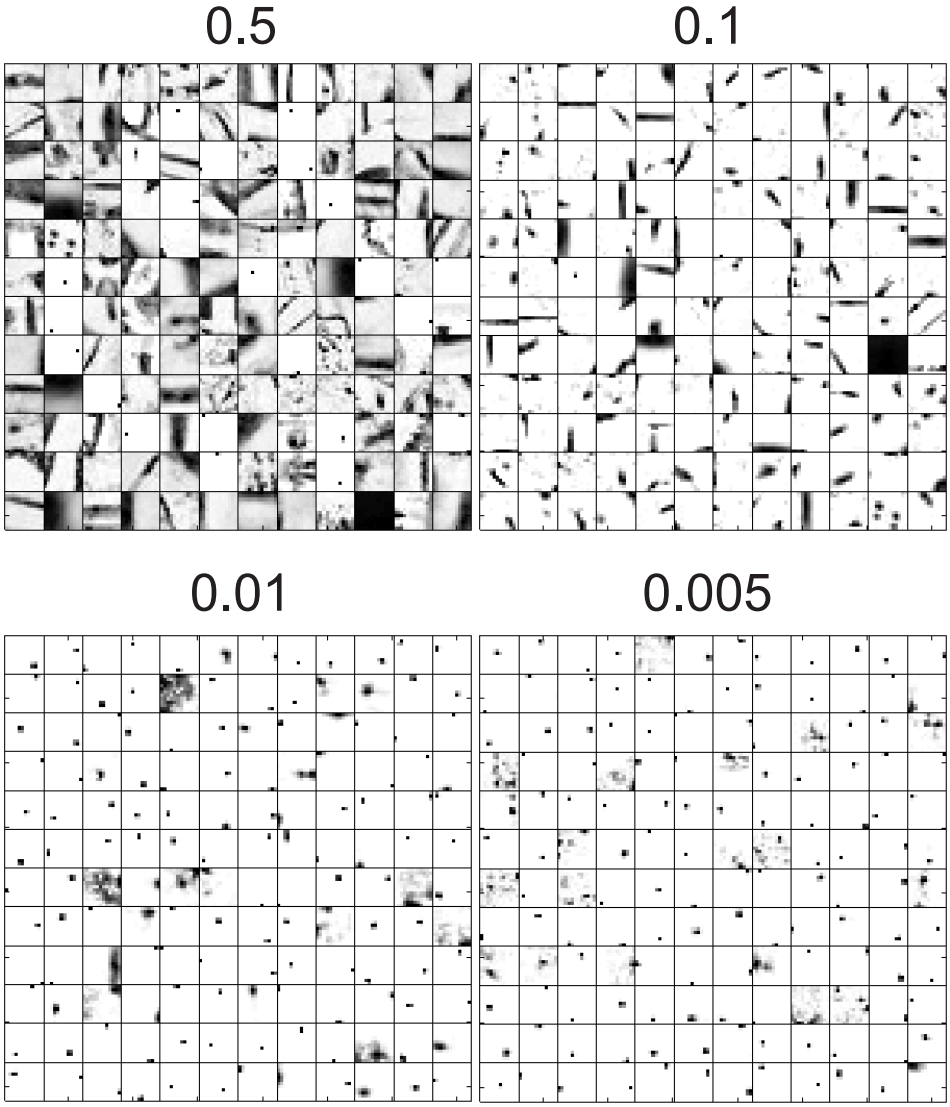


Figure 3.19: Four sets of weights obtained by training a spike-by-spike network on natural images with the spike-by-spike batch algorithm. Each set of weights consists of 144 vectors with 144 entries corresponding to 12x12 pixel values, shown for $\epsilon = 0.5$ (upper left), $\epsilon = 0.1$ (upper right), $\epsilon = 0.01$ (lower left), and $\epsilon = 0.005$ (lower right). For the training procedure 5000 images from the Corel image database were used. From each of these images one 12x12 patch (extracted with offset 100 x 100 from the upper left corner of the image) was taken. The three colour channels of each of these pixels were averaged for generating a gray value between 0 and 255. For each training step, 20.000 spikes were drawn from each pattern. The training procedure was repeated 300 times including all 5000 image tiles. The four examples show that with decreasing ϵ the weights emerge into a pixel basis. For larger ϵ values more spatially structured weights evolve.

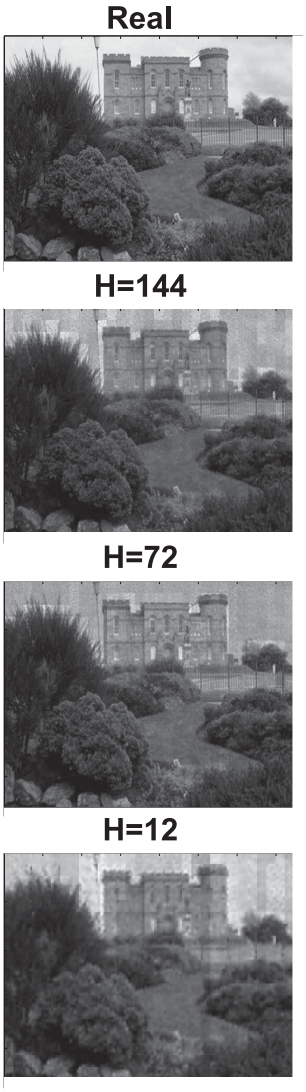


Figure 3.20: Reconstruction of natural images (consiting of gray values) with spike-by-spike networks ($\epsilon = 0.005$). The original input picture (titled 'real') was cut into a set of 12x12 pixel arrays. Weights (shown in Fig. 3.21) for 144 hidden neurons (second row), 72 hidden neurons (third row), and 12 hidden neurons (last row) similar to the procedure described in Fig. 3.19 were trained (with 20.000 spikes per array and trial). Since this procedure destroys the relative contrast between each array, an extra input channel for each array was introduced and used for coding a pixel value of 256. For the reconstruction, 20.000 input spikes were drawn from each 12x12 tile. The generated h_i-values were used for inferring the input tile via $\sum_i p(s \mid i) h_i$. The extra pixel was used for correcting the relative contrast between the tiles. Reducing the number of hidden neurons results in a reduction of reconstruction quality. The picture shown, was also taken from the Corel natural picture database.

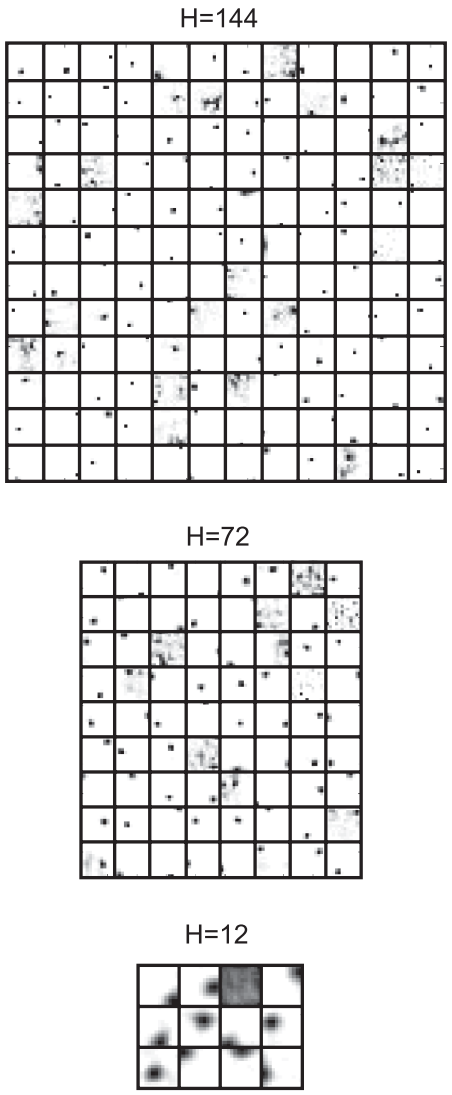


Figure 3.21: Weights ($\epsilon = 0.005$) for the reconstructions shown in Fig. 3.20, for 144 hidden neurons (fist row), 72 hidden neurons (second row), and 12 hidden neurons (last row).

Figure 3.22: Reconstruction of natural images spike-by-spike ($\epsilon = 0.005$). This figure is similar to Fig. 3.20 with the difference of using colour images with three colour channels instead of only one channel with gray values. The image was reconstructed using 432 hidden neurons (second row), 72 hidden neurons (third row), and 12 hidden neurons (last row). The corresponding weights are shown in Fig. 3.23.

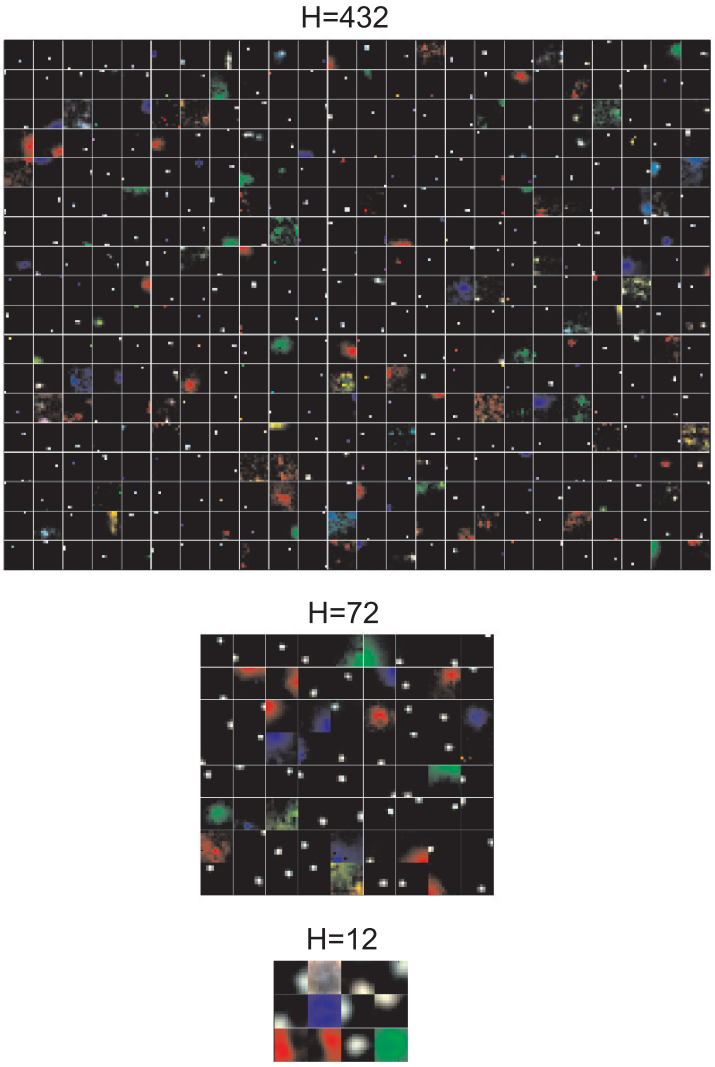


Figure 3.23: Weights ($\epsilon = 0.005$) for the reconstructions, shown in Fig. 3.22, for 432 hidden neurons (fist row), 72 hidden neurons (second row), and 12 hidden neurons (last row).

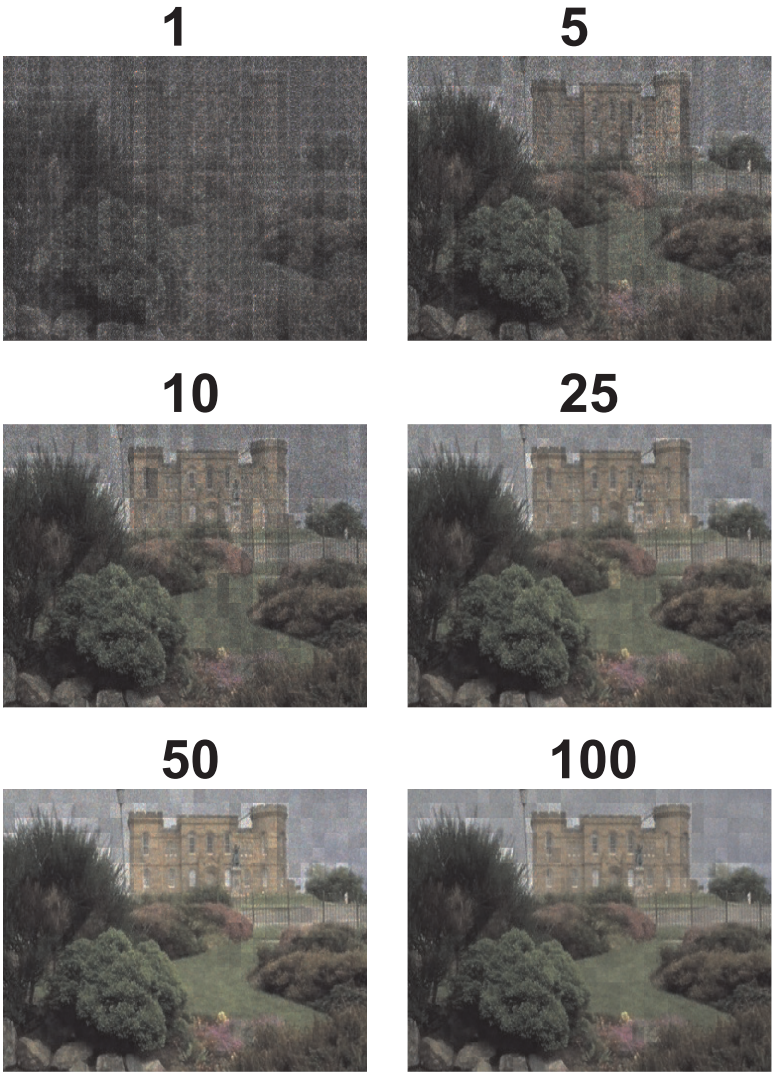


Figure 3.24: Reconstruction of natural images spike-by-spike in dependency of the number of spike events. This figure was generated like described in the caption of Fig. 3.22. The network was composed of 432 hidden neurons using the weights shown in Fig. 3.22 second row. Snapshots of the reconstruction after receiving 1, 5, 10, 25, 50, and 100 mean spikes per input neuron and array. With increasing numbers of spikes, the quality of the reconstruction increases. Fig. 3.25 shows the quality (Kullback-Leibler divergence) of the reconstruction in dependency of the used number of input spikes.

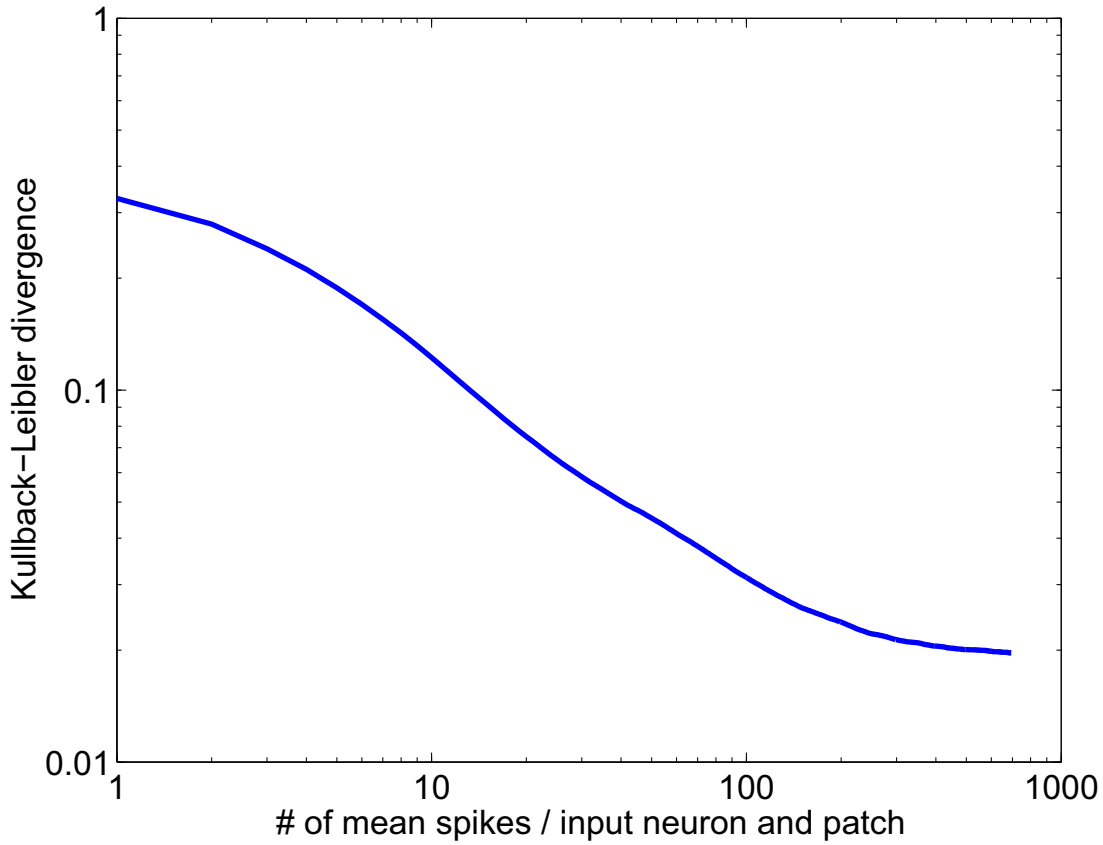


Figure 3.25: Kullback-Leibler divergence of the reconstruction shown in Fig. 3.24.

3.4 Summary and Discussion

A generative framework for neural computation that relies on spike signals given by Poissonian point processes was presented. In this framework probabilities for eliciting spikes capture neural activities and synaptic weights are correspondingly related to conditional probabilities to observe spikes given a putative elementary cause. While this scheme is equivalent to non-negative matrix factorization (NNMF) when used for the asymptotic case where mean rates represent the signals, the spike-by-spike on-line update for single action potentials leads to a fundamentally different dynamics of hidden states. In particular, for over-complete representations the algorithm favors sparse hidden layer representations in contrast to the basic algorithms for NNMF derived in (Lee and Seung, 1999), while the degree of sparseness can be tuned by a parameter. While ad hoc sparseness conditions for NNMF have been discussed in (Hoyer, 2004), in the spike-by-spike framework it emerges from the requirement of rapid convergence. A different idea to link sparseness to fast visual processing was put forward by Perrinet et al. (Perrinet et al., 2004). Here, the idea is to subtract with each spike an orthogonal projection of the best matching feature from the input scene (matching pursuit). This method has been shown to need very few spikes to achieve a good reconstruction of natural scenes.

Using the new algorithm it was demonstrated on simple paradigmatic examples that very small networks can rapidly and accurately perform complex computations using only few stochastic spikes per input neuron for achieving maximal performance.

As a proof of concept, learning and computation of random Boolean functions was implemented. Boolean functions are particularly challenging because a different input in one channel (bit) may switch from a specific function of the remaining bits to an arbitrary different function. In a neural environment, this dynamical property allows to change a computation performed in a certain brain area depending on a specific context. This context can be interpreted e.g. as attentional priming, additional sensory cues, or information provided by other brain areas. For Boolean functions with N input bits, 2^{N} hidden nodes are in general required for a complete representation. However, in this analysis it was found that optimal performance is in most cases possible with less than these 2^{N} hidden nodes. This reduction is due to the capability of the algorithm to find, and to represent redundancies in an effective manner. The error vanished completely in dramatic contrast to the algorithm for NNMF by Lee and Seung (Lee and Seung, 1999), which failed for this task. It is possible that the stochasticity of the spikes effectively implements a Monte-Carlo sampling over the posterior which helps to find a better solution as with NNMF. This is an interpretation being put forward by Hoyer and Hyvärinen (Hoyer and Hyvärinen, 2003).

The ability of this spike by spike network to learn and compute Boolean functions shows that highly nonlinear, non-smooth functions can efficiently be computed on the basis of very few spikes which underlines universality of this framework.

Applications of the model with $H \approx 500$ hidden nodes to handwritten digits demonstrate that on average less than one spike leads to recognition rates (5.8% error rate) which exceeded those of the nearest neighbor classifier that used the full training set with more than 7000 patterns (6.2% error rate). This high speed of recognition appears to be a general property of our framework, and depends only little on the input stimulus ensemble. For the original USPS data set, using a Support Vector Machine results in 4.0% error rate (Schölkopf et al., 1995) and humans have a 2.5% error rate (Bromley and Säckinger, 1991).

Also the spike by spike model was studied on the reconstruction of natural images. Training the network via the batch learning rule leads to weight vectors $p(s|i)$ whose appearances strongly depend on the value of ϵ. For very small ϵ, weight vectors approximate orthogonal pixel basis functions, specializing in explaining the spikes of few neighboring input channels. Consequently, the representation of one natural image is a linear, non-sparse superposition of these basis functions. In contrast, values of ϵ near one lead to weight vectors approximating templates of image patches. The representation then gets maximally sparse by activating only one hidden node whose weight vector matches most closely the presented input pattern. In between these extreme cases the weight vectors have strong similarities to the receptive fields shown in Olshausen anf Field (Olshausen and Field, 1996). An interpretation of the weight vectors learned with a fixed ϵ is one of a representation which is optimized to explain any input pattern with an average number of spikes being proportional to $1/\epsilon$.

In summary, the results challenge the notion, that the high speed of visual perception found in human and monkey psychophysics (Thorpe et al., 1996; Mandon and Kreiter, 2005) can only be explained by assuming that the exact timing of individual spikes conveys the information about the stimulus (van Rullen and Thorpe, 2001). The recognition of handwritten digits was also used to show that the framework is very robust to noise and data corruption.

Per construction, the spike-by-spike algorithm is invariant to scalings of overall intensities. This property has been found in primary visual cortex where the tuning of neurons is invariant to stimulus contrast (Skottun et al., 1987), which gives a hint to the potential relevance of this approach for explaining cortical computation. At first sight, the equations for update and on-line learning Eqs.(3.20) and (3.21) might appear biologically unrealistic. However, it may be expected that the algorithms can be implemented using known biophysical properties of real neural circuits. This implementation is not completely done yet, but we will give an outline of the underlying ideas in the following paragraph (some of these ideas, moving the spike-by-spike algorithm a step towards a biologically plausible implementation, are shown in the Results section of this chapter).

First, let us assume that the hidden variable $h^t(i)$ is represented in the excitability of a neuron, or, even more realistic, in the excitability of a neuronal group or population (Wilson and Cowan, 1972) which would be equal to the average over all membrane potentials. Second, we observe that the update term in Eq.(3.20) is proportional to

$h^t(i)p(s^t|i)$. One could interpret this term as a spike delivered over a synapse with efficacy $p(s^t|i)$, which is multiplied with the excitability $h^t(i)$. In fact, such multiplicative interactions have been observed in real neurons as e.g. in Chance et al. (Chance et al., 2002). Spikes in the populations representing the $h^t(i)$'s will be elicited with a probability proportional to $h^t(i)$. One inhibitory population would suffice to receive and to average those spikes, and feed them back to the other populations as a divisive normalization, which is a neural operation that has been extensively studied in early visual cortical areas (Heeger, 1992; Carandini and Heeger, 1994). This information transmission is not instantaneous but the simulations showed that is not a problem.

Taken together, this outline demonstrates that a biophysically plausible implementation of the spike-by-spike algorithm should be possible. We used networks composed of leaky integrate-and-fire neurons where the neurons are coupled by an implementation of the spike-by-spike update dynamics for information processing. The successful implementation of Boolean functions and the recognition of handwritten digits through such networks, showed that this approach is selfconsistent and that it can be used for constructing complex networks by stacking many layers of neurons, while communication is still done by single spikes. Furthermore it suggests that the spike-by-spike algorithm can be implemented as one part of a multi-compartment neuron model. Another handycap for implementing the spike-by-spike algorithm in a biological plausible fashion is the denominator of the update term in Eq.(3.20). We showed that recognition of handwritten digits by a spike-by-spike network is still possible even when the value of the denominator is estimated from Poissonian spike trains send from the other neurons. This suggests that a biophysically plausible implementation should be possible to formulate.

A different approach to spike-by-spike estimation was studied by Denève (Deneve, 2005), who derived a differential equation for the dynamics of a Bayesian neuron updating an estimate of the log-likelihood-ratio for the presence/absence of a stimulus in the receptive field. This study takes additionally the temporal dynamics of a stimulus into account, however, in its current state only for the presence/absence of a single cause and not for mixtures of causes. Denève also demonstrated that the corresponding differential equation is, assuming various approximations, similar to those of an integrate-and-fire neuron.

For demonstrating the biological plausibility of on-line learning, Eq.(3.21) can be interpreted such that the update of the weights is composed by two terms: one that is Hebbian, and another one which describes weight decay proportional to postsynaptic activity. The purpose of the second term is to put limits on the growth of the weights and thus to prevent 'weight explosions'.

$$\Delta p(s|i) \quad \propto \quad h(i)p(s|i)(\delta_{s,s^t} - p(s^t|i)) \; . \tag{3.27}$$

In previous approaches, simple cell responses were explained from ecological requirements (Barlow, 1960) like statistical independence of hidden node activations (Bell and Sejnowski, 1997). For natural image data it was argued that this requirement is

equivalent to sparseness constraints on the activities (Olshausen and Field, 1996). A different justification for sparseness may be found in the principle of minimal energy consumption (Levy and Baxter, 1996). In frameworks like the presented one, where the input data are modelled by a linear superposition of basis vectors, however, imposing sparseness conditions on the coefficients right from the beginning appears rather ad hoc. In contrast, this work points towards a new explanation of sparseness as a consequence of convergence time constraints. In Eq.(3.20) residual noise of the hidden variable estimations caused by non-vanishing values of the update speed parameter ϵ induces sparseness of hidden node activations. The relation of speed and sparseness becomes most clear when considering the update algorithm with the extreme value of $\epsilon = 1$. In this case, asymptotically only the hidden neuron obtaining the largest input will remain active. Here, sparseness becomes maximal and we effectively have a winner-takes-it-all network. With $\epsilon < 1$ sparseness is enforced, however, sparseness of the hidden representation is dominated by the necessity to explain the data.

These dependencies may also be interpreted in a different fashion: the parameter ϵ introduces a time scale which effectively determines how strongly the observation of an input spike changes the internal representation. Small ϵ increase the 'memory' for previous spikes, thus the internal representation can be adjusted more accurately to an input pattern at the cost of a larger observation time. It is obvious that a more accurate representation needs in general also more basic features to explain the input data. In contrast, with a large ϵ a sparse representation of few features already suffices to explain an input pattern. This representation has a high variability due to the small number of observed spikes (This tradeoff has a lot in common with lossy image compression used in computer science, where the number of basis functions is restricted.).

Taking all these observations together, this framework might provide a novel basis for understanding neural computation: Coming from a rather technical background where similar algorithms were introduced to achieve blind source separation and blind de-convolution (often referred to as the 'cocktail party problem'), the examples have shown that this approach is quite universal and can be used to efficiently perform general computations. In particular, it can perform multi-modal neural integration of sensory input with spike sequences from different sensory modalities or other brain regions (Deneve et al., 2001). Also it is straightforward to extend the approach to include statistical expectations, particular tasks, and attention.

- Statistical expectations (priors) can be included by a set of extra input neurons. The distribution over the spiking propabilities of these neurons represents the prior.

- Several tasks can also be represented by a set of extra input neurons. The selection of different tasks can done by different activity distributions over these neurons. These is similar to Boolean functions were one bit can change the rest of the function.

- Attention can be introduced into the spike-by-spike network like described for the prior or the selection of a task.

Chapter 4

Selective Visual Attention in V4/V1

4.1 Motivation

'Attention' is important for the visual system (see section 4.2.3). If attention is directed to an object embedded in a complex sensory scenery, attention is known to enhance the representation of the attended object. Some examples of improved aspects are faster responses, lower thresholds and better discriminability for attended in comparison to non-attended objects. These enhancements through attention suggest that the underlying neuronal representation and processing of information related to attended objects are improved. In this chapter we want to learn more about how attention modifies the neuronal representation of visually perceived shapes and on which mechanisms this improvement is based.

For example, selective visual attention has been found to induce modulations of neuronal firing rates (see section 4.2.3) (McAdams and Maunsell, 1999b; Moran and Desimone, 1985; Motter, 1993; Reynolds et al., 1999; Reynolds et al., 2000; Treue and Maunsell, 1996). Furthermore, neurons engaged in processing of an attended object tend to organize their response into synchronous firing patterns with oscillation frequencies in the gamma-band (Taylor et al., 2005; Steinmetz et al., 2000; Fries et al., 2001).

Several mechanisms how modulations of firing rates can influence and might improve representations have been proposed. One approach for explaining perceptual improvements is based on an increase of mean firing rates by attention like it was observed in several studies (see section 4.2.3). These rate modulations are accompanied by improved signal-to-noise ratios, thus providing a basis for enhancing discriminabilities of different stimuli (McAdams and Maunsell, 1999a). However, the corresponding improvements of representations appear to be limited since for stochastic spike

trains signal-to-noise ratios increase roughly with the square-root of the firing rate and attention-dependent increments of firing rate are often small or even missing (Reynolds and Chelazzi, 2004).

In addition, when multiple, spatially nearby stimuli are positioned within the same receptive field (RF), it was found that neurons tend to react as if only the attended stimulus was shown (Moran and Desimone, 1985; Treue and Maunsell, 1996). While this helps to disambiguate the representation with respect to such stimuli, it does not necessarily imply an improved representation of a single attended stimulus as compared to a single non-attended stimulus without other, competing stimuli within the RF.

In this chapter, it will be investigated whether additional mechanisms may explain the rather large perceptual effects of attention (Rock et al., 1992; Wolfe and Bennett, 1997). The following analysis is intended to improve the understanding how selective attention changes the discriminability of neuronal activity patterns in the visual cortex. The analysis is based on field potential signals which were recorded from an epidural electrode array implanted in macaque monkeys. During the experiment, animals attended to one of two spatially well-separated shapes placed in the right and left visual hemifields.

The goal of this analysis was to infer the perceived stimuli (shapes) from the measured neuronal data and to use the performance of this inference as tool for learning more about the neuronal representations of stimuli.

The recorded electrophysiological signals were pre-processed (e.g. current-source-density method) and decomposed into their frequency components by wavelet analysis. For evaluating the degree of discriminability, support vector machines (SVMs, see section 2.2.4) as state of the art classifier (Schölkopf et al., 2000; Schölkopf and Smola, 2001) were used. The SVMs were applied to identify the different classes of the shown shapes based on the measured neuronal activity patterns.

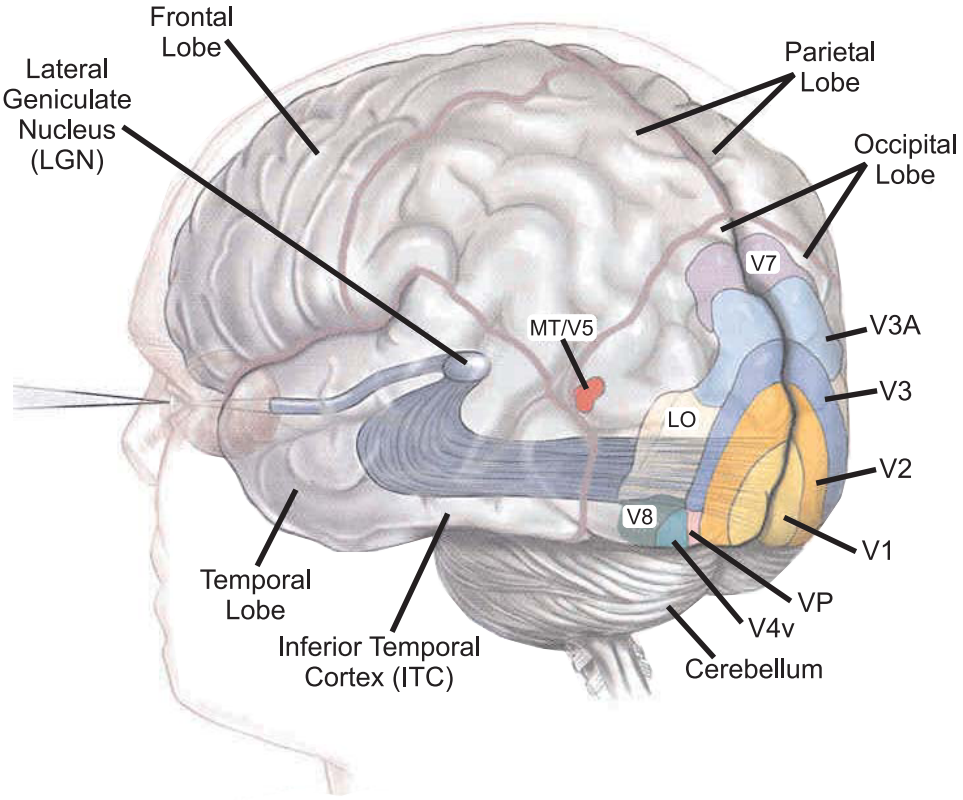


Figure 4.1: Overview of the visual pathways. Both retinas collect and transmit visual information over their optical nerve fibers to the lateral geniculate nucleus (LGN). From the LGN is the information transmitted to V1. V1 distributes the information to other areas of the brain. For a more detailed description of V1 and V4 (not shown) see section 4.2.2 and 4.2.2. V3A and MT/V5 are processing motion information. LO is involved in processing large-scale objects and V8 is linked to color vision. (The figure was adapted from (Logothetis, 1999), ©1999 Terese Winslow)

4.2 The visual system

For many species, processing visual information is crucial for their daily survival. The evolution of biological systems brought up many different types of visual systems (Land and Fernald, 1992). This overview will focus on the primate visual system and will discuss some features of area V1 and V4 of the visual cortex in more detail. Later in section 4 we will describe on data recorded in these areas and analyse the influence of attention on their dynamics. In Fig. 4.1 an overview of the human visual system is shown. We will start the following survey with the retina, continue with the LGN, and finally describe properties of the visual cortex.

4.2.1 Retina

The retina (Hubel, 1989; Goebel et al., 2003) can be characterised as the first part of the brain that comes in contact with visual information from the environment. It converts impinging lightwaves into patterns of neuronal activities and transmits these activities over the optical nerve to further processing stages. In the structure of the retina we find several layers of different types of cells.

The first layer of the human retina consists out of 'retinal pigment epithelium' cells as barrier to the tissue with the blood vessels, followed by a layer of photoreceptors. Thus the light has to pass all layers before the photons can reach the photoreceptors. The density of receptors is not constant over the whole retina. A higher density can be found in the central part of the retina in comparison with the peripheral part. Photoreceptors can be divided into two different populations, rods and cones. The rods are specialized for viewing in twilight and they will be deactivated by brighter illumination. In contrast to the rods, the cones cease to work if the illumination is too poor. The cones are specialised to different spectral bandwidths, which allows primates to perceive colors. Normally three types of cones with different absorption spectra can be found in the retina. Their maxima of sensitivity are located at blue ($\approx$ 420 nm), green ($\approx$ 540 nm), and red ($\approx$ 560 nm). Young presented first a concept in 1802 (Young, 1802) how to combine these three colors for explaining color vision. This basic concept was refined by von Helmholtz (von Helmholtz, 1860).

In the next retinal layer, three types of nerve cells are present: bipolar cells, horizontal cells, and amacrine cells. The bipolar cells get their input from the photoreceptors and many of them project directly to the subsequent layer containing retinal ganglion cells. The horizontal cells are positioned in parallel to the layer of receptors and provide long range connections between receptors and bipolar cells. The amacrine cells provide a similar structure for the bipolar cells and the ganglion cells. The last relevant retinal layer for information processing consists of retinal ganglion cells. The axons of these retinal ganglion cells form the optical nerve. An astonishing fact is that each eye typically comprises 125 million rods and cones, but only one million retinal ganglion

cells. This provokes the question how the perceived information is selected and pre-processed before it is routed over the optical nerve to the LGN.

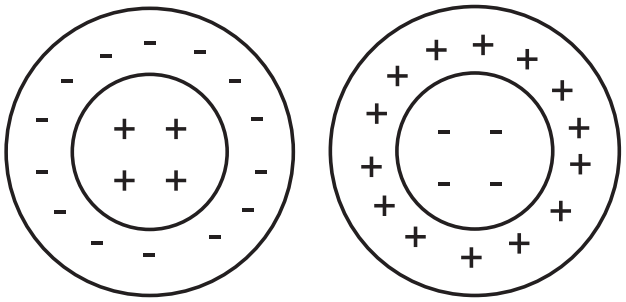


Figure 4.2: Illustration of center-surround on-center and off-center receptive fields. The regions marked with a plus-sign modify the neuronal activity through an excitatory influence when illuminated. Without illumination, these regions have an inhibitory influence on the neuronal activity. The regions marked with a minus sign react similar but with opposite sign.

In 1953 Kuffler (Kuffler, 1953) published a study about discharge pattern of ganglion cells in the retina. On- and off-center receptive fields, with a structure as shown in Fig. 4.2, were found. These center-surround on-center receptive fields react most strongly to stimuli of light surrounded by darkness, and the center-surround off-center receptive fields reveal their highest response to a dark area surrounded by light. Homogeneous brightness over the whole visual field will normally not cause a strong neuronal response. The ganglion cells can vary in their temporal response behaviour, size of receptive fields and sensitivity to contrast, color, and movement. In addition, the retina shows extremely complex patterns of neuronal activity for more natural and especially transient visual stimuli. The question, what neuronal response can be expected for a given (complex) stimulus is still topic of experimental and theoretical research (Hosoya et al., 2005; Fernandez et al., 2000; Berry et al., 1997; Greschner et al., 2002). The research regarding the visual system inspired the field of image processing and compression.

4.2.2 Pathways to and through the visual cortex

The axons of retinal ganglion cells project across the inner surface of the retina to the optical disk where the optical nerve is formed (Goebel et al., 2003). Both optic nerves from the retinas pass the optic chiasm (Jeffery, 2001) where the nasal fibers of one eye are fused with the fibers from the contralateral eye. The fibers are bundled into two optical tracts, of which $\approx 90\%$ continue to the lateral geniculate nucleus (LGN).

Lateral geniculate nucleus

The LGN (Malpeli and Baker, 1975; Connolly and van Essen, 1984; Goodale and Milner, 1992; Xu et al., 2001) consists of six layers of neurons separated by layers of very small nerve cells ('koniocellular' layers (Casagrande, 1999)). The cells in the first and second layer are relatively large ('magnocellular') while cells in the other four layers are smaller ('parvocellular'). The layers one, four, and six receive their input from the contralateral eye while the layers two, three, and five get their input from the ipsilateral eye. The cells of each layer form a retinotopic representation, which means that two neighboring points on the retina are represented by two neighboring cell populations. Evidence was found that the magnocellular, parvocellular and koniocellular neurons are parts of separate visual pathways (already starting at the retina ganglion cells), see e.g. (Livingstone and Hubel, 1987; Livingstone and Hubel, 1988; Xu et al., 2001). The magnocellular (M) pathway is relative fast and processes not many aspects from the perceived visual input (e.g. shows no wavelength selectivity). It seems that it is devoted to stereo vision and motion processing. The parvocellular (P) pathway, which is slower than the M pathway due to more thin axons, is accredited to transmit detailed information with high spatial resolution and it seems to process shape information and color. The latter in cooperation with the koniocellular (K) pathway. The P-pathway reacts only weakly to motion. The LGN projects to the primary visual cortex (V1) and the primary visual cortex sends connections back to the LGN. The connections from the LGN to V1 preserve the retinotopic representation, but the P-pathway innervates the primary visual cortex at a different layer of V1 than the M-pathway. The functional role of the LGN is not fully understood.

Area V1

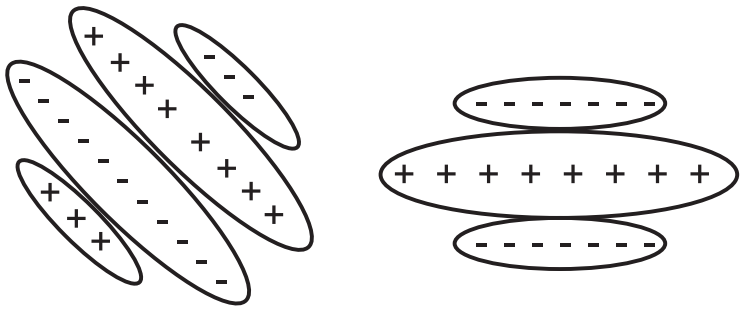


Figure 4.3: Illustration of the receptive fields (RF) of simple cells. (Figure adapted from (Hoyer, 2002))

Area V1 (Goebel et al., 2003), also known as Brodmann area 17 (Brodmann, 1909), can be understood as gateway to higher visual areas as well as an important processing stage of visual information. In neuro-anatomically descriptions (Goebel et al., 2003; Callaway, 1998), it is typically decomposed into 9 sub-layers: I, II, III, IVA, IVB, IVCα

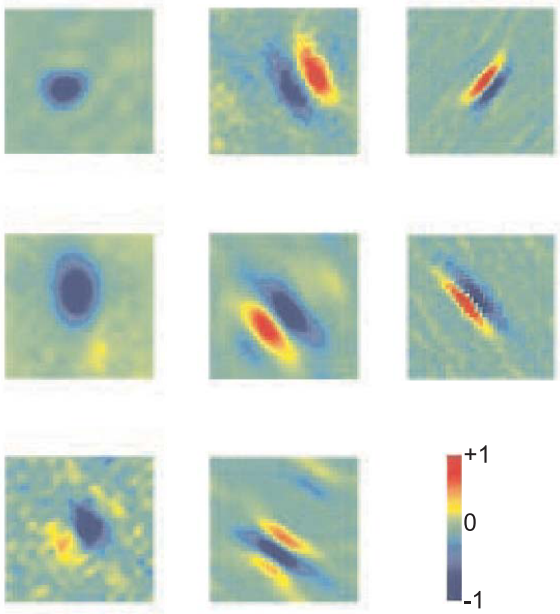


Figure 4.4: Examples of measured receptive fields in V1 (Figure adapted from (Ringach, 2002), used with permission from The American Physiological Society)

(receiving input from the M pathway), IVCβ (receiving input from the P pathway), V, and VI. 80% of the cells ('excitatory pyramidal cells') in V1 provide feedforward and feedback connections to other cortical areas. The axons of the non-pyramidal cells (more than 40 different types can be distinguished) do not leave V1. The complete functionality of V1 is unknown.

Regarding the functional architecture and neuronal response properties of (monkey) visual cortex, important research was done by Hubel and Wiesel (Hubel and Wiesel, 1968; Hubel and Wiesel, 1977). In contrast to retinal ganglion cells and LGN, the preferred stimuli of V1 are oriented line segments. A concentric tuning to simple light dots was mainly found in the sub-layer IVC, which gets direct input from the LGN.

Orientation-selective cells in V1 are typically classified as 'simple cells' or 'complex cells'. These cells are tuned to the orientations of elongated patches or lines, like depicted in Fig. 2.2. Simple cells favor stationary light bars of certain orientations. In Fig. 4.3 schematics for simple cells are shown. The receptive fields of these cells are divided into excitatory and inhibitory zones, similar to the center-surround receptive field organisation of the LGN or retinal ganglion cells, but different in shape. The receptive fields of complex cells are larger than those of simple cells and can be invariant to the exact position of the bar within the receptive field, as long as the orientation is matching the preferred orientation. In some cases the stimulus has to be spatially non-stationary for generating strong responses. Furthermore, cells (e.g. hypercomplex) exist, which are only selective to lines with a certain length.

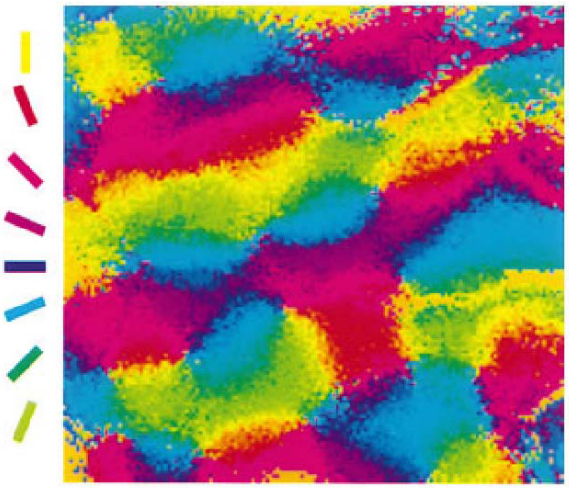


Figure 4.5: Orientation map from cat visual cortex. The colors encode the orientation of stimuli that generate the maximal response. (Picture taken from (Crair et al., 1997), with permission from Elsevier)

The receptive fields of cells in the primary visual cortex are organized such that neurons in 'cortical columns', aligned normal to the surface of the cortex, share nearly the same receptive field properties. In orthogonal direction to cortical columns, receptive field properties change gradually and form topographic maps. Maps for elementary features like e.g. orientation (see Fig. 4.5, (Bonhoeffer and Grinvald, 1991)), ocular dominance, direction, spatial frequency, and disparity were found. Evidence was found that contextual effects by stimuli shown outside of the (classical) receptive fields can alter the response properties of V1 (Zipser et al., 1996).

Other areas of the visual cortex

Classically two different visual pathways (Mishkin et al., 1983; Goebel et al., 2003) are distinguished, the ventral and the dorsal stream.

The ventral stream is regarded as a 'what' or 'vision for perception' information processing system. It is mainly driven by the P-Pathway and performs a fine-grained analysis of e.g. color and shape of visual scenes as well as pattern recognition and form analysis from faces or other local spatial structures. An illustration of the ventral stream is shown in Fig. 4.6. In contrast to the ventral stream, the dorsal stream is described as involved in 'where' or 'vision for action' information processing tasks. It is mainly driven by the M-pathway and processes spatial information of visual scenes, visually guided actions and visuomotor transformations.

The areas along both pathways are organized hierarchically and low-level representations are followed by more complex representations. For example V2 shares many

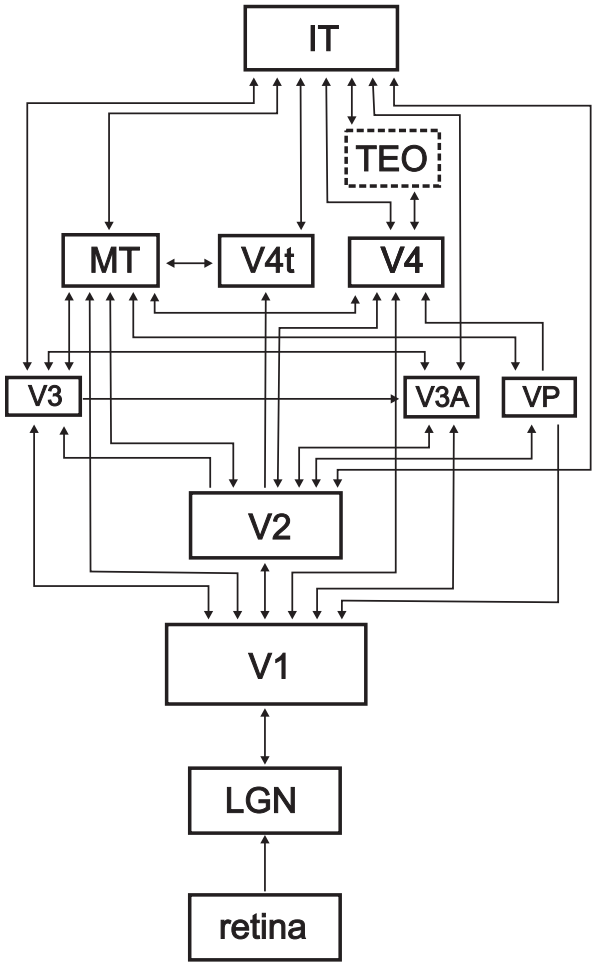


Figure 4.6: Simplified schematics of connections in the ventral loop. (Figure adapted from (Pollen, 1999))

receptive field properties with V1, but in addition the neurons in V2 can be sensitive to binocular disparity (depth information) or can have larger receptive fields. V3 is highly sensitive to contrast and selective for orientations, while V3A is motion-sensitive and V3B reacts to motion boundaries and kinetic contours. MT/V5 is tuned to directions and velocities of moving stimuli. It was shown that if this area is destructed, deficiencies in motion direction discrimination will occur (Goebel et al., 2003).

The idea of separate ventral and doral pathways is a simplification (see e.g. (Tolias et al., 2005; Ferrera and Maunsell, 2005; Zeki, 1993)). All of those visual areas are directly or indirectly (over other areas) linked by feedforward or feedback connections. The functional role of the feedback connections is still under debate (Pollen, 1999).

Area V4

The question what information the neurons from area V4 are representing by their activity patterns is still under research. In the following I will discuss some of these findings from experiments about the sensitivity of V4 neurons to stimulus properties.

For many V4 neurons a tuning of neuronal responses tor bar stimuli was measured. Selectivities for different length and width as well as different orientations and polarities of contrast have been confirmed. Many receptive fields of V4 neurons showed similarities to complex receptive fields in V1 (Desimone and Schein, 1987). They show a sensitivity (of some degree) to colour but also respond, with a lower activity, to white light (Schein and Desimone, 1990). Different populations of neurons are also selective to sinusoidal gratings and changes in the spatial frequency, orientation, phase, and size of the gratings (Gallant et al., 1996; Desimone and Schein, 1987). Furthermore, it was demonstrated that neurons in V4 are more strongly activated by non-sinusodial types of 'gratings', like e.g. concentric and radial patterns (Wilkinson et al., 2000; Gallant et al., 1996). Many V4 cells display a sensitivity to texture information of stimuli, which suggests that this visual area may be involved in the extraction of texture information (Hanazawa and Komatsu, 2001). If stimuli are too large (with respect to the receptive field size) then some of the cells react with strong suppression. Optical imaging detected a functional organisation linked to stimulus size (Ghose and Ts'O, 1997). Types of neurons were found, which were sensitive for direction of motion and kinetic patterns (Mysore et al., 2006; Desimone and Schein, 1987). In addition are V4 neurons also able code the absolute differences of objects (Dobbins et al., 1998), 3D orientations of slanted lines (Hinkle and Connor, 2002) and disparity (Hegde and van Essen, 2005; Hinkle and Connor, 2001). It has been found that inside the receptive field, the selectivity for binocular disparity is invariant for the position of stimuli. Neurons with similar disparity selectivity are clustered (Watanabe et al., 2002b).

Especially relevant for this work is the way V4 neurons code information about shape stimuli. It has been discovered that V4 cells respond stronger to complex shapes than to simple bar stimuli (Kobatake and Tanaka, 1994). Furthermore, it was revealed that cells in V4 are often tuned to contour features (like e.g. angles and curves), with the strongest variation of the neuronal response being correlated to the contour features orientation and convexity. Orientation of a contour means in this context, e.g. the direction of the angle of an edge-element (Pasupathy and Connor, 1999). Later it was found that tuning functions parametrised by curvature and angular position fitted neuronal responses with more accuracy than tuning functions parametrised by edge or axis orientation. For many cells, which responded to these types of stimuli, it was possible to characterize the strongest response by features of the stimulus' boundaries at specific angular positions (see Fig. 4.7). Other features of the shape had only minor influences on the neuronal response. In addition, it seems that these kind of selectivities of V4 can be modeled by two multiplied Gaussian functions, one parametrised by the curvature of the shape segment and the other one by the angular position of this shape segment (Pasupathy and Connor, 2001) (see Fig. 4.8). Furthermore, for silhouette-like

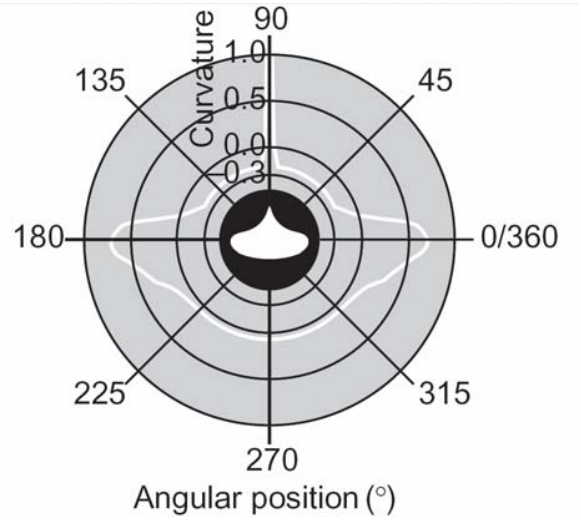


Figure 4.7: Curvature and angular position are important features for describing the response of a V4 neuron to a shape. Inside the black circle, an example shape is shown. Around the black circle, the boundary of the shape is shown in curvature and angular position space as white line. (Picture was taken from (Pasupathy and Connor, 2002), with permission from Macmillan Publishers Ltd / Nature Publishing Group)

stimuli it was shown that the boundary features can be partially reconstructed from the neuronal response of V4 populations (Pasupathy and Connor, 2002).

The neuronal responses of V4 neurons are modulated by visual attention (see section 4.2.3 for details about visual attention). It was demonstrated that attention can modify the sensitivity of neurons to contrast/salience in such a way that the magnitudes of neuronal responses to low-contrast stimuli are increased (Reynolds and Desimone, 2003). Spatial interactions between the attentional focus and visual stimuli show complex modulations of the neuronal responses. One component of this modulation was identified as response gradient around the attended target. The gradient decreases with increasing the distance between the attentional focus and a stimuli. (Connor et al., 1997).

4.2.3 Visual attention

Focusing our mind on the observation of one object in a crowded visual scene can make us nearly blind to changes that do not concern that object. This striking phenomenon is termed 'change blindness' (O'Regan et al., 1999) and is very likely resulting from 'visual selective attention' (Itti, 2002). Visual attention probably allows the visual system to blank out large amounts of incoming sensory input. It was also shown that visual selective attention is prerequisited for shape perception (Rock et al., 1992; Rock and Gutman, 1981). For reviews regarding visual attention see (Itti, 2002; Treue,

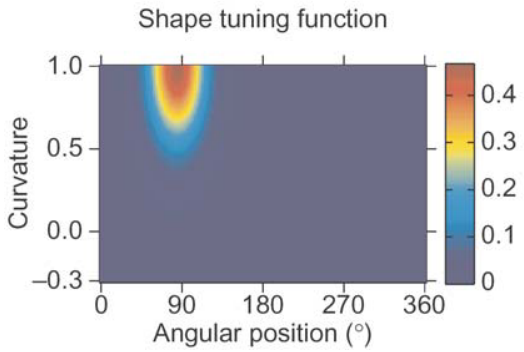


Figure 4.8: Shape tuning function of an example neuron in V4 for the features curvature and angular position, modeled by Gaussian shaped tuning functions for each stimulus dimension. The color scale represents the normalized predicted response. (Picture was taken from (Pasupathy and Connor, 2002), with permission from Macmillan Publishers Ltd / Nature Publishing Group)

2001; Desimone and Duncan, 1995; Reynolds and Chelazzi, 2004; van Rullen and Koch, 2005). Two different types of attention are discussed: bottom-up ('stimulus-driven') and top-down ('goal-directed','task-dependent') attention (Connor, 2004; Yantis, 1998; Egeth and Yantis, 1997).

Bottom-up attention seems to be a largely unconscious process and is considered to be driven by specific visual features of the perceived scene ('image-based'). Prominent examples for bottom-up attention are types of visual search experiments (Wolfe, 1998; Treisman and Gelade, 1980) where a target has to be localized that is hidden among other distracting elements. Depending on the features of the target and the distractors, either the target 'pops-out' from the background of distractors or it can become necessary to check all elements, one by one, to find the target. The pop-out effect is brought into association with a fast parallel search during which our attention is drawn onto the target. In the case of serial scanning, which is normally slower in comparison to search-tasks where pop-out effects occur, the focus of attention lies on one or a small number of elements from the whole visual scene. The shift of attention from one element to another is thought to be directed by higher cognitive levels. As a consequence, such shifts of attention are called top-down attention. Pop-out and serial search are the two extreme cases of an often more complicate behaviour (Itti, 2002).

One approach to understand bottom-up attention is the 'feature integration theory' (Treisman and Gelade, 1980) where simple features (like e.g. color, intensity, orientation, direction of movement, and disparity) are detected pre-attentively in a massively parallel way over the entire visual field and represented by topographical maps ('early representation' (Koch and Ullman, 1984)). This process is supposed to take place in the early visual processing areas (e.g. primary visual cortex). In a next step, these features can then be linked into more complex object representations. Before further processing begins, mechanisms using attention are applied to filter out most of the

non-attended representations through the so-called attentional bottleneck, (Itti, 2002).

Several computational models for explaining bottom-up attention have been proposed (see (Itti and Koch, 2001) for a review). Many of these models have in common that they use 'salience maps'. These salience maps are scalar topographical maps that combine early representations for different features such that the salience map represents the overall difference in feature space at one location in comparison with its surrounding locations. How these salience maps are computed differ from model to model. Typically salience maps are used to find the maximum of all saliencies and then to direct attention to that position.

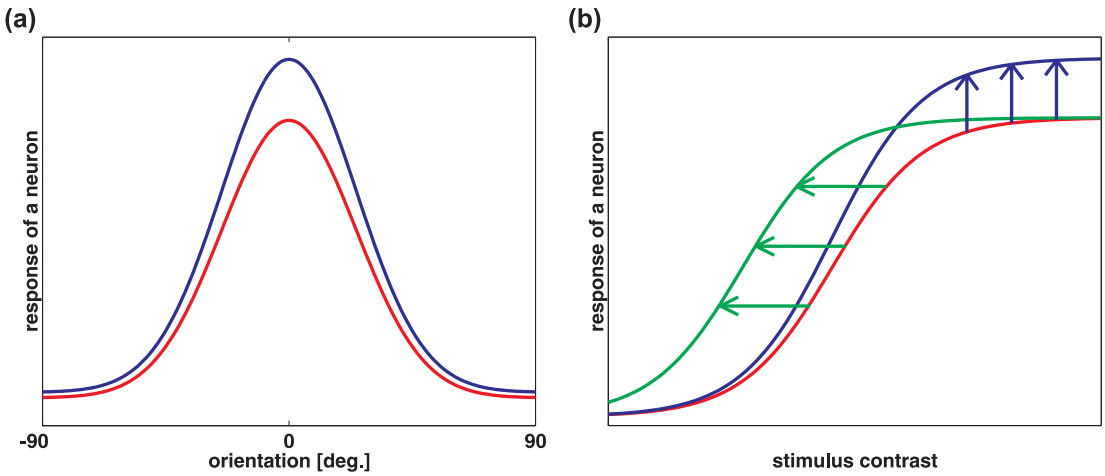


Figure 4.9: Illustration of two possibilities to of modify tuning functions through attention. The red line represents a tuning function without attention. The blue lines shows a multiplicative gain through attention, as it was found by (McAdams and Maunsell, 1999a) for orientation tuning, and the green line illustrates an increase of firing rates by a shift of the tuning function, as described in (Reynolds et al., 2000) for contrast-sensitive neurons.

Top-down attention seems to act differently than bottom-up attention (for a detailed review see (Treue, 2001). The following part of this section is based on this article.). It can be directed by cues from higher cognitive levels (like verbal cues) (Itti, 2002). Experiments revealed that already visual information processing in V1 is influenced by such effects, (Motter, 1993; Ito and Gilbert, 1999; Gilbert et al., 2000). Furthermore, modulations through attention were found in the ventral and dorsal pathway (Treue and Trujillo, 1999; McAdams and Maunsell, 2000). The modulation starts right at the beginning of the pathways and the effects are increasing along the pathways in direction to higher hierarchical levels (Tootell et al., 1998; Treue, 2001). For the dorsal pathway, experiments found that the attention focus, when directed into the receptive field of neurons, can alter the response properties of neurons. For the ventral pathway, experiments found only indications for a similar behaviour of cells. The findings for the ventral pathway need a verification (Treue, 2001).

Experiments suggest an attentional system which interacts with receptive fields that have an overlap with the so-called 'spotlight of attention'. In experiments with two stimuli inside of one receptive field (one attended and one non-attended) (Reynolds et al., 1999; Moran and Desimone, 1985), it was found that the spotlight of attention can act on spatial scales smaller than the corresponding receptive fields. The influence of attention on neuronal responses seems to be a combination of an enhancing effect for attended stimuli and an inhibitory effect on the non-attended stimuli ('push-pull') (Pinsk et al., 2004; Treue, 2001). This suggests a modification in the tuning properties of a neurons by attention. This can be interpreted in two ways: As attentional modification of the whole tuning function ('gain modulation' (Treue and Trujillo, 1999)) or as specific improvement of attended stimuli with a simultaneous penalty for the unattended stimuli (biased competition (Lee and Seung, 1999)). Furthermore, top-down attention can act in a non-spatial, feature-based fashion which allows to improve expected and behaviorally relevant visual features (Chelazzi et al., 1993; Chelazzi et al., 1998).

Fig. 4.9 shows two types of attentional modification (multiplicative gain (McAdams and Maunsell, 1999a) and gain through shifting (Reynolds et al., 2000)) to tuning functions, which were both found in experiments.

This chapter will continue with the presentation of results from a data analysis regarding attentional effects on the neuronal correlates, generated by complex shape stimuli (Rotermund et al., 2007a).

4.3 Experimental Setting, Preparations and Methods

4.3.1 The experimental setting

The following analysis is based on data measured from two macaque monkeys. These data sets are the result of experiments performed by Katja Taylor in the group of Andreas Kreiter (Institute for Theoretical Brain Research, University of Bremen). The experiments were designed to investigate the effect of selective attention on neural activity patterns in cortex, while processing visual information. Both animals, monkey F and monkey M, were trained to perform a so called 'delayed-match-to-sample' task (see Fig. 4.10).

For making the use of selective attention for this task necessary, two different stimuli (shapes) are presented simultaneously on a computer screen. During the beginning of the trial (1550 ms of the initial stimulus presentation period) the left or right shape is cued by a green colouring. The colour signalises the animal which side of the screen can be ignored. The green colour fades out within 600 ms after stimulus onset. Subse-

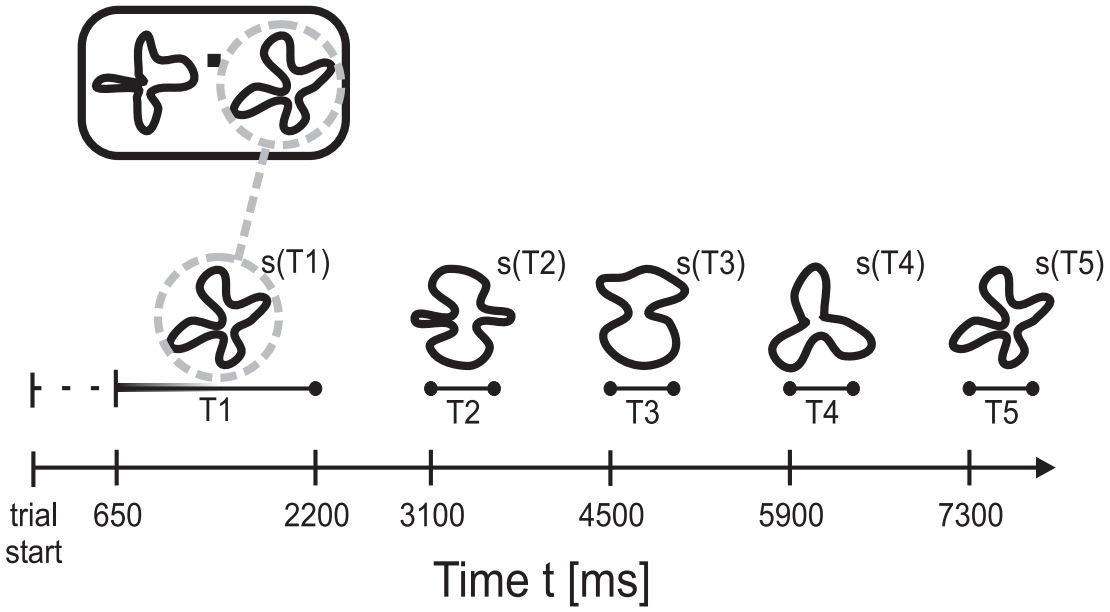


Figure 4.10: Schematic illustration of the shape-tracking task. Two sequences of shapes were presented in the left and right hemifield of a computer screen. (see an example in the upper left rectangle). While the monkey fixates to a dot in the centre of the screen, its attention was directed to one of the initial shapes presented during period T1 – in this example, to the shape on the right hand side. This was done by showing one shape in green colour during the period (in T1) which is marked with the shaded segment of the line. The task for the animal was to signal the re-occurrence of the initial shape in the attended hemifield during one of the following periods T2 - T5. Here, a correct response would be during or after presentation of shape s(T5). (Drawn by Katja Taylor)

quently, the memorized stimulus has to be compared to different test stimuli forming a stimulus sequence. In case that the memorised stimulus reappears, the animal has to signalize this event by releasing a lever. During the whole trial, the monkey has to fixate a fixation point, which is positioned between the two shapes, otherwise the trial is stopped with an error and will not be rewarded.

A stimulus sequence is composed of two to five stimuli (behaviourally relevant shape and distractor shape). After the initial shape, the stimuli were visible for a 500 ms period. One period of presentation was followed by a 900 ms delay phase, where only the fixation point remained on the computer screen. The distractor shapes were randomly selected from a set of ten different shapes. The behaviourally relevant shapes were selected from a subset of six of these ten shapes.

The chronically implanted epidural electrode arrays covered parts of area V4 and V1 (see Fig. 4.11), with 36 electrodes for monkey M and 37 electrodes for monkey F. When the cued stimulus was shown to the visual hemifield covered by the electrode array, the

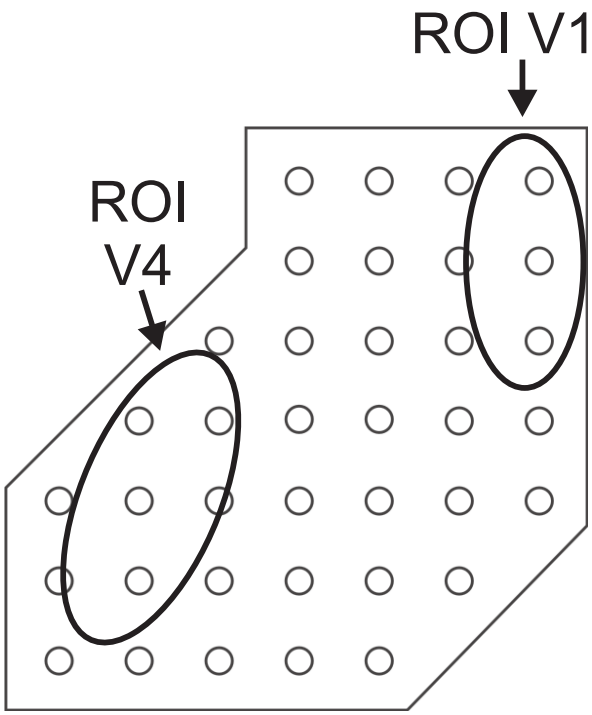


Figure 4.11: Map of the rough locations of the subdural electrodes (open circles) in relation to areas V1 and V4 of Monkey F. The regions were estimated by retinotopy. (Drawn by Katja Taylor)

situation is termed 'attended condition' (a-C). The other case, when the cued shape was shown to the visual hemifield without the electrode array, the condition will be referred to as 'non-attended condition' (n-C). It has to be noted that always one of the shapes was in the focus of selective visual attention (a-C and n-C symbolises only the fact whether the electrode array 'sees' the attended shape).

Average behavioural performance of monkeys during recording sessions was estimated from all but the longest trials in which a response would have been always correct. Disregarding fixation errors, the monkeys performed correct for 83,1% (monkey M) and 73,4% (monkey F) of the trials. Correct responses occurred 467ms (M) and 418ms (F) after target stimulus onset (median values). Errors were distributed roughly similar over different initial figures.

For more details of the behavioural training, visual stimulations, surgical preparations and recordings see (Taylor et al., 2005; Rotermund et al., 2007a).

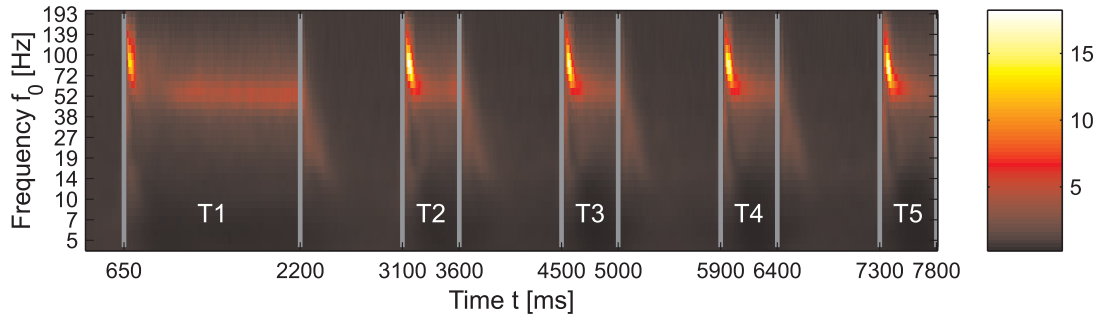


Figure 4.12: Example for a typical time-frequency plot on the same time axis as in Fig. 4.10, displaying the trial-averaged, normalised power spectral density $A(t, f_0)$ in the attended condition for an electrode over V1.

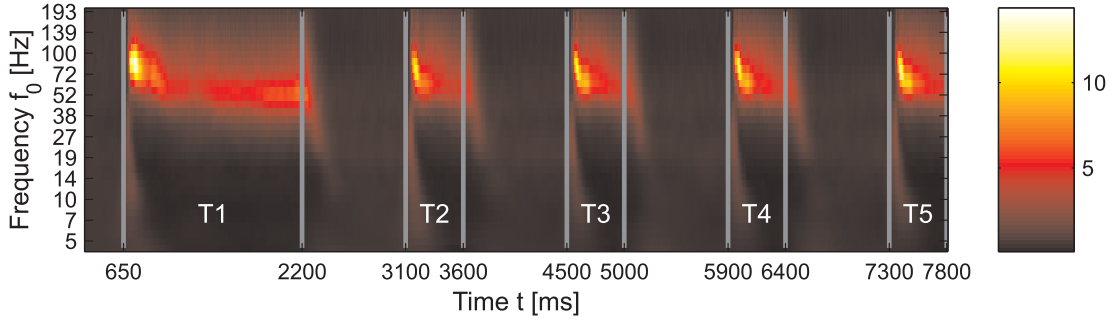


Figure 4.13: This figure shows, analog to Fig. 4.12, the time course of the trail averaged, normalised power spectral density $A(t, f_0)$ in the attended condition for an electrode over area V4.

4.3.2 Data Preprocessing

Presenting a visual stimuli in the visual hemifield contralateral to the implanted array produces local field-potential responses for the electrodes positioned over the visual areas V4 and V1. These field-potentials were high-passed filtered with a digital filter (for details see (Taylor et al., 2005; Rotermund et al., 2007a)). Trials containing artefacts were rejected. Furthermore, current source (and sink) density (CSD) (Gevins, 1984) was calculated for minimizing spatial smearing (Nunez et al., 1997): Assuming that the electrical field E is proportional to a gradient of a scalar function Φ

$$E = -\nabla\Phi \,,$$

we can use the two relevant Maxwell equations for an electro-statical field

$$\nabla \bullet E = 4\pi\rho$$
$$\nabla \times E = 0$$

with $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. This gives us the Poissonian equation

$$\nabla^2\Phi = -4\pi\rho$$

with charge carriers in the corresponding spatial region and the Laplacian equation

$$\nabla^2\Phi = 0$$

for spatial regions without any charge carriers (Jackson, 1993). This allows us to approximate $\rho(x, y)$ from the recorded date. For each millisecond of data the second spatial derivative of the field potentials was computed with the Laplacian operator (∇^2) (Perrin et al., 1987), using Gaussian radial basis functions (RBF) for interpolating the data, which was recorded at discrete electrode position, into a continuous surface (Moody and Darken, 1989). At the end of the calculation, for each electrode the CSD yields the signals $\nu_j(t)$ with j denoting trial number.

Based on $\nu_j(t)$, the signals were wavelet-transformed into time and frequency dependend amplitudes and phases. For this calculation a convolution with complex Morlet wavelets $w(t, f_0)$ (Kronlandt-Martinet et al., 1987) was performed. Using this method, wavelet power coefficients were obtained through the equation

$$a_j(t, f_0) = \left| \int_{-\infty}^{+\infty} w(\tau, f_0)\nu_j(t - \tau)d\tau \right|^2 . \tag{4.1}$$

For the following data analysis, only trials where the animals made no fixation errors were used. The spacing of frequency bands was logarithmic between 5 and 200Hz, chosen as $f_0(k) = \Omega^{k-1}f_0(1)$ for $k = 1, \ldots, 17$ frequency bands starting at $f_0(1) = 4.84\text{Hz}$. For a sufficiently tight coverage of frequency space, Ω was set to 1.2593. If we denote the power of a sinusoidal wave with frequency $f_0(k)$ at its frequency $f_0(k)$ with $P(f_0(k))$ then the power of this wave will decrease to $\approx 0.57P(f_0(k))$ at the nearest neighbouring frequencies $f_0(k \pm 1)$. For the following frequencies $f_0(k \pm 2)$ the power will drop to $\approx 0.059P(f_0(k))$. This ensures that the relevant part of frequency axis is covered tightly.

In Fig. 4.12 and Fig. 4.13 typical time frequency plots are shown for one electrode above V4 and one electrode above V1. For both figures the normalised mean spectra

$$A(t, f_0) = \frac{< a_j(t, f_0) >_j - n(f_0)}{n(f_0)} \tag{4.2}$$

were computed using normalisation coefficients quantifying the background activity $n(f_0)$, which was obtained from

$$n(f_0) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} < a_j(t, f_0) >_j dt$$

with $t_1 = 300\text{ms}$ and $t_2 = 350\text{ms}$. The time-frequency plots show that the increase of power through the stimuli vary over the frequency bands and is most pronounced between 40 and 100Hz.

4.3.3 Discriminating Stimuli with SVMs

Discriminating different dynamical states of the brain or its sub-areas using neural signals has been performed in various experimental settings, different species and with different types of recorded neural signals, ranging from single unit studies to LFP/ EEG recordings. See section 2.2.4 for a selection of applicable methods for such classification tasks. One prominent method is the support vector machine (see section 2.2.4, (Schölkopf et al., 2000; Schölkopf and Smola, 2001)). In this data analysis the widely used libsvm software package (Chang and Lin, 2001) was applied. It provides convenient data pre-processing routines and automatically searches through the parameter space for good SVM settings. Radial basis functions were used as kernels for the SVM classifier.

For classifying the presented shapes on the base of the measured data, represented by the wavelet power coefficients, the data sets were divided into two subsets of approximately equal size. The trials were alternately assigned to one training subset and to one test subset in an interleaved fashion. The classes were defined by the shapes $s_j(k)$ presented to the monkeys in selected intervals $k \in \{T1, T2, T3, \ldots\}$ of the stimulus sequence (see Fig. 4.10) displayed on the 'recorded' side of the visual field during trial j. Only the six classes for the behaviourally relevant shapes $s \in \{1, \ldots, 6\}$ were used as targets for the analysis. The remaining trials with the other four distractor-only shapes were ignored. In the following, where necessary, the index s will be used to distinguish variables, which were computed using only wavelet coefficients from trials j in which the shape shown in interval $k = T1$ was s , i.e. $s_j(T1) = s$. Likewise, data from trials where attention was directed to the visual hemifield represented in the recorded brain region will be distinguished with a superscript A, while using a superscript N otherwise.

From all coefficients $a_j(t, f_0)$ obtained within a period T for the centre frequency f_0, a subset of a's equally spaced in time were selected and used for computing averaged coefficients $\bar{a}_j(f_0)$. The spacing was adjusted to approximately twice the period $1/f_0$ which is sufficient to capture the typical rate of change in wavelet-analysed signals. Averaging led to a large decrease in computational complexity for the training of the SVMs. Data analyses with the original, full set of coefficients were also done and yielded in no substantial difference in classification performance, thus only averaged coefficients were used for the presented results.

The SVMs were trained on the training sets, and their classification performance was evaluated on the test sets. The resulting performances P were measured as the total percentage of shapes classified correctly by the SVM in the test sets. The chance level P_{chance} was computed as the ratio of the occurrences of the most frequently presented pattern in the training set to the total number of trials in this set. An increase (decrease) in performance P above chance level was considered to be significant as soon as the probability to obtain an equal or higher (equal or lower) performance by drawing from a binomial distribution around $P_{binom} = P_{chance}$ was smaller than

$p = 0.02$, respectively. A difference in performance $P^A - P^N$ was considered significant as soon as the probability to obtain P^A and P^N from two binomial experiments was smaller than p for any putative underlying probability P_{binom}.

4.4 Results

The classification performance of the SVMs, based on single trial classification, were used as an estimate for the amount of (usable) information about the presented stimuli (classes of presented shapes) or about the attentional state of the stimuli (whether the attended condition or the non-attended condition was used) contained in the recorded data. For selecting the necessary data for each set of data analyses, the relevant time interval was adapted to the regarding scientific question.

4.4.1 Discriminating shapes

The analysis started with the initial shape. In this segment of each trial, the animal has We started our analysis with classifying to memorize the target which it has to compare to the test stimuli recognised during the rest of the trial. The wavelet coefficients were selected from the time window 650-2200ms after trial start, and averaged over time. Using these coefficients from all electrodes of the implanted electrode array allowed to identify 93.1% of the inital stimuli correctly for monkey F and 84.9% for monkey M on the corresponding test data sets. These performances were evaluated using the trials from the attended condition. The chance level for classifing the six different initial shape classes were 18.1% for monkey F and 18.5% for monkey M, respectively.

In Fig. 4.14 and Fig. 4.15 the classification performance of signals from individual electrodes are shown. The figures reveal that two clusters of electrodes contribute the main explanatory power. One cluster was located above area V4 and the other one above area V1. Using the most discriminative set of four electrodes from the V4 cluster (marked by yellow crosses, for monkey F in Fig. 4.14a and for monkey M in Fig. 4.15a) resulted in a classification performance of 64.2% for monkey F and 54.0% for monkey M. A similar analysis was done with three electrodes from the V1 cluster (marked by green crosses, for monkey F in Fig. 4.14a and for monkey M in Fig. 4.15a). The signals of this electrode set from area V1 provided a classification rate of 85.6% (monkey F) and 79.4% (monkey M), respectively.

The classification performance of the signals from single electrodes from the V4 cluster reached up to 42.2% (monkey F) and 35.4% (monkey M). The electrodes from the V1 cluster with the most explainatory power reached 67.8% (monkey F) and 49.2% (monkey M). Again, this performance values were achieved by using the trials from the attended condition.

Fig. 4.16 shows the classification performances for the individual electrodes and trials in the attended-condition for the end period, where the animal has to recognize the reocurrence of the memorized shape and has the release the lever. Fig. 4.16a shows the performance of monkey F and Fig. 4.16b of monkey M. It has to be noted that the time window was a lot smaller than in the examples for the initial period with a 1550ms time window. For the end period the time window was beginning with stimulus onset of the last shape and had a length of 400ms. The cluster above area V4 revealed a classification performance of 49.1% for monkey F and 41.7% for monkey M, while the chance level was 18.0% (monkey F) and 18.6% (monkey M). The electrode cluster above V1 allowed to identify 80.5% (monkey F) and 67.3% (monkey M) of the shapes correctly.

Taken together, the results demonstrate that field potentials recorded at the surface of the dura are highly specific for the individual stimuli processed in the cortical columns underneath the electrodes.

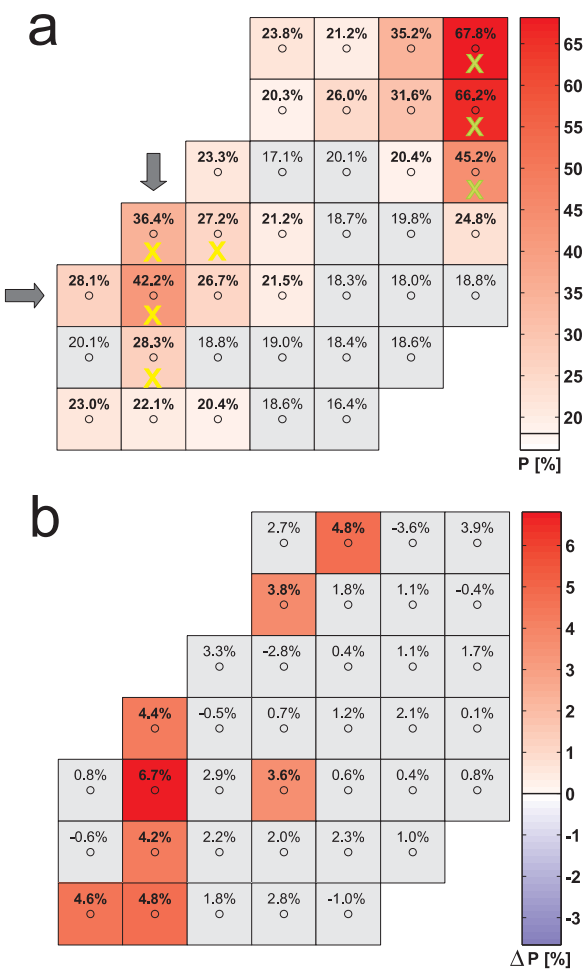


Figure 4.14: (a) Classification performance P of the initial shape s(T1) (data analysed from t=650 ms to t=2200 ms, see Fig. 4.10) for Monkey F in the attended condition. P is shown in dependence on the position of the electrodes in the array (small circles). The performance level is colour-coded according to the bar shown to the right of the array. For the grey coloured squares, classification performance did not differ significantly (p=0.02) from the chance level of 18% (indicated by the black horizontal line in the colour bar). Classification performance reaches peak values of 67.8% and 42.4% in two regions corresponding to areas V1 and V4, respectively (see Fig. 4.11). Grey arrows mark the main V4-electrode showing the highest performance. The combinations of electrodes in V4 selected for the further computational analysis are marked with yellow crosses (V1, green crosses). (b) Difference in classification performance between attended and non-attended stimuli, same display as in (a). The grey squares indicate electrodes where either the differences in performance deviated not significantly from zero, or where the performance under attention was not significantly different from chance level (p=0.02).

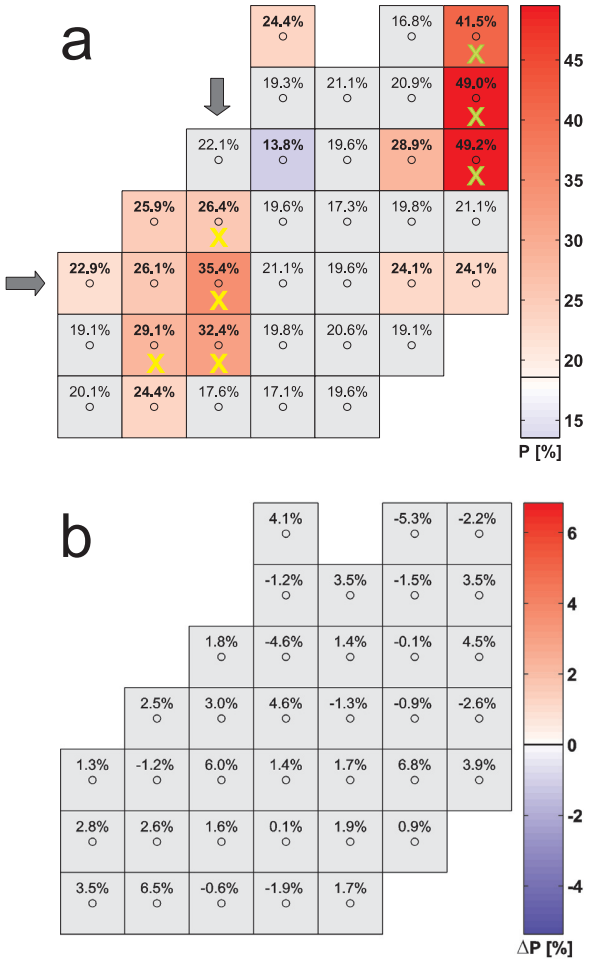


Figure 4.15: (a) Classification performance P of the initial shape s(T1) (data analysed from t=650 ms to t=2200 ms) for Monkey M in the attended condition. P is shown in dependence on the position of the electrodes in the array (small circles). (b) Difference in classification performance between attended and non-attended stimuli, same display as in (a). (see Fig. 4.14 for a more detailed description.)

Figure 4.16: Classification performance P of the 'end' shape (data analysed from a time window beginning with stimulus onset of the last shape and a duration of 400 ms) for monkey F (a) and monkey M (b) in the attended condition. P is shown in dependence on the position of the electrodes in the array (small circles).

4.4.2 Improvement of classification performances through attention

Up to now, results were reported only for trials from the attended condition. In the following, the focus will lie on the differences, between classification rates obtained from trials of the attended and the non-attended condition. During the initial period (650-2200 ms) the classification rate for the selected set of four electrodes from the V4 cluster was significantly improved for trials from the attended condition in comparison to the other condition. For monkey F, the performance increased from 55.5% to 64.2% ($p < 3E-06$, binomial test) and monkey M the performance improved from 41.6% to 54.0% ($p < 5E-04$) through attention. Examining the electrode from the V4 cluster displaying the most explanatory power under attention reveals a performance increase from 35.4% to 42.2% ($p < 3E-04$, monkey F) and from 29.5% to 35.4% ($p < 0.08$, monkey M). In Fig. 4.14b and Fig. 4.15b the absolute differences in classification performance for all single electrodes are shown. For monkey F significant differences cluster around the highly discriminative region in V4. A few scattered electrodes also reach significant differences, but only for very low classification rates close to chance level. For monkey M, the differences in classification performance between the trials from the attended and the non-attended condition reached values up to 6.5% and the electrodes from the V4 cluster showed a similar tendency as observed in monkey F. Due to the lower total number of trials recorded from monkey M, none of the single electrodes showed a significant difference (see Fig. 4.15b). Furthermore, in the data from both animals no significant differences were observed for electrodes located over V1.

Since it was necessary for the animals to attend to all stimuli presented during a stimulus sequence, an attention-dependent interaction between classification performance and the condition of attention can also be expected beyond the initial period. A similar effect like in the initial period was indeed found during all test stimulus presentations. For the final test period (end period), which is associated with the reoccurence of the memorized shape and the correct behavioural response, classification performance (the four electrode sets from the V4 clusters) measured in a 400ms time window (starting with stimulus onset) rose through attention from 40.6% to 49.1% ($p < 2E-04$, monkey F) and from 23.1 to 41.7% ($p < 7E-06$, monkey M). Chance levels were for monkey F 18.0% (a-C) / 18.3% (n-C) and for monkey M 18.6% (a-C) / 19.3% (n-c).

In Fig. 4.17 and Fig. 4.18, the stability of stimulus specific information and its attention-depended enhancement over time is demonstrated. The figures show the time course of classification performance for monkey F (see Fig. 4.17) and monkey M (see Fig. 4.18). In particular, for the attended condition there is little indication that stimulus on- or offset contain much more information on stimulus identity than the static periods between them. This shows that stimulus-specific activity patterns occur persistently while stimuli are presented. The curves for monkey M are more noisy due to the fewer trials performed during the experiment.

This leads to the question whether the specific signal characteristics, which support identification of presented shape by the resulting recorded neural activities, are similar over the whole trial or whether they change over time. This question will be examined by training SVMs on data from one stimulus presentation period and then use these SVMs on test data from another period. In the case that the classification performance is stable under this interchange, it can be assumed that therelevant signal characteristics remain unchanged.

Using this idea, one SVM was trained with the data from the first 400ms after stimulus onset in the initial period. A second SVM was trained on the data from the pre-terminal period (the period preceding the end period, using the first 400ms after stimulus onset of the test stimulus). Both SVMs were applied on test data sets from the initial and the pre-terminal period. The results are shown in Fig. 4.19. The performance values show that successful classification is also possible for test data taken from trial periods far apart in time. In general, the performance for different training and classification periods is comparable but somewhat smaller as compared to the performance achieved with test and training data from the same period. This indicates that the characteristics of the signals supporting stimulus identification are stable over time.

Another question is whether this attention-dependent enhancement of discriminability in cortical states is behaviourally relevant. If such a correlation exists, it can be expected that failures of such enhancements may result in behavioural errors. This hypothesis was tested by evaluating the classification rate in trials which were terminated by a behavioural error.

For trials in which monkeys responded to a wrong test stimulus or failed to respond to the test stimulus matching the sample, classification performance in the stimulus period immediately preceding the erroneous response fell significantly under the level achieved in correct trials. In monkey F, classification performance for the electrode combination in V4 was reduced significantly from 49.1% to 36.2% ($p < 1E-05$), which is even less than the 40.6% observed for correct trials in which no attention was paid to the stimulus. Similarly in monkey M, classification performance fell from 41.7% to 29.0% ($p < 0.04$). No significant difference in classifying non-attended shapes was found between the trials with correct and wrong responses (monkey F). Performance for non-attended stimuli in trials with a correct response in monkey M was 23.1% and again there was no significant difference to performance in error trials.

A similar reduction of the classification performance in error trials was found also for the temporally much earlier initial period. Here performance reduced from 64.2% to 51.9% ($p < 5E-06$, monkey F) and from 54.0% to 46.1% ($p < 0.18$, monkey M). These findings indicate a close relationship between attention-dependent enhancements of the discriminability of the cortical states associated with different stimuli and the behavioural performance in the delayed-match-to-sample-task.

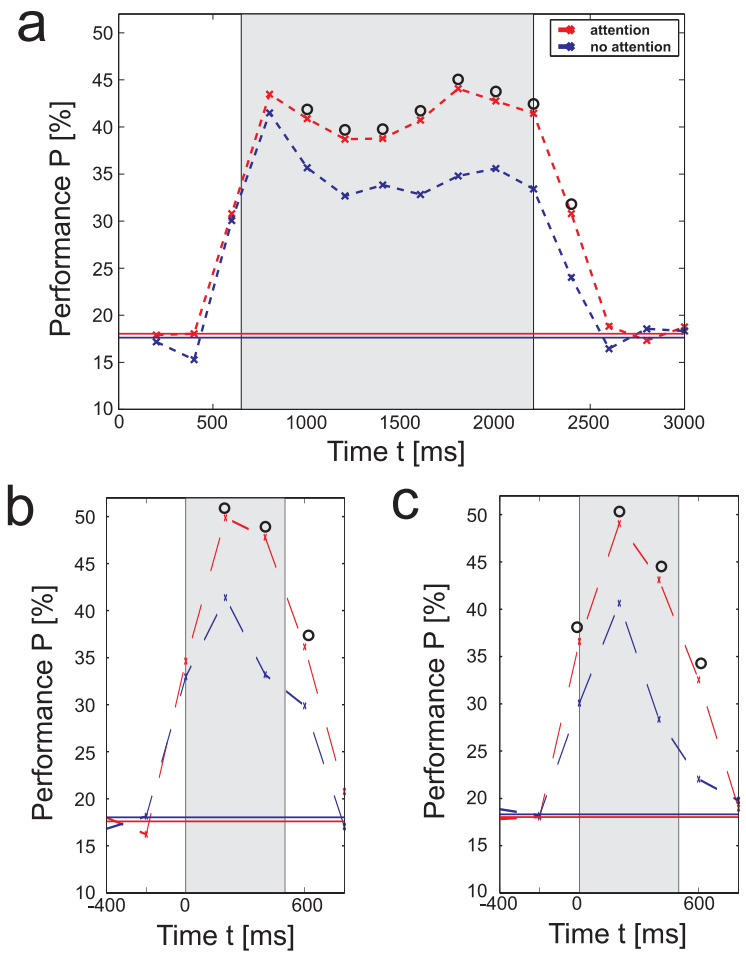


Figure 4.17: (a) Time course of classification performance for the selected set of electrodes above V4 (cf. Fig. 4.14a, yellow crosses), shown for the attended (red dotted line) and for the non-attended condition (blue dotted line). Data for the power coefficients in a frequency range between 5 and 200 Hz was taken from a time interval starting 200 ms before, and ending 200 ms after the times marked with the red and blue crosses, respectively. The black circles indicate a significant difference between the performances in both conditions (p=0.02), while solid lines depict the chance level for the corresponding condition. In (a), the SVM's were trained to classify the initial shape s(T1) presented to the monkeys during the period T1 shaded in light grey. Time t is measured relative to trial onset. In (b) and (c), the SVM's were trained to classify the second-last and the last shape (target) displayed in the sequence, respectively (stimulus display periods are again shaded in light grey). Time t is measured relative to the onset of the second-last shape in (b), and relative to the onset of the target shape in (c).

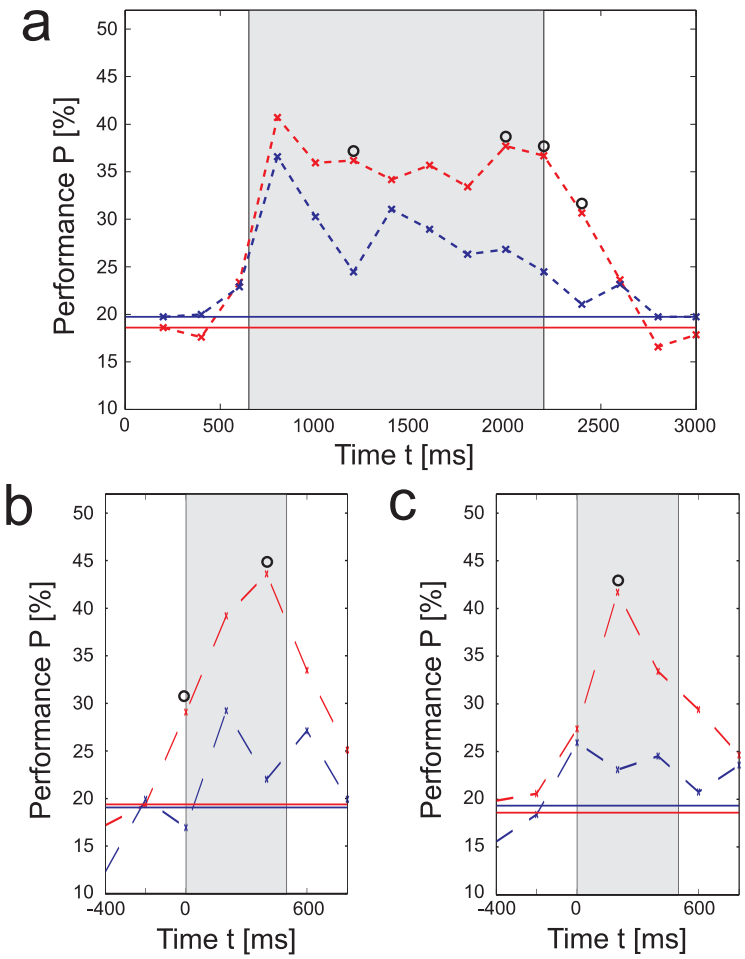


Figure 4.18: Time course of classification performance for Monkey M using the selected set of electrodes above V4 (see Fig. 4.15, yellow crosses), shown for the attended (red dotted line) and for the non-attended condition (blue dotted line). (see Fig. 4.17 for a detailed description.)

		data source for training SVM	
		initial period	**pre-terminal stimulus**
data source for classification	**initial period** chance level 18.1% (18.5%)	43.7% (38.7%)	37.5% (35.2%)
	pre-terminal stimulus chance level 17.6% (19.4%)	36.4% (33.5%)	49.9% (39.2%)

Figure 4.19: Similarity of stimulus-specific activity patterns supporting classification along trials. The table shows classification performance for data from the period for which the SVM was trained, in comparison to classification performance on data from a different period (shaded in grey) obtained from the selected V4 electrode combination (a-C, monkey F). Corresponding values for monkey M (a-C) are shown in brackets.

4.4.3 Stimulus-specific signals and coding

All presented results are based on all of the 17 wavelet coefficients as input for the SVMs. In the following, it will be discussed how the explanatory power is distributed over the 17 wavelet coefficients. For tackling this question, we test how classification performance changes when selecting different subsets of these coefficients as input data for the SVMs. Fig. 4.20a and Fig. 4.21a show the accuracy of classification in dependency of the selected intervals of the frequency spectrum, using data from the selected electrode combination over area V4, during presentation of the initial stimulus. The entries in both figures represent the classification rate achieved with the indicated number of wavelet coefficients (see the vertical axis) from a continuous frequency band interval. The upper frequency of these frequency bands is indicated on the horizontal axis. The figures show that most explanatory power for discriminating the shapes is located in the frequency range above 40Hz. For monkey F, Fig. 4.20a shows that maximally achieved classification rates of 64.6% can be approximated by only six of the 17 wavelet coefficients. The corresponding six wavelet coefficients were selected from the frequency interval between 38 Hz and 122 Hz. Furthermore, a classification performance of 59% is still possible using only three coefficients (with centre frequencies at 61 Hz, 76 Hz, and 96 Hz). Similar distributions of classification rates over certain intervals in the frequency spectrum, can be found in both animals for the attended as well as for the non-attended condition. The attention-dependent enhancement (see Fig. 4.20b and Fig. 4.21b) is largest within a similar frequency range (38 Hz - 122 Hz).

Another question is whether most of the explanatory power is contained in the spectral distribution of the wavelet coefficients or in the signal energy. Using the data from the V4 electrode with the highest classification rate, the signal energy was removed from each trial by dividing the selected wavelet power coefficients by their sum. In monkey F, classifying on the remaining features reduced performance over chance level on average by 3.1% (26.2%, monkey M). For comparison, classification was also performed on the signal energy, which was calculated by summing the selected wavelet power coefficients of each trial. Classifying on the signal energy reduced performance over chance level on average by at least 45.8% (monkey F) and 12.9% (monkey M), respectively, when selecting the most informative frequency range in terms of signal energy. Thus the discernability of field potentials caused by different stimuli is based on stimulus-dependent differences of spectral activity patterns and overall energy in the gamma-band. In monkey F, information in the spectral patterns is even predominant over information in signal energy.

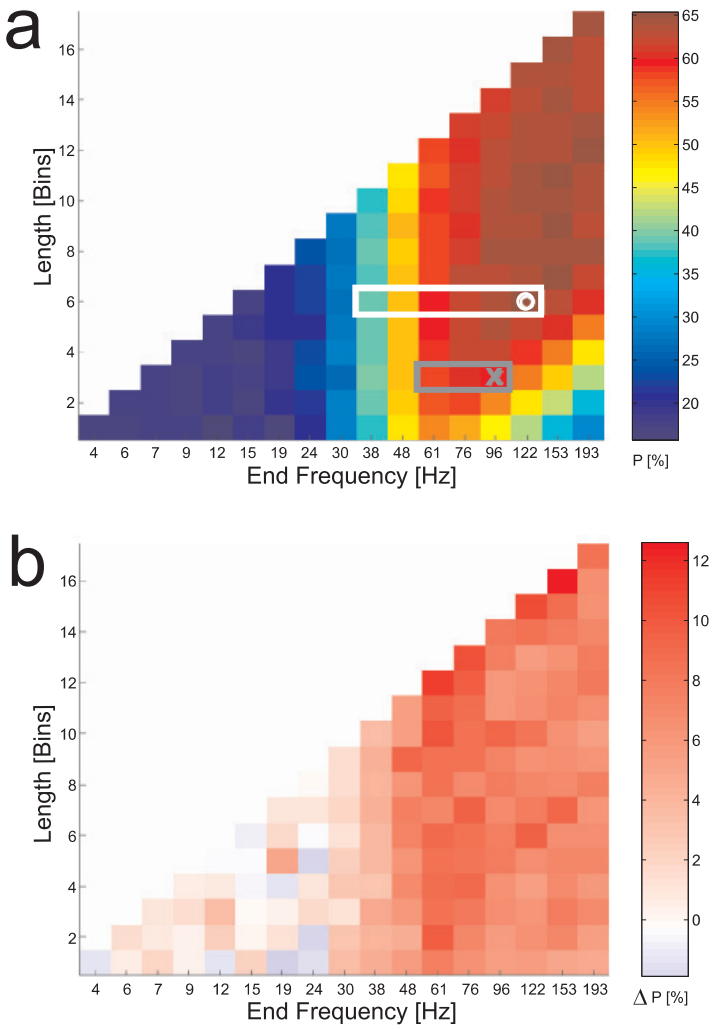


Figure 4.20: (a) Classification performance P for Monkey F, using different subsets of the power coefficients from the selected V4 electrode combination obtained during the initial period T1 (650 ms to 2200 ms after trial onset) under attention. Each square shows in colour-code the SVM performance on a combination of successive frequency bands, whose total number is indicated by the index on the vertical axis.The upper frequency band is indicated by the horizontal axis. For example, the performance value marked by the white circle was obtained using data from six frequency bands starting at frequency 38 Hz, and ending with 122 Hz (symbolised by the white rectangle). The performance shown marked by the grey cross was obtained using data from only 3 frequency bands at 61 Hz, 76 Hz, and 96 Hz (symbolised by the grey rectangle). (b) Percentage of increase in classification performance under attention, for the same combinations of frequency bands as in (a).

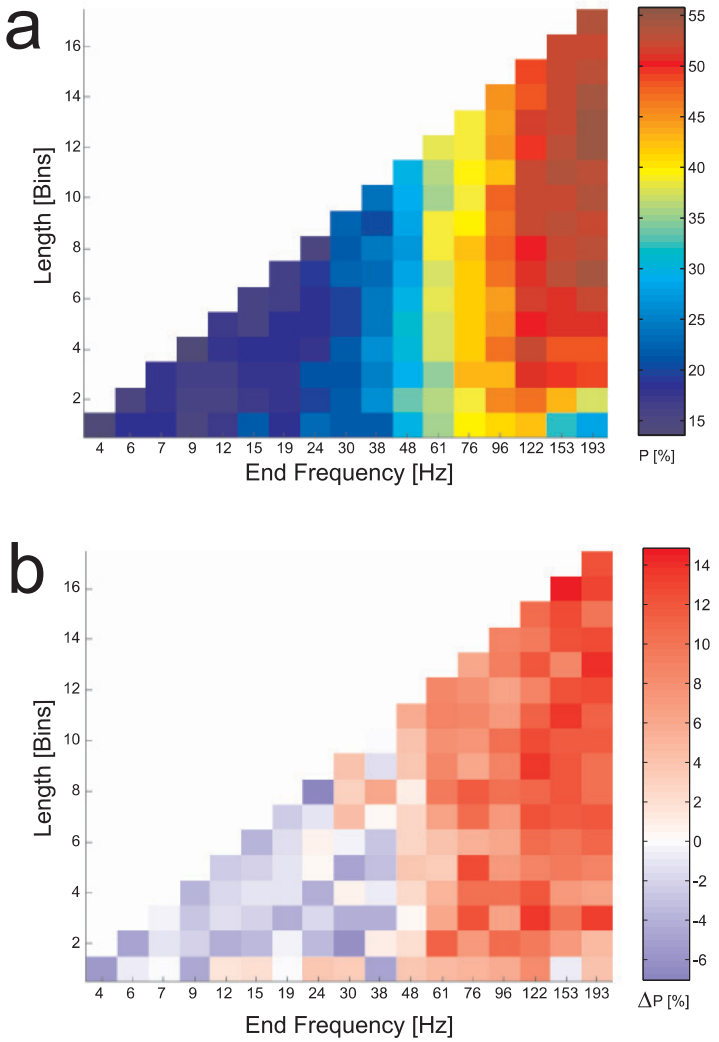


Figure 4.21: (a) Classification performance P for Monkey M and the corresponding attentional gain, using different subsets of the power coefficients from the selected set of V4 electrodes. (see Fig. 4.20 for a detailed description for the representation used)

4.4.4 Attention induced stimulus-specific signals changes

The preceding data-analysis revealed that attention improves the discriminability of the stimuli significantly. The question is how the signal characteristics of the neural responses become more distinct under the influence of selective attention.

The perfect discrimination of two stimuli classes requires that the corresponding signals fill different regions in data space. Errors made during classification can be related to data samples that fell into regions in data space which are occupied by more than one stimuli class. The number of errors made is related to the relative overlap between these regions (Fig. 4.22a). For reducing the number of errors, it is necessary to remove the overlap between these regions. One way to reduce the overlap is to shrink the regions in diameter (Fig. 4.22b). A second one is to increase the distance between the centres of the regions (Fig. 4.22c).

The mentioned shrinking effect can be produced by increasing the signal-to-noise-ratio (SNR). An increase of distance between the centers of the regions can be accomplished by a class-specific multiplicative scaling with constant SNR. It has to be noted that the combination of both effects and also including more complicated statistical changes in the data, could be used to further reduce the number of classification errors. For this data sets it turns out that it suffices to quantify these two basic effects to explain most of the effects of attention.

Analysing the distributions of the averaged wavelet power coefficients $\bar{a}_j(f_0)$, only small changes in the SNR η

$$\eta(f_0) = \frac{\mu(f_0)}{\sigma(f_0)}$$

and in the mean values μ

$$\mu(f_0) = < \bar{a}_j(f_0) >$$

were found. σ is defined by the standard deviation

$$\sigma(f_0) = \sqrt{< \bar{a}_j(f_0) - \mu(f_0) >}.$$

Fig. 4.23 (monkey F) and Fig. 4.24 (monkey M) show changes in the SNRs η and the scaling of the mean wavelet power coefficients μ through attention. The scatter-plots Fig. 4.23a and Fig. 4.24a demonstrate clearly that the changes in SNR are only minor, while the mean coefficients for the different shapes are scaled much stronger through attention (see Fig. 4.23b and Fig. 4.24b). The differential scaling is visible in the differences between the curves attributed to different shapes. The insets quantify the gain in classification performance under attention, demonstrating that it is not the strength of scaling for a single frequency band, but the concerted effect of changes in multiple frequency bands, which is causing the improvement in performance. Again, data from both monkeys shows qualitatively the same behaviour.

For the SNRs, an averaged change of $< \eta^A/\eta^N - 1 >= 4 \pm 7\%$ (monkey F) and $-1 \pm 9\%$ (monkey M) were calculated. For the mean values an absolute average change of $<| \mu^A/\mu^N - 1 |>= 14.9 \pm 10.4\%$ (monkey F) and $3.3 \pm 4.2\%$ (monkey M) were found (A: attended condition, N: non-attended condition, averages $< \ldots >$ are taken over all frequency bands and stimuli classes).

As it was outlined before, both changes may be responsible for the enhanced performance under attention. The average values alone do not allow to clarify to which extent the small changes in the mean SNR might explain the full attentional gain. Furthermore, it is not clear whether, by the differential scaling of the means, the class regions are really shifted away from each other than towards each other.

For quantifying the influence of the SNRs and mean values on the classification performance, two tests were performed on the data from the non-attention condition: In the first one, the SNR's were changed such that they match the SNR's of the data from the attended condition, while holding the mean values of the data set constant. This transformation was performed using the equation

$$a_{j,s}^{(I),qA}(f_0) = \mu_s^N(f_0) + \left(a_{j,s}^N(f_0) - \mu_s^N(f_0)\right) \frac{\eta_s^N(f_0)}{\eta_s^A(f_0)}\,. \tag{4.3}$$

A SVM was then trained and tested on this 'quasi'-attended data set to quantify how the separation of the shape classes improved through this transformation. In the second test, the SNR's of the data set were kept constant. This time the data was transformed such that the mean values are the same like the mean values of the data set from the attended condition.

$$a_{j,s}^{(II),qA}(f_0) = \mu_s^A(f_0) + \left(a_{j,s}^N(f_0) - \mu_s^N(f_0)\right) \frac{\mu_s^A(f_0)}{\mu_s^N(f_0)} \tag{4.4}$$

Again, a SVM was trained on this new 'quasi'-attended training data set and the classification performance was evaluated on the new 'quasi'-attended test data set.

A similar, 'inverse' test was performed on the attended data set using the transformation

$$a_{j,s}^{(I),qN}(f_0) = \mu_s^A(f_0) + \left(a_{j,s}^A(f_0) - \mu_s^A(f_0)\right) \frac{\eta_s^A(f_0)}{\eta_s^N(f_0)} \tag{4.5}$$

for changing the SNRs while retaining the mean wavelet coefficients, and

$$a_{j,s}^{(II),qN}(f_0) = \mu_s^N(f_0) + \left(a_{j,s}^A(f_0) - \mu_s^A(f_0)\right) \frac{\mu_s^N(f_0)}{\mu_s^A(f_0)}\,, \tag{4.6}$$

for changing the mean wavelet coefficients while retaining the SNRs. These two 'quasi'-non-attended data sets were then classificated with SVMs for determining the degradation of classification performance. Improvements and loss in performance were finally compared to the real differences in classification performance on the original data, and

expressed in percentages of these original differences being explained by the two scaling procedures.

Fig. 4.25 and Fig. 4.26 show the results using these scaling procedures for the V4 electrode combination in dependency of the selected frequency interval. Starting at 40 Hz the higher frequencies show the strongest effect of modulation. Scaling the SNRs produces only a minor increase in classification performances. In monkey M, the explanatory power of the data analyses using the scaling procedure is much weaker, showing the limiting effects of the trial statistics. Fig. 4.26 confirms the result displayed in Fig. 4.25 by downscaling the attention data set into the two 'quasi'-non-attended data sets. In summary, scaling the SNR explains only 1.6% (monkey F) and 28.7% (monkey M) of the original increase in performance under attention. Under the other transformation, where the SNRs are kept constant and the mean values were scaled, it was possible to explain 108.4% (monkey F) and 50.2% (monkey M). The result clearly indicates that attentional gain in performance is only to a minor extent caused by changes in SNR, but is to a large extent explained by shape-specific differential scaling of frequency components rendering the neural activity for different shapes more distinct from each other. This finding reveals a new mechanism of attention acting on coherent neuronal activity in area V4.

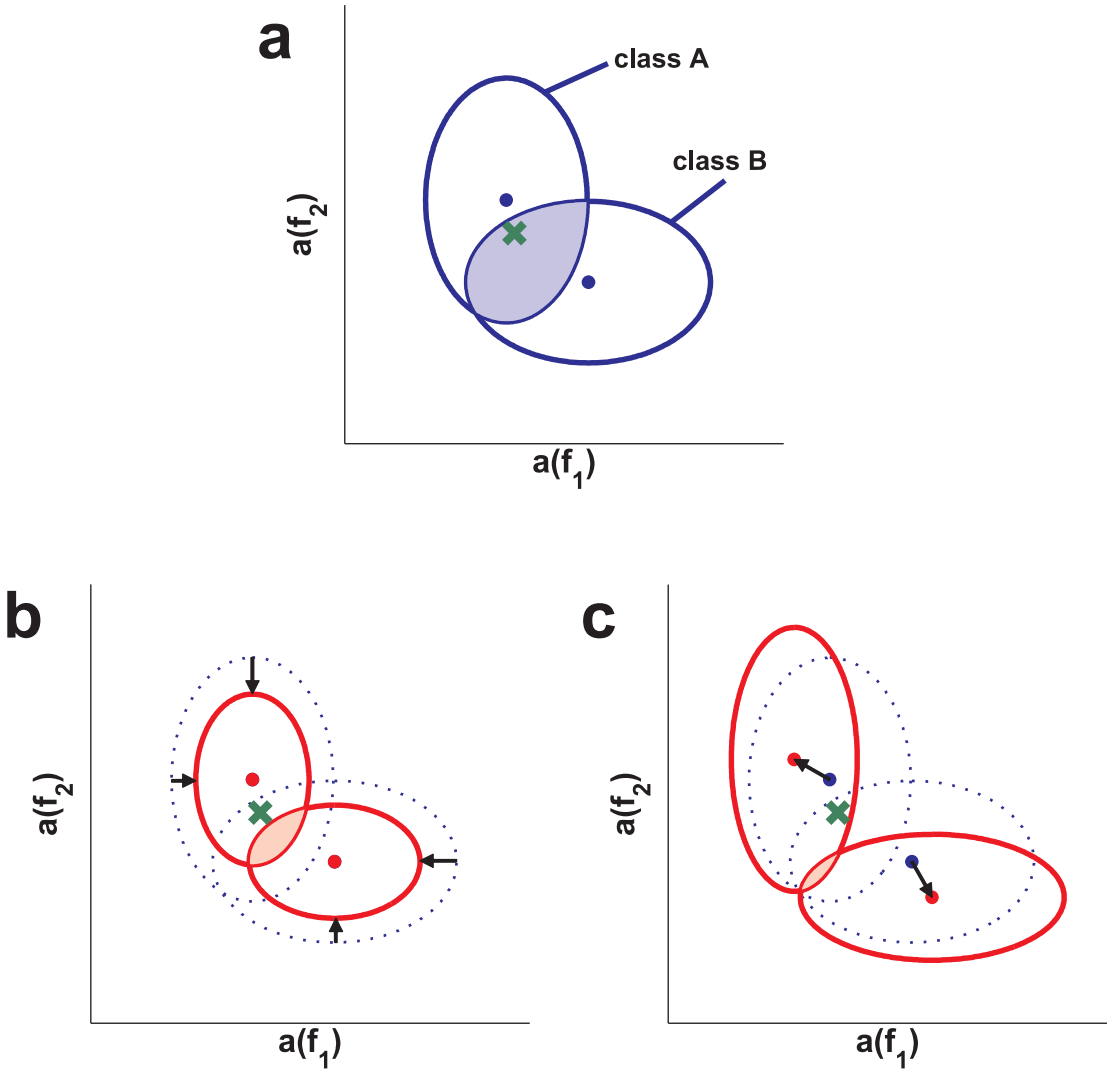


Figure 4.22: Examples for classification problems in a two-dimensional data space spanned by the variables a(f1) and a(f2) which could represent the wavelet coefficients for two different frequency bands. The regions indicated by the blue ellipsoids in (a) symbolize data from two classes' A and B. These two classes would correspond to ensembles of data points obtained by the repeated presentation of two shapes. When a new data point in the shaded region is observed (green cross), any classifier trained on the previously observed data is likely to make an error because the data may belong to either of the two classes. The total number of errors thus corresponds to the relative size of the shaded region where these classes overlap. (b) If attention would decrease the SNR, as indicated by the class boundaries shrinking around their centres (red ellipsoids), the same observation could now unambiguously be attributed to class A thus reducing the classification errors (shaded region). (c) If instead attention shifts the region centres (arrows), this change can likewise disambiguate the classification problem and reduce the number of errors (shaded region), even when the SNR remains constant.

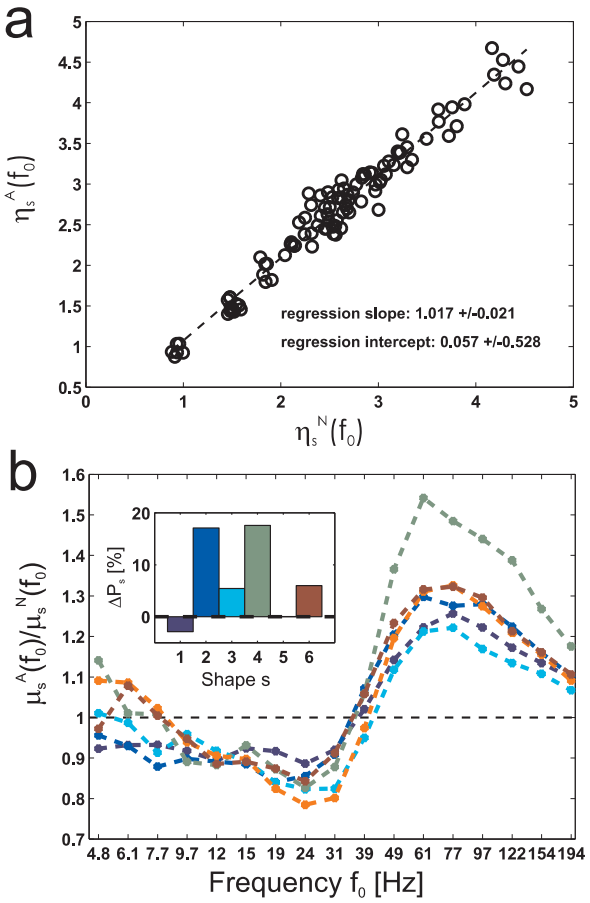


Figure 4.23: (a) Signal-to-noise ratios (SNR) of wavelet coefficients in the attended condition, $\eta_s^A(f_0)$, plotted against the SNR in the non-attended condition, $\eta_s^N(f_0)$, for monkey F. The SNRs were computed for each frequency band f_0 and shape s during the initial presentation period T1 (650ms to 2200ms). The data can be fitted with a linear regression (dashed line), resulting in a coefficient of 1.017 ± 0.021 for its slope. (b) Scaling factors or ratios $\mu_s^A/\mu_s^N(f_0)$ of the mean wavelet power coefficients for the attended and non-attended conditions, in dependence on the pattern class (shape) s and the frequency band f_0. The coefficients were calculated from the data obtained during the initial period from the main electrode in V4, for Monkey F. A scaling factor of 1 indicates no change in the mean power coefficients (dashed black line). The inset shows the relative change in classification performance through attention for each of the six different shapes s.

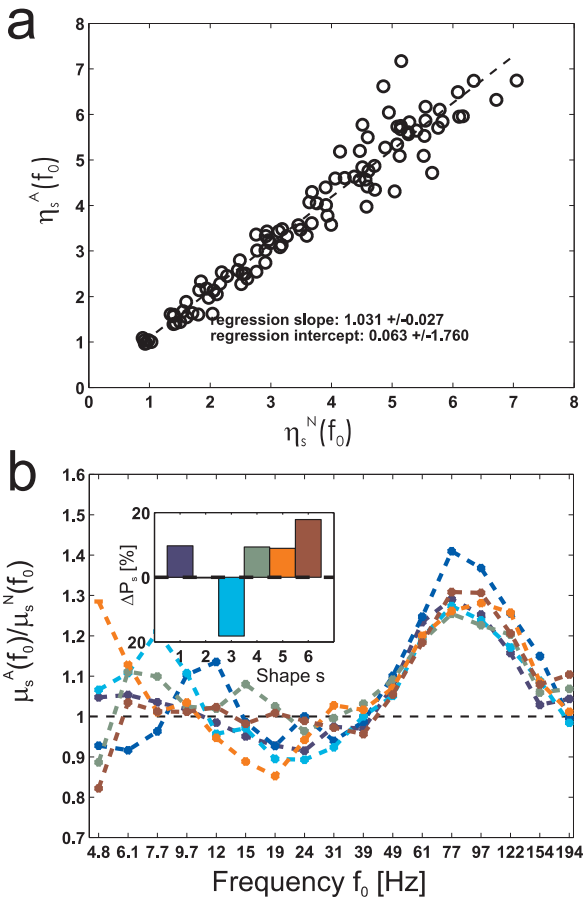


Figure 4.24: (a) Signal-to-noise ratios (SNR) of wavelet coefficients in the attended condition, $\eta_s^A(f_0)$, plotted against the SNR in the non-attended condition, $\eta_s^N(f_0)$, for monkey M. An analog analysis, like explained in Fig. 4.23, reveals a coefficient for the slope of a linear regression with 1.031 ± 0.027. (b) Scaling factors or ratios $\mu_s^A/\mu_s^N(f_0)$ of the mean wavelet power coefficients for the attended and non-attended conditions, in dependence on the pattern class (shape) s and the frequency band f_0, for Monkey M (for more details see Fig. 4.23).

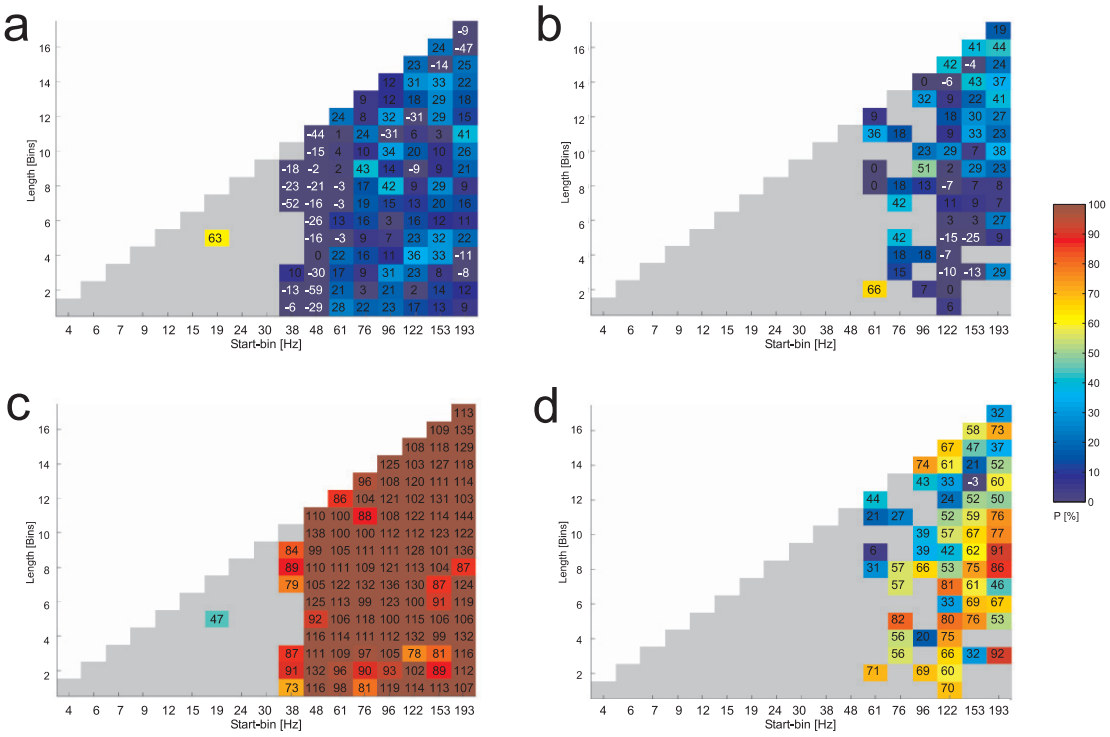


Figure 4.25: Explanatory power of hypothetical scaling of wavelet coefficients under attention. The upper row quantifies the effect of scaling the SNRs of the non-attended data as described in equation Eq.(4.3), shown for monkey F (a) and for monkey M (b). The lower row quantifies the effect of scaling the mean values as described in Eq.(4.4), shown for monkey F (c) and monkey M (d). The performance is plotted for different frequency ranges in the same schematics as used in Fig. 4.20 and Fig. 4.21. The colour code indicates how much of the percentage of the gain in performance under attention is explained by the respective scaling procedure. Values above 100% and below 0% are clamped to this range before being colour-coded (for the original value, see the numbers displayed in each coloured square). Frequency ranges in which there was no significant increase in performance in the original data are displayed in gray.

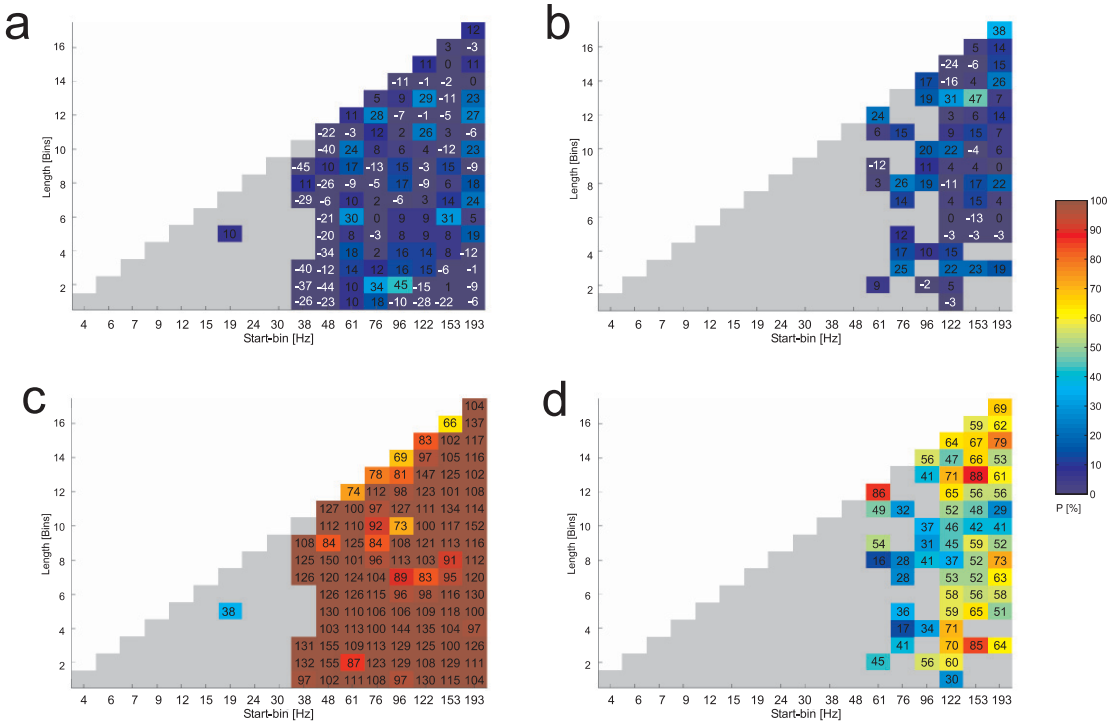


Figure 4.26: Explanatory power of hypothetical scaling of wavelet coefficients under attention. In contrast to Fig. 4.25, the scaling was this time applied to the attended data set, using scaling coefficients as in Eq.(4.5) and Eq.(4.6). Consequently, the plots show how much of the decrease in performance between the non-attended and attended condition can be explained by changes in the SNR (upper row) and by differential changes in the mean wavelet coefficients (lower row), respectively. Same presentation as in Fig. 4.25.

4.4.5 Attention effects in V1

In V1 we found no significant effects of enhancing classification performance through attention. This rises the question why we were not able to find such effects. Thus, we made a ROC analysis (see section 2.2.4) on the wavelet coefficients, in addition to the scaling procedures. Specifically, for any two different shapes and each specific frequency band, we obtain two distributions of wavelet coefficients. A standard ROC analysis can then estimate how well these two distributions can be distinguished from each other, i.e. with which probability one can correctly classify an observation as caused by the one or the other shape, given a suitably decision criterion. This analysis was performed for each combination of the six shape classes and for each frequency band, in the attended as well as in the non-attended condition. The results are summarized in Fig. 4.27 (monkey F) and Fig. 4.28 (monkey M). It turns out that in V1 many combinations of frequencies exist, which allow for an almost perfect discrimination between two shapes already in the non-attended condition (integral over ROC curve between 0.9 and 1), while there are no such combinations of frequencies in V4. In fact, for V4 almost all combinations of frequencies and shapes remain below 0.8. This gives a modulation by attention the necessary room' for increasing the discriminability between the shapes. This dependence can also be seen in the mean attentional gain displayed in the same figure, which is highest for medium and low values of the ROC integral. Taken together, these figures demonstrate why in V1 attention does not have a significant effect on discrimination performance, as revealed by the SVMs in Fig. 4.14b and Fig. 4.15b.

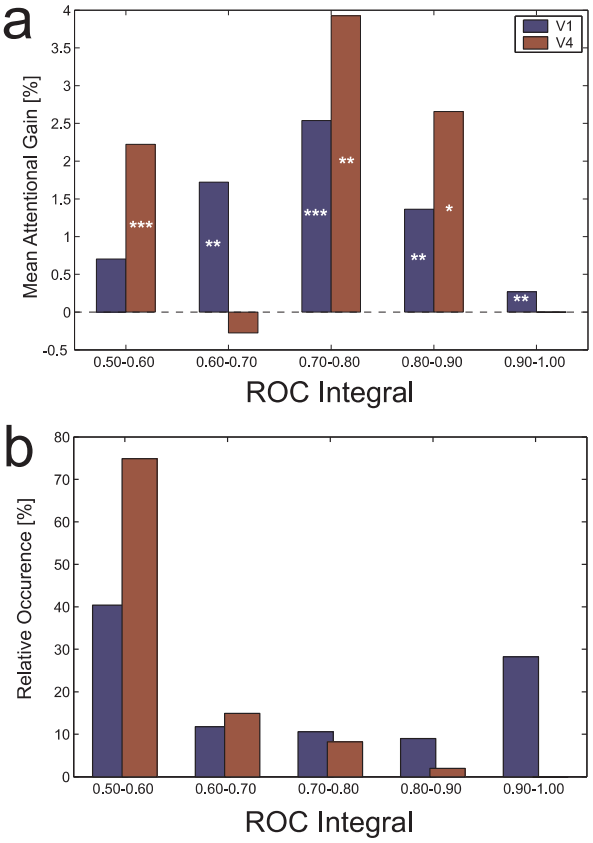


Figure 4.27: Summary of analysis of the receiver-operator-characteristics for any two distributions of wavelet power coefficients. The integrals over the ROC curves quantify how well the wavelet coefficients for one frequency band allowed to distinguish between two shapes in the non-attended condition, taking values between 50% and 100%. In (a) and (b) these integrals were sorted into five performance classes equally spaced, the histogram (b) quantifies the distribution of the occurences of these values for monkey F. The corresponding mean gain in performance under attention is quantified for each of the performance classes in (a). The white stars denote mean gains, which are significantly different from 0, while the number of the stars indicates the significance level in terms of standard deviations (three stars for three and more standard deviations). Data was taken from the electrodes with the highest performances in V1 (blue) and V4 (red).

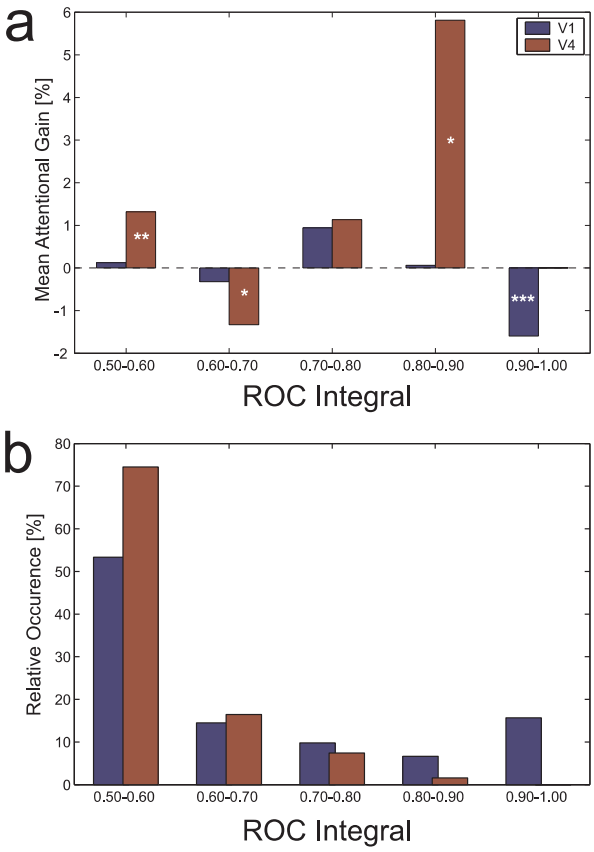


Figure 4.28: Summary of analysis of the receiver-operator-characteristics for any two distributions of wavelet power coefficients, for Monkey M (for details see Fig. 4.27).

4.4.6 Modelling stimulus-specific signals

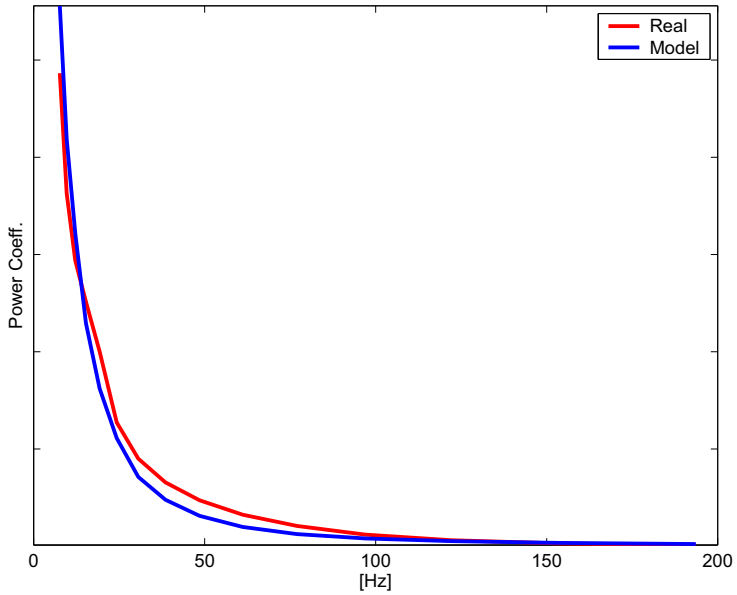


Figure 4.29: The red curve shows the power coefficients of the measured data for the 17 frequency bands before the initial stimulus period, where the screen was still free from shape stimuli. These power coefficients were averaged over the whole training data set (Monkey F). The blue curve displays the corresponding results from a simple computational model. In this model a population of 100 neurons created binomial spike-trains with mean rate of 10 Hz. These spike-trains were averaged and then folded with an artificial post synaptic potential ($PSP(t) = \exp(-t/\tau)$, with $\tau = 20ms$) to mimic the creation of LFP-like signals. For display reasons, the area under each curve was normalised to 1.

One approach to understand more about underlying biological mechanism is to build a computational model and compare the model's output to real measured data. Such a model is typically based on several assumptions. In the following, the electrode of monkey F with the highest classification performance will be used. The goal is to model the mean power spectra for the different shapes and both attentional conditions.

Assuming that the stimulus-specific information, stored in the wavelet coefficients, can be described by mean power coefficients and that the variations over trials are just noise, may in this case allow to search for a simple underlying model. A second assumption for this analysis will be that spike-trains can be transformed into LFP-like singals, just by convoluting the averaged spiketrains from a population of neurons with post-synaptic potentials (PSPs). These PSP's are modeled by an exponentially decreasing function

$$PSP(t) = e^{-\frac{t}{\tau}}.$$

These assumptions are very strong simplifications of the real mechanisms which created the epi-dural local fieldpotentials in the first place.

However, performing this transformation on spiketrains generated by binomial random processes representing a population of neurons, produces a power characteristics that shows similarities to recorded data from the period before the initial stimuli were shown. Fig. 4.29 shows the comparison between the real data and this simple model. The output of the model (except the absolute scale, which was removed) is similar for a large range of mean rates used in the binomial process. The model approximates roughly the real data but was not able to reproduce the recorded data perfectly. This may be due to the fact that the computational model was in detail too simple.

In a next step, the generation of the spike-trains by the population of neurons in the model was replaced by a random process. This random process uses based on gamma distributions for generating the time-interval between two spikes. This type of distribution can be tuned by two parameters. In a large number of simulations, we searched for the best fitting parameter sets which can explain the power wavelet coefficient curves for the six different shapes under the two attentional conditions for the initial period. The results can be found in Fig. 4.30, using the gamma distributions shown in Fig. 4.31. As a control how strongly the results are influenced by the spike-train generating process, another simulation with a population of uncoupled leaky integrate-and-fire (IaF) neurons was also performed. For the IaF neurons, the strength of the input and the amplitude of the additive noise on the membrane potential were used for fitting the model to the stimulus-specific characteristics of the real data. All IaF neurons received exactly the same external input. The noise process created positive/negative values, drawn from a uniform distribution for each neuron and timestep. Fig. 4.32 shows the results.

In comparison, both approaches were able to approximate the shape of the curves. They also show that it is a problem to infer reliable information about the underlying neuronal system only on the basis of the mean power wavelet coefficient curves. It is also not clear which characteristics of the real curves are the important ones. In addition, applying SVMs on the data generated by the models showed extremely (and much too) good classification performances. Comparing real data and data from the model, revealed that the variances of both data sets differ strongly. This can be compensated by applying external noise (e.g. multiplicative noise) to the calculated power wavelet coefficients but this seems not to allow to get new insights.

Taken together, this analysis should remind us that even if a model approximates the data well it may show only weak connections to the real biological process. In this example we have two different models which can reproduce the measured power spectra with nearly the same quality. For a better understanding of the correlations between perceived stimuli and the resulting neuronal activity patterns, data from additional and more specialised experiments are necessary. Another interesting question is whether it is possible to reproduced the changes made by attention in a framework with a biological plausible network where the attentional state is modeled by a global parameter

(e.g. the level of synchronicity in the network).

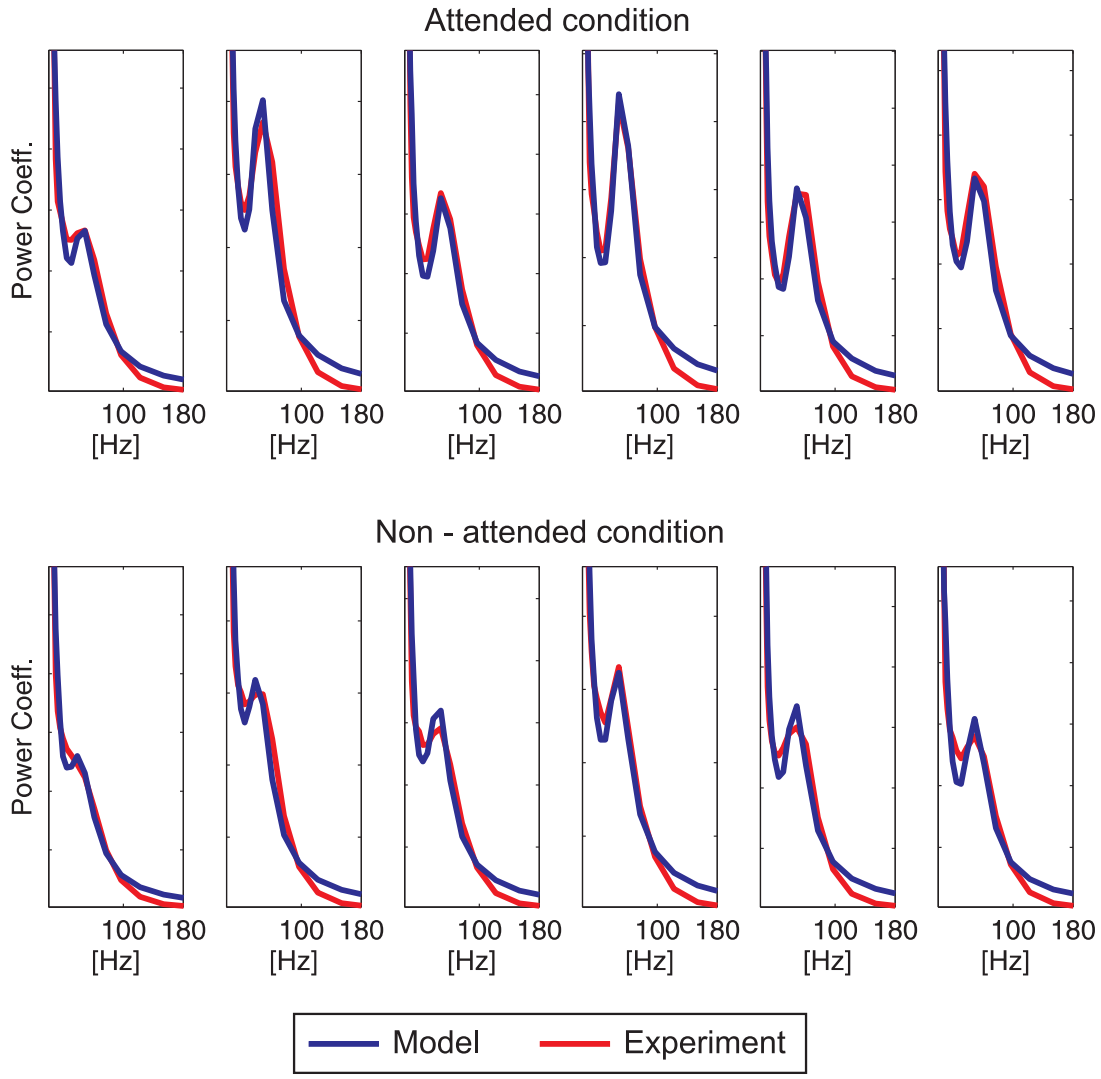


Figure 4.30: Like in Fig. 4.29 the red curves show the mean power coefficients calculated from the wavelet-transformed recorded data (training data set from Monkey F, initial period) for all six different shapes (from left to right). The upper row shows the data for the attended condition while the lower row shows the data for the non-attended condition. The blue curves are the corresponding magnitudes calculated from a simple model where the spike-trains of a population of 100 neurons were averaged and then convoluted with $\mathsf{PSP}(t) = \exp(-t/\tau)$ (with $\tau = 20\mathrm{ms}$), before the power coefficients were calculated. Each of these spike-trains was generated using a gamma distribution for modeling the Inter-Spike-Interval (ISI) distribution. See Fig. 4.31 for the ISI distributions which were used for calculating the blue curves in this figure. Like in Fig. 4.29, the curves are normalised.

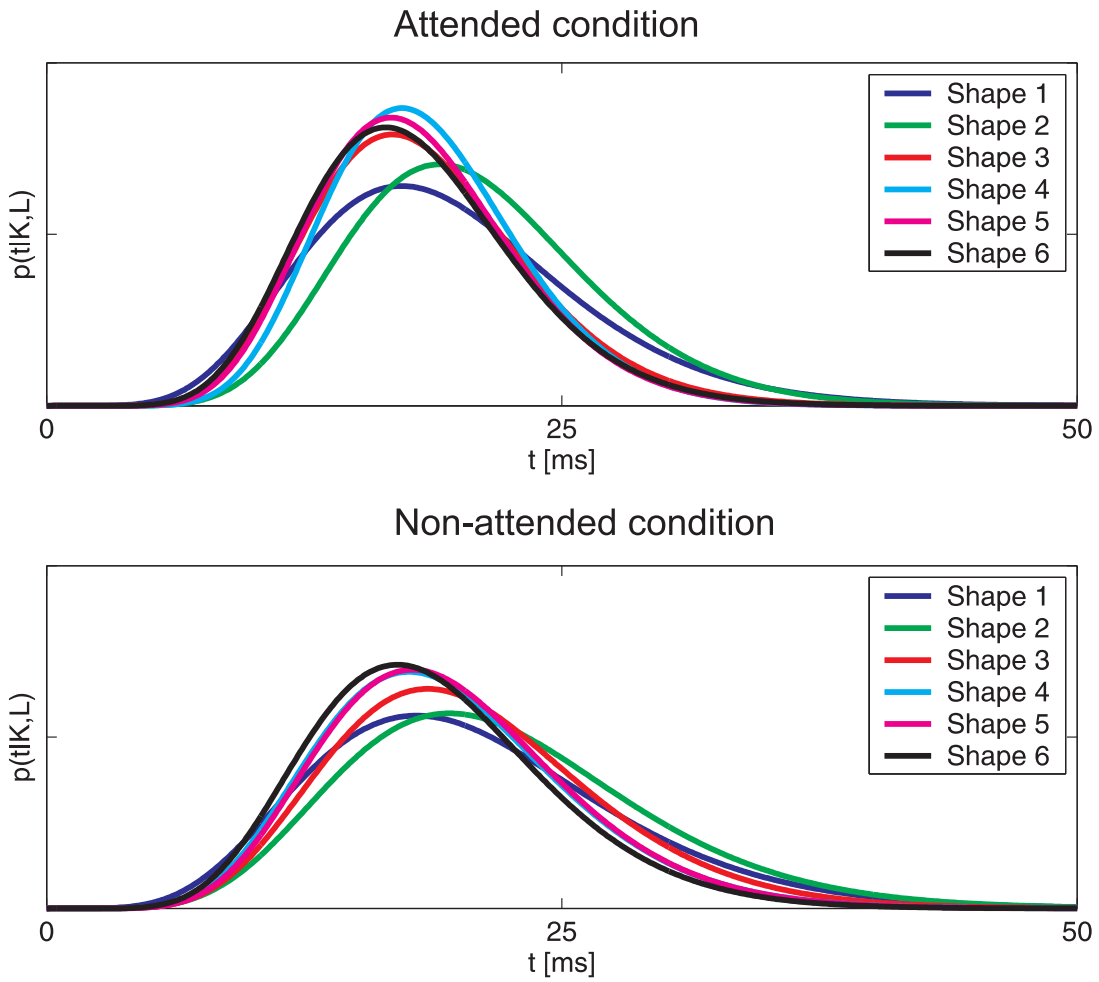


Figure 4.31: Inter-Spike-Interval distributions used for simulating Fig. 4.30. The distributions are gamma distributions $p(t|K,L) = t^{K-1} L^K \exp(-t \cdot L)/\Gamma(k)$ where the K's are taken from $[7.56, 15.25]$ and the L's are taken from $[0.37, 0.83]$. The upper set of curves are for the attended condition and the other set represents the non-attended condition.

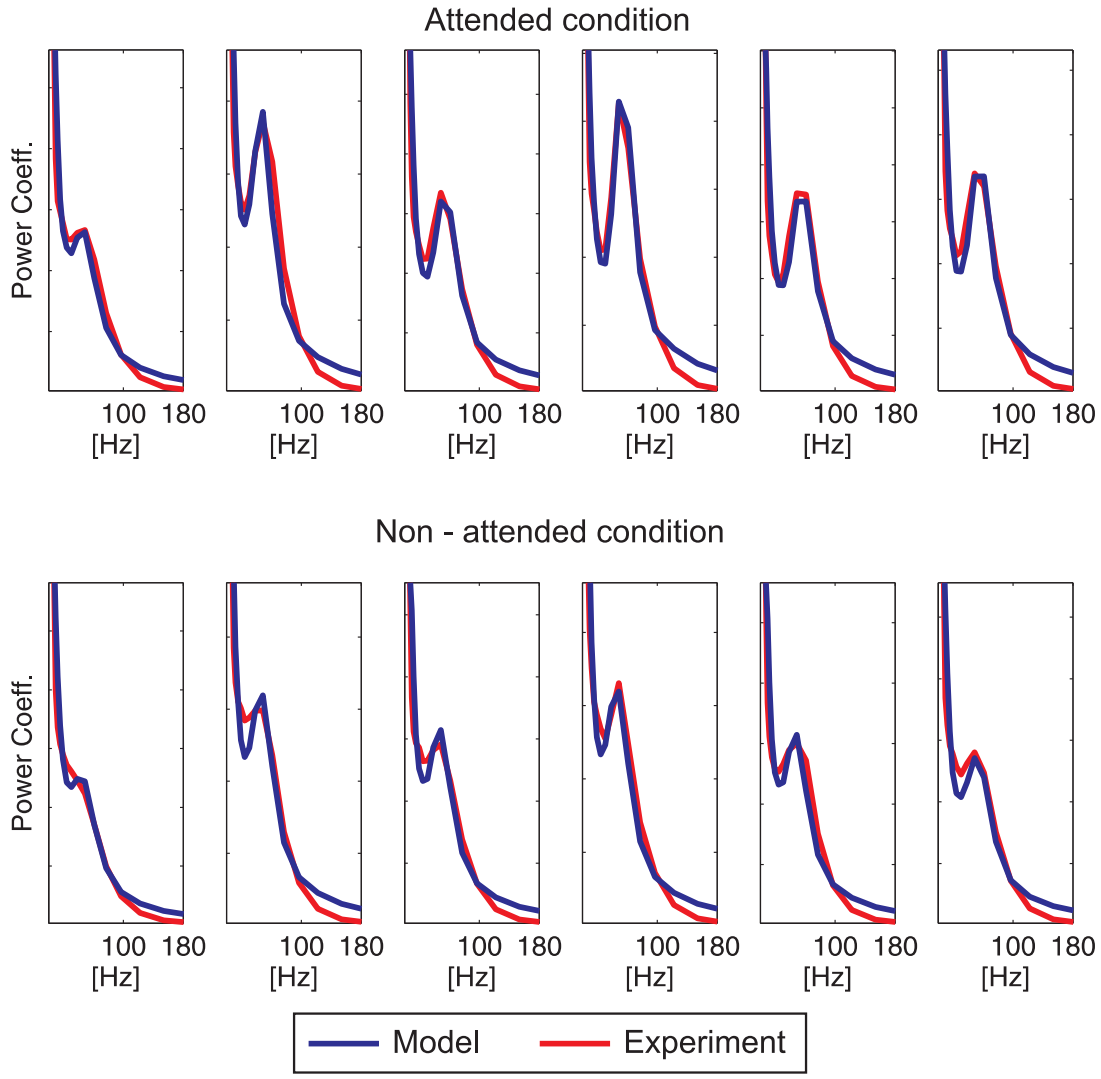


Figure 4.32: The red curves show the recorded power coefficients for the six shapes under both conditions in the initial period. The blue curves show the corresponding results of a model. See Fig. 4.30 for more details how the red curves and blue curves were calculated. For this figure, the spiketrains of the computational model were generated by a population of 100 uncoupled leaky integrate-and-fire (IaF) neurons, instead of using a gamma distribution. The strength of input into the IaF neurons (all neurons received the same input) and the noise amplitude (drawn from a uniform distribution which equally distributed positive and negative values) were used as parameters for fitting the models to the different shapes and conditions.

4.4.7 Discriminating the Attentional Condition

Since our analyses revealed that the different attentional conditions change the stimulus-representation, it is an interesting question whether it is possible to classify the attentional condition on the basis of averaged power wavelet coefficients. In case of high classification performances it would allow to use selective visual attention e.g. as reliable BCI signal for controlling a speller.

In Fig. 4.33 the classification rate of this two-class-discrimination problem for all single electrodes of the electrode array are shown (initial period). While the chance level for this estimation lies around 50%, it was possible to reach a performance up to 73% (Monkey F) / 78% (Monkey M) with single electrodes over the V4 region. The classification rates from the electrodes above V1 are not far from chance level. Using the combinations of V4 electrodes (depicted by yellow crosses in Fig. 4.14 for monkey F and in Fig. 4.15 for Monkey M) a classification performance of 80.3% for Monkey M and 79.4% for Monkey F was achieved (initial period).

The figures Fig. 4.34 and Fig. 4.35 show the time course of classification performance for the sets of V4 electrodes and both monkeys. These pictures reveal two insights:

1. It is possible to classify the attentional condition during the initial and the terminal period with nearly the same precision.
2. Using the data selected from a 400ms time window, the classification rate can be nearly as high as with a large time window (650 - 2200 ms after trial onset).

Quantifying the contributions from different frequency bands, Fig. 4.36 shows that for both monkeys the information about the attentional condition is stored differently in area V4. For reaching a classification rate of 81.5% monkey M requires only the power coefficients from the frequency bands 61 Hz and 76 Hz. For Monkey F, the power coefficients of the frequency band range from 24 Hz to 122 Hz are required to reach a classification performance of 79%.

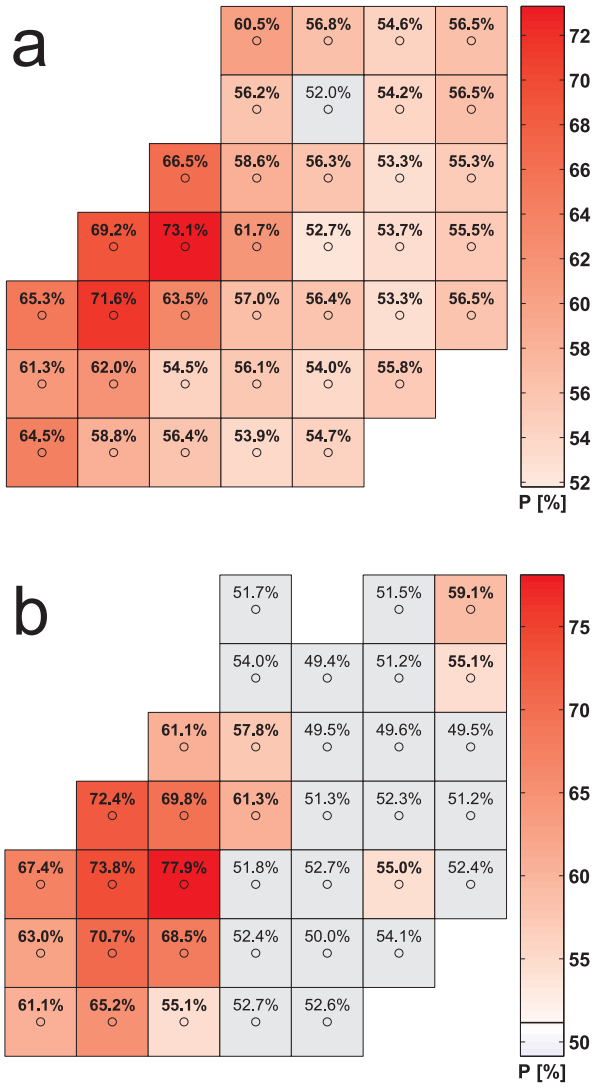


Figure 4.33: Classification performance P for the attentional condition (recorded data used from t=650 ms to t=2200 ms, see Fig. 4.10) for Monkey F (a) and Monkey M (b). The presentation is similar to the one used in Fig. 4.14, including the level of significance (p=0.02) and marking classification performance that did not differ significantly from the chance level with grey coloured squares. The chance level for classifying the two possible classes lies at 50% for Monkey F and 51% for Monkey M. The electrodes over region V4 show a classification rate up to 73% for Monkey F and 78% for Monkey M. The contributions from the electrodes above V1 are in comparison to V4 rather small.

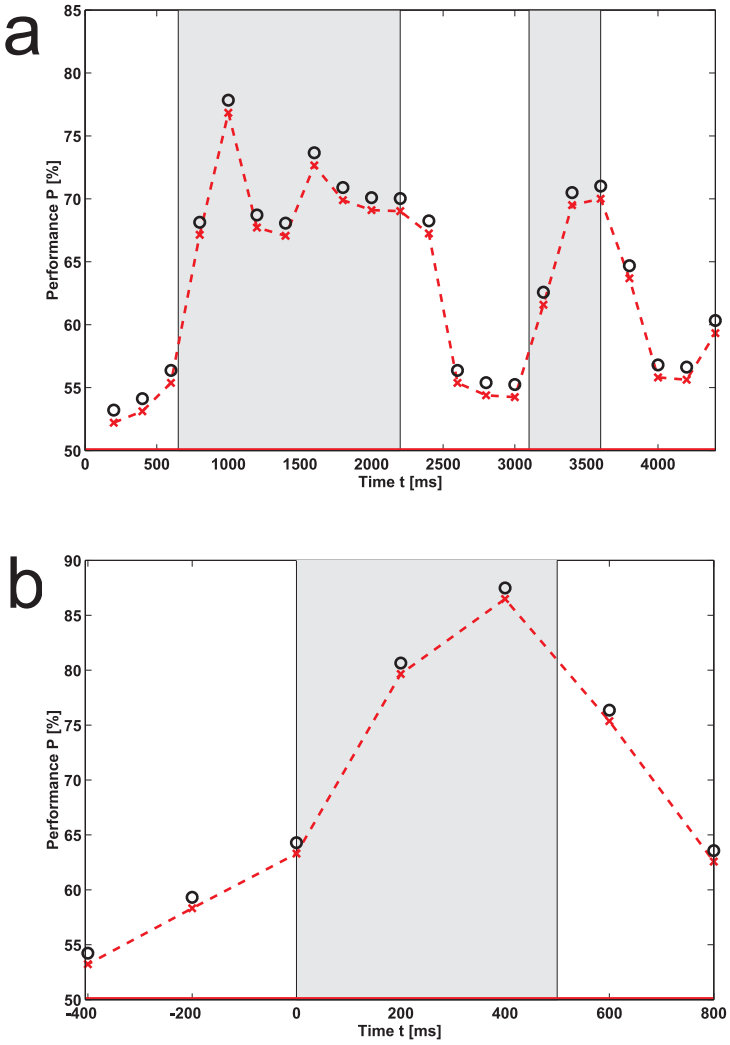


Figure 4.34: Time course of classification performance for the selected set of electrodes above V4 (cf. Fig. 4.14a, yellow crosses) for Monkey F. The condition of attention was classificated by SVMs. Power coefficients in a frequency range between 5 and 200 Hz were taken from a range starting 200 ms before, and ending 200 ms after the times marked with the red crosses. The black circles indicate a significant difference between the performance and the chance level (p=0.02). The chance level is depicted by the solid red line. In (a) the performance for T1 is shown. In (b), the period of the last shape (target) is analysed. Time in (b) is measured relative to the onset of the target shape.

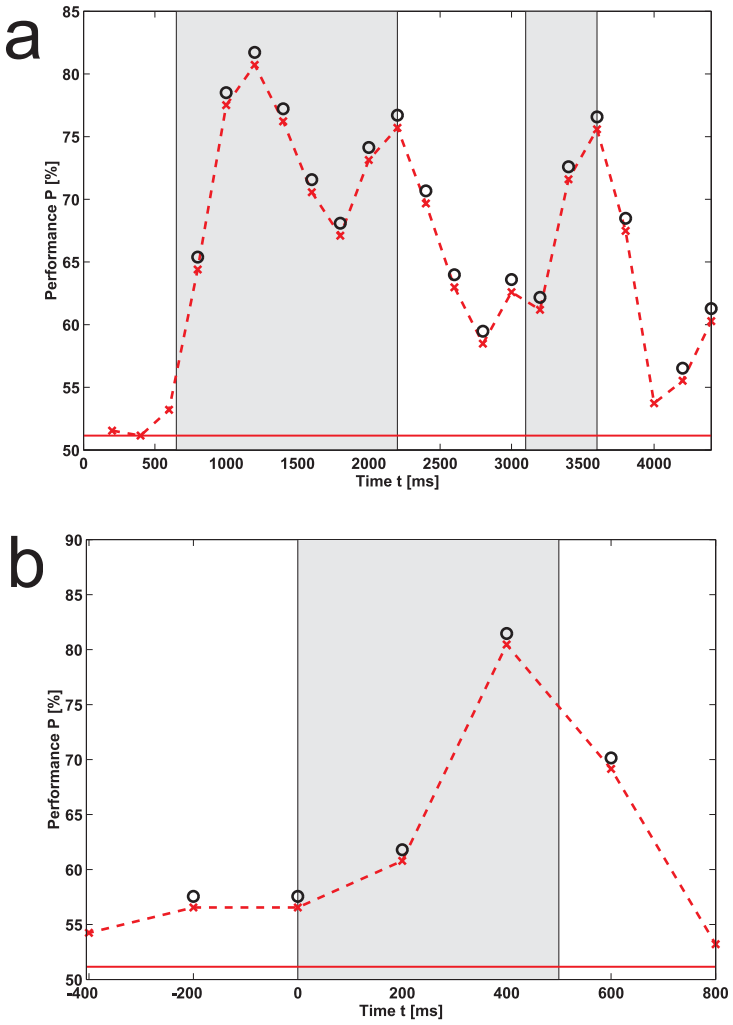


Figure 4.35: Time course of classification performance for Monkey M (see Fig. 4.34 for a detailed description). The condition of attention was classificated by SVMs. The data for the classification procedure was taken from the set of electrodes above V4 (see Fig. 4.15, yellow crosses).

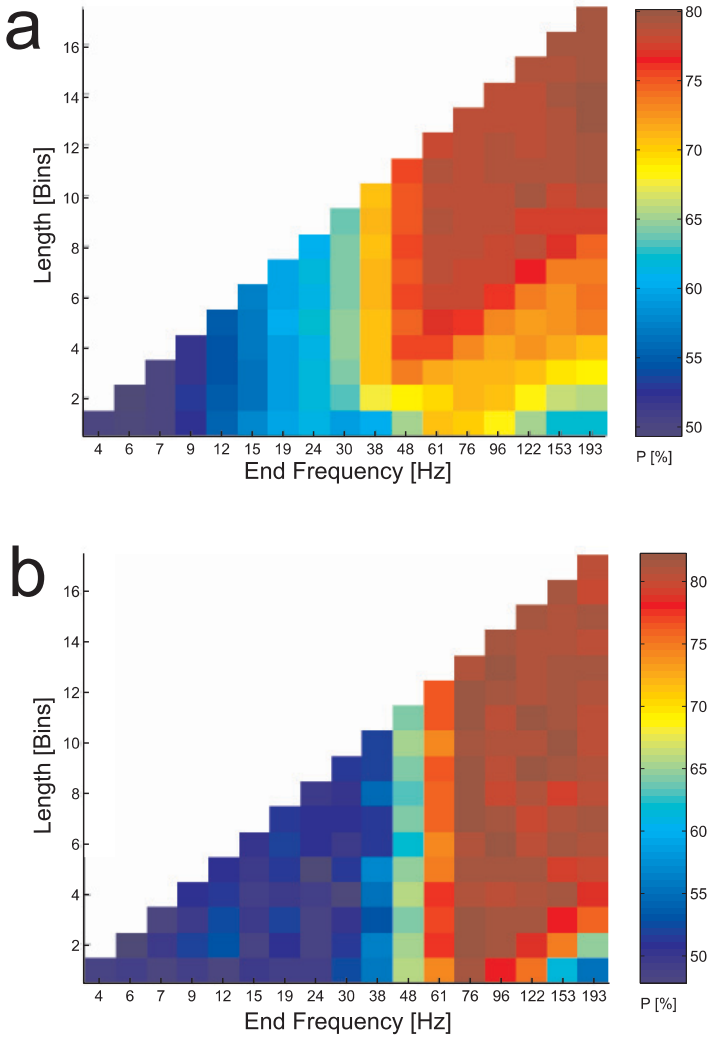


Figure 4.36: Classification performance P for Monkey F (a) and Monkey M (b), using different subsets of power coefficients from the selected V4 electrode combination during the initial period T1 (650 ms to 2200 ms after the trial onset). The task of the classification was to decide whether the was recorded in the attentional or non-attentional condition. (see Fig. 4.20 for a detailed description of this type of figure). Monkey M reached a performance P up to 82.3%. Using only the coefficients for 76 Hz and 61 Hz allows a classification performance of 81.5%. Monkey F reaches only a classification performance of 80.1% and the distribution over the frequency is different. E.g. for achieving a classification rate of 79%, the frequency bands from 24 Hz up to 193 Hz are necessary.

4.4.8 Attention on Morphing Shapes

In addition, the data from a second shape-tracking task was to investigate the discrimination of the two attentional conditions. Fig. 4.37 illustrates the modified task was performed in detail. It differs from the previous task in presenting sets of continuously morphing shapes insted of presenting sequences of static shapes with blank periods between two sets of shapes. In Fig. 4.38 (monkey F) and Fig. 4.39 (monkey M) the time course of classification performance is shown for the single electrodes. Both monkeys show good classification rates for signals form around area V4, while data from area V1 yields only low classification rates. This trend, regarding V1, is supported by the figures Fig. 4.40 and Fig. 4.41, which show the classification performance for the combined electrodes over V1. Again, only a low classification performance is reached, in comparison to the V4 electrode combination. Using data from available electrodes for the SVM yields a classification performance beyond 90%. The V4 electrode combination, used in the previous analyses, show a relative bad performance for monkey M when compared to the performance based on all electrodes. Selecting an alternative set of electrodes near V4, based on the information of Fig. 4.38 and Fig. 4.39 allows an increase in classification performance close to the performance when using all electrodes.

Using the alternative set of V4 electrodes, Fig. 4.42 displays how the explanatory power is distributed over different frequency bands. Monkey M showed a classification performance up to 91.3% and monkey F a classification performance up to 87.2%. Reaching 90.8%, with the data from monkey M, was even possible with the three frequency bins 122Hz, 96 Hz, and 76 Hz. Using only 96 Hz in combination with 76 Hz, a classification performance of 89.8% was still possible. Providing the coefficients for 96 Hz as input for the SVMs gives a classification rate of 88%, while using only the data for 76Hz is enough to get 87%. For monkey F the frequency bins 122 Hz - 38 Hz lead to a classification performance of 86.8%.

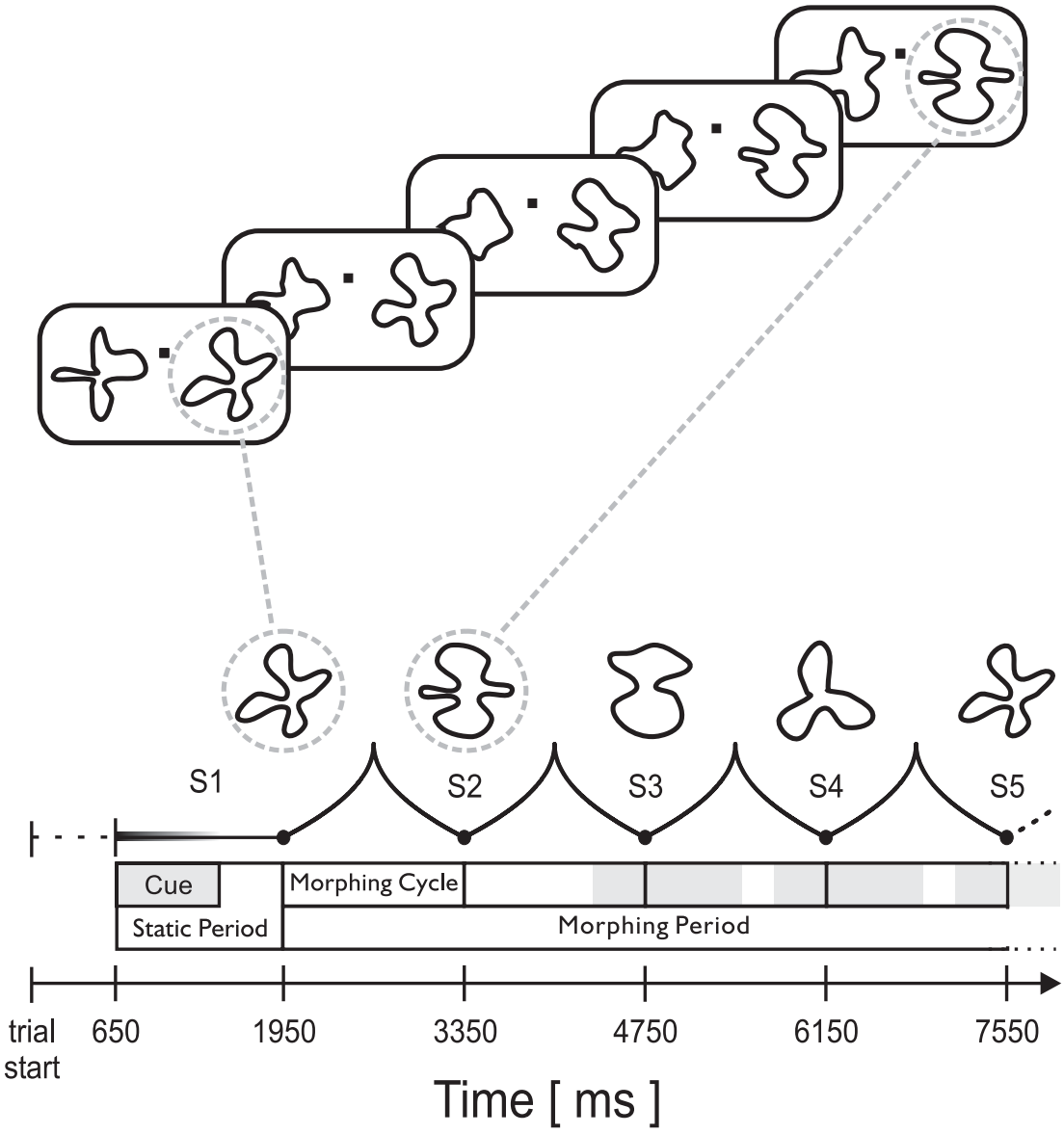


Figure 4.37: Schematic illustration of a modified version of the shape-tracking task. The difference to the task shown in Fig. 4.10 is: Instead of separating the presentation of two static shapes by periods of empty screens, now the two shapes are morphed continuously over the trial. For example, between 1950 ms and 3350 ms after trial start shape S1 is continuously morphed into shape S2. For more details on this task, see (Taylor et al., 2005).

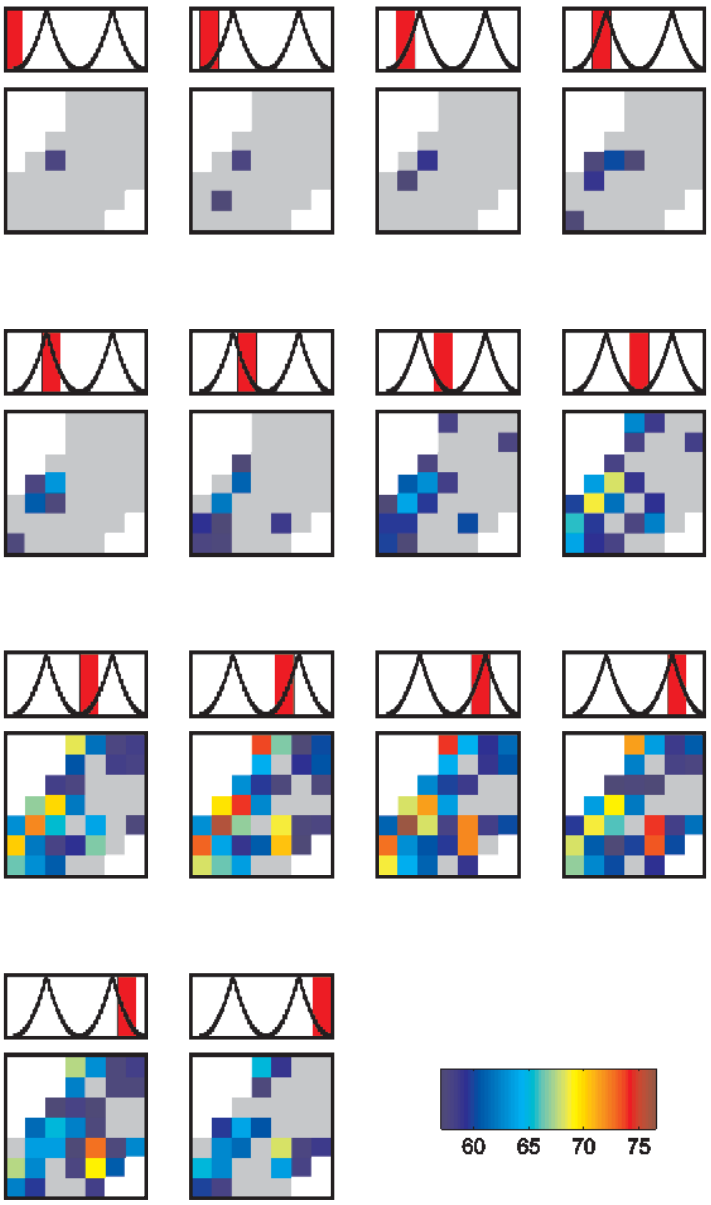


Figure 4.38: Time course of classification performance in discriminating the two attentional conditions, in the modified shape tracking task (monkey F). Data for calculating the power coefficients was taken from the time intervals marked in red, which are displayed above the classification performance maps. The black curves schematically display to the two morph cycles between 1950 ms and 4750 ms after trail onset (see Fig. 4.37). The red interval has always a length of 400 ms and starts in the first sub-picture (upper left) at 1750 ms. The interval was shifted by 200 ms between subsequent performance maps. For the grey coloured electrode positions, classification performance did not differ significantly (p=0.02) from chance level 55%. For all other electrodes, the performance is colour-coded in percentage of classification performance according the bar shown in the lower right corner.

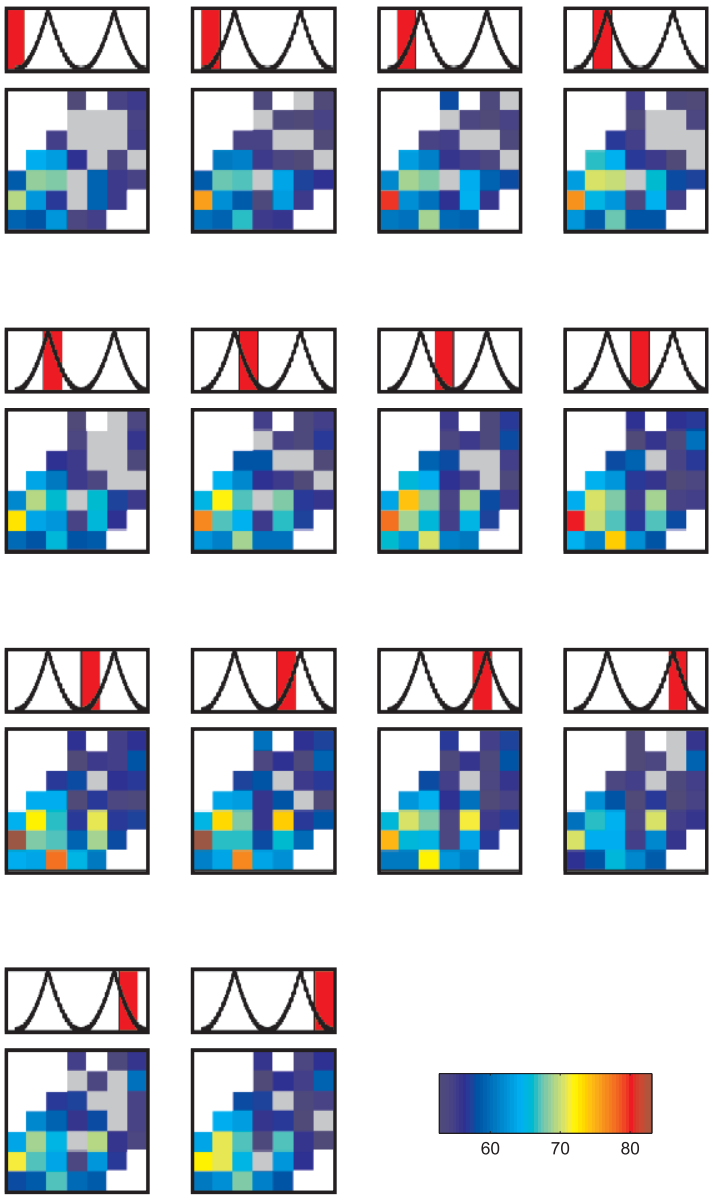


Figure 4.39: Time course of classification performance in discriminating the two attentional conditions in the modified shape tracking task (monkey M). See Fig. 4.38 for a more detailed distribution. For this data set, the chance level was $\approx 51\%$.

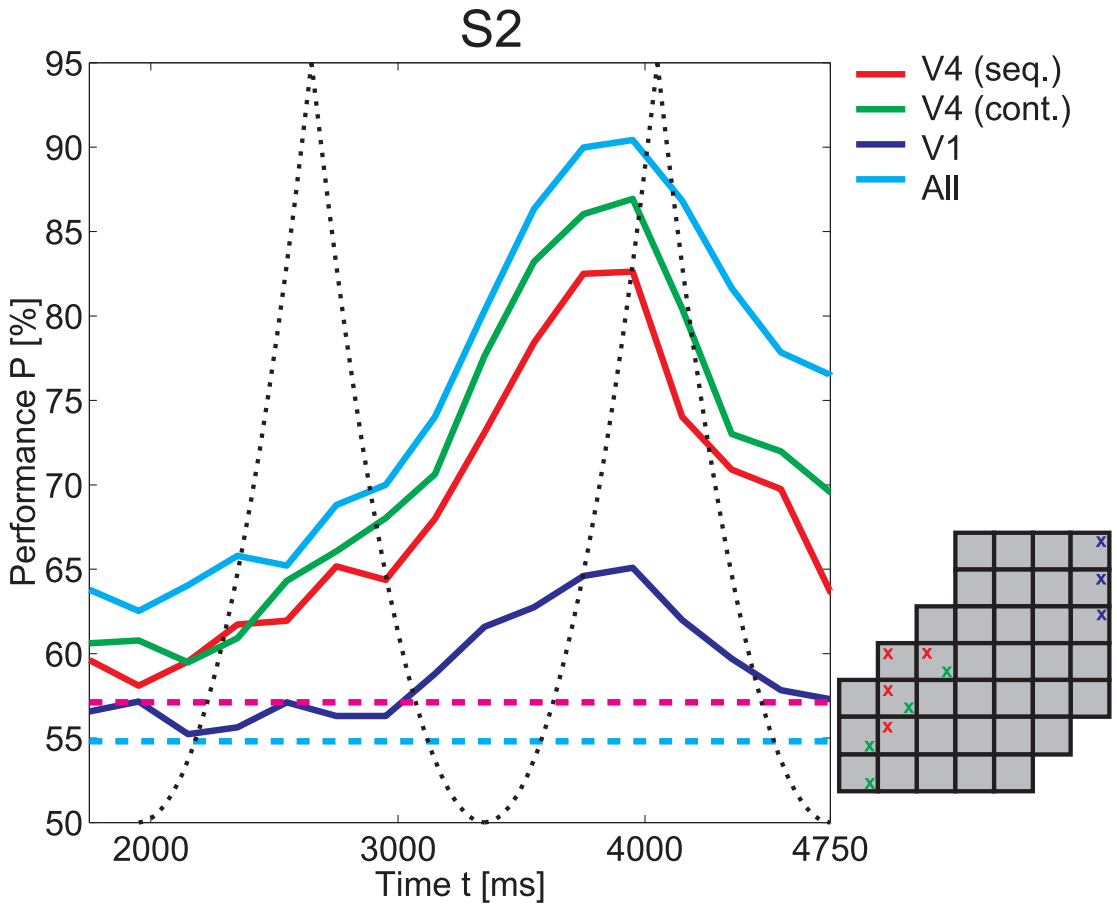


Figure 4.40: Time course of classification performance for discriminating the two attentional conditions for monkey F in the modified shape-tracking task, for different sets of electrodes. The red curve represents the classification performance for the 'old' V4 electrode combination. The green curve shows data from a combination of electrodes in the proximity of V4, selected on the basis of Fig. 4.38 (with the data from 3750 - 4150 ms after trial onset). The dark blue curve expresses the classification rate for a combination of three V1 electrodes. The position of the recruited electrodes are depicted in the electrode array map (lower right). The used electrodes are marked with crosses. The colour of the crosses is the same for the corresponding curves. In addition, the light blue curve represents the performance when all electrodes of the electrode array are used. The dashed cyan line shows the chance level and the dashed magenta line quantifies the border. Below this border, the performances are not significant different from the chance level (p=0.02). The black dashed line schematically visualizes the morph cycles of the shapes (see Fig. 4.37). For each point in a performance curve the data recorded in an interval beginning 200 ms before and ending 200 ms after, was used for the SVMs.

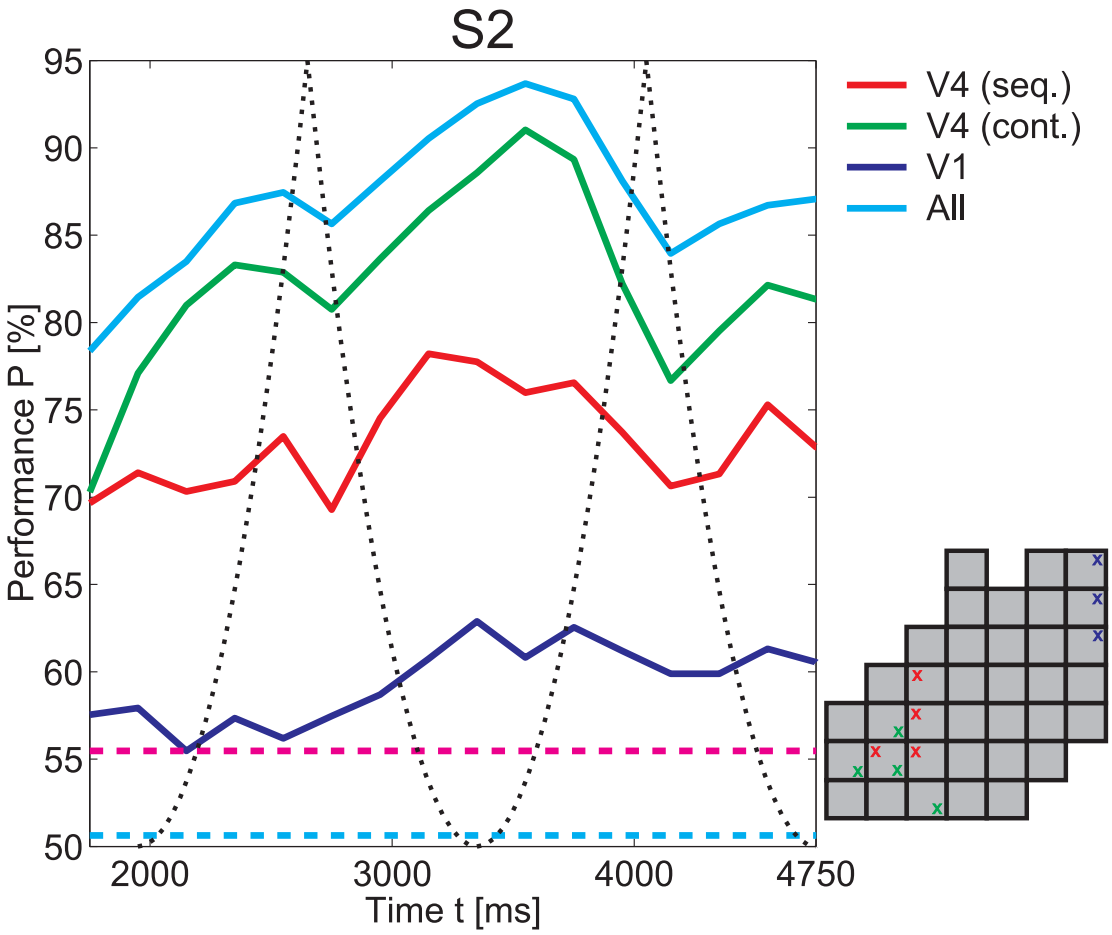


Figure 4.41: Time course of classification performance in discriminating the two attentional conditions, for monkey M and different combinations of electrodes in the modified shape tracking task. See Fig. 4.40 for a more detailed distribution. The alternate set of V4 electrodes was selected according to Fig. 4.39 (using the data from 3350 - 3750 ms after trial onset) .

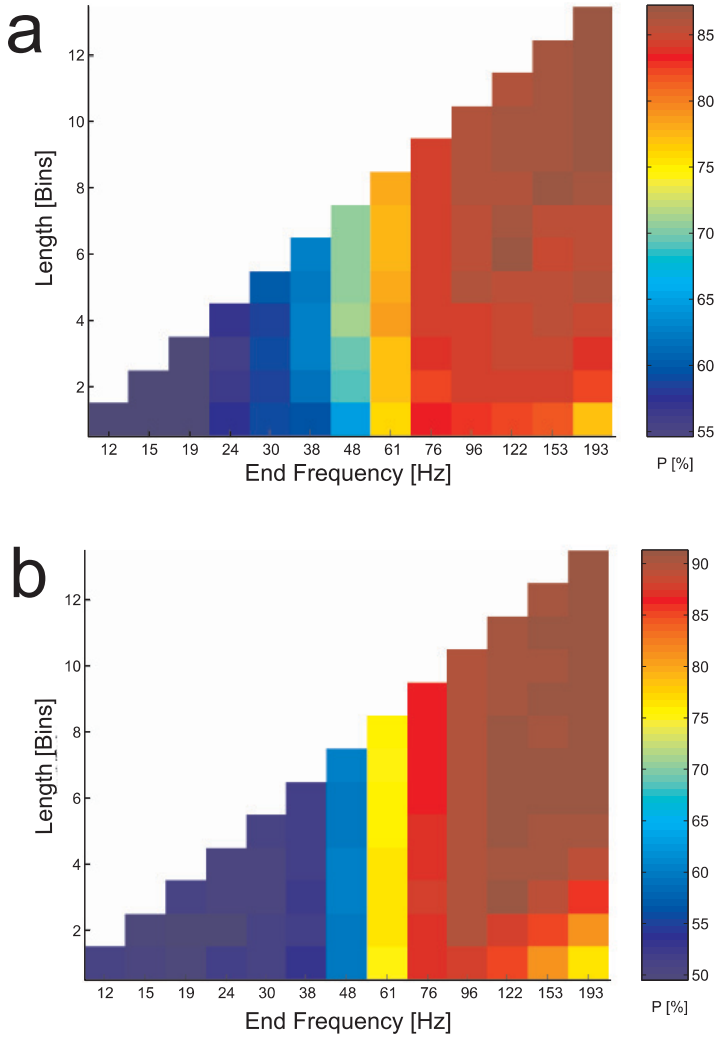


Figure 4.42: Classification performance P for monkey F (a) and monkey M (b), using different subsets of the power coefficients from the alternative V4 electrode combination (see Fig. 4.40 and Fig. 4.41) obtained during the modified shape-tracking task. The target of the classification was to discriminate the two different attentional conditions. For monkey F the data from 3750 - 4150 ms (after trial onset) was used and for monkey M the power coefficients from 3350 - 3750 ms (after trial onset) were selected. These time windows were selected on basis of Fig. 4.41 and Fig. 4.39. The used time intervals showed the highest classification performance and were selected for this reason.

4.5 Summary and Discussion

The presented analysis (Rotermund et al., 2007a) revealed three main insights:

1. Activity patterns created during the processing of different shape stimuli, contained in the local field potentials measured over area V4 allow surprisingly well to distinguish the underlying shape classes as well as the attentional condition.
2. Selective attention substantially enhances stimuli dependent differences of these neural activity patterns in comparison to conditions where the stimuli was not attended.
3. Behavioural failures are related to reductions in classification performances.

Most of the information which allow to discriminate different shapes was found in the frequency components in the γ-band above 40 Hz. Furthermore, the stimulus-specific characteristics of the signal is similar during different stimulation periods in a trial. The attention dependent enhancement of stimulus discriminability can not be explained by a simple increase of the SNR, but turns out to be most strongly related to a stimulus-specific differential scaling of the frequency components. This scaling results in an enhanced separation between the characteristic frequency patterns for different stimuli.

The enhanced discriminability under conditions of attention could in principle be traced back to two different changes in the signals. First, there is a small but statistically significant improvement of the signal-to-noise ratio. This finding is in line with a study by McAdams and Maunsell (McAdams and Maunsell, 1999b) describing an attention-dependent improvement of the SNR for spike count data from area V4. In their data, the rise of the SNR resulted from the attention dependent increase of stimulus responses together with a less than proportional increase of the standard deviation which rises approximately only as the square root of the response. Together with the enhanced absolute difference between the responses for different stimuli being amplified by a stimulus-independent gain factor, the increased SNR resulted in an improved orientation discriminability. In contrast, increases of the SNR in the present study explained only a minor part of the entire enhancement of shape discriminability under attention. The major part of the effect is based on an attention-dependent increase of differences between responses to different stimuli, which allow for improved stimulus discriminability even though SNRs stay almost unchanged. Since removing any differences in signal power/strength from all stimulus responses has only minor effects on classification accuracy and its attention dependent enhancement, the effect of attention does not rely on a stimulus-independent gain in response strength. The results rather indicate that attention changes the spectral composition and spatial distribution of the neural signals in different ways for different stimuli. Not only were these changes different for different stimuli, but in addition their direction was such as to increase distinctiveness of the respective neural activity patterns (in spatial and frequency composition) (see

Fig. 4.22 for illustration). Arbitrary directions of changes would not necessarily have caused an improvement of stimulus discriminability. Thus the major part of the effect is therefore not explained by a uniform, stimulus-independent effect as in gain models for single cell firing rate data, but by more complex changes in the composition and distribution of neural activity.

The underlying neuronal mechanisms are not known. In literature two candidates of putative neuronal mechanisms can be found which could contribute to an explanation:

- Feature-specific changes in attentional modulation have been observed in area MT (Martinez-Trujillo and Treue, 2004), which could play a significant role in scaling neural signals differentially depending on stimulus shape.
- Recruiting additional neuronal populations to encode a stimulus by dynamically changing their receptive field properties (Connor et al., 1997; Womelsdorf et al., 2006) could also render neural signals for different shapes more distinct from each other.

What do these findings imply with respect to cortical stimulus processing and its dependence on attention? Classification in the present study depends on the pattern of frequencies in the local field potential caused by neural processing of different stimuli. While field potentials clearly lack the specificity of information contained in the activity of the contributing single neurons, they reflect synchronised parts of neuronal activity patterns (Elul, 1972; Nunez, 1995; Robbe et al., 2006). Synchronous activity is known to be particularly effective in driving other neurons (Bruno and Sakmann, 2006; Usrey et al., 2000; Segev and Rall, 1998; Azouz and Gray, 2003) and has therefore been implicated in structuring effective connectivity (Aertsen et al., 1989) and defining in a transient and flexible manner neuronal assemblies (Singer and Gray, 1995; Abeles, 1991; Aertsen et al., 1986; Kreiter and Singer, 1996; von der Malsburg, 1985). Previous studies already demonstrated that attention enhances specifically such oscillatory activity in the visual system (Taylor et al., 2005; Fries et al., 2001). In addition, the results show that with attention the structural organization of field potentials systematically changes, indicating an attention-dependent change of composition and dynamic state in the network of synchronized neurons processing the stimulus in area V4. The more distinct patterns of neural activity associated with processing of an attended as compared to non-attended stimuli suggest a more differentiated and specific composition and state of synchronous neuronal assemblies if they process shape under conditions of attention. Such indications for enhanced modes of processing of attended stimuli are well in line with psychophysical findings demonstrating improved processing of attended stimuli and the particular importance of attention for shape perception (Rock et al., 1992; Rock and Gutman, 1981).

Further evidence for the behavioural relevance of the attention-dependent changes of cortical processing reflected by the observed changes in the pattern of field potentials comes from the reduced discriminability of signals preceding behavioural errors. In

fact, the findings imply that one could predict the occurrence of behavioural errors from such less distinct signals. In summary, the present data provide evidence that selective attention improves processing of attended stimuli by enhancing the distinctiveness and discriminability of cortical network states involved in the representation and processing of individual stimuli already in cortical area V4.

While attention caused a clear improvement of the stimulus classification achieved for field potentials from area V4, no significant effect was found for area V1. A likely reason for this lack of effect is the very high classification performance observed for the V1 recording sites, even without attention. It is based on the large size of the stimuli in comparison to the small size of the RFs for local field potentials in V1 (Eckhorn et al., 1993). This results in massive, shape-dependent differences in the coverage of these RFs by different stimuli. Consequently strong differences of the overall activation of the cortical columns underneath a V1 electrode allow distinguishing stimuli very reliably, even if they are not attended. This high stimulus discriminability was in addition confirmed by a ROC-analysis testing how well pairs of two stimuli can be distinguished by a single frequency component from one electrode. Performing this analysis for all possible combinations of stimulus pairs and frequency components for V1 revealed that almost 30% (monkey F) or 17% (monkey M) of combinations permitted an already 90-100% correct differentiation between two stimuli. The presence of these simple, almost perfectly discriminative signals in the non-attended condition strongly reduces the possibility to observe in V1 any further attention-dependent classification enhancements based on more complex indicators of attention dependent changes in cortical network states.

In contrast, in V4 there was no combination which allowed for such a high classification performance. This leaves room to observe substantial attention-dependent improvements. Thus the results do not exclude the possibility that similar changes as in V4 could also be observed for V1, if stimuli would be as similar with respect to the small V1 RFs as they were with respect to the large RFs of V4.

The high stimulus discriminability achieved with local field potentials recorded from the surface of the dura is also of interest in the context of Brain Computer Interfaces (BCI). In this study the electrode arrays were carried over years and recordings were pooled from recording sessions over several weeks. While BCIs based on single- or multi-unit recordings (Wessberg et al., 408; Taylor et al., 2002) typically need an initial calibration for each session, the recordings for the present study are sufficiently stable to allow for demanding stimulus discriminations with the same classifier over months. This is even more remarkable since the stimuli were not constructed to be easily distinguishable, but to require considerable effort by the monkeys for successful discrimination. The findings therefore suggest that the comparatively simple shapes of letters and many other symbols could be detected with high reliability in the spatial distribution of field potentials recorded from visual cortex.

Another interesting aspect for BCI applications is the high classification performance in discriminating the two attentional conditions. With the data from the modified shape-

tracking task it was possible to identify the attentional condition with a precision of up to 93.7% using the data from a 400 ms time window. For further research projects it will be interesting to learn more about the spatial aspects of selective visual attention. In the presented experiments attention was only applied to the left or right part of a computer screen. However, an important question in the field of BCIs is the possibility to partition the visual field even further into more and smaller regions while retaining a similar classification performance under these new conditions. Furthermore the analysis revealed that the mode of presentation (stationary shapes vs. morphing shapes) can significantly enhance the classification rate for the attentional condition. This finding should guide the design of BCIs using visual selective attention.

Chapter 5

Stabilizing Decoding Against Non-stationaries

5.1 Motivation

In chapter 5.2.3 we briefly discussed the idea of brain computer interfaces and functional neuro-prostheses, as well as different ways of acquiring information from activity patterns in the brain (action potentials, LFP's, EEG's) for controlling external devices. Many experiments have successfully demonstrated that external devices can in principle be controlled by brain signals (Andersen et al., 2004a; Schwartz, 2004). One application is controlling prosthetic devices for restoring lost body functions. Especially for voluntary use of neuro-protheses in normal life outside a laboratory, it is important to provide long-term stability of both, the information acquisition process and the successive translation of measured data into control signals. As antagonists to the requirement of stability, different processes induce on-going changes in the observed signal characteristics and in the physical properties of the prosthesis. These include relative movement of the recording array with respect to the brain, neuron death and growth processes at the recording site, chemical changes at the contact surface between the electrode tip and the neuronal tissue, or adaptation of the neuronal response properties to the task. Taken together, non-stationarities imply that over some time the situation will differ more and more from the one to which decoding algorithms were trained when the prosthesis was installed. Accordingly, errors between actual and desired movements might increase and render the prosthesis useless after some time. For counteracting on-going changes, it is necessary to adapt the algorithms for reconstructing the relevant information from the measured activity patterns (see section 2.2).

A standard solution for this problem is to re-train the prosthesis by requiring the subject to perform a well-defined task at regular intervals (Hinterberger et al., 2004; Tillery et al., 2003). In these tasks the desired movement is known, the performance can

be accessed by an external observer, and subsequently be used to adapt the mapping between brain activity and the control signal for executing the estimated, intended action. The drawbacks of this scheme are obvious: Everyday use of the prosthesis must be interrupted by retraining sessions, which are likely to be conducted under supervision in a hospital or specialized laboratory. Between training epochs, the performance of the prosthetic device will still decrease gradually, and it is particularly unlikely that it can recover from abrupt changes of the recordings induced e.g. by extreme movements of the head. If only the brain would be required to compensate for these non-stationarities, it is likely that the network creating the recorded activity becomes stressed by requiring it to operate at the brink of, or beyond its functional capabilities.

For voluntary movements, the error between desired and actual movement is unknown to an external observer. In this chapter I will present a new method (Rotermund et al., 2006a) of using an extra error signal measured directly from the brain for adapting the estimation process in an on-line fashion, and thus protecting it against non-stationaries. This idea provides an efficient alternative to current methods of re-learning procedures and has several potential advantages for a disabled person: although being technically demanding, it can eliminate the need for performing tedious training tasks in a clinical environment, and both adaptation and even initial calibration can be done during the everyday use of the prosthesis. In summary, this scenario may dramatically increase independence, life quality, and – most importantly – motivation of the patient. It has to be noted that this scenario can already be used for actual prostheses but it mainly aims at neuro-prosthetic devices with high quality for reconstructing intended movements which can be expected for the near future.

The required additional error signal has to represent the user's affective evaluation of the current neuro-prosthetics performance. Several possible candidates for signals representing affective evaluations of performance are already known (Ridderinkhof et al., 2004b; Musallam et al., 2004), and it has been demonstrated that such error signals can in principle serve for reinforcement learning (Sutton and Barto, 1998). Through numerical simulation, using realistic assumptions about the origin and quality of neural signals, it will be shown that even if the error signal has a low information content and shows high noise levels, like it is typically encountered in recording brain activities, it is possible to counteract the effect of a variety of non-stationaries. Even in the case when the error signal itself is effected by (a slightly reduced class of possible) non-stationaries, the procedure can be still applied.

5.2 Neuronal and Computational Background

5.2.1 Motor system and movements of arms

This section will start with a very short overview of the neuro-anatomical structures. Then selected rules of arm movement will be discussed, followed by a discussion about

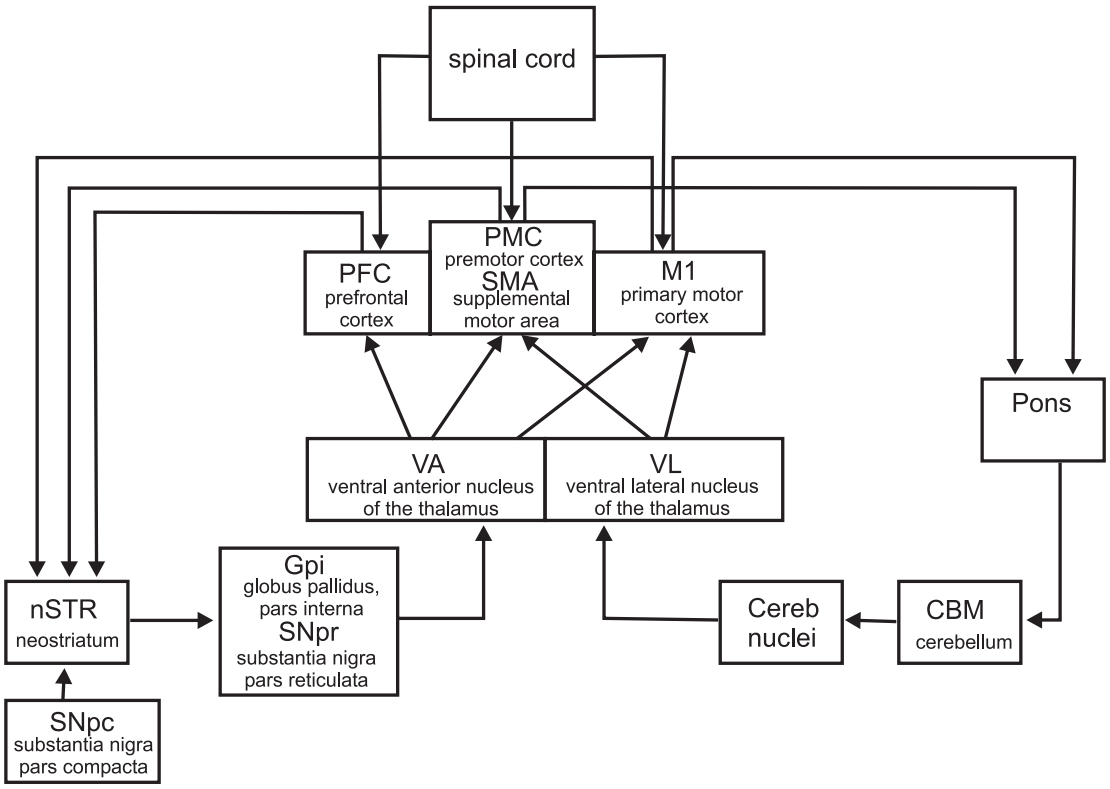


Figure 5.1: Schematic wiring diagram between spinal cord, subcortical, and cortical motor areas. (Figure adapted from (Johansen-Berg, 2001))

correlations between neuronal activities and movements as well as properties of neuronal activity patterns representing errors of movements. This section will close with a glance on current research in neuronal prosthetics.

Motor cortex

The human motor system (Rizzolatti and Wolpert, 2005) and corresponding brain areas (Roland and Zilles, 1996; Chouinard and Paus, 2006) show a degree of complexity similar to the visual system (see Fig. 5.1 for a simplified illustration and (Johansen-Berg, 2001) for more details). The primary motor cortex (M1, Brodmann area 4) receives its main input from the parietal cortex and also gets input from e.g. the premotor cortex and the thalamus (Passingham, 1993). Intracortical stimulations in the motor cortex often generate reactions in groups of muscles and individual muscles can often be addressed from more than one position in the motor cortex, which indicates that regions of the motor cortex rather represent groups than individual muscles (Johansen-Berg, 2001; Donoghue et al., 1992). Current studies in M1 indicate that this cortical area may be composed of several modules. Each of these modules seems to be controlling one physical system like e.g. a hand, an arm, or a leg. Furthermore, it is generally

believed that these modules are adaptive to the actual conditions of the system. In total, the primary motor cortex is important for generating motor action but it is also involved in learning new actions. Furthermore, connections between the neuronal activity of M1 and other cognitive variables (Sanes and Donoghue, 2000) exit.

The motor system consits of more areas than the primary motor cortex, like e.g. the posterior parietal cortex, the premotor cortex, the cerebellum, the basal ganglia, and the spinal cord.

The neurons in posterior parietal cortex show correlations between their discharging behaviour and specific motor acts like e.g. grasping. In addition, it was found that actions seem to be organized in 'chains' of elementary motor commands. Also several neurons were found that show mirror properties, which means that they not only fire during performing an action, but also react while only observing a specific motor action (Rizzolatti and Wolpert, 2005).

The premotor cortex (PMC, parts of Brodmann area 6) (Fulton, 1935; Woolsey et al., 1952) is brought into association with sensory guidance of movement and preparation of motor actions (Wise, 1985) while the supplementary motor area (SMA, parts of Brodmann area 6) (Penfield and Rasmussen, 1950) seems to be concerned with planning, execution and memorization of whole sequences of motor actions (Tanji and Shima, 1994).

Other functionally important parts of the motor system are the cerebellum, the basal ganglia, and the spinal cord. If the cerebellum is damaged then dysfunctions in coordinated movements can occur as well as disturbances in balance (vertigo), jerking involuntary movements, loss of power and tone in the limbs, and abnormal posture and staggering gait. It also plays a role in motor learning. A damage of the basal ganglia does not typically generate a paralysis or ataxia but leads to abnormal movement patterns, like e.g. involuntary movement, poverty of movement, and altered muscle tone (Johansen-Berg, 2001). Furthermore, basal ganglia play a role in learning action sequences (Rizzolatti and Wolpert, 2005).

The spinal cord is not just a simple connection between brain and nerves projecting to the muscles. Experiments where the spinal cord was stimulated directly, suggested that it generates motor primitives (patterns of muscle activations) itself (Wolpert and Ghahramani, 2000; Tresch et al., 1999; Giszter et al., 1993).

In summary, the motor system is very complex and still under research. Also the question how the motor system can handle alterations of the system's dynamics is not answered. For example, if we transport an object with our hand then the physical properties of the arm are changed due to extra load. The properties of the hand can also be changed by e.g. growth and fatigue. It seems that, depending on the sources of these changes, different adaptation strategies are applied. It makes a difference if the change is caused by changes of intrinsic properties or due to external influences (Lackner and DiZio, 2000; Miall, 2002; Rizzolatti and Wolpert, 2005).

Some rules of moving an arm

If a person reaches for an object then the hand can follow, on its way, a nearly arbitrary trajectory with a nearly arbitrary velocity profile. One movement of the hand can be composed of a huge number of possible posture configurations. Each posture configuration can be generated by different combinations of muscle activations, and the necessary muscle tensions can be created by various neuronal activities. Taken together all these degrees of freedom, it seems that it is extremely unlikely that two persons would use similar movement strategies for reaching the same target. However, it was found that a set of rules for movements is applied and that these rules are even common to different species (Schaal, 2002; Flash and Sejnowski, 2001). These regularities were mainly investigated in isolated arm or hand movement studies. If arm and hand are moved at the same time, then both movements influence each other, making the situation more complex. A review regarding this topic, which I will follow in this section, can be found in (Schaal, 2002).

One recurring type of movement characteristics was found for point-to-point reaching movements in humans and other primates. The trajectories are composed of approximately straight movements into the target direction combined with an approximately symmetric bell-shaped velocity profile (Bullock and Grossberg, 1988). Deviations from the shape of the velocity profile depend on movement speed. Different models were proposed for explaining these variations: e.g. as result of perceptual distortion and dynamical properties of feedback loops in motor planning (Bullock and Grossberg, 1988). Other models explain the findings by using different optimization criteria. They e.g. include the dynamics of the arm (Kawato, 1999; Uno et al., 1989) or different noise levels for different velocities generated by the stochastic properties of neuronal activities (Harris and Wolpert, 1998).

Another approach is to compose movements from so-called movement primitives, also known as units of action, basis behaviors, or gestures, see (Mussa-Ivaldi and Bizzi, 2000; Flash and Sejnowski, 2001; Flash and Hochner, 2005; Thoroughman and Shadmehr, 2000; Sternad and Schaal, 1999). For example, a movement could be constructed from a number of static straight-lined movements with symmetric bell-shaped velocity profiles, or a movement could be a combination of dynamic movement primitives (e.g. point or limit cycle attractors). It is still not clear if movement primitives are used for the generation of movements, but from a theoretical point of view the existence of such basic units of actions would reduce the complexity of motor learning problems (Sternad and Schaal, 1999).

For movements restrained to two dimensions (in planar space with Cartesian coordinates) it was found (Lacquaniti et al., 1983; Viviani and Flash, 1995; Moran and Schwartz, 1999a) that the angular velocity $a(t)$ follows power law with exponent 2/3,

$$a(t) = k \cdot c(t)^{\frac{2}{3}} ,$$

where $c(t)$ is the curvature of the trajectory and k is a proportionality constant. The

origin of this law is still under debate. It is suspected that it reflects an important principle of moment generation in the central nervous system. Violations of this law have been found if the restrainment to two dimensions is loosened (Schaal and Sternad, 2001).

A robust phenomenon in human arm and hand movement, even used as a behavioral benchmark for testing models, is Fitts' Law (Mottet and Bootsma, 2001; Bullock and Grossberg, 1988) or speed-accuracy trade-off. It quantifies the required time T for rapidly reaching a target in dependency of the required accuracy

$$T = a + b \log_2 \left(\frac{2 \cdot A}{W} \right) ,$$

with a and b as fitting constants, A being the distance between start and end point of the movement, and W describing the target width, which defines the necessary precision of reaching. The physiological causes for the Fitts' Law are still unknown.

It is possible to use these rules as *a priori* knowledge for estimation algorithms, used for controlling prosthetic arm devices. This allows to increase the precision of predicting the next possible positions of the prosthetic device.

5.2.2 Error signals in the brain

Behavioral actions can lead to errors if something went wrong. The central nervous system relies on information about errors, which are useful for adapting and optimizing its representations and functions as, e.g. the adaptation of the motor system to changes in physical properties of the limbs (Lackner and DiZio, 2000; Miall, 2002; Rizzolatti and Wolpert, 2005). Later in this chapter, we will develop a framework that uses information about perceived errors, resulting from actions performed with prosthetic devices. The error signals will be used for modifying the control system of neuroprostheses. This signal has to be acquired from the brain's neuronal activity. Since this is new to neuronal protheses, it is not clear where this error signal can be acquired from. In the following an overview about neuronal correlates regarding errors and related signals in the brain is given. The focus will lie on error signals created by the movement of arms. This overview will start with a brief description of reward signals found in the mammalian brain. The neuronal correlates of some of these rewards may also be useful for improving the performance of a prosthetic device.

Rewards in the brain

Rewards and punishments, as a kind of negative reward, are important for behavioural learning. Rewards can act as behavioural goals if information about required actions for obtaining rewards can be deduced by the subject (Dickinson and Balleine, 1994). For a review on reward and neuronal coding see (Schultz, 2004), which I will use as a basis for this part of this introduction.

Interestingly, it has been found that rewards which have been fully predicted will not contribute to learning (Kamin, 1969; Schultz et al., 1997) and that errors between predicted and received rewards are often more important (Rescorla and Wagner, 1972; Pearce and Hall, 1980). In game theory, the relevance of a reward is depending on the magnitude of an reward multipiled by the probability of an expected reward. The longer the estimated time until the reward is expected to be achieved, the lower is the value of the reward (Ho et al., 1999). However, often 'reward' is not a good description and the term 'utility' is used instead, which also includes e.g. the preferences of the subject (Schultz, 2004).

Experiments revealed a large number of sites in the brain that are involved in representations of rewards and related quantities. Some of these may be suitable for enhancing the long term performance of neuronal protheses. In the following list several types of reward signals and the corresponding brain regions will be enumerated:
Depending on the subject's preferences to a particular type of reward, neurons in the orbitofrontal and striatal neurons are activated. Furthermore, structures in the striatum, orbitofrontal cortex, dorsolateral prefrontal cortex, anterior cingulate cortex, perirhinal cortex, superior colliculus, pars reticulata of substantia nigra, and dopaminergic pars compacta of substantia nigra showed differences in their neuronal activity when

a trial was or was not rewarded. Neurons in striatum, dorsolateral prefrontal cortex, orbitofrontal cortex, parietal cortex, posterior cingulate cortex and dopaminergic pars compacta of substantia nigra revealed correlations between their activity and the magnitude of reward (liquid). Orbitofrontal and striatal neurons react to some types of conditioned stimuli while their activity has in this case a correlation to the predicted reward. Furthermore, the neurons of orbitofrontal cortex as well as the amygdala and striatum including the nucleus accumbens are sensitive to reception of different liquid and food rewards (Schultz, 2004). The activities of prefrontal cortex (Matsumoto et al., 2003; Kobayashi et al., 2002), intraparietal area (Platt and Glimcher, 1999), cingulate motor area (Shima and Tanji, 1998), frontal and supplementary eye fields and premotor cortex are also influenced by expected reward (Roesch and Olson, 2003), and midbrain dopaminergic neurons are processing the difference between expected reward and received reward (Nakahara et al., 2004). Prefrontal cortex (Watanabe et al., 2002a), midbrain dopamin-containing neurons (Satoh et al., 2003), ventral striatum (Shidara et al., 1998), and for voluntary movements (Tremblay and Schultz, 1999) orbitofrontal neurons are in addition coding motivational aspects of a reward. The probability of receiving a reward modulates the response of neurons in superior colliculus and in the pars reticulata of substantia nigra, while the activity of some parietal neurons can better be described in terms of the product between the magnitude and the probability of the reward. Some dopamine neurons show a decay of their activity in the interval between the stimulus that advertises a reward and the reward itself, while neurons in midbrain (Fiorillo et al., 2003) and posterior cingulate cortex seem to code for the risk and/or the uncertainty in receiving reward. Other neurons in dorsolateral prefrontal cortex, anterior cingulate cortex, posterior cingulate cortex, and frontal eye fields represent events where the animal made a behavioural error that caused a missing reward. It seems that these neuronal representations of reward-related variables are used by the brain to direct other processes like e.g. arm and eye movements (Schultz, 2004).

Conflict monitoring

Not all actions result directly in a reward. This is one reason why the absence of a reward after a performed action can not always be used as an indicator for a behavioural error. However, it is still important for an animal to monitor its own performance. This allows the animal to react with counteractive measures in situations where problems occur. A hypothesis that addresses this monitoring of conflict situations is the so-called 'conflict-monitoring' hypothesis (Botvinick et al., 2004; Botvinick et al., 2001; Rushworth et al., 2004; Yeung et al., 2004b). The idea behind this hypothesis is that regions in the brain monitor occurring errors and conflicts (Niki and Watanabe, 1979), and that this information in turn is used for adaptation processes in the cognitive control system such that conflicts can be solved or circumvented in the future. Studies suggest that anterior cingulate cortex (ACC) (Ridderinkhof et al., 2004a; Ridderinkhof et al., 2004b) is important for the detection of conflict situations and seems to be used for representing at least three different types of conflicts (Botvinick et al., 2004): First, situations that require overriding a running action with a better response. This creates

a conflict between the new and the obsolete behaviour. Second, there are cases where a number of possible actions is available but the situation is underdetermined, which results in a problem for selecting the best option. A conflict between the possible actions occurs. This can occur e.g. during simple motor tasks where several equally good responses are available (Frith et al., 1991). The third class of conflicts encompasses situations that are involved in the 'commission of errors'. An alternative hypothesis is that ACC rather encodes the effort for achieving a goal than a conflict situation (Walton et al., 2003).

An observable result of conflict monitoring seems to be the increase in reaction times after making errors (Botvinick et al., 2001; Gehring et al., 1993; Gehring et al., 1995; Garavan et al., 2002). Here, the magnitude of error and activation of ACC are connected (Kerns et al., 2004). Other studies (Picard and Strick, 1996; Paus, 2001) refer to the strong connectivity between the ACC and motor structures which suggest that ACC influences response selection. An activation of ACC was also found when subjects got feedback based on their decisions (gambling tasks) (Nieuwenhuis et al., 2004; Holroyd and Coles, 2002). It has been revealed that transient activities of ACC are related to the detection of errors. In EEG recordings, event related potentials (ERP) are observed, which peak between 80 and 130 ms after responding to a task with an error (Rushworth et al., 2004; Falkenstein et al., 2000; Gehring et al., 1993; Krigolson and Holroyd, 2006). Similar responses of ACC to errors were found in studies using fMRI (Carter et al., 1998; Menon et al., 2001) and electrophysiological recordings (Ito et al., 2003). Examples are tasks where errors and response conflicts are associated to fast correction movements (overriding actions) during erroneous actions (Yeung et al., 2004b). Computational models describing the connection between conflit monitoring theory and details of ERP (Yeung et al., 2004b) have been published. An alternative study (van Veen et al., 2004) showed disagreements between the role of the ACC in conflict-monitoring hypothesis and their experimental results. Furthermore, it has to be noted that ERP can also be influenced by the observation of an error (van Schie et al., 2004).

Errors in arm movements

For using error signals in neuro-prosthetic devices, quantification of the deviation between intended actions and realized actions is necessary. Depending on the type of neuro-prosthetic device (e.g. artificial limbs or a communication device like a speller) the possible actions are different. This generates an extra difficulty in finding putative sites in the brain which represent the corresponding errors because for different types of prostheses the error signal may be represented at different positions in the brain. We will now focus our discussion of regarding such error signals on prostheses for arms.

Possible actions for an artificial arm are e.g. execution of reaching movements. At the end of an erroneous movement we can quantify the error of such an action for example by the euclidean distance in external Cartesian coordinates. Actually it is not possible

to send back direct feedback, about the actual configuration of a prosthetic device, into the central nervous system. Thus the error has to be perceived via the visual system. Since we need both, the intended position and the visually observed actual position, candidates for such an error signal are neurons in areas which code correlations between planned movements and observations of movement. Putative brain areas include the superior colliculus (Stuphorn et al., 2000), dorsal pre-motor area (Boussaoud et al., 1998; Musallam et al., 2004; Lee and van Donkelaar, 2006; Fujii et al., 2000), ventral pre-motor area (Mushiake et al., 1997), parietal reach regions (Batista et al., 1999; Musallam et al., 2004), or parietal cortex (Culham and Valyear, 2006). An advantage of these signal sources could be their proximity to the motor system: one can expect that the influence of other variables on the firing rate like e.g. the general emotional state of the subject, remain small. In addition, a strong activation of cerebellum and motor cortex correlated to execution errors was found (Diedrichsen et al., 2005; Chen et al., 2006).

Despite this body of accumulated information, many questions are still unanswered, especially where to find sites in the primate brain which represent errors of actions and the corresponding neuronal coding of error signals. Furthermore, it is unclear how these neural correlates of errors will change when an artificial device replaces the original body part.

5.2.3 Brain computer interfaces

A brain-computer interface (BCI) is a system that identifies the user's intentions from suitable brain signals, see (Lebedev and Nicolelis, 2006; Lebedev et al., 2005; Andersen et al., 2004a; Schwartz, 2004; Wolpaw et al., 2002; Wolpaw et al., 2000; Kuebler et al., 2001; Curran and Stokes, 2003). The extracted information can be used to control computer applications or electromechanical devices (e.g. wheelchairs or prostheses). BCIs bypass the normal pathways between brain and muscles, allowing handicapped users to control external information processing devices. Experimental techniques for acquiring information about the desired action from the brain encompass non-invasive methods and invasive methods.

Non-invasive methods

Non-invasive methods base on data acquisition technology which does not require to open the skull. A prominent example is the recording of EEG signals (electroencephalogram) (Wolpaw and McFarland, 2004a; Fabiani et al., 2004), which records electric activity generated by the brain from the scalp. Data acquisition via fMRI (Weiskopf et al., 2004; Yoo et al., 2004) falls into the same category, but due to size and costs for fMRI systems, it is not applicable for the daily use. Also the temporal resolution of a fMRI scan is low and lies in the order of magnitude of a second.

One main advantage of EEG recordings is that it can be tested on healthy subjects because it does not require surgery. The risk of using an invasive method, including necessary surgery at head and brain, is typically (with the actual state of technology) too risky for most users of BCI. Another advantage is that EEG-based BCIs are affordable, portable and easy to install. The main disadvantage of EEG lies in the limited information bandwidth of EEG recordings, due to a coarse spatial resolution and a small amplitude of high-frequency signals. These low-pass filtering properties in space and time are a consequence of the properties of fluid, tissue and skull between the brain and the sensors of an EEG system.

In EEG signals, different types of neuronal correlates were found that correspond to different functions of the central nervous system. Four interesting correlates for BCI applications are:

- Signals generated by the motor system
  It was shown that the representations of motor actions in the EEG (e.g. like finger movements (Xu et al., 2004; Astolfi et al., 2005)) and also the imaginations of motor actions (Wolpaw and McFarland, 2004a; Pfurtscheller and Neuper, 2001) can be used to control computer applications. It has to be noted that patients with limb amputations show a lower performance than healthy subjects. Performance decreases with the time interval since the limb was lost (Blankertz et al., 2006).

- P300
  Another characteristic of EEG, that can be used for single trial analysis in the context of BCI applications, is the so called 'P300' (or 'P3') (Farwell and Donchin, 1988; Paulus et al., 2002; Spencer et al., 2001; He et al., 2001; Meinicke et al., 2003; Sutton et al., 1965). The P300 is a positive component of the event-related potential (ERP) which can be detected approximately 300 ms after stimulus onset. It shows correlations to sound and light stimuli. These correlations depend on the subject's degree of uncertainty, its attentional state, the frequency of occurrence and the novelty of the stimulus. This type of BCI signal is used for communication BCIs like spellers but is not very helpful for reconstructions of intended movements for prostheses.

- Steady-state visually evoked potentials
  If flickering visual stimuli with frequencies of approximately 8-15 Hz (e.g. periodic full-field flashes) are presented, EEG recordings reveal oscillatory neuronal activities with similar periodicity in the visual cortex. This phenomenon is called Steady-State Visually Evoked Potential (SSVEP) (Müller et al., 1985) and it can be modulated by visual attention (Vanni et al., 1999).

- Multifocal visually evoked potential (mfVEP)
  Instead of stimulating the retina with a full-field flash and then measuring the EEG-response, it is also possible to stimulate different segments of the retina independently from each other. This method uses the correlations between these different regions of the visual field and their contribution to the whole EEG signal. This technique is termed multifocal VEP (Nobre et al., 2000; Teder-Salejarvi et al., 1999) and shows a correlation to the position of the focus of attention (Seiple et al., 2002).

Invasive methods

While EEG is used predominantly for humans, invasive methods are tested almost exclusively on animals (e.g macaque monkeys) (Lebedev and Nicolelis, 2006; Carmena et al., 2003; Musallam et al., 2004; Schwartz, 2004; Andersen et al., 2004a). However, it was recently shown (Hochberg et al., 2006) that these methods can be adapted to human users. A patient with an implanted BCI was shown to be able to control his TV and draw simple pictures with his cortical signals. It was estimated that it will still need 10-20 years before such invasive BCI will be usable for clinical applications (Lebedev and Nicolelis, 2006). It is unclear how long it will take until the expectations in science-fiction literature (like e.g. (Shirow, 1991)) can be met and artificial body parts will fully replace lost ones.

Sending information into the other direction, from external sensors into the central nervous system, reached clinical application. In some cases it is possible to place so-called 'cochlear implants' into the inner ears of deaf patients. A set of electrodes stimulates the acoustic nerve directly by emitting electric impulses such that the user

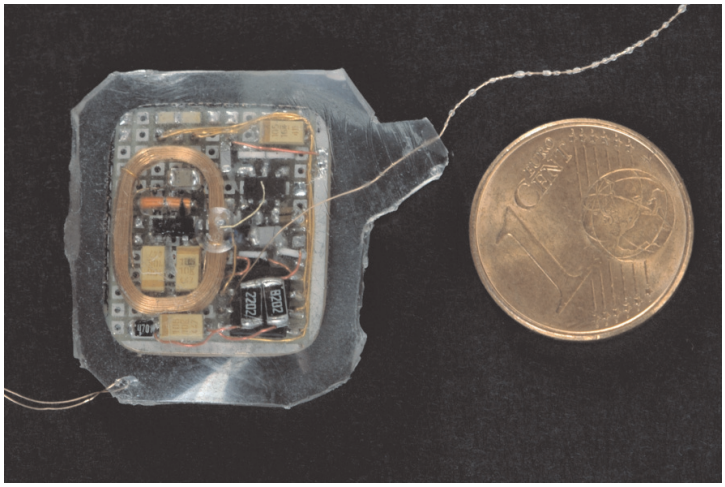


Figure 5.2: Prototype of a wireless implant based on our ideas (Pawelzik and Rotermund, 2005) build by the Fraunhofer IMS (Duisburg). This implant receives energy via electromagnetic waves, measures the electric potential, and sends this information to a receiver outside the skull via electromagnetic waves. Photo was taken by Harald Rehling.

gets an impression of its acoustic environment (Grayden and Clark, 2005). Retinal implants and visual cortical prostheses fall also into the category of 'sensory information transmitted into the CNS' interfaces. Both types of prostheses are not ready yet for being usable in clinical applications. The idea of retinal implants (Zrenner, 2002; Javaheri et al., 2006) functions as a bypass of degeneration of the retina, like e.g. retinitis pigmentosa, by feeding a visual signal directly into the optic nerve. Visual cortical prostheses aim at directly stimulating areas in the visual cortex. The first tests on human patients started recently (Dagnelie, 2006).

Another question is how an external artificial device is integrated into the 'body representation'. Does the brain interpret a neuro-prosthetic arm just as a tool or does it really replace the lost limb (Iriki et al., 1996; Maruishi et al., 2004; Gurfinkel et al., 1991; Maravita et al., 2003)? This question is not answered completely yet. Especially, it is unclear what happens if sensory feedback from the limb (e.g. touch or position information) is given to the brain. First experiments with monkeys performing a reaching task while stimulating primary somatosensory cortex by electrical impulses in order to induce task cues showed that such a feedback can improve the performance of an BCI (Fitzsimmons et al., 2005).

The advantage of invasive methods lies in the high spatial resolution over a large frequency range resulting in a large information bandwidth. High transfer rates are typically very desirable but invasive methods, like noted earlier, have the disadvantage that they require to open the skull and to place electrodes on top of, or inside the brain. Another drawback of invasive recordings is that cables connecting the implanted

electrodes to external devices have to leave the skull through an opening, which is a potential source of infections and a potential hazard of getting stuck with the cables. We proposed an idea to eliminate the problem with the cables by using wireless information and energy transmission (Pawelzik and Rotermund, 2005) (see Fig. 5.2). Our contributions focus on scalability in size (as small as possible) and on the number of measurement sites (as many as possible). We expect that if we use analog information processing and information coding then this will enable us to build very small implants with a low number of components. The smaller the implants are, the more implants can be used in the same region. But not only the size is important for the question how many implants can be installed. Also the aspect of the available information bandwidth for sending the measured information out of the skull restricts the number of implants. Our approach deals with this problem by using a different carrier frequency for each of the implants and thus allows to transmit the information from many implants in parallel and realtime. Since this type of wireless information transmission is a key technology in the field, several other groups also develop wireless invasive measurement systems for neuronal activities, e.g. (Mohseni et al., 2005; Knutti et al., 1979; Morizio et al., 2005; Chien and Jaw, 2005; Irazoqui-Pastor et al., 2003; Martel et al., 2001; Irazoqui-Pastor et al., 2005).

Neuro-prosthetic arms

In the following, I want to discuss in more detail neuro-prosthetic devices for arms. For reconstructing intended movements from neuronal activities, it is helpful to understand how information about movements is expressed in neuronal activities of motor cortex cells. After introducing the main theories regarding this topic, I will continue presenting some techniques to extract information about the intended movement from measured data.

Neuronal coding of movements For movements in two dimensions, different studies investigated the response behavior of neurons from the motor system (mainly motor cortex). Typical paradigms are e.g. center-out tasks. There the subject has to move its hand from a start marker (at the origin) to a target. These targets are circularly aligned around the start marker. Moran and Schwartz (Moran and Schwartz, 1999b) suggested that the cortical activity $f(t)$ in dependency of (the absolute value of) velocity ν and direction ϑ of a moving finger evolves as

$$f\left(\nu(t-\tau),\vartheta(t-\tau)\right) \;=\; \nu(t)\left(b_0 + b_1\sin(\vartheta(t)) + b_2\cos(\vartheta(t))\right)\,, \qquad (5.1)$$

with b_0, b_1, and b_2 as constants, and τ as the typical delay between the neuronal activity of a movement and performing the movement itself.

In (Paninski et al., 2004) it was proposed to describe the neuronal response for hand movements in first order by

$$f\left(\nu(t),\vartheta(t)\right) \;=\; b_0 + b_1\nu(t)\cos(\vartheta(t)-\vartheta_{PD})\,, \qquad (5.2)$$

where ϑ_{PD} is the 'preferred direction' of the neuron. ϑ_{PD} denotes the direction where the neuron shows its strongest response. b_0 and b_1 are both constants. A similar conclusion can be found in (Georgopoulos et al., 1986; Georgopoulos et al., 1982) as well as its extension to 3D movements (Georgopoulos et al., 1988). In this publication they are focused on the correlation between the direction ϑ and the neuronal activity. The influence of the velocity is ignored. It seems that simple cosine-shaped functions might be an oversimplification because asymmetric and bimodal tuning functions were also found in this context (Amirikian and Georgopoulos, 2000).

It has to be noted that if load is applied to an arm then tuning changes (Kalaska et al., 1990). Experiments with primates working against 3D force-fields (Taira et al., 1996) suggest that the connection between neuronal activities and a force (with F as magnitude and ϑ_F as direction of the force) can be described by

$$f(F, \vartheta_F) = b_0 + b_1 \cos(\vartheta_F - \vartheta_{F,PD}) + b_2 F + b_3, \tag{5.3}$$

with $\vartheta_{F,PD}$ as the preferred directions of the neurons for forces and b_0, b_1, b_2 and b_3 as constants.

It is important to understand that the real situation is more complex than described by these equations. The neurons are typically not tuned exclusively to one parameter. Ashe and Georgopoulos (Ashe and Georgopoulos, 1994) revealed that most of the tested neurons have a strong correlation between direction and magnitude of the response, like described earlier. In addition, there exists a weaker influence of the velocity and the actual position on the neuron's response properties. Acceleration also affects the neuronal activities of those cells, but its influence is less intense. These results motivated Todorov (Todorov, 2000) to propose that these 'high-level' parameter correlations with neuronal responses are just a secondary effect from correlations between 'low-level' muscles commands and their neuronal representation.

Information about arm movements can also be extracted from epicortical field potentials (Mehring et al., 2004; Ball et al., 2004) and local field potentials (LFP) (Rickert et al., 2005; Scherberger et al., 2005). The LFPs can contain information that is complementary to the information carried by simultaneously measured spiketrains (Mehring et al., 2003).

Decoding of movements from neuronal signals After measuring neuronal activities, the user's intended action has to be extracted from that data. This can be a complex problem. Often, the first step is to 'clean' the recorded data and divide the data, as best as possible, into independent information channels. This can be done by e.g. applying the current source density method on EEG or LFP data for reducing spatial perturbations (Mitzdorf and Singer, 1979; Mitzdorf, 1985) or by performing 'spike-sorting' (Lewicki, 1998; Kreiter et al., 1989) for identifying and assigning the corresponding neuron identifiers to spikes recorded from extracellular electrodes. The idea behind this procedure is to improve properties of the data for enhancing the extraction of the relevant features. The last step is to extract intended actions from the

informative features on the data using suitable estimation algorithm (e.g. the support vector machine (Cortes and Vapnik, 1995), see section 2.2.4). The estimator must be trained on a training-data set before it can be used. When using support vector machines, the learning process segments data space, assigning different possible actions to different segments. After learning, the SVM can be used to select the intended action from the features while recording the data. The selection of the pre-processing steps, feature extraction methods, and estimation algorithms depends highly on the statistical properties of the data and how the required information is hidden in the recorded signals. These properties of the data may not be stable over time which can adaptation of the estimation algorithm necessary.

A popular method is to reconstruct movements from neuronal population activities assuming cosine-tuned neurons (Georgopoulos et al., 1986; Georgopoulos et al., 1988). For this method one assumes that one measures the rate f from N different neurons and that the tuning function is given, for each of these neurons, by

$$f_i(\vartheta) \;=\; b_{0,i} + b_{1,i}\cos(\vartheta - \vartheta_{PD,i})\,.$$

The population vector (Dayan and Abbott, 2001) is defined by

$$\hat{x} = \sum_{i=1}^{N} \frac{f_i - b_{0,i}}{b_{1,i}} \cdot (\cos(\vartheta_{PD,i}), \sin(\vartheta_{PD,i}))\ .$$

With perfect cosine tuning with uniformly distributed preferred directions, the population vector for decoding direction is equivalent to maximum likelihood estimation (Serruya and Donoghue, 2004). Differences to this type of tuning cause a non-optimal decoding. In particular, tests showed that the use of population vectors for reconstructing absolute values of velocity may lead to unbeautiful results. It is often better to use the OLE (Salinas and Abbott, 1994), but the number of alternative estimators is huge (Serruya and Donoghue, 2004; Andersen et al., 2004a).

In addition, a strategy to improve the performance of on-going reconstructions of intended movements is to use prior information about the expected limb dynamics or about the typical ranges of the variables to be estimated. For example, it is reasonable to assume that arm movements are continuous in space. For incorporating this information into the simulation it is possible to use e.g. sequential Monte-Carlo methods (or also called 'particle filters') (Arulampalam et al., 2002) which are an extension of Kalman filters (Welch and Bishop, 2004; Black et al., 2003). Or to include information about the next possible action by (hidden) Markov models (Rabiner, 1989).

5.3 The model for the simulations

For demonstrating the capabilities of the new adaptation procedure, a situation is simulated, where a 2D robotic arm has to be controlled by an adaptive decoding of

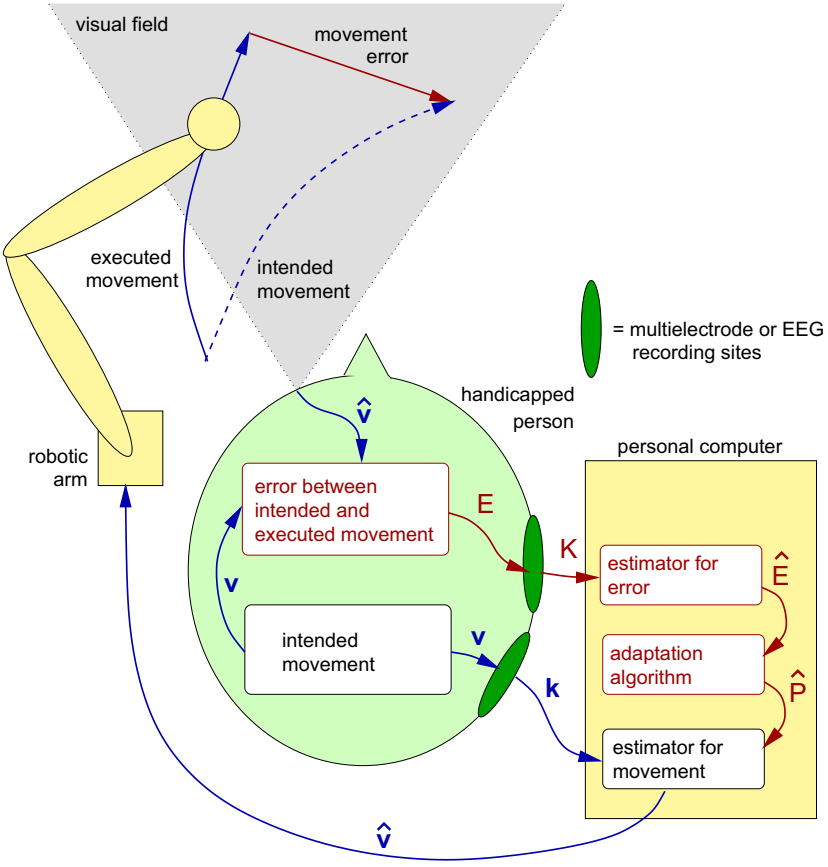


Figure 5.3: Schematic view of the model. A handicapped person is intending to execute a reach movement. In the brain of the subject (light green), the velocity $\boldsymbol{\nu}$ of the intended movement is encoded in the spikes of neurons from motor cortex being recorded as a spike count vector $\mathbf{k}$. $\mathbf{k}$ is fed into an algorithm running on a personal computer (yellow, right) utilizing the parameters $\hat{\mathbf{P}}$ to make an estimation $\hat{\boldsymbol{\nu}}$ of the intended movement. $\hat{\boldsymbol{\nu}}$ is subsequently used to control a robotic arm (yellow, left) executing the movement. The resulting mismatch E between intended and executed movement is perceived by the subject (to whom the originally intended movement is known), and encoded by neurons in error-monitoring brain regions. This error signal is simultaneously recorded by a second electrode array delivering a total spike count K (from which an estimate $\hat{\mathsf{E}}$ of the original error can be computed). An adaptation algorithm similar to reinforcement learning schemes (Sutton and Barto, 1998) employs a Monte-Carlo procedure for changing the parameters $\hat{\mathbf{P}}$ of the motion estimator, seeking to improve the performance of the prosthesis by minimizing E. This scheme differs significantly from a normal prosthetic application because of its use of an internal error source (additional components and signals required are drawn in red).

brain signals (see Fig. 5.3). It should be noted that the on-line adaptation strategy is not limited to this special kind of device. It can also be expected to be appliable prostheses where the user of the device can evaluate its performance and deliver a suitable error signal about this evaluation. In setups where we can apply the proposed on-line adaptation schema, the system typically consists of two parts, namely one 'internal', and one 'external' part. The internal part refers to two neuronal populations. One population is supposed to evoke activity patterns that correlate with the intended action. E.g. for arm movements, we can find such neuronal populations in the posterior parietal reach region or the dorsal premotor cortex (Andersen et al., 2004a). The second population of nerve cells (from an error-monitoring brain region (Ridderinkhof et al., 2004b)) is supposed to deliver an error signal that is correlated to the performance or mismatch between the intended and perceived actions of the external device. The activity patterns of both neuronal populations are recorded and delivered as input to the 'external' part.

The external part of the model is the controlling system for the prosthetic device and the device itself. It comprises three algorithms for processing the incoming data. One estimation algorithm decodes the intended movement signal. A second estimator interprets the error signal. The third component is the on-line adaptation algorithm. The estimated intended movement is used to control the external device. In parallel, the error signal is used to adapt the decoding algorithm of the intended movement signal.

5.3.1 Neural Encoding of Intended Movement

For the simulation we use, as the variable of interest, a movement velocity vector in 2 dimensions. This vector

$$\boldsymbol{\nu} \;=\; \{\nu\cos(\varphi), \nu\sin(\varphi)\} \tag{5.4}$$

with direction φ and length ν (ranging from 0 to $\nu_{\max}$) is representing the intended movement of the robotic arm. In section 5.2.1 and 5.2.3 we discussed several possible sources for neuronal activity correlated with intended movement signals. A good source for such signals (Schwartz, 2004; Andersen et al., 2004a; Andersen et al., 2004b) are neurons in the posterior parietal reach region or the dorsal premotor cortex which have shown to display a substantial velocity tuning suitable to be exploited in neural prosthesis. We will approximate the shape of the neurons' tuning to the direction of movement by cosine functions (Schwartz et al., 2001). Data about the intended movement is virtually recorded from N_ν neurons. Regarding the absolute value of the velocity we will use an approximation by a linear function (Moran and Schwartz, 1999b; Paninski et al., 2004; Black et al., 2003).

The mean firing rate f_i of neuron i can then be written as

$$f_i(\boldsymbol{\nu}) = f_i^{\text{off}} + \frac{f_i^{\text{mod}}}{2}\left(1 + \frac{\nu}{\nu_{\max}}\cos\left(\varphi - \varphi_i\right)\right) . \tag{5.5}$$

f_i^{off} denotes the baseline, and f_i^{mod} the maximum modulation of the firing rate, respectively. φ_i indicates the preferred direction of the neuron. In order to simulate a situation like it is encountered in real experimental settings, we require a substantial set of neurons to be not tuned at all. Furthermore, we assume that a large fraction of the N_ν neurons are nearly silent (the choice of the corresponding parameters will be discussed later). The firing of the neurons is described by a Poissonian process, which is a good approximation of the in-vivo firing properties of cortical neurons (Shadlen and Newsome, 1998). It follows that the probability p to observe k_i spikes in a time window of length T is given by

$$p(k_i \mid f_i(\nu, \varphi), T) = \frac{1}{k_i!} \left(f_i(\nu, \varphi) T\right)^{k_i} \exp\left(-f_i(\nu, \varphi) T\right) . \tag{5.6}$$

For simplicity, time in these simulation is sliced into intervals of duration T. For each interval, a vector of spike counts $\mathbf{k} = \{k_1, \ldots, k_{N_\nu}\}$ is drawn from the Poissonian distribution and used for the estimation of the intended movement. The real parameter vector describing neuronal encoding $\mathbf{P} = \{f_1^{\text{off}}, \ldots, f_{N_\nu}^{\text{off}}, f_1^{\text{mod}}, \ldots, f_{N_\nu}^{\text{mod}}, \varphi_1, \ldots, \varphi_{N_\nu}\}$ is unknown to the estimation algorithm. The task of the adaptation procedure is to find parameters $\hat{\mathbf{P}}$ which are as close as possible to the real values $\mathbf{P}$.

5.3.2 Estimation of Intended Movement

In section 2.2.3 and A.2.2 we discussed how to derive the optimal linear Bayesian estimator for the minimum mean squared error between estimated velocity $\hat{\boldsymbol{\nu}}$ and intended velocity $\boldsymbol{\nu}$ for tuning functions with linear speed modulation and cosine-tuning for the direction of velocity under Poisson noise. Given the spike count vector $\mathbf{k} = \{k_1, \ldots, k_{N_\nu}\}$ and the real (unknown) set of parameters $\mathbf{P}$ for the velocity coding tuning curves, the estimator is given by the expression

$$\hat{\boldsymbol{\nu}}(\mathbf{k}) \;=\; \sum_j^{N_\nu} \left(k_j - T \left(\frac{f_j^{\text{mod}}}{2} + f_j^{\text{off}} \right) \right) \mathbf{D}_j \,, \tag{5.7}$$

with the vector coefficients $\mathbf{D}_j$ defined by

$$\begin{aligned} \mathbf{D}_j \;&=\; \sum_i^{N_\nu} f_i^{\text{mod}} \nu_{\max} \{\cos(\varphi_i), \sin(\varphi_i)\} \\ &\times \left[\left\{ \frac{T}{2} f_i^{\text{mod}} f_j^{\text{mod}} \cos\left(\varphi_i - \varphi_j\right) + 8\delta_{i,j} \left(\frac{f_i^{\text{mod}}}{2} + f_i^{\text{off}} \right) \right\}^{-1} \right]_{i,j} \end{aligned} \tag{5.8}$$

Since the tuning properties $\mathbf{P}$ of the neurons are unknown, we have to use the adapted, approximate parameter set $\hat{\mathbf{P}}$ instead for the estimation with Eqs.(5.7),(5.8). In the typical case where both parameter sets do not match, we can use one particularly useful feature of the linear velocity estimator for calculating the error made through using the

wrong parameter set. If we know $\mathbf{P}$ and $\hat{\mathbf{P}}$, then it is possible to compute an analytical solution for the mean squared error in closed form, given that coding takes place with parameters $\mathbf{P}$ while decoding uses the parameters $\hat{\mathbf{P}}$,

$$\begin{aligned}\chi(\hat{\mathbf{P}}|\mathbf{P})^2 &= \frac{v_{max}^2}{2} - 2\sum_{i}^{N}\left(L_{i,x}\hat{D}_{i,x} + L_{i,y}\hat{D}_{i,y}\right) \\ &\quad + \sum_{i}^{N}\sum_{j}^{N}\left(Q_{i,j} + \hat{M}_i\hat{M}_j - 2M_i\hat{M}_j\right)\left(\hat{D}_{i,x}\hat{D}_{j,x} + \hat{D}_{i,y}\hat{D}_{j,y}\right) . \quad (5.9)\end{aligned}$$

Using the equations from the appendix C.1, $\mathbf{L}_i$, $Q_{i,j}$ and M_i are computed with the real parameters $\mathbf{P}$ of the coding system, while $\hat{\mathbf{D}}_i$ and $\hat{M}_i$ are calculated with the parameter set $\hat{\mathbf{P}}$ used in the adaptation process. $\chi(\hat{\mathbf{P}}|\mathbf{P})^2$ can be used as a tool for testing adaptation algorithms during their development. In addition, $\chi(\mathbf{P}|\mathbf{P})^2$ gives the smallest mean squared error which can be achieved under the neuronal noise by any conceivable adaptation process (see Fig.5.5).

5.3.3 Neural Encoding of Perceived Error

The on-line adaptation procedure requires a measureable neuronal representation of the error between intended and performed actions of the prothesis. The neural basis of error representation, error-monitoring and error encoding is still a subject of intense research (see for example (Diedrichsen et al., 2005; Ridderinkhof et al., 2004b; Schultz, 2004; Schultz, 1998; van Veen et al., 2004; Holroyd and Coles, 2002; Yeung et al., 2004a; Nakahara et al., 2004; van Schie et al., 2004; Musallam et al., 2004)). In section 5.2.2 we briefly discussed different types of reward and error representations in the CNS. Actually, it is not clear which regions in the brain can act as suitable signal sources. Therefore it is essential to specify the key properties of an error signal required for adapting the parameters of neural prostheses. This will allow experimentalists to search specifically for a signal with these properties.

The on-line adaptation process is exclusively based on a binary decision: A suitable error signal should reliably indicate whether a potential new set of parameters for the estimator of the movement (or of any other action) results in a better performance (as measured e.g. by the difference between the intended velocity and the performed velocity, the difference of the intended position and the real position, or the mismatch between intended and executed trajectory). The error signal should hereby reflect the performance averaged over a representative set of single actions. For the adaptation process an averaged performance is necessary because the error value has to represent the quality of the estimator for all possible movements. Using only the error value of one single movement for the adaption process can lead to an optimisation of the decoding for the most frequent movements with simultaneous disintegration of the quality for less frequent actions. Either the signal itself is already an averaged quantity when being recorded, or the averaging process must be carried out after recording in a pre-processing step by the controller.

Since the putative error signal is recorded from a neurophysiological signal source it may also suffer from non-stationarities. For a correct binary decision during adaptation, it is crucial that this error signal depends monotonically on the performance of the prosthesis. In any other respect, the signal is allowed to be non-stationary on a time scale larger than the time scale on which the performance is averaged and the binary decision is made. It is also permitted that the error signal is delayed to the actually executed limb movement, as long as the delay is smaller than the averaging time interval. However, it is not allowed that monotonicity reverses its sign over time, because this would then lead to error maximization.

For the concrete implementation, let us assume an error signal coding the squared error between intended and executed velocity vector. Furthermore, let us assume a linear dependency between the firing rate of error-monitoring neurons and the squared error. It should be noted that these assumptions are not essential for the success of the adaption process.

In detail, the simulation is based on the following equations: The difference between the intended movement $\boldsymbol{\nu}$ and its estimate $\hat{\boldsymbol{\nu}}$ is quantified by the squared error $E(\boldsymbol{\nu}, \hat{\boldsymbol{\nu}})$ given by

$$E(\boldsymbol{\nu}, \hat{\boldsymbol{\nu}}) = \sqrt{(\nu_x - \hat{\nu}_x)^2 + (\nu_y - \hat{\nu}_y)^2}\,. \tag{5.10}$$

We will assume that recordings in an error-monitoring brain region are made from N_E neurons with similar tuning functions increasing monotonously with increasing error E. In particular, let us assume that the *population* firing rate f_E is given by a linear function of the error E,

$$f_E(E) = F^{\text{off}} + F^{\text{mod}} E\,, \tag{5.11}$$

with offset F^{off} and maximum firing rate modulation F^{mod}. As for the motor cortical neurons, the population firing rate f_E is observed as a stochastic spike count K drawn from a Poissonian distribution according to Eq.(5.6).

5.3.4 Adaptation

During adaptation, the parameters $\hat{\mathbf{P}}$ of the movement estimator are optimized with respect to a loss function. The ultimate goal of an adaptation process is to reach a global minimum of the loss function with respect to the adapted parameter set, which in this simulation is the case if $\hat{\mathbf{P}} = \mathbf{P}$. If this set of parameters was found then the prosthesis would operate with maximal performance.

In theory, a well-defined loss function incorporates the statistics of the intended movements, an error metrics (a function that quantifies the difference between the real movement and the estimated movement in a scalar value) and constraints on the system. This information can then be used to find the optimal estimator. In reality, the loss function may be unknown to external observers and therefore also to the adaptation algorithm. That is not a problem as long as the performance of the prosthesis

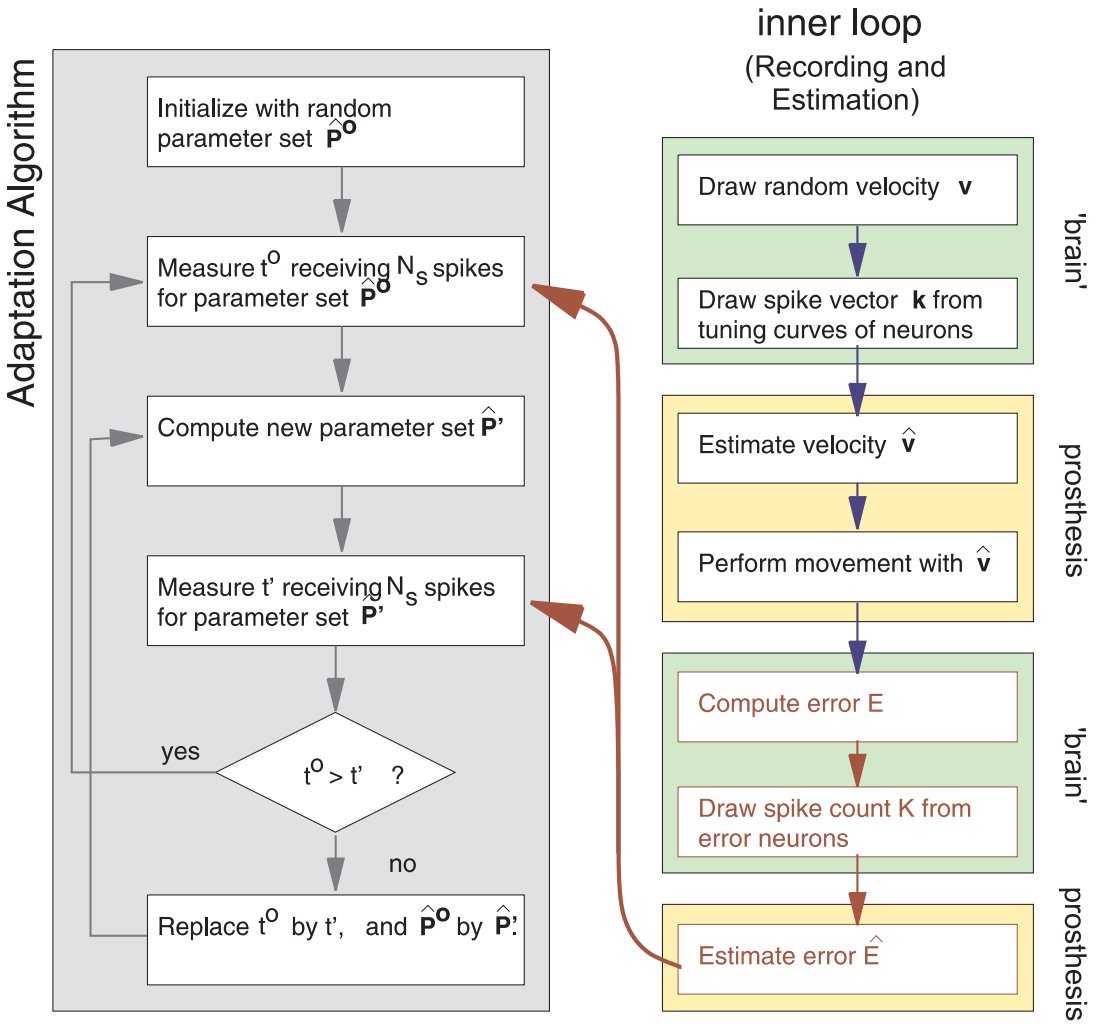


Figure 5.4: Schematic diagram of the simulations (for details, see the text). An 'inner loop' executed each time interval T draws an intended movement velocity from a distribution, computes the corresponding spike counts of the velocity and error neurons, and performs the estimations of the velocity and error signals. The adaptation algorithm evaluates different parameter sets constructed by a Monte-Carlo procedure, by averaging the error signal over a succession of periods T. Hereby, it realizes a stochastic gradient descent on the error signal E.

is represented monotonically through the recorded error signal and therefore allows to find better sets of parameters while minimizing the value of the error signal (for the on-line adaptation process it was assumed that a lower performance results in a higher neuronal activity of the error-coding neurons). By performing only a binary decision, we may sacrifice adaptation speed but gain robustness against non-stationarities in the error coding.

The realization of the adaptation algorithm embraces the on-going estimation of $\boldsymbol{\nu}$ and the observation of the corresponding error signal K (the number of spikes from the population of error coding neurons) in each time interval T, by changing the estimation parameters $\hat{\mathbf{P}}$ on a larger time scale. For an overview of the complete procedure, see the diagram in Fig. 5.4.

The on-going estimation of $\boldsymbol{\nu}$, which uses the equations introduced in the previous subsections and the measurement of the error-related neuronal activity is executed in a so-called 'inner loop'. Within this loop, at first an intended movement velocity $\boldsymbol{\nu}$ is drawn from a uniform distribution on a unit disc ($\nu_{\max} = 1$). The resulting, randomly drawn spike count vector $\mathbf{k}$ from the model's motor cortical area is then used to estimate the velocity $\hat{\boldsymbol{\nu}}$ with the current parameter set $\hat{\mathbf{P}}$. $\hat{\boldsymbol{\nu}}$ controls the movement of the robotic arm, leading to an error E which is encoded in a population spike count K from the model error-monitoring brain region. This loop is then repeated.

The actual adaptation algorithm ('outer loop') then proceeds as follows:

1. Prior to the adaptation, the parameter set $\hat{\mathbf{P}}^0$ for the estimation of intended movement velocity is initialized with random values.
2. The current parameter set $\hat{\mathbf{P}}$ used in the inner loop is set to $\hat{\mathbf{P}}^0$, and the inner loop is executed until a pre-defined number of spikes N_s has been recorded from the error neurons. The necessary time for receiving the N_s spikes is stored in a variable t^0.
3. A new parameter vector $\hat{\mathbf{P}}'$ is randomly drawn from a small neighborhood around $\hat{\mathbf{P}}^0$, whose size is scaled in dependence on t^0 (for details, see the appendix C.2).
4. The current parameter set $\hat{\mathbf{P}}$ is set to $\hat{\mathbf{P}}'$, and the inner loop is executed until again N_s spikes have been recorded from the error neurons. The length of the necessary time interval for accumulating these N_s spikes is stored in a variable t'.
5. If $t^0 > t'$, the new parameter set $\hat{\mathbf{P}}'$ is discarded and the algorithm goes back to step two. If $t^0 \leq t'$, parameter set $\hat{\mathbf{P}}^0$ will be replaced by $\hat{\mathbf{P}}'$, t^0 by t', and the algorithm will proceed with step three.

In regular time intervals, re-evaluation of the currently used parameter set $\hat{\mathbf{P}}^0$ through step 2 is necessary. This is important, because t^0 can become corrupted due to non-stationarities, stochastic neurons, and sampling noise.

Choosing N_s is subject to a tradeoff between precision and speed. With low N_s, adaptation speed will be high, but the performance is only averaged over a non-representative set of few movements which may lead to the acceptance of a less optimal parameter set. With high N_s, performance is evaluated more accurately but fewer adaptation steps can be made in a fixed amount of time. Saturation on the quality of the parameters occurs when the potential gain of an adaption step is of the same order of magnitude as the noise level on the evaluation of the performance. This problem can be alleviated by increasing N_s (e.g. dynamically in dependence on t^0).

Besides using a Monte-Carlo sampling, there are several other putative adaptation strategies in reinforcement learning problems (for an overview, see (Sutton and Barto, 1998)), like e.g. gradient-based methods. However, in this setting the objective function is not known analytically. Numerical computation of a N-dimensional gradient requires sequential evaluation of the local variation of the loss function in all N dimensions. Due to the noise, this procedure takes a very long time and thus renders gradient-based methods more vulnerable to non-stationarities. For these reasons, and because of its general applicability, a Monte-Carlo algorithm was chosen which is more robust against non-stationarities and performs a faster adaptation, but only into a random direction.

5.3.5 Choice of Parameters

Recent progress in multi-electrode recording techniques allows to record from up to 64 largely independent channels in the motor cortex of a macaque monkey (Schwartz, 2004). However, typically only a fraction of all channels will contain signals which are strong enough and reliable enough for use in a neural prosthesis (Mehring et al., 2003). We decided to choose parameters which establish a sort of worst-case scenario, in order to test for the computational limits of the adaptation and estimation algorithms.

For the simulation, let us assume that it is possible to record from $N_\nu = 64$ neurons. The tuning properties of these neurons are drawn from a probability distribution which characterizes the typical variations when inserting a real multielectrode array into a brain: with probability $p = 0.5$, neurons are chosen to be nearly silent (firing rates below 3 Hz). With probability $p = 0.25$, neurons are barely tuned, but fire with random frequencies between 0 and 50 Hz (Moran and Schwartz, 1999a), independently on the movement velocity. Note that taken together, these neurons make up a percentage of 75% of all neurons, and that they are of no use for decoding the angle and absolute velocity of the movement. With probability $p = 0.25$, neurons will have a random firing rate offset f_i^{off} and a random modulation f_i^{mod}, choosen such that $f_i^{\text{mod}} + f_i^{\text{off}}$ does not exceed 50 Hz. Direction tuning preference φ_i is assigned randomly, which yields a very inhomogeneous coverage of the movement angles between 0 and 2π. We further set $\nu_{\max} = 1$ and $T = 1$ s in the simulations.

When recording from error-monitoring brain regions, it may be necessary to restrict recording to only one or a few channels. The reason for this limitation is that many of

these areas are located deep below the brain's surface (Ridderinkhof et al., 2004b). In this deeper regions, recording is more difficult. Moreover, the literature on these signals reveals that we can assume as a worst case that typical, maximum firing rates of the corresponding neurons often do not exceed 15 Hz, with a spontaneous baseline firing rate of about 5 Hz (for example, reward-related neural activity (Tremblay and Schultz, 2000)). Using these parameters, a simultaneous recording from $N_E = 5$ neurons with a mean maximum firing rate of 10 Hz was simulated, thus leading to the maximum population firing rate of $F^{mod} + F^{off} = N_E \cdot 15$ Hz $= 75$ Hz with a baseline of $F^{off} = 25$ Hz.

Taken together, the parameter's choice reflects the harsh conditions under which a neural prosthesis must work reliably: synaptic and neuronal noise, neurons which display no response or have no tuning properties, and large offsets in the firing rate due to spontaneous activity.

5.4 Results from the Simulations

The main motivation for this chapter was to show that neural prostheses can successfully be adapted with a suitable, internal error signal, counteracting strong non-stationarities in neural coding and signal recording. Consequently, a scenario in which a complete change of the tuning properties of the motor cortical neurons took place (Fig. 5.5) was simulated first. In the shown example, before $t = 0$ the adaptation of $\hat{\mathbf{P}}$ had reached a stationary state, and the decoding of a sample intended movement trajectory was almost perfect (see leftmost inset in Fig. 5.5). At $t = 0$, the tuning of the neurons was changed completely, corresponding to an entire re-initialization of the parameters $\mathbf{P}$ with new, random values drawn from the typical distribution of tuning values. The prediction error E instantly increased to values where a successful decoding of the intended movement was impossible (see center inset in Fig. 5.5). However, with a half-life period of about 90 minutes, the Monte-Carlo reinforcement learning algorithm succeeded in re-adapting $\hat{\mathbf{P}}$ and in decreasing E. Reconstruction of an intended movement was again possible after only a couple of hours of adapting $\hat{\mathbf{P}}$ (rightmost inset in Fig. 5.5).

In Fig. 5.6, a typical distribution of tuning curves for the 64 neurons described by $\mathbf{P}$ is shown. Only few neurons are substantially tuned, and are on that account useful in decoding the intended movement. In Fig. 5.7, the corresponding estimated tuning curves with adapted parameters $\hat{\mathbf{P}}$ are shown. From 64 signals available, the adaptation algorithm has chosen only 8 substantially tuned neurons for movement estimation. In comparison, Fig. 5.7 shows that the original tuning curves of the selected neurons, depicted as the blue bars, do not differ significantly from the estimated ones. In both figures (Figs. 5.6,5.7) only tuning for 8 directions is shown, like it is observed in a typical 2D 'center-out' reaching task with 8 targets (Mehring et al., 2003).

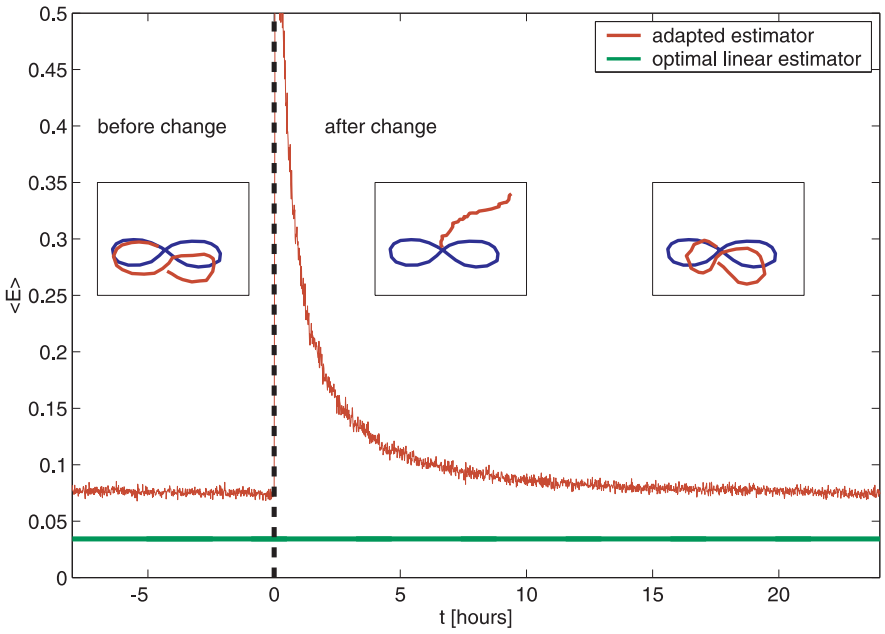


Figure 5.5: Simulation demonstrating that adaptation is possible even after a complete, radical change in neuronal tuning: the red curve shows the mean error $< E >$ in estimating intended velocity, averaged over 1000 trials (chance level is ≈ 0.9) with on-going adaptation of estimation parameters during the whole simulation period. At $t = 0$ the direction preferences, spontaneous firing rates, and the maximum firing frequencies of all 64 'recorded' neurons with cosine-shaped tuning curves and linear speed modulation are completely re-initialized with random values. For $t \neq 0$, tuning properties are held constant. After some hours of re-adapting the estimator's parameters, the performance prior to the change has been restored. The green curve displays the minimal mean error achievable if the velocity estimation would have been made with the real tuning parameters (which are unknown to an external observer). The three insets visualize the prosthetic's performance prior to the tuning change (left), immediately after the change (center), and 24 hours after the change (right). Adaptation has been successful if the estimated trajectory (in red) closely approximates the intended trajectory (in blue).

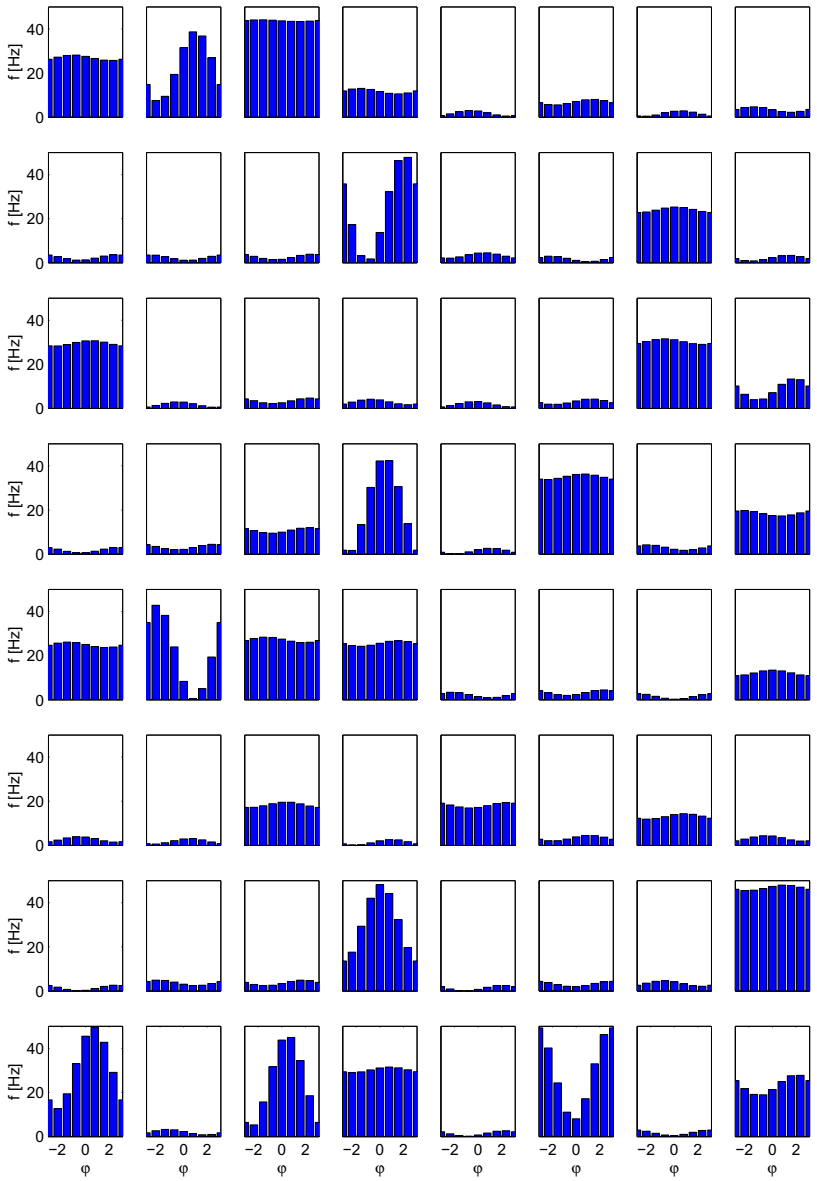


Figure 5.6: Directional tuning of the responses to an intended movement with maximum velocity v_{max} and varying angle φ, for $i = 1, \ldots, 64$ motor neurons. The heights of the blue bars indicate the mean number of spikes recorded in the corresponding conditions. Note that in this example, many neurons barely respond at all.

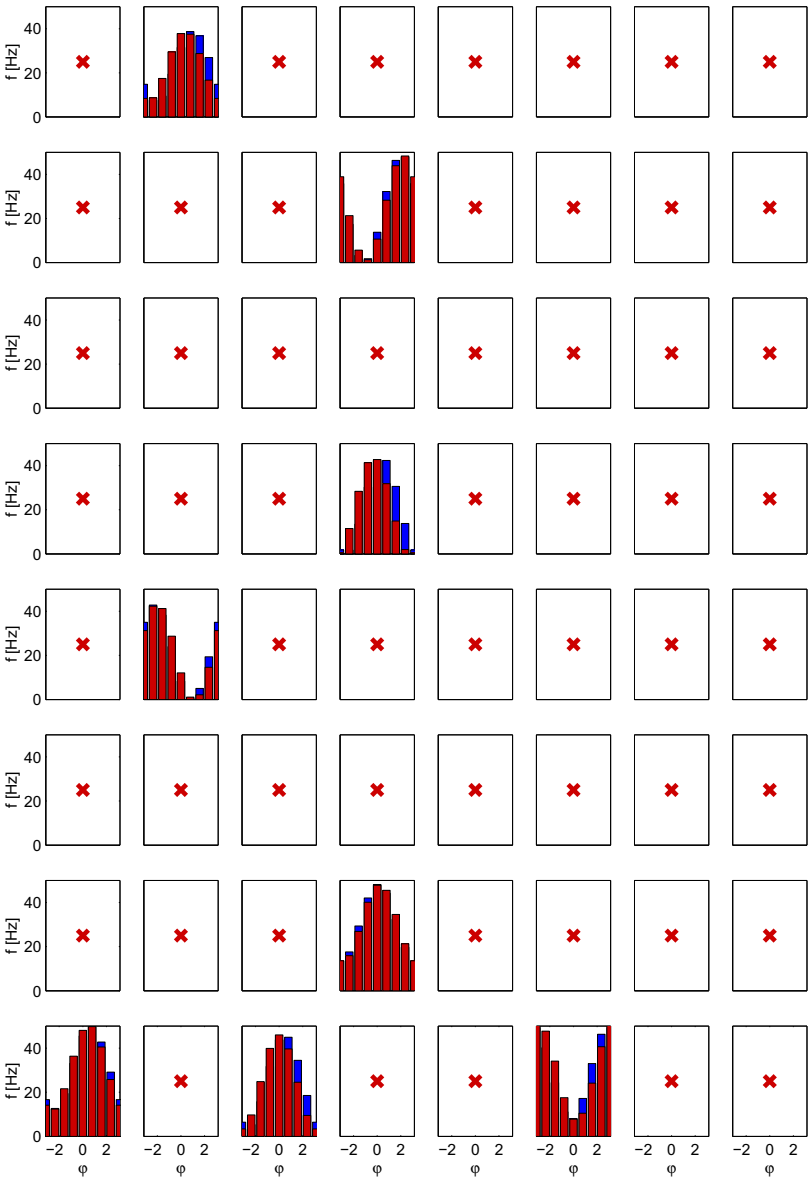


Figure 5.7: Directional tuning estimated by the Monte-Carlo adaptation algorithm. Recording sites excluded because of lack in variation of the spike count are marked with red crosses. The height of the red bars denotes the expected firing frequency at the corresponding site, if a movement of orientation φ with velocity $\nu_{\max}$ is intended by the subject. The blue bars show the real tuning curves at the corresponding recording sites.

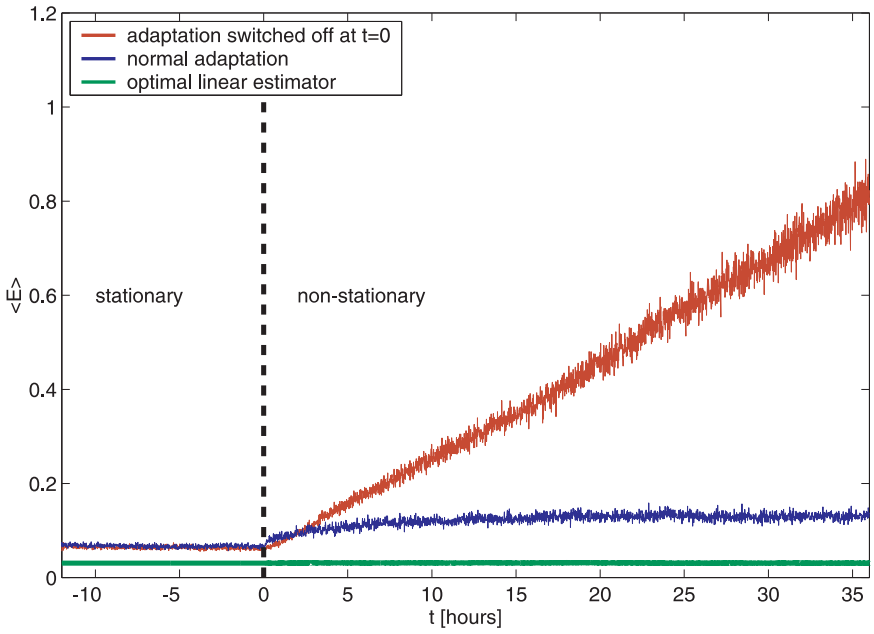


Figure 5.8: Error in decoding intended velocity under non-stationarities in the recording from the motor neurons. Before $t = 0$, tuning properties were held constant. Starting at $t = 0$, on average every 30 minutes one recording site changed its tuning properties. At the end of the simulation, tuning has changed completely for all 64 sites. Blue, mean error of 500 trials with adaptation of estimation parameters during the full simulation period. Red, mean error of 500 trials with adaptation switched off at time $t = 0$. For comparison, the minimum mean error $\chi(\mathbf{P}|\mathbf{P})^2$ achievable using the 'correct' parameters $\mathbf{P}$ with the optimal linear estimator is shown as the green line.

Furthermore we simulated a scenario where recording from the neurons in motor cortex is subjected to an on-going change in tuning (Lebedev et al., 2005; Andersen et al., 2004a). In the simulation depicted in Fig. 5.8, starting at time $t = 0$, tuning properties of the neurons change randomly according to a Poissonian distribution: on average, every 30 minutes of simulated time one recorded site changed its tuning properties completely. The new tuning function at the corresponding site was randomly drawn from the same distribution from which the tuning curves were drawn initially. For a reinforcement learning algorithm this situation is typically more difficult to handle: if the change of the encoding system is faster than the algorithm is able to find better parameter values for the velocity estimation, the resulting error will increase substantially, instead of decreasing. In the example, however, the non-stationarities influence adaptation only marginally, and the estimation error remains low. If no error signal would have been used, as is the typical case in current prosthetic applications, decoding of an intended velocity would have been impossible after only a few hours of simulation time.

In all the presented simulations, the intended velocities are restricted to a maximal length of ν_{max}. This restriction was not enforced for the estimated velocities, thus the error of an estimation done with a bad estimator is not bound.

5.5 Conclusion and Summary

In the context of neural prosthetics, one can distinguish between three approaches of choosing and optimizing the parameters of an estimator:

- *Learning*
  Learning starts with an initial, possibly guessed or randomly selected set of parameters. For optimizing the parameters, a task with well-defined goal is performed repeatedly. The mismatch between this goal and the outcome of the task is used to optimize the parameters of the prosthesis. After the subject has done sufficient training trials, the parameters of the prosthesis remain fixed (for example: co-adaptation (Taylor et al., 2002)).

- *Re-Learning*
  Re-Learning, which is also termed 'Offline Adaptation', employs essentially the same procedures as learning. The only difference is that after some time interval of using the prosthesis, the learning procedure is repeated. This approach can cope with non-stationarities in data acquisition and in the neural coding itself. However, between the special training sessions, the performance of the prosthesis degrades continuously.

- *(On-line) Adaptation*
  In contrast to the other methods, (On-line) Adaptation is done during the normal

use of the prosthesis, and it does not require a task with pre-defined goal. As a consequence, the actual performance of the prosthesis must be evaluated under the condition that the goal of an action is unknown. For this reason adaptation is only possible if an indirect, and task-independent signal source for the actual performance is available. If such a signal can be recorded, it offers the possibility to perform a freely chosen task while simultaneously optimizing the estimation parameters.

The most useful signal for this example would directly encode the difference between the target location or speed of an intended and actually executed movement. For a motor prosthesis, candidates for such an error signal could be neurons in or nearby areas which code the correlations of motor cortical plans of movement and observations of movement (putative areas include the superior colliculus (Stuphorn et al., 2000), dorsal pre-motor area (Boussaoud et al., 1998; Musallam et al., 2004; Fujii et al., 2000), ventral pre-motor area (Mushiake et al., 1997) or parietal reach regions (Batista et al., 1999; Musallam et al., 2004)). An advantage of these types of signals could be their proximity to the motor system: one can expect that the influence of other variables as e.g. the general emotional state of the subject on the firing rates remains small. But even if the error signal does not exclusively represent the performance of the prosthesis, the required binary decision in the adaptation process can still be made because all influences on the error (like for example through other internal mental states), which are acting on a larger time scale than the averaging and decision process, will be filtered out.

However, these brain areas may not be accessible for Brain-Computer Interfaces using electrophysiological recordings because of their position deep inside the brain. Other types of signals relate to the perceived difference between the intention of a subject, and the outcome of one of its actions (reward). Conveniently, such a signal would also be useful for other types of neural prostheses which barely involve the motor system. But most brain signals encoding reward are differential in nature: they depend heavily on the expected reward which yields a baseline for the activities in the corresponding neurons (Ridderinkhof et al., 2004a; Kobayashi et al., 2002; Shima and Tanji, 1998; Platt and Glimcher, 1999; Ito et al., 2003; Shidara et al., 1998). Depending on this baseline, firing rates are positively or negatively modulated, if the reward is higher or lower than expected, respectively (Tobler et al., 2005; Schultz et al., 1997; Matsumoto et al., 2003; Fiorillo et al., 2003). In principle, a differential signal poses no problem for the algorithm, but the success of using such an error signal depends on the dynamics of its baseline. As explained earlier (see Section 5.3.3), the baseline has to remain fairly constant while calculating the representative averaged error for the old and the new parameter set for the comparison. A further complication may be that reward signals are strongly influenced by the emotional state and motivation of the subjects (Satoh et al., 2003; Watanabe et al., 2002b; Roesch and Olson, 2003; Tremblay and Schultz, 1999) or the context (Tremblay and Schultz, 1999), and may therefore prove inappropriate for a reliable source of error coding. The error signal needs to depend monotonically on the perceived mismatch, which is certainly not the case for neurons

whose responses increase independently of whether the reward is larger or smaller than anticipated.

An EEG Brain-Computer Interface could serve as a non-electrophysiological information source for the error value by using event-related brain potentials (Schalk et al., 2000; Gehring et al., 1993; van Schie et al., 2004) to extract such a signal. Alternatively, the user can be trained (Wolpaw and McFarland, 2004b; Blankertz et al., 2002) to generate an error signal which is transmitted through EEG recording to the adaptation system of the prosthesis.

Through the described adaptation scheme, failures can effectively be used to increase the future performance of the neuro-prosthetic device during voluntary, non-instructed use. The origins and properties of error signals are not yet known precisely, thus only weak assumptions were made and signals with low information content were used. The simulations demonstrate that such hypothetical error signals suffice for on-line-adaptation, preserving and enhancing the performance of a prosthesis subjected to non-stationarities. Several important issues have to be kept in mind when this approach is considered for a real prosthetic application:

- Monotonicity is a crucial property of the error signal, and a change of its sign is not permitted. But since an error signal will be recorded from a population of neurons it is still very probable that the whole population retains the correct sign of the original dependency, even if single neurons occasionally violate the monotonicity condition.

- Reinforcement learning is difficult and very slow if many parameters have to be adapted. Thus, great effort must be put into an appropriate selection of the signals from a multitude of available recording sites. It is not the best strategy to use as many signals as possible. Instead the most predictive signals should be used.

- Using prior knowledge may substantially increase adaptation performance. Incorporating this knowledge into more sophisticated reinforcement approaches (as e.g. in (Xie and Seung, 2004a; Murata et al., 2002)) may accelerate learning and may lead to lower errors.

- On the contrary to using these more sophisticated approaches, the adaptation algorithm which was proposed is capable to adapt the decoding parameter set even without explicit information about the general shape of the neuron's tuning curves. However, it is important to reduce the number of parameters as far as possible to avoid a 'curse of dimensionality'.

- One putative problem may be competition between the adaptation processes in the brain and the on-line adaptation of the control system. It is unclear how these two processes will interact. The behaviour of this interaction will strongly depend on the two time scales of adaptation.

It has to be noted that before testing this approach in an experimental environment, it is necessary to identify reliable neural sources for an error signal with the required properties. Only then it would be possible to apply our procedure to a real prosthetic device. At the same time, this more detailed information about the attributes of such error signals could be used to improve the design of algorithms.

Chapter 6

Summary and Conclusion

Interacting with our dynamic environment requires to process huge amounts of sensory data in short time. This incoming stream of information is combined with internal states (e.g. memories or intentions) and results in actions. The fundamental mechanisms behind this fast information processing are still not understood. Even how information is stored in, and transmitted with sequences of action potentials is still under heavy debate. This thesis provides novel ideas to accomplish fast information processing, to understand adaptive coding strategies, and to perform unsupervised online learning of non-stationary representations.
In its first, genuinely theoretical part (chapter 3 - Information Processing Spike by Spike) this thesis develops a new concept in the field of fast information processing with single action potentials. The framework is based on stochastic generative models using Poissonian spike trains as input. It is capable of realizing arbitrary input-output functions, updating an internal representation with each incoming spike, for performing computations as fast as possible.
Leaving those purely theoretical considerations behind, the second part of this thesis (chapter 4 - Selective Visual Attention in V4/V1) investigates principles of adaptive neural coding in real data, focusing on the question how an internal cortical state, evoked by selective visual attention, modifies information processing in the brain. In collaboration with monkey neuro-physiologists we studied the influence of attention on the discriminability of visual stimuli through their neuronal correlates recorded as epidural field potentials.
The final part in this thesis (chapter 5 - Stabilizing Decoding Against Non-Stationaries) takes us towards a medical application for extracting internal brain states from neuronal activities. For controlling prosthetic devices with brain signals, reliable algorithms for estimating the intended actions of a person are required. A method was designed which allows to stabilise the estimator of a neuro-prosthesis against disruptions from non-stationarities in the characteristics of coding the intended actions, and from changes in their representations in the measured neuronal correlates.

In the following I will summarize and discuss the results from these three parts:

Information Processing Spike by Spike

For fast information processing, the Spike-by-Spike framework presented in the first part 3 is a promising alternative to approaches using rank-order coding (Thorpe et al., 2001). In contrast, this framework is very robust even under strong noise. Temporal information, describing the timing between two spikes, is ignored completely and only the relative number of spikes emitted by a population of neurons is used for information transmission and information processing. Nevertheless, simulations showed that already this noisy input, together with a clever algorithm, allows to compute given tasks rapidly and accurately (Ernst et al., 2007b; Pawelzik et al., 2006a; Rotermund et al., 2005; Ernst et al., 2004; Pawelzik et al., 2004; Pawelzik et al., 2003).

The information is processed within the framework of a generative model for Poissonian spike trains. In this approach probabilities for eliciting spikes can be interpreted as neuronal activities and synaptic weights are related to conditional probabilities to observe spikes given a putative cause. The structure of the Spike-by-Spike network allows to extend the approach to include e.g. statistical expectations, task-relevant information, and attentional modulations. Two types of learning strategies for training the weights were developed: A batch learning rule, which first collects the information about many input patterns and then updates the weights, and an on-line learning rule that is formally equivalent to a Hebbian-like learning rule with weight normalization. The on-line rule updates the weights after each received input spike, but with an update strength being some orders of magnitudes smaller than the update of the internal representation.

The new algorithm was able to learn and compute Boolean functions. This suggests that the network in principle can implement arbitrary computations. It was also able to learn to ignore input bits, which were not relevant for realizing a desired input-output-relation, during training and then to perform the computation with the (for this Boolean function) smallest possible network. It was even possible to construct a hierarchical network with more than one layer that can be used to compute Boolean functions. In addition to demonstrating the self-consistency of the basic framework, using more than one layer allows to reduce the total number of hidden nodes dramatically. Unfortunately, it is actually not clear how to learn these multi-layer Spike-by-Spike networks consistently. This is partially the consequence of not knowing an objective function for the intermediate layers. One idea, that still has to be tested, is to apply a wake-sleep-algorithm (see section A.4.2). Another approach could be to reduce the dimensionality of the learning problem by using Spike-by-Spike versions of Boolean primitives (like e.g. AND, NAND, XOR, and OR) as predefined modules. These primitives are then combined to implement the desired Boolean function, like in any modern digital computer.

Spike-by-Spike networks were also trained to recognise handwritten digits from the USPS database. With roughly 500 hidden nodes (and the corresponding set of weights), the Spike-by-Spike information processing algorithm was capable to exceed the recog-

nition rate of a nearest neighbour classifier (whose performance provides a good benchmark for classification problems) which uses more than 7000 patterns. Also the computation was done nearly as rapid as with the benchmark algorithm. In addition, the recognition performance turned out to be relatively robust against pixel noise and occlusions in the digit images.

The Spike-by-Spike algorithm has a free parameter ϵ for tuning the strength of the influence that a new spike will have on the actual internal hidden representation. In the example, where the handwritten digits had to be recognised, it was found that precision can be increased if this free parameter was adapted to the number of received input spikes. In a neuro-biological context this scaling of ϵ could maybe be realised by neural mechanisms like e.g. synaptic depression. In an additional study it would be interesting to investigate the improvements, which can be obtained by using biological plausible adaptation process for ϵ.
Furthermore, for different values of ϵ the appearance of the weights changes. Using natural images as input for training Spike-by-Spike networks resulted for small ϵ in an orthogonal basis function set for pixel data. Thus, in the limit of very small ϵ, the natural images are represented by linear superpositions of pixels. Larger ϵ lead to spatially extended weight structures and more sparse activation distributions over the hidden nodes. With the extreme value of $\epsilon = 1$ (maximal sparseness on the distribution of hidden activities) a winner-takes-all network is realised, and only one of the weights is used for explaining the input. The behaviour of the network in dependency of the ϵ-values between these two extreme situation has still to be thoroughly analysed and categorized.
Implementing a realistic pre-processing for the natural images, like it is known from the visual system, would allow to compare the weights generated by the Spike-by-Spike network with measured receptive field properties from the mammalian visual system. Using a more realistic pre-processing would let us expect a better match between the learned weights for different ϵ-values and the measurements.

Another interpretation of the ϵ-value is that it introduces a time scale for the validity of the internal representation and regulates the length of the 'memory' in the network. Smaller ϵ induce a longer memory than larger ϵ. This allows for small ϵ to include more features or basis functions, achieving higher accuracy in the representation of the input. However, building this more detailed representations needs also more input spikes and thus more time. This creates a trade-off between speed and accuracy. Speed and accuracy can also be improved by increasing the number of hidden neurons. In this context, comparing the speed of Spike-by-Spike networks with Thorpe's rank order hypothesis (Thorpe et al., 2001) would be interesting. This will be a subject in future research.

An important question for models explaining information processing in the CNS is whether it can be implemented by biologically realistic means. The short answer for this model is: It is not sure, but could be possible.
Some problems related to this question are non-local interactions between the hidden

nodes introduced by the denominator in the dynamics of the hidden neurons. The required information comprising analog values of weights and hidden activity, has to be exchanged between all hidden nodes. This data must be transmitted via spikes. One workaround is to represent the local information of one hidden node by the population activity of many neurons and use the spike count as an estimate for the products between the hidden values and the corresponding weights. Simulations showed that this is possible, but information processing is then not instantaneous and slows down considerably, because some number of spikes has to be collected to reliably represent the analog values.
The next concern regarding biological plausibility is raised by the necessary operations to be performed by the network: multiplications and divisive normalisation of the hidden node dynamics. Multiplicative interactions and divisive normalisation mediated by inhibitory interactions have been observed in real neurons. But it is not clear if these 'implementations' are pure realisations of the required calculation primitives, or if they are combinations of e.g. additive and subtractive interactions.
One final open question to be mentioned is how the hidden variable could be stored in a real biological system. This quantity is not allowed to change between two spikes, so the membrane potential of a neuron is not a good candidate. One working hypothesis is that the hidden variable may be implemented as one part of a multi-compartmental neuron model. This idea was tested in a simulation where the spike-by-spike model was successfully combined with a leaky integrate-and-fire model. A second approach would be to allow for decaying hidden activities. Preliminary tests (not shown) revealed that this approximation has the potential for delivering a partial solution.
However, for really finding satisfying answers to all these remaining questions concerning the biological plausibility of the model, more research has to be done.

All presented results were produced for static input distributions. Another approach solving partially this problem assumes that the observed input was generated by only one cause. The model was proposed by Deneve (Deneve, 2007a; Deneve, 2007b) and describes the dynamics of the external world as a hidden Markov model. In this algorithm, one neuron estimates this hidden Markov model. An on-line learning algorithm, allowing to train the parameters of the model (the weights of the connections and the time-constants) for this system is available (Mongillo and Deneve, 2007). A comparison between this model and the presented model was done by Ernst et al. (Ernst et al., 2007a). Other more complex models were developed (Beck and Pouget, 2007; Rao, 2004), whose implementation in neuronal models require very strong approximations. For further research activities, it will be interesting to extend the Spike-by-Spike model to dynamic input distributions. This would allow to apply the Spike-by-Spike network to movies (e.g. for compressing movies) and controlling tasks (e.g. balancing an inverted pendulum).

Selective Visual Attention in V4/V1

For a better understanding of the neuronal mechanisms of selective visual attention, the laboratory of Prof. Andreas Kreiter (University of Bremen) trained two monkeys to perform shape-tracking tasks while field potentials from visual cortex were recorded. We analysed this data by classifying the presented shapes and conditions of attention using estimation algorithms from machine learning (Support Vector Machines) (Rotermund et al., 2007a; Rotermund et al., 2007c; Rotermund et al., 2007b; Pawelzik et al., 2006d; Rotermund et al., 2006b; Rotermunda et al., 2005). With these methods, we gained valuable insight into information-carrying and behaviourally relevant aspects of the measured data, and identified the influence of selective visual attention on the neuronal activity patterns.

Hereby it was possible to reach a classification performance of up to 93% correct using the data from all electrodes and of up to 64.6% using only the data from a combination of electrodes above area V4.

Analysing the experimental data showed that the neuronal correlates of the shape's representation contain re-occurring activity patterns that allow to identify the corresponding shapes or conditions of attention with high precision. After a period of training, these classifications can be made in real-time. This allows to use these types of electro-physiological signals as data source for Brain Computer Interfaces. The analyses revealed that it is possible (with a precision of up to 93.7%) to decide within a 400 ms time window whether a shape in the left or right visual hemifield was attended or not. This could allow to construct a spelling device where each 400 ms a binary decision towards an intended word or letter can be made. It is possible to use such a device as communication neuro-prosthesis, allowing handicapped persons (e.g. totally locked-in patients) to communicate with their environment. This putative application rises directly further questions about useful or usable properties of selective visual attention and their neuronal correlates, e.g.: How fast can selective visual attention be transferred from one object to another? How is performance in inferring the object which was intended influenced by the spatial density or the number of objects being present?
While the first question is relevant for finding out how many decision steps can be made within a time interval, the second question is interesting for determining how many decisions (bits) can be made within one of those steps. For answering the latter one has to find out into how many different attendable parts the visual space can be segmented. Unfortunately, these questions can not be answered with the existing data and thus require new experiments addressing these other aspects of selective visual attention.

A main result of this thesis is that, in addition to being a well discriminable state, the condition of attention improves the performance of classifying the underlying shapes.This result motivated us to investigate the changes in the corresponding activity patterns induced by attention, which make the power coefficients from different shapes more distinct.

It was possible to quantify the contributions of two different enhancing effects. It is

known that attention can improve the signal-to-noise ratio (SNR) of spike count data from area V4 (through a multiplicative and stimulus-independent gain for orientation-selective activity) (McAdams and Maunsell, 1999b). A significant but small improvement through an increased SNR was also found for our data. But the major enhancement was found to be caused by an attention-dependent, intricate increase of differences between the responses to different stimuli. This mechanism makes the power coefficient vectors of different shapes more distinct by moving the corresponding data clouds in the data space of the classes more apart while mainly conserving the SNR.

The underlying neuronal mechanisms for this attention-dependent shape-selective modification are not known (a detailed discussion of putative mechanisms can be found in chapter 4). For studying the problem from a theoretical point of view, the idea arose that this question could be studied within a framework where neuronal networks with a global control parameter (representing the condition of attention) may produce a similar behaviour in their 'neuronal correlates'. The target was to reproduce the characteristics of the curves, describing the mean power coefficients in dependency of the frequencies, for the data measured from the electrodes over areas V4. In a first step, two types of models were constructed. Both systems were able to mimic these averaged power wavelet spectra. We sought for a simple connection between the parameters of the models with properties of the shapes. It was not possible to find such a simple linkage. However, even if we would have found a simple correlation between the properties of the shapes and the parameters of the model, it would not be for sure that this connection has some meaning or can tell us something about the underlying neuronal mechanisms.
As a future project, it will be interesting to take one step back and first try to answer this questions: Is it possible to construct a biological plausible neuronal network with a global control parameter (which e.g. switches the strength of the connections between two states) that maps its input more distinctly onto its output space when the control parameter is set to its value corresponding to the state of attention? What restrictions on the input and output space are compatible for obtaining an enhancement effect? What limitations on the improvement are imposed if the control mechanism is constrained to e.g. a multiplicative gain or an offset added to the weights?

One caveat of the measured epi-dural field potentials is that they are signals that lost a lot of information about more localized neuronal processes because of spatially averaging over the activities of large neuronal populations. Thus we can only identify that oscillatory activity in the γ-band above 40 Hz is important, but we can not exclude that even more information may be contained in asynchronous activity not captured by this recording technique.A better alternative would be to perform more localized intra-cortical recordings providing a higher spatial resolution. This would allow us to constrain the phenomenology to be reproduced by computational models in more detail, and thus increase the chance of identifying the underlying neuronal mechanisms in simulation studies.

As reported above, the classification using data from area V4 showed a significant

enhancement in performance under attention. The data from area V1 did not show such an effect. A ROC analysis revealed that the classification performance of discriminating one class out of two classes is already very high in the non-attended condition. These high classification rates leave no room to observe an attention-induced improvement, while for area V4 this enhancement is possible. New experiments revealed that if the two simultaneously presented shapes are moved more closely together (with still spatially separated representation in V1 but largely overlapping representations in V4), the activity in V1 synchronizes with the activity in area V4 (Smiyukha et al., 2006). How these inter-areal synchronization phenomena influence the discriminability of the underlying shapes has still to be investigated.

Stabilizing Decoding Against Non-Stationaries

Realizing that the own body lost possibilities of interacting with the environment would be a big problem for all of us, causing the understandable desire to circumvent these handicaps. Over the years, a large research industry for finding solutions for this problem has emerged, which allows to combine the search for a better understanding of the CNS with helping disabled persons to re-gain autonomy or quality of life.

Actually a huge number of different types of functional neuronal prostheses are under development. Prominent examples are retina implants and motor cortex prostheses. Today, only cochlea implants (for re-gaining a limited sense of hearing by replacing the input to the hearing nerve through artificially produced electrical impulses) and pacemaker for the brain (for treating e.g. Parkinson disease and dystonia by electrical stimulations) have success in medical applications. All other systems are not ready yet.

From all the available approaches of creating an interface for interchanging information between the CNS and external devices, I focused my research on the aspect of 'reading thoughts'. Here the goal is to identify reoccurring patterns in the correlates of neuronal activities that can be controlled voluntarily by the user of the system. Using this information allows to control e.g. computers. The applied information extraction procedures are similar to those we used for analysing the effects of attention (chapter 4) or for the search for optimal coding strategies, given a set of constraints (Bethge et al., 2003b; Bethge et al., 2003a; Bethge et al., 2002a; Bethge et al., 2002c; Bethge et al., 2002b; Bethge et al., 2001).

Most researchers are working on the question how to increase the bandwidth of data acquisition or how to extract as much information content as possible from the recorded data. I deviated from these goals and tried to understand how it would be feasible to protect the information extraction performance as long as possible against degeneration by non-stationaries like e.g. moving electrodes or changes in the regarding neuronal information processing networks. This question will increase its importance when these Brain Computer Interface technologies, which are currently under development, are

transferred to long-term medical applications (Pawelzik and Rotermund, 2005).

My solution of counteracting these degenerative processes is based on an extra error signal extracted from the neuronal activities of the CNS, which describes the perceived actual performance of the prosthetic device (Rotermund et al., 2006a; Pawelzik et al., 2006c). Using this performance measure, it is possible to apply a reinforcement type of on-line adaptation strategy that allows to compensate different types of non-stationaries. The error signal is used for guiding a stochastic search in the parameter space of the estimation algorithm towards an optimal set of parameters (the state with the minimal error). This adaptation procedure has always to keep track of the ongoing changes induced by the non-stationaries applied to the 'real' parameters, used for coding the information about the intended action into the corresponding correlates of neuronal activities.

For testing the idea, simulations were made where a population of motor neurons controlled a robotic arm in two dimensions. A second population of neurons represented the error between the intended movements of the robotic arm and the executed movements. The envisioned on-line adaptation schema was applied to this setting. The presented simulations showed that it is possible to reverse the degeneration of performance caused by a complete reset of all parameters (like we can expect through a movement of a whole electrode array) or random changes in a subset of parameters (if the changes are not occurring to frequent).

Despite the clear evidence from the simulations that the idea works, transferring it to a real experimental setup is absolutely not trivial. The major problem lies in the lack of a suitable error signal source. As discussed in chapter 5 in great detail, for arm movements some potential areas in the brain are known (Diedrichsen et al., 2005). But even for these regions, the correlates of error values in the neuronal activity are not thoroughly researched yet. Thus, finding suitable areas in the brain has to be the next step. The necessary properties of the signal are weak: It has to depend monotonically on the perceived mismatch between the intended action and the executed action. It should be as independent as possible from other influences. Even non-stationaries on the error signal are allowed if their effects are slower than the non-stationaries degenerating the performance of the estimator. However, even after a suitable source for such an error signal has been found, recording the neuronal activity by multi-electrode recordings remains as a problem, which has still to be solved.
Depending on new knowledge about the coding of intended actions and their corresponding error signals, the estimator and the on-line adaptation algorithm can be optimised. For this development it is always helpful to remember two aspects: It is important to keep the system as low dimensional as possible because of the 'curse of dimensionality', and the on-line adaptation schema should use as less information as possible for an update step, otherwise the reaction-time to changes due to non-stationaries would be increased. In addition it may be helpful to include the adaptation processes of the CNS into the on-line adaptation schema.
Thus, it is important to gather through experiments more information before a next

step on the algorithmic side toward a real medical application can be done.

Taken together, this thesis presented three new contributions:

- A theoretical method of processing information spike by spike in a fast and efficient fashion. This study also showed that it is sufficient to use neurons, generating Poissonian spike trains, for performing fast and efficient information processing.
- A new mechanism, produced through selective visual attention, was revealed that renders information about different visual stimuli, represented in γ-band oscillatory activity of neuronal populations, more distinct for an external observer and probably for the brain itself. It also showed that internal states of the brain can alter the neuronal activity pattern in a complex manner and it demonstrated that the power of the γ-band contains significant information about visually perceived shapes.
- A method for neuronal-prostheses capable of protecting estimators of intended actions (like arm movements) against non-stationaries, for the cost of an extra error signal describing the mismatch between the intended and executed action.

Appendix A

Additional Background

A.1 Modeling of neurons

A.1.1 Hodgkin and Huxley model

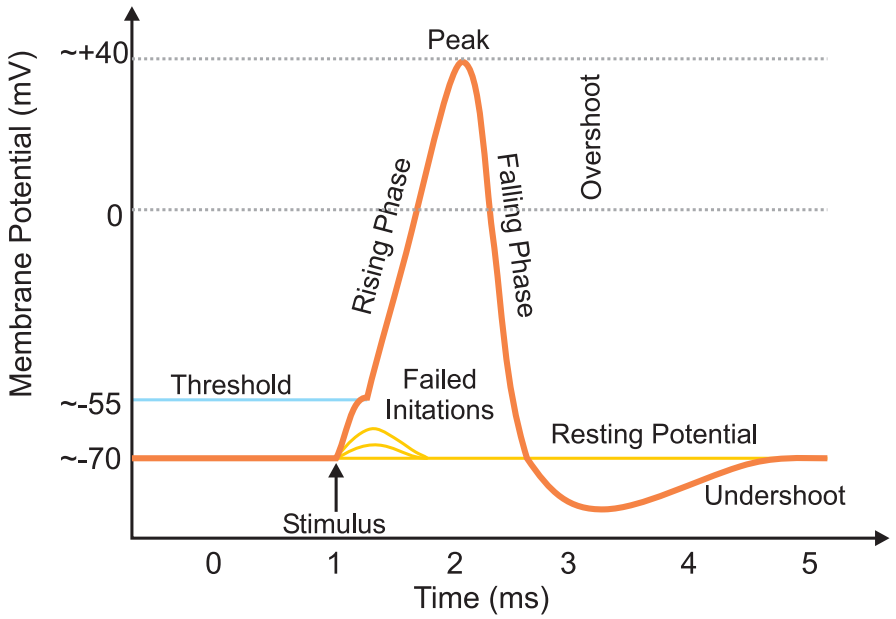


Figure A.1: Schematic action potential. If the membrane potential passes the threshold, the creation of an action potential is initiated. (Figure was adapted from wikipedia.org)

Action potentials that are measured in real experiments show relative complex functional characteristics (see Fig. 2.1 or the schematic illustration in Fig. A.1). The

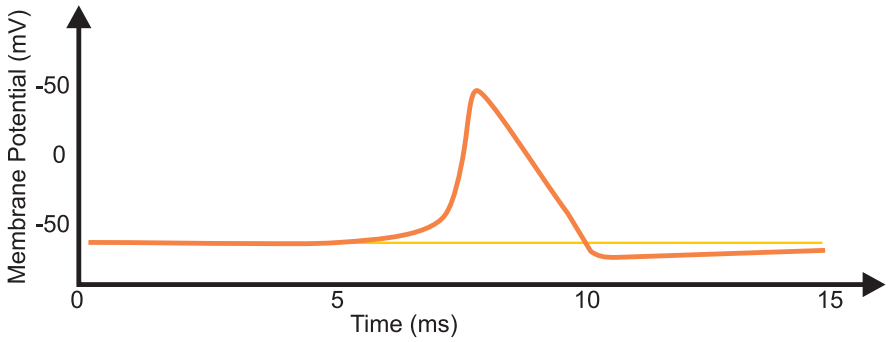


Figure A.2: Illustration of an exemplary solution of the Hodgkin and Huxley model.(Figure was adapted from (Dayan and Abbott, 2001))

voltage-clamp experiments on the squid giant axon (e.g. (Cole and Curtis, 1939)) inspired Hodgkin and Huxley (Hodgkin and Huxley, 1952) to develop a model that can mimic the experimental data. An action potential is the result of an exchange of ions between the inside and the outside of the nerve cell. The driving force for these exchanges is a difference between the concentrations of ions outside and inside of the cell. For one type of ions, the potential can be quantified by the Nernst equation

$$E = \frac{k_b T}{z_{\text{ion-type}} q} \log\left(\frac{\text{concentration}_{(\text{outside,ion-type})}}{\text{concentration}_{(\text{inside,ion-type})}}\right) \tag{A.1}$$

where T is the temperature in Kelvin, k_B the Boltzmann constant and $z_{\text{ion-type}} q$ the charge of the ion. Ion-pumps in the membrane are creating these differences in ion concentrations. During generating an action potential selective ion channels regulate the exchange through the membrane.

The Hodgkin Huxley model is based on a differential equation for the membrane potential V, which accounts different ionic currents

$$C \frac{dV}{dt} = I_{ext} - I_K - I_{Na} - I_{leak}\,. \tag{A.2}$$

I_{ext} represents an external electric current, e.g. an electrical current injected into the neuron. I_K and I_{Na} quantify the in- and outflow of K- and Na- ions into or out of the cell through ion channels. Furthermore, I_{leak} describes a leak of charge carriers and C denotes the capacitance of the membrane. In this model, the membrane potential is only described by a single variable V. These models are termed single compartment models.

For generating the temporal dynamics of an action potential, the flow of sodium and potassium ions as well as the leak current are modeled by the following equations

$$\begin{aligned} I_K &= g_k(V,t) \cdot (V - E_K) \\ I_{Na} &= g_{Na}(V,t) \cdot (V - E_{Na}) \\ I_{leak} &= g_{leak} \cdot (V - E_{leak})\,. \end{aligned}$$

E_K, E_{Na} and E_{leak} are called equilibrium potentials. $g_k(V,t)$ and $g_{Na}(V,t)$ represent the dynamics of the conductance of the ion channels, while the conductance g_{leak} is constant. Using conductances for modeling the ionic flow, shows that the Hodgkin-Huxley model is a conductance-based model. The dynamics of the conductances of the sodium and potassium channels are simplified and approximated by

$$\begin{aligned} g_k(V,t) &= \bar{g}_K \cdot n^4 \\ g_{Na}(V,t) &= \bar{g}_{Na} \cdot m^3 \cdot h\,. \end{aligned}$$

$\bar{g}_K$ and $\bar{g}_{Na}$ are constants and the following differential equations describe the temporal evolution of n,m and h

$$\begin{aligned} \frac{dn}{dt} &= \alpha_n(V) \cdot (1-n) - \beta_n(V) \cdot n \\ \frac{dm}{dt} &= \alpha_m(V) \cdot (1-m) - \beta_m(V) \cdot m \\ \frac{dh}{dt} &= \alpha_h(V) \cdot (1-h) - \beta_h(V) \cdot h \end{aligned}$$

where $\alpha(V)$ and $\beta(V)$ are exponential functions (Dayan and Abbott, 2001; Johnston and Wu, 1997). One trace of the Hodgkin-Huxley model is shown in Fig. A.2.

A.1.2 McCulloch and Pitts neurons

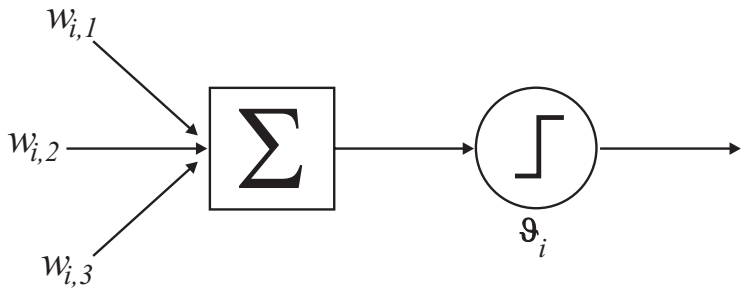


Figure A.3: Schematics of a McCulloch and Pitts neuron. The outputs from other neurons j are multiplied by weights $w_{i,j}$ and then summed. This value is then compared with a threshold ϑ_i. The output of the neuro i is 1 if the threshold is transgressed and 0 otherwise.

Models like e.g. the Hodgkin and Huxley model or the Connor Stevens model (Connor and Stevens, 1971) focus on explaining the shape of action potentials and to mimic the dynamics of a 'real' neuronal network. But it is also interesting to use only the information about the timing of action potentials for computations. One of the simplest neuron models of this type is the McCulloch and Pitts model (McCulloch and Pitts, 1943). It calculates a binary response r from a weighted sum of inputs from other neurons. In every timestep, this 'artificial' neuron sums the weighted inputs $r_j w_{i,j}$ and

compares the result to a threshold. If the total input is above threshold, the neuronal unit delivers 1 (firing), otherwise it delivers 0 (not firing). This behavior is described by

$$r_i(t+1) = \Theta\left(\sum_j w_{i,j} r_j(t) - \vartheta_i\right)$$

with ϑ_i as threshold for neuron i and

$$\Theta = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.3}$$

being the Heavyside function Θ introducing a non-linearity. The weights are allowed to have positive or negative values. If $w_{i,j}$ is zero, then neuron j has no connection to neuron i (Hertz et al., 1991).

A.2 Propability based estimators

A.2.1 Minimum mean squared error estimator

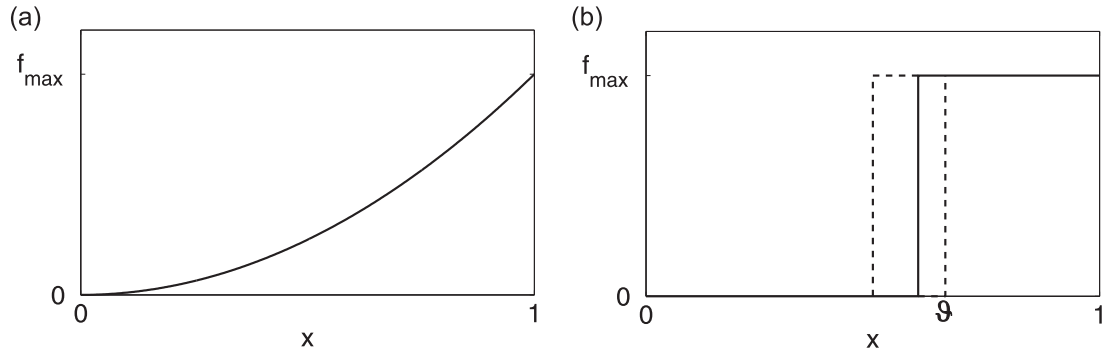


Figure A.4: (a) In the asymptotic limit ($f_{max} \cdot T \to \infty$), it can be shown that the parabolic tuning function $f^{asymp}(x)$ is optimal (here shown with $f_{min} = 0$). For small $f_{max} \cdot T$ it can be advantageous to use tuning functions like displayed in (b). Depending on $f_{max} \cdot T$ (and for $f_{min} = 0$), the optimal threshold ϑ lies between $\frac{2}{3}$ and $\frac{1}{2}$.

In the following example (Bethge et al., 2003a), it was assumed that one neuron has to encode as much information as possible about a scalar into its rate. Furthermore, the rate has to be transmitted by a channel with Poisson noise Eq.(2.1) and the information has to be decoded by a MMSE estimator.

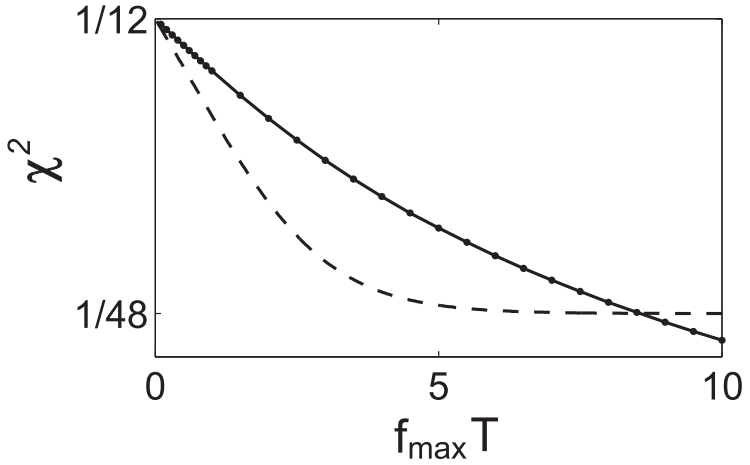


Figure A.5: Comparison of the minimum mean squared error for the the step function (dashed line) and the parabolic tuning function (solid line). The χ^2- axis has a logarithmic scale.

If it is allowed to use infinite long time intervals ($T \to \infty$) for transmitting the information, then the optimal tuning function for such a neuron can be determined in the

asymptotic limit by the Fisher information (for a discussion see (Bethge et al., 2002c)). As result we obtain an optimal tuning function with parabolic shape (see Fig. A.4a)

$$f^{asymp}(x) \;=\; \left((\sqrt{f_{max}}-\sqrt{f_{min}})x+\sqrt{f_{min}}\right)^2 , \tag{A.4}$$

where f_{min} represents the minimum firing rate (also called baseline) and f_{max} denotes the maximum firing rate of the neuron. The MMSE of the asymptotically optimal tuning function is given by

$$\chi^2[f^{asymp}] = \frac{1}{3} - \frac{1}{2\,(\sqrt{f_{max}\,T})^3} \sum_{k=0}^{\infty} \frac{1}{k!}\, \frac{\Gamma^2_{0,f_{max}\,T}(k+1)}{\Gamma_{0,f_{max}\,T}(k+\frac{1}{2})} \quad . \tag{A.5}$$

In the case where $f_{max}T$ is finite, it is not guaranteed that the tuning curve with the parabolic shape is still optimal. This is especially true for the case $f_{max}T \to 0$, where the Poisson distribution converges to a Bernoulli distribution

$$p(k \mid fT) \;=\; (fT)^k(1-fT)^{1-k} \quad \text{for } k \in \{0,1\}\,.$$

For this case, it can be shown that the optimal tuning function can be described by

$$f^{binary}(x) = f_{min} + (f_{max}-f_{min})\,\Theta\left(x-\vartheta_{f_{min}}(f_{max}T)\right) , \tag{A.6}$$

where $\Theta(z)$ is the Heaviside function (Fig. A.4, right)

$$\Theta(z) \;=\; \begin{cases} 1 & \text{if } z>0 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{A.7}$$

For $f_{min}=0$, it can analytically be calculated that the optimal threshold $\vartheta_{f_{min}}(f_{max}\,T) \in [\frac{1}{2},\frac{2}{3}]$ is a function of $f_{max}\,T$ given by

$$\vartheta_0(f_{max}\,T) = 1 - \frac{3-\sqrt{8e^{-f_{max}\,T}+1}}{4(1-e^{-f_{max}\,T})} \tag{A.8}$$

Furthermore, we can calculate the corresponding MMSE (for details see (Bethge et al., 2003a)):

$$\chi^2[f^{binary}] = \frac{1}{12}\left(1 - \frac{3\,\vartheta_0^2(f_{max}\,T)}{[(1-\vartheta_0(f_{max}\,T))(1-e^{-f_{max}\,T})]^{-1}-1}\right) . \tag{A.9}$$

Now we know the optimal tuning functions for very small and infinite large $f_{max}\,T$, (see Fig. A.5 for a comparison between both tuning functions) but the question remains what type of tuning functions are optimal between the two extreme cases and in which range of $f_{max}\,T$ we can use the solutions for the extrem cases.

One idea for tackling this question is to construct a new tuning function composed out of both tuning functions

$$f^{ramp}_{\alpha,\beta}(x) = \begin{cases} f_{min} & , \quad 0<x<\alpha \\ ((\sqrt{f_{max}}-\sqrt{f_{min}})\frac{x-\alpha}{\beta-\alpha}+\sqrt{f_{min}})^2 & , \quad \alpha<x<\beta \\ f_{max} & , \quad \beta<x<1 \end{cases} . \tag{A.10}$$

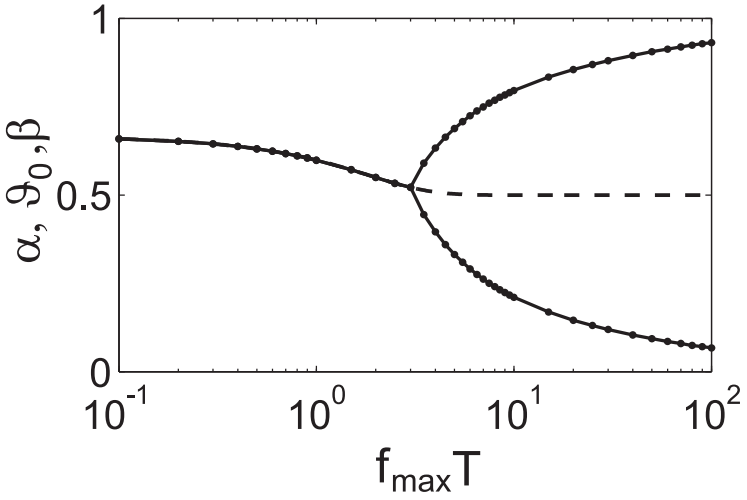


Figure A.6: Optimal parameters for the step tuning function (dashed line) and the $f^{ramp}_{\alpha,\beta}$ tuning function in dependency of $f_{max}T$.

Eq.(A.10) includes, with $\alpha = \beta$, the step function and turns into the optimal parabolic shape for the asymptotic limit, with $\alpha = 0$ and $\beta = 1$. Using extensive numerical simulations, it was possible to calculate optimal α and β values for the $f^{ramp}_{\alpha,\beta}$ tuning function. The results are shown, together with the optimal threshold for the step function, in Fig. A.6. In the region of $f_{max}T \approx 3$ the f^{ramp} function starts to differ from the step tuning function and exceeds the performance of the step function. For the family of f^{ramp} where $\alpha = \vartheta_0 - w$ and $\beta = \vartheta_0 + w$, we were able to show analytically the existence of a phase transition (Bethge et al., 2003b).

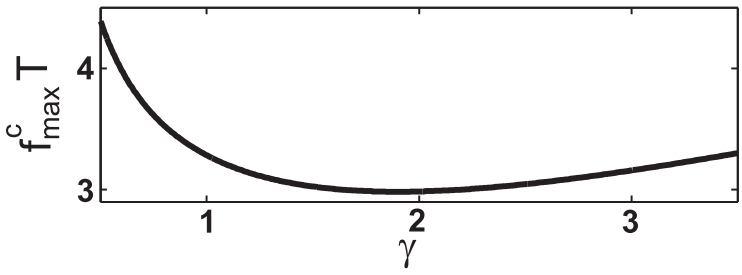


Figure A.7: The critical $f^c_{max}T$, where the phase transition occurs, is shown as a function of γ for the $f^{ramp,\gamma}_{\alpha,\beta}$ tuning function.

Another interesting result in this context is, that we were able to analyse numerically an extended version of this restrained f^{ramp} tuning function

$$f^{ramp,\gamma}_{\alpha,\beta}(x) = \begin{cases} f_{min} & , \quad 0 < x < \alpha \\ ((\sqrt{f_{max}} - \sqrt{f_{min}})\frac{x-\alpha}{\beta-\alpha} + \sqrt{f_{min}})^{\gamma} & , \quad \alpha < x < \beta \\ f_{max} & , \quad \beta < x < 1 \end{cases} . \tag{A.11}$$

In Fig. A.7, the dependency of the $f^c_{max}T$, where the phase transition occurs, from γ is shown. The minimum of $f^c_{max}T \approx 3$ can be found for $\gamma = 1.9$.

An analysis of these and some other tuning functions as well as a detail discussion, including the phase transition, can be found in (Bethge et al., 2003a; Bethge et al., 2003b; Bethge et al., 2002a).

A.2.2 Linear minimum mean squared error estimator

In chapter 5 the linear minimum mean squared error estimator is a major part of the presented simulations. In that simulations, regarding the possibilities of stabilising a neuro-prosthetic device against non-stationaries, we will assume that neurons fire independently from each other, and that their spikes are drawn from Poissonian distributions with mean values $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$. With these conditions, we can write $p(\mathbf{k} \mid \mathbf{x}, \mathbf{P})$ as $p(\mathbf{k} \mid T\Lambda(\mathbf{x}, \mathbf{P}))$ which due to the independence condition factorises into

$$p(\mathbf{k} \mid T\Lambda(\mathbf{x}, \mathbf{P})) = \prod_{i=1}^{N} \frac{1}{k_i!} \left(T\lambda_i(\mathbf{x}, \mathbf{P})\right)^{k_i} \exp\left(-T\lambda_i(\mathbf{x}, \mathbf{P})\right) . \tag{A.12}$$

Using Eq.(A.12) together with the first and second momentum of the Poisson distribution

$$\begin{aligned} \sum_{\mathbf{k}} p(\mathbf{k} \mid \mathbf{x}, \mathbf{P}) k_i &= T\lambda_i(\mathbf{x}) \\ \sum_{\mathbf{k}} p(\mathbf{k} \mid \mathbf{x}, \mathbf{P}) k_i^2 &= T\lambda_i(\mathbf{x}) + T^2\lambda_i(\mathbf{x})^2 , \end{aligned}$$

the expressions for g_i and $Q_{i,j}$ simplify to

$$g_i(\mathbf{x}) = T\lambda_i(\mathbf{x}) \tag{A.13}$$

$$Q_{i,j} = \int_{\mathbf{x}} \rho(\mathbf{x}) \left(T^2\lambda_i(\mathbf{x})\lambda_j(\mathbf{x}) + T\lambda_i(\mathbf{x})\delta_{i,j}\right) . \tag{A.14}$$

In the study about the neuro-prosthetic controlling system, we will select $\lambda_i(\mathbf{P})$ as tuning function for neurons from the motor cortex. These model neurons will encode in their neuronal response information about velocities for movements. To be more precise, each of the simulated velocity-coding neurons will be modeled by a function with cosine-shaped tuning for the direction φ (Georgopoulos et al., 1982; Fu et al., 1997), and with a linear tuning for the absolute velocity v,

$$\lambda_i(\mathbf{P}) = f_i(\boldsymbol{v}) = f_i^{\text{off}} + \frac{f_i^{\text{mod}}}{2} \left(1 + \frac{v}{v_{\max}} \cos\left(\varphi - \varphi_i\right)\right) . \tag{A.15}$$

In this special case, the parameter vector $\mathbf{P}$ comprises the maximum mean firing rates above threshold f_i^{mod}, the offset (spontaneous) firing rates f_i^{off}, and the directions with

the highest response φ_i. After a long analytical calculation (see C.1 for details) it can be shown that the OLE for the velocity is defined by

$$\hat{v}(\mathbf{k}) = \sum_i^N \left(k_i - T \left(\frac{f_i^{\text{mod}}}{2} + f_i^{\text{off}} \right) \right) \mathbf{D}_i \tag{A.16}$$

with

$$\begin{aligned} \mathbf{D}_j &= \sum_i^N \mathbf{L}_i \left(Q_{ij} - M_j M_i \right)^{-1} \\ &= \sum_i^N f_i^{\text{mod}} v_{\max} \{ \cos(\varphi_i), \sin(\varphi_i) \} \\ &\times \left[\left\{ \frac{T}{2} f_i^{\text{mod}} f_j^{\text{mod}} \cos\left(\varphi_i - \varphi_j \right) + 8 \delta_{i,j} \left(\frac{f_i^{\text{mod}}}{2} + f_i^{\text{off}} \right) \right\}^{-1} \right]_{i,j} . \end{aligned} \tag{A.17}$$

A.3 Recurrent networks

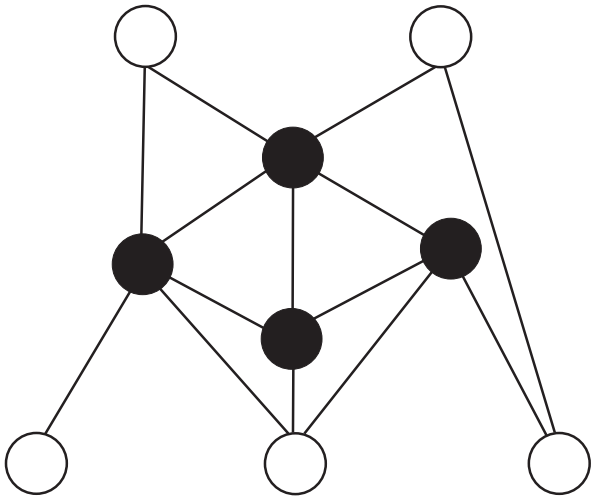


Figure A.8: Example of a recurrent network. The shaded cycles represent hidden units. The open cycle can be output neurons (e.g. the upper units) and input neurons (e.g. the lower units). In general, it is not necessary that a recurrent network contains all three types of neurons (input, output and hidden).

In comparison with feedward networks, recurrent networks have (Hertz et al., 1991; Pearlmutter, 1990) a loop structure. For this class of networks, it is allowed that neurons can have connections to all other neurons in the network, even a connection to themselves. Fig. A.8 shows an example of a recurrent networks.

The number of different types of recurrent network models and training methods is large. It is beyond the scope of this thesis to give a representative overview over this kind of models. For this reason I will only introduce Hopfield networks as an example for implementing associative memory through an attractor dynamics, Boltzmann machines and liquid state machines. The liquid state machine is an example for recurrent networks capable of processing time series as input.

A.3.1 Hopfield networks

The Hopfield network (Hopfield, 1982; Hertz et al., 1991) can be seen as a member of the family of recurrent networks. Hopfield networks are examples for associative memory systems and provide an implementation of the idea of storing information in dynamical attractors. Given an incomplete input pattern from a set of trained patterns, the network may complete the input pattern using the stored information. This information about learned patterns is stored in the connection weights and can be retrieved by the dynamics of updating neuronal activities. The neurons are McCulloch-

Pitts units (see section A.1.2). The input- output relation is given by

$$S_i(t+1) = \Theta\left(\sum_j w_{i,j} S_j(t) - \vartheta_i\right) \tag{A.18}$$

where $\Theta(.)$ is the Heavyside function (Eq.(A.3)) with threshold ϑ_i for unit i and $w_{i,j}$ are the weights between neuron i and j. The Heavyside function is sometimes replaced by the $sgn(.)$ function, which results in the output states $\{-1,+1\}$ instead of $\{0,+1\}$. An introduction, which I use as a source for this discussion, can be found in (Hertz et al., 1991).

The update of the units in a Hopfield network can be done in a synchronous or asynchronous fashion. With the synchronous update strategy, all units are updated in parallel at the same timestep. In the asynchronous update, one single unit is selected randomly and updated. Alternatively, an update step can be performed on a random subset from the neurons. The asynchronous update mode is said to be more biologically realistic because the other type needs a kind of global time frame for coordinating the update.

For symmetrical sets of weight matrices $w_{i,j} = w_{i,j}$ it is possible to define a cost function (Lyapunov function). The retrievable patterns can be understood as local minima of the cost function, which is given by

$$E = -\frac{1}{2}\sum_{i,j} w_{i,j} S_i S_j \,. \tag{A.19}$$

Setting a subset of units to initial values identical to one trained pattern and allowing the system to evolve according to its dynamics, it may settle in one of those local minima (attractors). Depending on the properties of the set of memorized patterns, similarity of the patterns, and storage capacity of the Hopfield network, the results can differ from the trained patterns because the emergence of additional unwanted local minima.

Assuming that the energy of the system is minimized by the configuration where the overlap between the training patterns and the actual state of the neurons is maximized, it is possible to derive a (Hebb-like (Hebb, 1949)) learning rule for the weights:

$$w_{i,j} = \frac{1}{N}\sum_{\mu} O_i^{\mu} O_j^{\mu} \,,$$

with N as the number of neurons in the network and O_i^{μ} as the input values of pattern μ for neuron i. Often the weights $w_{i,i}$ are set to zero because this can help to reduce spurious states which do not correspond to any of the learned patterns.

We can extend the Hopfield model by replacing the deterministic states of the units r_i by stochastic units. Now the actual states can be sampled using the probabilities

$$p_{A,i} \;=\; \frac{1}{1+e^{-2\beta(\sum_j w_{i,j} S_j - \vartheta_i)}} \tag{A.20}$$

for state A, e.g. 0 and

$$p_{B,i} = \frac{1}{1 + e^{2\beta(\sum_j w_{i,j} s_j - \vartheta_i)}} \tag{A.21}$$

for state B, e.g. 1.

β is called (inverse pseudo-) temperature, which gives a hint to the similarity of the Ising model used in statistical mechanics to this type of Hopfield model. This link is one reason for the popularity of Hopfield models because it allows to interchange computational tools between two scientific disciplines. In the following, we will continue with Boltzmann machines, which are related to Hopfield networks.

A.3.2 Boltzmann machines

In a Hopfield network only the properties $< r_i >$ and $< r_i \cdot r_j >$ can be chosen freely ($< . >$ denotes the expectation value) Higher order correlations can not be trained independently. Boltzmann machines are an extension of stochastic Hopfield networks which solve this problem. In a Hopfield network all neurons can be observed. In Boltzmann machine networks some neurons may be hidden and some neurons may be dedicated to input and output tasks (Hertz et al., 1991).

The input-output relation for the units Eq.(A.18), the cost function Eq.(A.19) and the state probabilities Eqs.(A.20) and (A.21) are, for Hopfield networks and Boltzmann machines, the same. The difference lies in learning because of visible and hidden units in Boltzmann machines. Learning can be done e.g. by minimizing the Kullback-Leibler divergence with gradient descent method. The Kullback-Leibler divergence is calculated between the intended and actual, over the hidden units marginalized, Boltzmann distributions. We start with Eq.(A.19) and split the visible and hidden components of the network:

$$H(\boldsymbol{\nu}, \mathbf{h}) = -\left(\frac{1}{2}\boldsymbol{\nu}^T \mathbf{V}\boldsymbol{\nu} + \frac{1}{2}\mathbf{h}^T \mathbf{W}\mathbf{h} + \boldsymbol{\nu}^T \mathbf{J}\mathbf{h}\right) .$$

$\boldsymbol{\nu}$ and $\mathbf{h}$ denote the visible and hidden units in vectorial notation while $\boldsymbol{\nu}^T$ and $\mathbf{h}^T$ are their transposed vectors. $\mathbf{V}$ are weights between the visible nodes, $\mathbf{W}$ between the hidden units and $\mathbf{J}$ between the hidden and visible neurons. The probabilistic parameters of the system are described by the Boltzmann distribution

$$p(\boldsymbol{\nu}, \mathbf{h}) = \frac{1}{Z} e^{-\beta H(\boldsymbol{\nu}, \mathbf{h})} .$$

The Kullback-Leibler divergence (see section 2.2.2) is given by

$$D_{KL}(p_0(\boldsymbol{\nu}) \parallel p_{EQ}(\boldsymbol{\nu})) = \sum_{\boldsymbol{\nu}} \left(\sum_{\mathbf{h}} p(\mathbf{h} \mid \boldsymbol{\nu}) \tilde{p}_0(\boldsymbol{\nu}) \right) \log \left(\frac{\sum_{\mathbf{h}} p(\mathbf{h} \mid \boldsymbol{\nu}) \tilde{p}_0(\boldsymbol{\nu})}{\sum_{\mathbf{h}} p_{EQ}(\mathbf{h}, \boldsymbol{\nu})} \right) .$$

$p_0(\boldsymbol{\nu}, \mathbf{h}) = p(\mathbf{h} \mid \boldsymbol{\nu})\tilde{p}_0(\boldsymbol{\nu})$ is the data distribution, while $\tilde{p}_0(\boldsymbol{\nu})$ is the empirical data distribution which can be observed at the visible units. $p_{EQ}(\mathbf{h}, \boldsymbol{\nu})$ is the equilibrium distribution of all units after iterating the system for a long time. As result of the gradient descent method we obtain the following update rules

$$\begin{aligned} \Delta \boldsymbol{W} &= <\mathbf{h}\mathbf{h}^T>_0 - <\mathbf{h}\mathbf{h}^T>_{EQ} \\ \Delta \boldsymbol{V} &= <\boldsymbol{\nu}\boldsymbol{\nu}^T>_0 - <\boldsymbol{\nu}\boldsymbol{\nu}^T>_{EQ} \\ \Delta \boldsymbol{J} &= <\boldsymbol{\nu}\mathbf{h}^T>_0 - <\boldsymbol{\nu}\mathbf{h}^T>_{EQ} \ . \end{aligned} \tag{A.22}$$

$<\ .\ >_0$ is calculated with setting the visible units to the intended input and output values and $<\ .\ >_{EQ}$ is evaluated while all units can evolve freely. For the averaging process the network needs to be in equilibrium. Taken together, this learning rule is slow and the numerical evaluation of the average correlations can be inaccurate even for small β because of strong fluctuations. Other learning methods have been developed to compensate these problems, e.g. mean field methods, recurrent back-propagation, contrastive divergence learning and reinforcement learning (see section 2.4.4) (Welling and Hinton, 2002; Hertz et al., 1991).

A.3.3 Liquid state machine

Another concept of analysing of time series is the liquid state machine (Maass et al., 2002; Natschläger et al., 2002). The idea behind this framework is based on a medium for storing the received input as perturbations or echos (like rain falling on a quiet sea causing ripples on the water) that decay over time but on different time scales. The desired information can be read out by networks similar to perceptrons observing the states of the medium.

More formally, the input is denoted by $\mathbf{u}(t)$. For a liquid state machine M the internal states are denoted by $\mathbf{x}^M(t)$ and based on $u(s)$ with $s \leq t$. The internal state is calculated by

$$\mathbf{x}^M(t) = (L^M u)(t)\,,$$

where L^M is the so called 'liquid filter', which maps the series of inputs onto internal states. For extracting the desired information $y(t)$ from $\mathbf{x}^M(t)$, memory free readout functions $f^M(.)$ can be used

$$y(t) = f^M(\mathbf{x}^M(t))\,.$$

These read out maps are typically task-specific and for extracting different properties of the signal, different maps are applied to the same liquid state. A liquid state machine can be built e.g. out of randomly and recurrently connected integrate-and-fire neurons with non-linear synapses. It was suspected that liquid state machines can have 'universal power of computations with fading memory on functions of time' (Maass et al., 2002).

A.4 Generative models

Generative models are typically used for modeling probability distributions of observable data. These generative models may contain hidden parameters for describing observed data. Examples for such models are hidden Markov models and Helmholtz machines, which we will discuss in some more detail.

A.4.1 Hidden Markov model

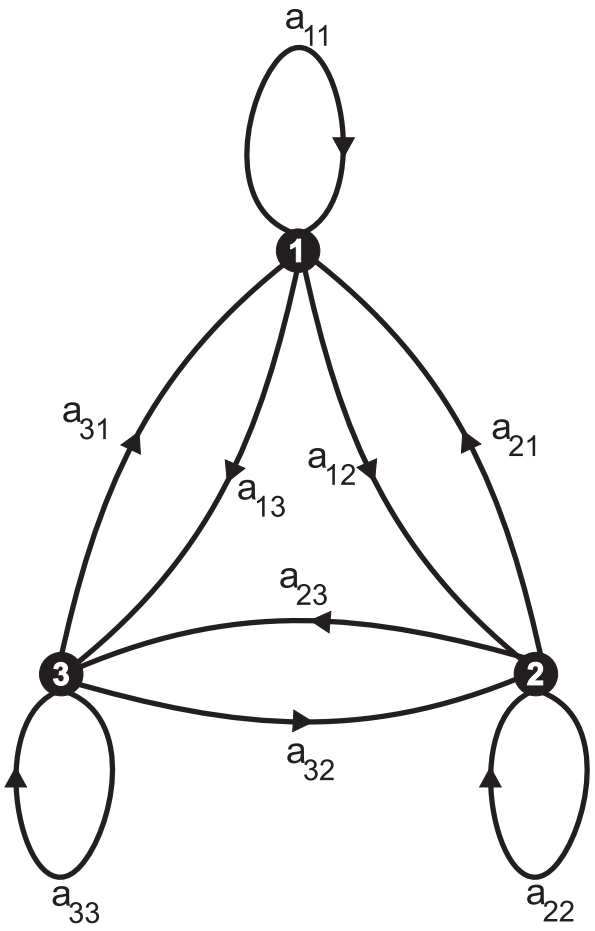


Figure A.9: Illustration of a Markov model with three states. The transition probabilities between the states are denoted by the $a_{i,j}$'s.

Hidden Markov models (HMM) are very popular and often used in technical applications. A good review of these models can be found in (Rabiner, 1989), which I will use as basis for this overview. HMMs are based on Markov models (Fig. A.9). The probability $a_{i,j}$ for choosing the next state S_j depends only on the actual state S_i,

$$a_{i,j} = p(q_{t+1} = S_i \mid q_t = S_i) \,.$$

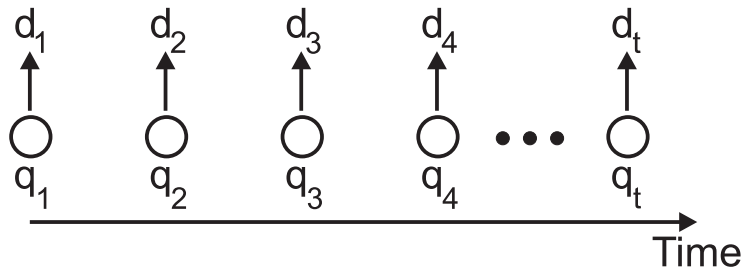


Figure A.10: A hidden Markov process, using the transition probabilities $a_{i,j}$, generates a sequence of hidden states q_t, from which observations d_t are drawn from the conditional probabilities b.

$S = \{S_1, \cdots, S_N\}$ represents the available states of the model, while q_t denotes the realisation at time t. In HMMs the actual state q_t is hidden. The state changes randomly in each time step according the transition probabilities a. In addition, an observable symbol d_t is drawn for each time step out of a probability distribution b. Fig. A.10 illustrates the whole process. The probability distribution for drawing the observable symbols is defined by

$$b_i(k) = p(d_t = v_k \mid q_t = S_i)\,.$$

Possible observable symbols are $V = \{v_1, \cdots, v_M\}$. Not all symbols may be available at each state. The realisation of the observable process at time t is represented by d_t. By introducing the initial state distribution π_i, the HMM is now fully specified,

$$\pi_i = p(q_1 = S_i)\,.$$

All information about the model can be summarised by

$$\lambda = (\{a_{i,j}\}, \{b_i(k)\}, \{\pi_i\})\,.$$

Typically three different types of computational problems are considered in the context of HMMs:

- to evaluate the probability that a sequence of observations was generated by a hidden Markov model
- to find the optimal sequence of hidden states that explains a sequence of observations
- to learn a hidden Markov model from data

For solving these problems, it is helpful to reduce the computational costs by introducing the so called forward and backward variables. These variables show structural

similarities to the forward and backward-propagated components used for training feed-forward networks by using the backpropagation-learning rule. The forward variables are defined by

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) a_{i,j} \right) b_j(d_{t+1})$$

for $1 \leq t \leq T-1$, starting with

$$\alpha_1(j) = \pi_j b_j(d_1)\,.$$

This forward variable propagates forward in time, in contrast to the backward variable β which is initialized at the end of the sequence, $\beta_T(i) = 1$, and then evaluated backwardly via

$$\beta_t(i) = \sum_{i=1}^{N} a_{i,j}\, b_j(d_{t+1})\, \beta_{t+1}(j)$$

for $1 \leq t \leq T-1$.

The first of the three computational problems is to evaluate the probability that a sequence of observations was generated by a hidden Markov model with the parameters λ. For this we have to calculate

$$p(\{d_1, \cdots, d_T\} \mid \lambda) \;=\; \sum_{q_1, \cdots, q_T} p(\{d_1, \cdots, d_T\} \mid \{q_1, \cdots, q_T\}, \lambda) \cdot p(\{q_1, \cdots, q_T\} \mid \lambda)$$

with

$$\begin{aligned} p(\{d_1, \cdots, d_T\} \mid \{q_1, \cdots, q_T\}, \lambda) &= \prod_{t=1}^{T} p(d_t \mid q_t, \lambda) \\ &= b_{q_1}(d_1) \cdot b_{q_2}(d_2) \cdots b_{q_T}(d_T) \end{aligned}$$

and

$$p(\{q_1, \cdots, q_T\} \mid \lambda) = \pi_{q_1} \cdot a_{q_1, q_2} \cdots a_{q_{(T-1)}, q_T}\,.$$

Using the forward variables, the equation can be written as

$$p(\{d_1, \cdots, d_T\} \mid \lambda) \;=\; \sum_{i=1}^{N} \alpha_T(i)\,.$$

The second computational problem is to find the optimal sequence of hidden states that explains a sequence of observations, given a model λ. As usual, 'optimality' depends on the applied criteria. One possible choice is to always select the most probable state $\hat{q}_t$ in each time step. This can be calculated by using

$$\gamma_t(i) = p(q_t = S_i \mid \{d_1, \cdots, d_T\}, \lambda)\,,$$

being expressed by the forward and backward variables as

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \cdot \beta_t(j)} .$$

The individually most likely states can be determined by

$$\hat{q}_t = \underset{i}{\operatorname{argmax}}\, \gamma_t(i) .$$

Another criteria, that can be used in this context, is the most likely path through the states $p(\{q_1, \cdots, q_T\} \mid \{d_1, \cdots, d_T\}, \lambda)$. This can be realised by the Viterbi-algorithm. This algorithm uses auxiliary variables $\delta_t(j)$ and $\Psi_t(j)$, defined by

$$\delta_t(j) = \max \left(\left\{ \delta_{(t-1)}(i)\, a_{i,j} \right\}_{i=1,\cdots,N} \right) b_j(d_t) \tag{A.23}$$

and

$$\Psi_t(j) = \underset{i}{\operatorname{argmax}} \left(\left\{ \delta_{(t-1)}(i)\, a_{i,j} \right\}_{i=1,\cdots,N} \right) . \tag{A.24}$$

Eq.(A.23) and Eq.(A.24) are used for $2 \leq t \leq T$. The initial values are given by

$$\delta_1(j) = \pi_j\, b_j(d_1)$$

and

$$\Psi_1(j) = 0 .$$

Using this information we can define a backtracking procedure, beginning with

$$\hat{q}_T = \underset{i}{\operatorname{argmax}} \left(\{\delta_T(i)\}_{i=1,\cdots,N} \right)$$

and continuing by

$$\hat{q}_t = \Psi_{(t+1)}(\hat{q}_{(t+1)})$$

until the estimate of the whole sequence is complete.

The last class of problems in HMMs is learning, where it is necessary to find the 'best' parameter set $\lambda = (\{a_{i,j}\}, \{b_i(k)\}, \{\pi_i\})$ with respect to $p(\{d_1, \cdots, d_T\} \mid \lambda)$. For finding these parameter sets the 'Baum-Welch method' (Baum et al., 1970) can be used. Alternatively, gradient descent methods and other EM algorithms (see section 2.4.3) can be applied. The task shows similarities to learning in Bayesian belief networks as discussed in section 2.4.2.

One way to extend Hidden Markov models are the so called 'Hidden semi-Markov' models (Murphy, 2002). This modification allows to model the generation of symbol sequences instead of single symbols while being in a hidden state.

A.4.2 Helmholtz machines

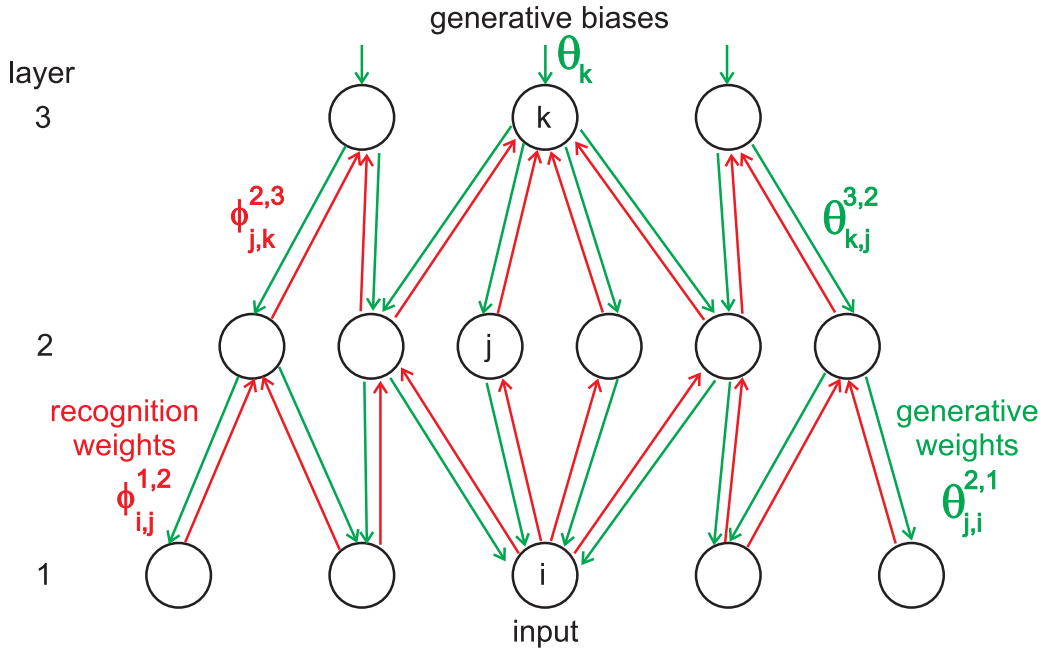


Figure A.11: Example of a three-layer Helmholtz machine. (Illustration adapted from (Dayan et al., 1995))

Training the network for a Boltzmann machine is a complex problem and sometimes not practical. If intra-layer connections are not necessary for performing the aimed information processing, then it is possible to switch to a similar network structure, termed Helmholtz machine. For this type of neuronal network a feasible training algorithm is available.

The Helmholtz machine (Dayan et al., 1995; Dayan and Hinton, 1996) is a hierarchical neuronal network with binary stochastic units including a training algorithm that allows self-supervised learning. This units are connected by two sets of weights. Connection sets θ in top-down direction implement a generative model while the bottom-up connection sets ϕ realise a recognition model (see Fig. A.11). Using this network as generative model can be interpreted as reconstructing the input on the basis of the given information, while the recognition model can be interpreted as mapping the input onto 'representations' (activities of neurons in the hidden layer).

No intra-layer connections are allowed (otherwise this machine would be a Boltzmann machine). If the network is used with weights from the generative model or the recognition model, information flows in only one direction. This means that the following layer in the information processing hierarchy does not project its results backwards. The connections can bypass one or several layers.

The Helmholtz machine is used in two different versions: The deterministic Helmholtz

machine (Dayan et al., 1995), using mean values for the recognition model, and the stochastic Helmholtz machine. For the stochastic version, a learning procedure called 'wake-sleep' algorithm can be used. In the wake phase, the network is used in recognition mode and the representation inferred from the input is used to update the weights of the generative model. During the second, 'sleep' phase, the network produces (fantasizes) representations and inputs, without external input. This information is used to update the weights used by the recognition model. The update rule (local 'delta rule') is similar to Eq.(A.22).

For some examples it is possible to use EM on $\log(p(Data \mid \theta))$ for finding suitable weight sets θ. Using the generative model part of a Helmholtz machine, we can calculate the log probability of generating an example '$Data$', which is given by

$$\begin{aligned} \log(p(Data \mid \theta)) &= \log\left(\sum_{\alpha} p(Data, \alpha \mid \theta)\right) \\ &= \sum_{\alpha} P_{\alpha}(Data) \log(p(Data, \alpha \mid \theta)) - \sum_{\alpha} P_{\alpha}(Data) \log(P_{\alpha}(Data)) \end{aligned} \tag{A.25}$$

with $P_{\alpha}(Data) = p(\alpha \mid Data, \theta)$. In general, the posterior probability $P_{\alpha}(Data)$ can be complex and may be not decomposed into products of simpler distributions. This can make the computational evaluation difficult. Dayan and Hinton (Dayan and Hinton, 1996) suggest to use $Q_{\alpha}(Data)$ instead, which is an arbitrary probability distribution that need not to depend on θ. Using this distribution in Eq.(A.25) yields

$$\begin{aligned} \log(p(Data \mid \theta)) &= \sum_{\alpha} Q_{\alpha}(Data) \log(p(Data, \alpha \mid \theta)) - \sum_{\alpha} Q_{\alpha}(Data) \log(Q_{\alpha}(Data)) \\ &\quad + D_{KL}(Q(Data) \parallel P(Data)) \\ &= -F(Data; \theta, Q) + D_{KL}(Q(Data) \parallel P(Data)) \end{aligned} \tag{A.26}$$

with

$$D_{KL}(Q(Data) \parallel P(Data)) = \sum_{\alpha} Q_{\alpha}(Data) \log\left(\frac{Q_{\alpha}(Data)}{P_{\alpha}(Data)}\right).$$

Eq.(A.26) is especially interesting because it was shown (Neal and Hinton, 1999) that $F(Data; \theta, Q)$ is minimised, where $Q_{\alpha}(Data)$ equals $P_{\alpha}(Data)$ and $D_{KL}(Q(Data) \parallel P(Data)) = 0$.

For the Helmholtz machine many extensions were proposed, see (Dayan and Hinton, 1996) for a detailed discussion.

Appendix B

Information processing spike by spike

B.1 Pattern Pre-Processing

Before using patterns as input for the spike-by-spike algorithm, it is necessary to convert the raw input data into suitable probability distributions.

To avoid negative firing rates, the raw image or bit patterns $\mathbf{B}_\mu$ with N components each were pre-processed to form the positive input patterns $\mathbf{B}_\mu{}^+$ finally applied to the network. In a first step, the $b_{\mu,s'}$, $s' = 1, \ldots, N$ were normalized by subtracting the individual mean value $< b_{\mu,s'} >= 1/N \sum_{s'=1}^{N} b_{\mu,s'}$,

$$\tilde{b}_{\mu,s'} = b_{\mu,s'} - < b_{\mu,s'} > \quad . \tag{B.1}$$

Then, the N components were duplicated and assigned pairwise to the even and uneven input node pairs, yielding $S = 2N$ non-negative components $b_{\mu,s}{}^+$ according to the expressions

$$b_{\mu,2s'-1}{}^+ = \begin{cases} +\tilde{b}_{\mu,s'} & \text{for } \tilde{b}_{\mu,s'} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{B.2}$$

$$b_{\mu,2s'}{}^+ = \begin{cases} 0 & \text{for } \tilde{b}_{\mu,s'} > 0 \\ -\tilde{b}_{\mu,s'} & \text{otherwise} \end{cases} \quad . \tag{B.3}$$

This pre-processing is motivated by the properties of early visual processing in the brain: splitting the mean-value corrected input into negative and positive values resembles the analysis of visual stimuli by on- and off-cells in the lateral geniculate nucleus (LGN).

B.2 Training Procedures

For the training procedure (see Fig. 3.2), the S input nodes are split into two sets of sizes S_p (indexed by $s = 1, \ldots, S_p$) and S_c (indexed by $s = S_p + 1, \ldots, M = S_p + S_c$). Correspondingly, there also exist two sets of conditional probabilities $p(s|i)$. Each item in the training set comprises a non-negative input pattern $\mathbf{B}_\mu^{tr+}$ together with its correct classification c_μ^{tr}. The pattern $\mathbf{B}_\mu^{tr+}$ is applied to the first set of S_p input nodes, while the correct classification c_μ^{tr} activates the c_μ^{tr}-th node of the second set of input nodes. Pattern and classification inputs are in addition weighted by a factor λ. It controls the strength of the 'input' and 'output' arguments through the number of spikes and thus balances the combination of two streams of information during training the network. Using these assignments, the final training scenes $\mathbf{V}_\mu^{tr}$ are given by the expressions

$$\nu_{\mu,s}^{tr} = \lambda\, b_{\mu,s}^{tr} \Big/ \sum_{s'=1}^{S_p} b_{\mu,s'}^{tr} \quad \text{for } s \in [1, S_p] \tag{B.4}$$

$$\text{and} \quad \nu_{\mu,s}^{tr} = (1-\lambda)\delta_{s-S_p, c_\mu^{tr}} \quad \text{for } s \in [S_p + 1, S_p + S_c]. \tag{B.5}$$

During training, spikes are drawn randomly from $\mathbf{V}_\mu^{tr}$ according to Eq.(3.2), while both $h(i)$ and $p(s|i)$ are updated with the SbS-online or SbS-batch algorithms.

B.3 Classification and Computation Procedures

For the classification run, input scenes $\mathbf{V}_\mu^{ts}$ are composed solely of the pre-processed patterns $\mathbf{B}_\mu^{ts+}$ via $\nu_{\mu,s}^{ts} = b_{\mu,s}^{ts} / \sum_{s'=1}^{S_p} b_{\mu,s'}^{ts}$. (see Fig. 3.3). The first set of conditional probabilities $p(s|i)$ from the learning procedure with $s = 1, \ldots, S_p$ is re-normalized, yielding the *pattern weights*

$$p^*(s|i) = p(s|i) / \sum_{s'=1}^{S_p} p(s'|i)\,. \tag{B.6}$$

For each presented pattern or scene μ, now only the internal states $h_\mu(i)$ but not the weights $p(s|i)$ are updated using Eq.(3.20). The remaining S_c weight vectors are normalized, yielding the *classification weights*

$$\hat{p}(c|i) = \frac{p(c + S_p|i)}{\sum_{c'=1}^{S_c} p(c' + S_p|i)} \quad \text{for } c = 1, \ldots, S_c\,. \tag{B.7}$$

From $h(i)$ and $\hat{p}(c|i)$, quantities $q_\mu(c) = \sum_{i=1}^{H} \hat{p}(c|i) h_\mu(i)$ are computed. These $q_\mu(c)$ denote the probabilities for each of the M_{ts} test patterns $\mathbf{V}_\mu^{ts}$ to belong to the class c. From the $q_\mu(c)$, a prediction for the classification $\hat{c}_\mu^t$ is computed and updated within each time step t,

$$\hat{c}_\mu^t = \mathrm{argmax}_c\; q_\mu^t(c)\,. \tag{B.8}$$

The mean classification error e^t over all M_{ts} test patterns is finally computed as

$$e^t = 1/M_{ts} \sum_{\mu=1}^{M_{ts}} \delta_{c_\mu^{ts}, \hat{c}_\mu^t} \ . \tag{B.9}$$

B.4 Details and Parameters for the Computation of Boolean Functions

From all possible 2^{32} Boolean functions of 5 input and 1 output bits, $F_o = 100$ Boolean functions were chosen randomly for the SbS network (with on-line learning). By splitting the inputs arguments into on-off channels as described in section B.1, we will therefore have $S = S_p + S_c = 12$ input nodes.

During *SbS with on-line learning*, the 32 pattern were presented sequentially with a number of 4620 spikes per pattern and with $\lambda = 5/6$. This procedure was repeated $Z = 20$ times. Before each learning step, and for each different pattern μ, the h's were reset to a flat prior of $h_\mu^0 = 1/H$. Before the first learning step, the $p^z(s|i)$ were uniformly initialized with $p^0(s|i) = 1/S$. After learning, the h's were again reset to h_μ^0, and each input pattern was presented for $T = 1000$ spikes for testing the classification performance. Learning constants were chosen as $\epsilon = 1000/1001$ and $\gamma = 0.01/1.01$, unless stated otherwise.

B.5 Details and Parameters for the Classification of Handwritten Digits

The data base of handwritten digits from the United States Postal Service (USPS) contains $M_{tr} = 7291$ training and $M_{ts} = 2007$ test patterns, each one consisting of $N = 16 \times 16$ grey scale values (pixels) ranging from -1 (black) to 1 (white). According to the data preprocessing described before, the network then comprises $S_p = 512$ pattern input channels, and $S_c = 10$ classification input channels for the training.

Using the SbS algorithm with batch learning, one learning step included the parallel presentation of all training patterns with $\lambda = 0.5$, $\epsilon = 0.1$, $T = 4620$, $\Delta = 4620$, and $Z = 20$ similar to the procedure employed with the Boolean functions. During classification, each digit pattern was presented for $T = 10000$ spikes.

In order to test for the robustness of the algorithms, the recognition was made more difficult in two different ways. The first challenge was introduced by an occlusion of a variable number of rows or columns. Starting from the center columns or rows of the digit pattern, pixel values of whole rows and columns were set to an intermediate value

of 0. For example, a vertical occlusion of 6 columns on a pattern of size 16×16 was realized by affecting the pixel values of columns 6 to 11 in all rows 1 to 16. A horizontal occlusion of 4 rows was realized by setting all pixel values of rows 7 to 10 of all columns 1 to 16 to a value of 0. The second challenge was introduced by superimposing random noise on the digit pattern. In detail, for each digit to be noisified, random pixel values $\mathbf{B}_{\mu}^{\mathrm{rnd}}$ were drawn from a uniform distribution between -1 and 1. By means of a parameter $\eta \in [0, 1]$, the original pattern was linearly combined with the noise pattern to a new input pattern $b_{\mu,s}^{new}$ according to the rule

$$b_{\mu,s}^{new} = (1 - \eta) b_{\mu,s} + \eta b_{\mu,s}^{\mathrm{rnd}} \,. \tag{B.10}$$

The parameter η regulates the amount of noise on the original pattern. $\eta = 0$ represents the noise-free original pattern, and $\eta = 1$ denotes the case when the original pattern has been fully replaced by the noise pattern.

Appendix C

Stabilizing decoding against non-stationaries

C.1 The estimator for the velocity

The simulations shown in chapter 5 use optimal linear Bayesian estimators for inferring velocities of movements from spike-counts. In the following, we will see how this estimators can be derived for tuning-functions with cosine tuning for the direction of the movement and linear tuning for the length of the vector describing the movement. Furthermore, a Poissonian random process is assumed for drawing the corresponding spike-counts.

For the computation of the optimal linear estimator according to Eq.(2.33)

$$\begin{aligned}
\mathbf{D}_j = \sum_i (\mathbf{L}_i - M_i \mathbf{Z}) \left[\mathbf{R}^{-1}\right]_{i,j} , \qquad & \mathbf{R} = \{Q_{i,j} - M_i M_j\}_{i,j=1,\ldots,N} , \\
M_i = \int_{\mathbf{x}} \rho(\mathbf{x}) g_i(\mathbf{x}) , \qquad & \mathbf{L}_i = \int_{\mathbf{x}} \rho(\mathbf{x}) g_i(\mathbf{x}) \mathbf{x} , \\
\mathbf{Z} = \int_{\mathbf{x}} \rho(\mathbf{x}) \mathbf{x} , \qquad & Q_{i,j} = \sum_{\mathbf{k}} \int_{\mathbf{x}} \rho(\mathbf{x}) p(\mathbf{k} \mid \mathbf{x}, \mathbf{P})) k_i k_j , \\
\text{and} \qquad & g_i(\mathbf{x}) = \sum_{\mathbf{k}} p(\mathbf{k} \mid \mathbf{x}, \mathbf{P}) k_i ,
\end{aligned}$$

we assume that the velocity $\boldsymbol{\nu}$ is defined on a disk-shaped set with radius $\nu_{\max}$ (maximum velocity). For simplicity, we further assume that all directions and absolute values for the velocities are uniformly distributed on this set. It follows directly that $\mathbf{Z} = 0$.

The three remaining quantities $\mathbf{L}_i$, M_i and $Q_{i,j}$ from Eq.(2.33), for determining M_j,

$\mathbf{D}_j$, and $\mathbf{Z}$ which compose the estimator Eq.(2.32)

$$\hat{\mathbf{x}}(\mathbf{k}) = \sum_{j=1}^{N} (k_j - M_j)\,\mathbf{D}_j + \mathbf{Z}\,,$$

are calculated analytically using the following equations

$$\begin{aligned}
\mathbf{L}_i &= \frac{1}{\pi {\nu_{max}}^2} \int_0^{2\pi} d\varphi \int_0^{\nu_{max}} d\nu\, \nu^2 \{\cos(\varphi), \sin(\varphi)\} \\
&\times \left(\frac{f_i^{mod} T}{2} \left(1 + \frac{\nu}{\nu_{max}} \cos\left(\varphi - \varphi_i\right) \right) + f_i^{off} T \right) \\
&= \frac{T}{8} f_i^{mod}\, \nu_{max} \{\cos(\varphi_i), \sin(\varphi_i)\} \qquad \text{(C.1)} \\
M_i &= \frac{1}{\pi {\nu_{max}}^2} \int_0^{2\pi} d\varphi \int_0^{\nu_{max}} d\nu\, \nu \left(\frac{f_i^{mod} T}{2} \left(1 + \frac{\nu}{\nu_{max}} \cos\left(\varphi - \varphi_i\right) \right) + f_i^{off} T \right) \\
&= T \left(\frac{f_i^{mod}}{2} + f_i^{off} \right) \qquad \text{(C.2)} \\
Q_{ij} &= \frac{1}{\pi {\nu_{max}}^2} \int_0^{2\pi} d\varphi \int_0^{\nu_{max}} d\nu\, \nu \left(\frac{f_i^{mod} T}{2} \left(1 + \frac{\nu}{\nu_{max}} \cos\left(\varphi - \varphi_i\right) \right) + f_i^{off} T \right) \\
&\times \left(\frac{f_j^{mod} T}{2} \left(1 + \frac{\nu}{\nu_{max}} \cos\left(\varphi - \varphi_j\right) \right) + f_j^{off} T \right) + \delta_{i,j} M_i \\
&= \frac{T^2}{16} \left(8 \left(\frac{f_i^{mod}}{2} + f_i^{off} \right) \left(\frac{f_j^{mod}}{2} + f_j^{off} \right) + f_i^{mod} f_j^{mod} \cos\left(\varphi_i - \varphi_j\right) \right) \\
&+ T \delta_{i,j} \left(\frac{f_i^{mod}}{2} + f_i^{off} \right) \qquad \text{(C.3)}
\end{aligned}$$

Inserting these expressions into the definition for $\mathbf{D}_j$ yields

$$\begin{aligned}
\mathbf{D}_j &= \sum_i^N \mathbf{L}_i \left(Q_{ij} - M_j M_i \right)^{-1} \\
&= \sum_i^N f_i^{mod} \nu_{max} \{\cos(\varphi_i), \sin(\varphi_i)\} \\
&\times \left[\left\{ \frac{T}{2} f_i^{mod} f_j^{mod} \cos\left(\varphi_i - \varphi_j\right) + 8 \delta_{i,j} \left(\frac{f_i^{mod}}{2} + f_i^{off} \right) \right\}^{-1} \right]_{i,j} \qquad \text{(C.4)}
\end{aligned}$$

which can be directly inserted into the estimator equation (2.32) reading

$$\hat{\mathbf{v}}(\mathbf{k}) = \sum_i^N \left(k_i - T \left(\frac{f_i^{mod}}{2} + f_i^{off} \right) \right) \mathbf{D}_i\,. \qquad \text{(C.5)}$$

This equations assume that we know the real parameter of the tuning functions $\mathbf{P}$. But $\mathbf{P}$ are not known. Thus makes it necessary to use the learned approximate parameter set $\hat{\mathbf{P}}$ for the computation of $\hat{\mathbf{v}}(\mathbf{k})$.

C.2 Parameter adaptation

The real parameters $\mathbf{P}$ of the system that encodes the information about the intended movement into spike trains are unknown. This makes it necessary to find suitable approximations of $\mathbf{P}$ instead. Errors made by the reconstruction of the intended movement are used as a measure how good the approximation of the parameters is. The preferred directions φ_i are approximated by a random walk which is guided by the size of the error signal. Other subsets of parameters (f_i^{mod} and f_i^{off}) can be inferred from the mean activities and variance of the activities of the neurons itself. In the following, the whole procedure is described in detail.

A new parameter set $\hat{\mathbf{P}}'$ was computed in three different steps.

1. The mean $<k_i>$ and variance ν_i of the firing rate of each neuron was computed by averaging over the last $T_{avg} = 900$ seconds

$$<k_i>(t) = 1/T_{avg} \sum_{t'=t-T_{avg}+1}^{t} k_i(t') \tag{C.6}$$

$$\nu_i(t) = \sum_{t'=t-T_{avg}+1}^{t} (k_i(t') - <k_i>(t))^2 . \tag{C.7}$$

   An estimate of f_i^{mod} and f_i^{off} was then obtained by the expressions

$$\hat{f}_i^{\mathrm{mod}\prime} = 4\sqrt{\nu_i(t) - <k_i>(t)} \tag{C.8}$$

$$\hat{f}_i^{\mathrm{off}\prime} = <k_i>(t) - \hat{f}_i^{\mathrm{mod}\prime} . \tag{C.9}$$

   It is important to note that these quantities may also suffer from sampling noise, thus one has to select a T_{avg} which is not too small.

2. Only neurons which respond well, or showed a substantial variation in their firing rates are chosen for the estimation procedure. This criterion was implemented by ignoring activities from neurons with $\hat{f}_i^{\mathrm{mod}\prime} < f_T$, with the modulation threshold parameter f_T set to 10 Hz. Selecting the tuning curves from the random distribution introduced in the main text body, one typically obtains between $N_\nu = 6 \ldots 15$ neurons whose signals will be used in the estimation process.

3. A new estimate for the preferred directions φ_i' was then obtained by a random angle shift on the current φ_i's,

$$\hat{\varphi}_i' = \hat{\varphi}_i^0 + 2\pi\epsilon_\phi\eta_i , \tag{C.10}$$

   where η_i denotes a random number drawn from a normal distribution with mean 0 and standard deviation 1. The scaling variable ϵ_ϕ has the purpose of decreasing the adaptation speed with decreasing movement error, and vice versa. It was

adjusted by hand, and subsequently found to yield good results if chosen to be $\epsilon_\phi = 0.1\sqrt{<\hat{E}>}$. $<\hat{E}>$ denotes the estimate of the perceived error averaged over either the second or fourth step of the adaptation algorithm (whichever came last). The number of spikes over which the newly constructed estimator $\hat{\mathbf{P}}' = \{\hat{f}_1^{\text{off}\prime}, \ldots, \hat{f}_{N_v}^{\text{off}\prime}, \hat{f}_1^{\text{mod}\prime}, \ldots, \hat{f}_{N_v}^{\text{mod}\prime}, \hat{\varphi}'_1, \ldots, \hat{\varphi}'_{N_v}\}$ is to be tested was then set to $N_s = 1000/<\hat{E}>$.

In section 2.2.3 Eq.(2.29)

$$\hat{f}(K) = \frac{\Gamma(2+K, F_{min}, F_{max})}{(F_{max}-F_{min})\,\Gamma(1+K, F_{min}, F_{max})} - \frac{F_{min}}{(F_{max}-F_{min})}$$

shows how an optimal Bayesian estimator for this type of error signal (with linear tuning) looks like. It should be noted that Eq.(2.29) was derived for a uniform distribution of error values in the interval $[0, 1]$. Any deviation from this assumed value distribution leads to a bias in the estimation.

Appendix D

Additional information sources

In addition to the literature which was cited in the text, I used (Bethge, 2003; Schulzke, 2006; Hoyer, 2002; MacKay, 2003; Dayan and Abbott, 2001; Goebel et al., 2003; Hubel, 1989; Johansen-Berg, 2001) and the 'Antrag auf Finanzierung des Sonderforschungsbereiches 517 2005-2007' of the University of Bremen and Carl von Ossietzky University of Oldenburg. These sources were used e.g. as guidelines for deciding which topics could be interesting for my introduction into the field of computational neuroscience as well as how the presentation can be structured.

Furthermore, I used various articles from wikipedia.org, especially from the field of neuro-science and information-theory, for getting a more broader impression of the sub-fields.

Another type of additional sources for whole text segments and images were publications which I published with other scientists (see my publication list for a more detailed list).

Bibliography

Abeles, M. (1991). Corticonics. *Cambridge University Press*.

Aertsen, A., Gerstein, G., Habib, M., and Palm, G. (1989). Dynamics of neuronal firing correlation: modulation of effective connectivity. *J. Neurophysiol.*, 61:900–917.

Aertsen, A., Gerstein, G., and Johannesma, P. (1986). Brain theory. pages 7–24.

Aldrich, J. (1997). R.A. Fisher and the making of maximum likelihood 1912–1922. *Statist. Sci.*, 12(3):162–176.

Amirikian, B. and Georgopoulos, A. (2000). Directional tuning profiles of motor cortical cells. *Neurosci Res.*, 36(1):73–79.

Andersen, R., Burdick, J., Mussallam, S., Pesaran, B., and Cham, J. (2004a). Cognitive neural prosthetics. *TRENDS in Cognitive Science*, 8(11):486–493.

Andersen, R., Mussallam, S., and Pesaran, B. (2004b). Selecting the signals for a brain-machine interface. *Current Opinion in Neurobiology*, 14:720–726.

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussianbayesian tracking. *Signal Processing*, 50(2):174–188.

Ashe, J. and Georgopoulos, A. (1994). Movement parameters and neural activity in motor cortex and area 5. *Cerebral Cortex*, 4:590–600.

Astolfi, L., Cincotti, F., Babiloni, C., Carducci, F., Basilisco, A., Rossini, P., Salinari, S., Mattia, D., Cerutti, S., Dayan, D., Ni, L. D. L. Y., He, B., and Babiloni, F. (2005). Estimation of the cortical connectivity by high-resolution eeg and structural equation modeling: simulations and application to finger tapping data. *IEEE Trans Biomed Eng*, 52(5):757–768.

Azouz, R. and Gray, C. (2000). Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo. *PNAS*, 97(14):8110–8115.

Azouz, R. and Gray, C. (2003). Adaptive coincidence detection and dynamic gain control in visual cortical neurons in vivo. *Neuron*, 37:513–523.

Ball, T., Nawrot, M., Schulze-Bonhage, A., Aertsen, A., and Mehring, C. (2004). Towards a brain-machine interface based on epicortical field potentials. *Biomed. Eng.*, 49 (Suppl. 2):756–759.

Barlow, H. (1960). *The coding of sensory messages*, pages 331–360. Cambridge University Press.

Batista, A., Buneo, C., Snyder, L., and Andersen, R. (1999). Reach plans in eye-centered coordinates. *Science*, 285:257–260.

Baum, L., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, 41(1):164–171.

Beck, J. M. and Pouget, A. (2007). Exact inferences in a neural implementation of a hidden markov model. *Neural Computation*, 19:1344–1361.

Bell, A. and Sejnowski, T. (1997). The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338.

Bernardo, J. (1979). Expected information as expected utility. *Annals of Statistics*, 7(3):686–690.

Berry, M., Warland, D., and Meister, M. (1997). The structure and precision of retinal spike trains. *Proc. Natl. Acad. Sci.*, 99:5411–5416.

Bertsekas, D. (1995). *Non-linear programming*. Athena Scientific, Belmont, MA.

Bethge, M. (2003). *Codes and Goals of Neuronal Representations*. PhD thesis, University of Bremen.

Bethge, M., Rotermund, D., and Pawelzik, K. (2001). Optimal short-term population coding: When fisher information fails. *Proceedings of the 28th Göttingen Neurobiology Conference*.

Bethge, M., Rotermund, D., and Pawelzik, K. (2002a). Binary tuning is optimal for neural rate coding with high temporal resolution. *Advances in Neural Information Processing Systems*, 15.

Bethge, M., Rotermund, D., and Pawelzik, K. (2002b). Optimal neural population coding and fisher information. *Verhandlungen der Deutschen Physikalischen Gesellschaft DPG (VI) 37, 1, V.*

Bethge, M., Rotermund, D., and Pawelzik, K. (2002c). Optimal short-term population coding: when fisher information fails. *Neural Comput.*, 14(10):2317–2351.

Bethge, M., Rotermund, D., and Pawelzik, K. (2003a). Optimal neural rate coding leads to bimodal firing rate distributions. *Network: Comput. Neural Syst.*, 14:303–319.

Bethge, M., Rotermund, D., and Pawelzik, K. (2003b). A second order phase transition in neural rate coding: Binary encoding is optimal for rapid signal transmission. *Phys. Rev. Lett.*, 90(8):088104–1.

Bilmes, J. (1997). A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, University of Berkeley.

Binder, J., Koller, D., Russell, S., and Kanazawa, K. (1997). Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2-3):213–244.

Black, M., Bienenstock, E., Donoghue, J., Serruya, M., Wu, W., and Gao, Y. (2003). Connecting brains with machines: The neural control of 2D cursor movement. In *1st International IEEE/EMBS Conference on Neural Engineering*, pages 580–583.

Blahut, R. (1987). *Principles and practice of information theory.*

Blankertz, B., Curio, G., and Mueller, K.-R. (2002). Classifying single trial EEG: Towards brain computer interfacing. *Advances in Neural Inf. Proc. Systems*, 14:157–164.

Blankertz, B., Dornhege, G., Krauledat, M., Müller, K., Kunzmann, V., Losch, F., and Curio, G. (2006). The berlin brain-computer interface: Eeg-based communication without subject training. *IEEE Trans Neural Syst Rehabil Eng.*, 14(2):147–152.

Bliss, T. and Collingridge, G. (1993). A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, 361:31–39.

Bonhoeffer, T. and Grinvald, A. (1991). Iso-orientation domains in cat visual cortex are arranged in pinwheel-like patterns. *Nature*, 353:429–431.

Botvinick, M., Braver, T., Barch, D., Carter, C., and Cohen, J. (2001). Conflict monitoring and cognitive control. *Psychol Rev.*, 108(3):624–652.

Botvinick, M., Cohen, J., and Carter, C. (2004). Conflict monitoring and anterior cingulate cortex: an update. *Trends Cogn Sci.*, 8(12):539–546.

Boussaoud, D., Jouffrais, C., and Bremmer, F. (1998). Eye position effects on the neuronal activity of dorsal premotor cortex in the macaque monkey. *J. Neurophysiol.*, 80:1132–1150.

Britten, K., Shadlen, M., Newsome, W., and Movshon, J. (1993). Responses of neurons in macaque mt to stochastic motion signals. *Visual Neuroscience*, 10:1157–1169.

Brodmann, K. (1909). *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues.*

Bromley, J. and Säckinger, E. (1991). Neural- network and k- nearest-neighbor classifiers. Technical report, Tech. Rep. 11359-910819-16TM, AT&T.

Bruno, R. and Sakmann, B. (2006). Cortex is driven by weak but synchronously active thalamocortical synapses. *Science*, 312:1622–1627.

Bryson, A. and Ho, Y. (1969). *Applied optimal control: optimization, estimation, and control*.

Bullock, D. and Grossberg, S. (1988). Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychol Rev.*, 95(1):49–90.

Buntine, W. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225.

Buracas, G., Zador, A., DeWeese, M., and Albrigh, T. (1998). Efficient discrimination of temporal patterns by motion-sensitive neurons in primate visual cortex. *Neuron*, 20(5):959–969.

Burkitt, A. (2006a). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics*, 95(1):1–19.

Burkitt, A. (2006b). A review of the integrate-and-fire neuron model: Ii. inhomogeneous synaptic input and network properties. *Biological Cybernetics*, 95(2):97–112.

Burns, B. and Webb, A. (1976). The spontaneous activity of neurons in the cat's cerebral cortex. *Neuron*, 20:959–969.

Callaway, E. (1998). Local circuits in primary visual cortex of the macaque monkey. *Annual Review of Neuroscience*, 21:47–74.

Carandini, M., Demb, J., Mante, V., Tolhurst, D., Dan, Y., Olshausen, B., Gallant, J., and Rust, N. (2005). Do we know what the early visual system does? *Journal of Neuroscience*, 25(46):10577–10597.

Carandini, M. and Heeger, D. (1994). Summation and division by neurons in primate visual cortex. *Science*, 264:1333–1336.

Carmena, J., Lebedev, M., Crist, R., O'Doherty, J., Santucci, D., Dimitrov, D., Patil, P., Henriguez, C., and Nicolelis, M. (2003). Learning to control a brain-machine interface for reaching and grasping by primates. *Public Library of Science Biology*, 1(2):193–208.

Carter, C., Braver, T., Barch, D., Botvinick, M., Noll, D., and Cohen, J. (1998). Anterior cingulate cortex, error detection, and the online monitoring of performance. *Science*, 280(5364):747–749.

Casagrande, V. (1999). The mystery of the visual system k pathway. *The Journal of Physiology*, 517(3):630.

Chance, F. S., Abbott, L. F., and Reyes, A. D. (2002). Gain modulation from background synaptic input. *Neuron*, 35:773–782.

Chang, C. and Lin, C. (2001). Libsvm: a library for support vector machines.

Chelazzi, L., Duncan, J., Miller, E., and Desimone, R. (1998). Responses of neurons in inferior temporal cortex during memory-guided visual search. *J Neurophysiol*, 80:2918–2940.

Chelazzi, L., Miller, E., Duncan, J., and Desimone, R. (1993). A neural basis for visual search in inferior temporal cortex. *Nature*, 363:345–347.

Chen, H., Hua, S., Smith, M., Lenz, F., and Shadmehr, R. (2006). Effects of human cerebellar thalamus disruption on adaptive control of reaching. *Cerebral Cortex*, 16(10):1462–1473.

Chien, C. and Jaw, F. (2005). Miniature telemetry system for the recording of action and field potentials. *J Neurosci Methods.*, 147(1):68–73.

Chouinard, P. and Paus, T. (2006). The primary motor and premotor areas of the human cerebral cortex. *The Neuroscientist*, 12(2):143–152.

Cole, K. and Curtis, H. (1939). Electric impedance of the squid giant axon during activity. *The Journal of General Physiology*, 22:649–670.

Connolly, M. and van Essen, D. (1984). The representation of the visual field in parvicellular and magnocellular layers of the lateral geniculate nucleus in the macaque monkey. *The Journal of Comparative Neurology*, 226(4):544–564.

Connor, C. (2004). Visual attention: Bottom-up versus top-down. *Current Biology*, 14:R850–R852.

Connor, C., Preddie, D., Gallant, J., and van Essen, D. (1997). Spatial attention effects in macaque area v4. *Journal of Neuroscience*, 17(9):3201–3214.

Connor, J. and Stevens, C. (1971). Inward and delayed outward membrane currents in isolated neural somata under voltage clamp. *J Physiol.*, 213(1):1–19.

Cooper, G. (1990). The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42:393–405.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Crair, M., Ruthazer, E., Gillespie, D., and Stryker, M. (1997). Relationship between the ocular dominance and orientation maps in visual cortex of monocularly deprived cats. *Neuron*, 19:307–318.

Cramer, H. (1946). A contribution to the theory of statistical estimation. *Skandinavisk Aktuarietidskrift*, 19:85–94.

Culham, J. and Valyear, K. (2006). Human parietal cortex in action. *Curr Opin Neurobiol.*, 16(2):205–212.

Curran, E. and Stokes, M. (2003). Learning to control brain activity: A review of the production and control of eeg components for driving braincomputer interface (bci) systems. *Brain and Cognition*, 51(3):326–336.

Dagnelie, G. (2006). Visual prosthetics 2006: assessment and expectations. *Expert Rev Med Devices*, 3(3):315–325.

Dayan, P. and Abbott, L. (2001). *Computational and Mathematical Modeling of Neural Systems*.

Dayan, P. and Hinton, G. (1996). Varieties of helmholtz machine. *Neural Networks*, 9(8):1385–1403.

Dayan, P. and Hinton, G. (1997). Using expectation-maximization for reinforcement learning. *Neural Computation*, 9:271–278.

Dayan, P., Hinton, G., Neal, R., and Zemel, R. (1995). The helmholtz machine. *Neural Comput.*, 7(5):889–904.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B 39:1–38.

Deneve, S. (2005). Bayesian inference in spiking neurons. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 353–360. MIT Press, Cambridge, MA.

Deneve, S. (2007a). Bayesian spiking neurons i: Inference. *not published yet*.

Deneve, S. (2007b). Bayesian spiking neurons ii: Learning. *not published yet*.

Deneve, S., Latham, P. E., and Pouget, A. (1999). Reading population codes: a neural implementation of ideal observers. *Nature Neuroscience*, 2(8):740–745.

Deneve, S., Latham, P. E., and Pouget, A. (2001). Efficient computation and cue integration with noisy population codes. *Nature Neuroscience*, 4(8):826–831.

Desimone, R. and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annu. Rev. Neurosci.*, 18:193–222.

Desimone, R. and Schein, S. (1987). Visual properties of neurons in area v4 of the macaque: sensitivity to stimulus form. *Journal of Neurophysiology*, 57(3):835–868.

Devroye, L. (1986). *Non-Uniform Random Number Generation*.

Dickinson, A. and Balleine, B. (1994). Motivational control of goal-directed action. *Learn. behav.*, 22(1):1–18.

Diedrichsen, J., Hashambhoy, Y. L., Rane, T., and Shadmehr, R. (2005). Neural correlates of reach errors. *Journal of Neuroscience*, 25(43):9919–9931.

Dobbins, A., Jeo, R., Fiser, J., and Allman, J. (1998). Distance modulation of neural activity in the visual cortex. *Science*, 281(5376):552–555.

Donoghue, J., Leibovic, S., and Sanes, J. (1992). Organization of the forelimb area in squirrel monkey motor cortex: representation of digit, wrist, and elbow muscles. *Experimental Brain Research*, 89(1):1–19.

Eckhorn, R., Frien, A., Bauer, R., Woelbern, T., and Kehr, H. (1993). High frequency (60-90 hz) oscillations in primary visual cortex of awake monkey. *Neuroreport*, 4:243–246.

Eckhorn, R. and Poeppel, B. (1974). Rigorous and extended application of information theory to the afferent visual system of the cat. i. basic concepts. *Biological Cybernetics*, 16(4):191–200.

Edwards, F., Konnerth, A., Sakmann, B., and Busch, C. (1990). Quantal analysis of inhibitory synaptic transmission in the dentate gyrus of rat hippocampal slices: A patch clamp study. *J. Physiol.*, 430(1):213–249.

Egeth, H. and Yantis, S. (1997). Visual attention: Control, representation, and time course. *Annual Review of Psychology*, 28:269–297.

Elul, R. (1972). The genesis of the eeg. *International Review of Neurobiology*, 15:227–272.

Ernst, M. O. and Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415:429–433.

Ernst, U., Gutkin, B., Pawelzik, K., and Deneve, S. (2007a). Dynamics of probabilistic neuronal computation. *not published yet.*

Ernst, U., Rotermund, D., and Pawelzik, K. (2004). An algorithm for fast pattern recognition with random spikes. *26th DAGM Symposium Proceedings*, pages 399–406.

Ernst, U., Rotermund, D., and Pawelzik, K. (2007b). Efficient computation based on stochastic spikes. *Neural Computation.*

Fabiani, G., McFarland, D., Wolpaw, J., and Pfurtscheller, G. (2004). Ieee conversion of eeg activity into cursor movement by a brain-computer interface (bci). *Trans Neural Syst Rehabil Eng*, 12(3):331–338.

Falkenstein, M., Hoormann, J., Christ, S., and Hohnsbein, J. (2000). Erp components on reaction errors and their functional significance: a tutorial. *Biol Psychol.*, 51(2-3):87–107.

Farwell, L. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr Clin Neurophysiol.*, 70(6):510–523.

Fernandez, E., Ferrandez, J., Ammermller, J., and Normann, R. (2000). Population coding in spike trains of simultaneously recorded retinal ganglion cells. *Brain Res.*, 887:222–229.

Ferrera, V. and Maunsell, J. (2005). Motion processing in macaque v4. *Nature Neuroscience*, 8:1125.

Fiorillo, C., Tobler, P., and Schultz, W. (2003). Discrete coding of reward probability and uncertainty by dopamine neurons. *Science*, 299:1898–1902.

Fisher, R. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society A*, 222:309–368.

Fitzsimmons, N., Drake, W., Sandler, A., Hanson, T., Lebedev, M., and Nicolelis, M. (2005). Long-term behavioral improvements in a reaching task cued by microstimulation of the primary somatosensory cortex. *Society for Neuroscience Abstracts*, 31:402.7.

Flash, T. and Hochner, B. (2005). Motor primitives in vertebrates and invertebrates. *Curr Opin Neurobiol.*, 15(6):660–666.

Flash, T. and Sejnowski, T. (2001). Computational approaches to motor control. *Current Opinion in Neurobiology*, 11:655–662.

Fourcaud, N. and Brunel, N. (2002). Dynamics of the firing probability of noisy integrate-and-fire neurons. *Neural Computation*, 14:2057–2110.

Friedman, N. (1998). The Bayesian structural EM algorithm. In *UAI*, pages 129–138.

Fries, P., Nikolic, D., and Singer, W. (2007). The gamma cycle. *TINS special issue*, 30(7):309–316.

Fries, P., Reynolds, J., Rorie, A., and Desimone, R. (2001). Modulation of oscillatory neuronal synchronization by selective visual attention. *Science*, 291:1560–1563.

Frith, C., Friston, K., Liddle, P., and Frackowiak, R. (1991). Willed action and the prefrontal cortex in man: a study with pet. *Proc Biol Sci.*, 244(1311):241–246.

Fu, Q., Flament, D., Coltz, J., and Ebner, T. (1997). Relationship of cerebellar purkinje cell simple spike discharge to movement kinematics in the monkey. *Journal of Neurophysiology*, 78:478–491.

Fujii, N., Mushiake, H., and Tanji, J. (2000). Rostrocaudal distinction of the dorsal premotor area based on oculomotor involvement. *J. Neurophysiol.*, 83:1764–1769.

Fulton, J. (1935). *A Note on the Definition of the motor and premotor Areas.*

Gallant, J., Connor, C., Rakshit, S., Lewis, J., and Essen, D. V. (1996). Neural responses to polar, hyperbolic, and cartesian gratings in area v4 of the macaque monkey. *J Neurophysiol*, 76:2718–2739.

Garavan, H., Ross, T., Murphy, K., Roche, R., and Stein, E. (2002). Dissociable executive functions in the dynamic control of behavior: inhibition, error detection, and correction. *Neuroimage.*, 17(4):1820–1829.

Gehring, W., Coles, M., Meyer, D., and Donchin, E. (1995). A brain potential manifestation of error-related processing. *Journal of Electroencephalography and Clinical Neurophysiology*, Supplement 44:287–296.

Gehring, W. J., Goss, B., Coles, M. G. H., Meyer, D. E., and Donchin, E. (1993). A neural system for error detection and compensation. *Psychological Science*, 4:385–390.

Geman, S., Bienenstock, E., and Doursat, R. (1991). Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Georgopoulos, A., Kalaska, J., Caminiti, R., and Massey, J. (1982). On the relations between the direction of two-dimensional arm movement and cell discharge in primate motor cortex. *J. Neurosci.*, 2(11):1527–1537.

Georgopoulos, A., Kettner, R., and Schwartz, A. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. ii. coding of the direction of movement by a neuronal population. *J Neurosci.*, 8(8):2928–2937.

Georgopoulos, A., Schwartz, A., and Kettner, R. (1986). Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419.

Gerstner, W. and Kistler, W. (2002a). *Spiking Neuron Models*. Cambridge University Press.

Gerstner, W. and Kistler, W. (2002b). *Spiking Neuron Models - Single Neurons, Populations, Plasticity*.

Gevins, A. (1984). Analysis of the electromagnetic signals of the human brain: milestones, obstacles, and goals. *IEEE Trans Biomed Eng*, 31:833–850.

Ghose, G. and Ts'O, D. (1997). Form processing modules in primate area v4. *J Neurophysiol*, 77:2191–2196.

Gilbert, C., Ito, M., Kapadia, M., and Westheimer, G. (2000). Interactions between attention, context and learning in primary visual cortex. *Vision Res.*, 40(10–12):1217–1226.

Giszter, S., Mussa-Ivaldi, F., and Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13:467–491.

Goebel, R., Muckli, L., and Kim, D.-S. (2003). *The Human Nervous System, 2nd Edition - Chapter: The Visual System.*

Goodale, M. and Milner, A. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, 15:20–25.

Grayden, D. and Clark, G. (2005). *Cochlear Implants: A Practical Guide, 2nd Edition – Chapter: Implant design and development.*

Greschner, M., Bongard, M., Rujan, P., and Ammermller, J. (2002). Retinal ganglion cell synchronization by fixational eye movements improves feature estimation. *Nature Neuroscience*, 5:341–347.

Gurfinkel, V., Levick, Y., and Lebedev, M. (1991). Body scheme concept and motor control. body scheme in the postural automatisms regulation. *Intellectual Processes and Their Modelling*, pages 24–53.

Hämäläinen, M., Hari, R., nad J. Knuutila, R. I., and Lounasmaa, O. (1993). Magnetoencephalographytheory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.*, 65(2):413–497.

Hanazawa, A. and Komatsu, H. (2001). Influence of the direction of elemental luminance gradients on the responses of v4 cells to textured surfaces. *The Journal of Neuroscience*, 21(12):4490–4497.

Harris, C. and Wolpert, D. (1998). Signal-dependent noise determines motor planning. *Nature*, 394:780–784.

Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

He, B., Lian, J., Spencer, K., Dien, J., and Donchin, E. (2001). A cortical potential imaging analysis of the p300 and novelty p3 components. *Hum Brain Mapp.*, 12(2):120–130.

Hebb, D. (1949). The organization of behavior: A neuropsychological theory.

Heckerman, D. (1995). A tutorial on learning with bayesian networks. Technical report, Microsoft Research.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Heeger, D. (1992). Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, 9:181–198.

Heeger, D. and Ress, D. (2002). What does fmri tell us about neuronal activity? *Nat Rev Neurosci*, 3:142–151.

Hegde, J. and van Essen, D. (2005). Stimulus dependence of disparity coding in primate visual area v4. *J. Neurophysiol.*, 93:620–626.

Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the theory of neuronal computation.*

Herzog, M. H. and Fahle, M. (2002). Effects of grouping in contextual modulation. *Nature*, 415:433–436.

Hinkle, D. and Connor, C. (2001). Disparity tuning in macaque area v4. *Neuroreport.*, 12(2):365–369.

Hinkle, D. and Connor, C. (2002). Three-dimensional orientation tuning in macaque area v4. *Nature Neuroscience*, 5:665–670.

Hinterberger, T., Schmidt, S., Neumann, N., Mellinger, J., Blankertz, B., Curio, G., and Birbaumer, N. (2004). Brain-computer communication with slow cortical potentials: Methodology and critical aspects. *IEEE Trans. Biomed. Eng.*, 51(6):1011–1018.

Hinton, G., Osindero, S., and Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554.

Ho, M., Mobini, S., Chiang, T., Bradshaw, C., and Szabadi, E. (1999). Theory and method in the quantitative analysis of impulsive choice behaviour: implications for psychopharmacology. *Psychopharmacology*, 146(4):362–372.

Hochberg, L., Serruya, M., Friehs, G., Mukand, J., Saleh, M., Caplan, A., Branner, A., Chen, D., Penn, R., and Donoghue, J. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171.

Hodgkin, A. and Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544.

Holroyd, C. and Coles, M. (2002). The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological Review*, 109:679–709.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci*, 79(8):2554–2558.

Hosoya, T., Baccus, S. A., and Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, 436:71–77.

Hoyer, P. (2002). *Probabilistic Models of Early Vision.* PhD thesis, Helsinki University of Technology.

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469.

Hoyer, P. O. and Hyvärinen, A. (2003). Interpreting neural response variability as monte carlo sampling of the posterior. In *Advances in Neural Information Processing Systems 15 (2002)*, volume 15, pages 277–284. MIT press.

Hubel, D. (1989). *Auge und Gehirn – Neurobiologie des Sehens.*

Hubel, D. and Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. *J Physiol.*, 195(1):215–243.

Hubel, D. and Wiesel, T. (1977). Functional architecture of macaque monkey visual cortex (ferrier lecture). *Proc. R. Soc.*, 198:1–59.

Irazoqui-Pastor, P., Mody, I., and Judy, J. (2003). In-vivo eeg recording using a wireless implantable neural transceiver. *Neural Engineering, 2003*, 1:622–625.

Irazoqui-Pastor, P., Mody, I., and Judy, J. (2005). Recording brain activity wirelessly. inductive powering in miniature implantable neural recording devices. *IEEE Eng Med Biol Mag.*, 24(6):48–54.

Iriki, A., Tanaka, M., and Iwamura, Y. (1996). Coding of modified body schema during tool use by macaque postcentral neurones. *Neuroreport.*, 7(14):2325–2330.

Ito, M. and Gilbert, C. (1999). Attention modulates contextual influences in the primary visual cortex of alert monkeys. *Neuron*, 22:593–604.

Ito, S., Stuphorn, V., Brown, J., and Schall, J. (2003). Performance monitoring by the anterior cingulate cortex during saccade countermanding. *Science*, 302:120–122.

Itti, L. (2002). *The handbook of brain theory and neural networks - Chapter: Visual Attention.*

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews: Neuroscience*, 2(3):194–203.

Jackson, J. (1993). *Klassische Elektrodynamik, 2. Auflage.*

Javaheri, M., Hahn, D., Lakhanpal, R., Weiland, J., and Humayun, M. (2006). Retinal prostheses for the blind. *Ann Acad Med Singapore*, 35(3):137–144.

Jeffery, G. (2001). Architecture of the optic chiasm and the mechanisms that sculpt its development. *Physiol. Rev.*, 81:1393–1414.

Johansen-Berg, H. (2001). *Reorganisation and modulation of the human sensorimotor system: implications for recovery of motor functions after stroke.* PhD thesis, University of Oxford.

Johansson, R. and Birznieks, I. (2004). First spikes in ensembles of human tactile afferents code complex spatial fingertip events. *Nature Neurosci.*, 7(2):170–177.

Johnston, D. and Wu, S.-S. (1997). *Foundations of Cellular Neurophysiology.*

Jordan, M. (1999). *Learning in Graphical Models*.

Kalaska, J., Cohen, D., Prud'homme, M., and Hyde, M. (1990). Parietal area 5 neuronal activity encodes movement kinematics, not movement dynamics. *Experimental Brain Research*, 80(2):351–364.

Kamin, L. (1969). *Fundamental issues in associative learning Chapter: Selective association and conditioning*.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9:718–727.

Kerns, J., Cohen, J., 3rd MacDonald, A., Cho, R., Stenger, V., and Carter, C. (2004). Anterior cingulate conflict monitoring and adjustments in control. *Science*, 303(5660):1023–1026.

Kistler, W. and van Hemmen, J. (2000). Modeling synaptic plasticity in conjunction with the timing of pre- and postsynaptic action potentials. *Neural Comput.*, 12:385–405.

Knutti, J., Wildi, E., Marshall, J., Allen, H., and Meindl, J. (1979). Totally implantable dimension telemetry. *Biotelem Patient Monit.*, 6(3):133–146.

Kobatake, E. and Tanaka, K. (1994). Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. *J Neurophysiol.*, 71(3):856–867.

Kobayashi, S., Lauwereyns, J., Koizumi, M., and Sakagami, M. (2002). Influence of reward expectation on visuospatial processing in macaque lateral prefrontal cortex. *J. Neurophysiol.*, 87:1488–1498.

Koch, C. and Ullman, S. (1984). Selecting one among the many: A simple network implementing shifts in selective visual attention. Technical report, MIT A.I. Memo 770.

Koechlin, E., Anton, J. L., and Burnod, Y. (1999). Bayesian interference in populations of cortical neurons: a model of motion integration and segmentation in area mt. *Biol. Cyb.*, 80:25–44.

Körding, K. P. and Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. *Nature*, 427:244–247.

Kreiter, A., Aertsen, A., and Gerstein, G. (1989). A low-cost single-board solution for real-time, unsupervised waveform classification of multineuron recordings. *J Neurosci Methods*, 30(1):59–69.

Kreiter, A. and Singer, W. (1996). Brain theory. pages 201–227.

Krigolson, O. and Holroyd, C. (2006). Evidence for hierarchical error processing in the human brain. *Neuroscience*, 137(1):13–17.

Kronlandt-Martinet, R., Morlet, J., and Grossmann, A. (1987). Analysis of sound patterns through wavelet transforms. *Intern J Pattern Recognit Artif Intell*, 1:273–302.

Kuebler, A., Kotchoubey, B., Kaiser, J., Wolpaw, J., and Birbaumer, N. (2001). Brain-computer communication: Unlock the locked in. *Psychol. Bull.*, 127:358–375.

Kuffler, S. (1953). Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology*, 16(11):37–68.

Kullback, S. and Leibler, R. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Lackner, J. and DiZio, P. (2000). Aspects of body self-calibration. *Trends in Cognitive Science*, 4:279–288.

Lacquaniti, F., Terzuolo, C., and Viviani, P. (1983). The law relating the kinematic and figural aspects of drawing movements. *Acta Psychol*, 54(1-3):115–130.

Land, M. and Fernald, R. (1992). The evolution of eyes. *Annual Review of Neuroscience*, 15:1–29.

Land, M. and McLeod, P. (2000). From eye movements to actions: how batsmen hit the ball. *Nat Neurosci.*, 3(12):1340–1345.

Lanteri, H., Roche, M., and Aime, C. (2002). Penalized maximum likelihood image restoration with positivity constraints: multiplicative algorithms. *Inverse Problems*, 18:1397–1419.

Lanteri, H., Roche, M., Cuevas, O., and Aime, C. (2001). A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints. *Signal Processing*, 81:945–974.

Lapicque, L. (1907). Recherches quantitatives sur lexcitation electrique des nerfs traitee comme une polarization. *Journal of Physiology and Pathololgy*, 9:620–635.

Lebedev, M., Carmena, J., O'Doherty, J., Zacksenhouse, M., Henriquez, C., Principe, J., and Nicolelis, M. (2005). Cortical ensemble adaptation to represent velocity of an artifical actuator controlled by a brain-machine interface. *The Journal of Neuroscience*, 25(19):4681–4693.

Lebedev, M. and Nicolelis, M. (2006). Brain-machine interfaces: past, present and future. *Trends Neurosci.*, 29(9):536–546.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791.

Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *NIPS – Advances in Neural Information Processing Systems*, volume 13, pages 556–562. The MIT Press.

Lee, J. and van Donkelaar, P. (2006). The human dorsal premotor cortex generates on-line error corrections during sensorimotor adaptation. *J Neurosci.*, 26(12):3330–3334.

Lee, T. and Mumford, D. (2003). Hierarchical bayesian inference in the visual cortex. *J. Opt. Soc. Am.*, 20(7):1434–1448.

Lehmann, E. and Casella, G. (1999). *Theory of point estimation*.

Levy, W. and Baxter, R. (1996). Energy-efficient neural codes. *Neur. Comp.*, 8:531–543.

Lewicki, M. (1998). A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9(4):53–78.

Linden, D. and Connor, J. (1995). Long-term synaptic depression. *Annu. Rev. Neurosci.*, 18:319–357.

Livingstone, M. and Hubel, D. (1987). Psychophysical evidence for separate channels for the perception of form, color, movement, and depth. *Journal of Neuroscience*, 7:3416–3468.

Livingstone, M. and Hubel, D. (1988). Segregation of form, color, movement, and depth: Anatomy, physiology, and perception. *Science*, 240:740–749.

Logothetis, N. (1999). Vision: a window on consciousness. *Sci Am.*, 281(5):69–75.

Logothetis, N., Pauls, J., Augath, M., Trinath, T., and Oeltermann, A. (2001). Neurophysiological investigation of the basis of the fmri signal. *Nature*, 412:150–157.

Lucy, L. (1974). An iterative technique for the rectification of observed distributions. *Astron. J.*, 74:745–754.

Maass, W. and Bishop, C. (2000). *Pulsed Neural Networks*.

Maass, W., Natschlager, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.*, 14(11):2531–2560.

MacKay, D. (1995). Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505.

MacKay, D. (2003). *Information Theory, Inference and Learning Algorithms*.

Madigan, D., York, J., and Allard, D. (1995). Bayesian graphical models for discrete data. *International Statistical Review*, 63(2):215–232.

Mainen, Z. and Sejnowski, T. (1995). Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506.

Malpeli, J. and Baker, F. (1975). The representation of the visual field in the lateral geniculate nucleus of macaca mulatta. *J. Comp. Neurol.*, 161:569–594.

Mandon, S. and Kreiter, A. K. (2005). Rapid contour integration in macaque monkeys. *Vision Research*, 45:291–300.

Maravita, A., Spence, C., and Driver, J. (2003). Multisensory integration and the body schema: close to hand and within reach. *Curr Biol.*, 13(13):R531–539.

Martel, S., Hatsapoulos, N., Hunter, I., Donoghue, J., Burgert, J., Malasek, J., Wiseman, C., and Dyer, R. (2001). Development of a wireless brain implant: the telemetric electrode array system (teas) project. *Engineering in Medicine and Biology Society, 2001*, 4:3594–3597.

Martinez-Conde, S., Macknik, S., and Hubel, D. (2000). Microsaccadic eye movements and firing of single cells in the striate cortex of macaque monkeys. *Nature Neuroscience*, 3:251–258.

Martinez-Trujillo, J. and Treue, S. (2004). Feature-based attention increases the selectivity of population responses in primate visual cortex. *Curr Biol*, 14(9):744–751.

Maruishi, M., Tanaka, Y., Muranaka, H., Tsuji, T., Ozawa, Y., Imaizumi, S., Miyatani, M., and Kawahara, J. (2004). Brain activation during manipulation of the myoelectric prosthetic hand: a functional magnetic resonance imaging study. *Neuroimage*, 21(4):1604–1611.

Mason, S. and Graham, N. (2002). Areas beneath the relative operating characteristics (roc) and relative operating levels (rol) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society*, 128(584):2145–2166.

Matsumoto, K., Suzuki, W., and Tanaka, K. (2003). Neuronal correlates of goal-based motor selection in the preforntal cortex. *Science*, 301:229–232.

McAdams, C. and Maunsell, J. (1999a). Effects of attention on orientation-tuning functions of single neurons in macaque cortical area v4. *The Journal of Neuroscience*, 19(1):431–441.

McAdams, C. and Maunsell, J. (1999b). Effects of attention on the reliability of individual neurons in monkey visual cortex. *Neuron*, 23:765–773.

McAdams, C. and Maunsell, J. (2000). Attention to both space and feature modulates neuronal responses in macaque area v4. *Neurophysiol.*, 1751-1755:83(3).

McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*, 5:115–133.

Mehring, C., Nawrot, M., de Oliveira, S. C., Vaadia, E., Schulze-Bonhage, A., Aertsen, A., and Ball, T. (2004). Comparing information about arm movement direction in single channels of local and epicortical field potentials from monkey and human motor cortex. *J Physiol*, 98(4-6):498–506.

Mehring, C., Rickert, J., Vaadia, E., de Oliveira, S. C., Aertsen, A., and Rotter, S. (2003). Inference of hand movements from local field potentials in monkey motor cortex. *Nature Neuroscience*, 6:1253–1254.

Meinicke, P., Kaper, M., Hoppe, F., Heumann, M., and Ritter, H. (2003). Improving transfer rates in brain computer interfacing: A case study. *Advances in Neural Information Processing Systems*, 15:1107–1114.

Menon, V., Adleman, N., White, C., Glover, G., and Reiss, A. (2001). Error-related brain activation during a go/nogo response inhibition task. *Hum Brain Mapp.*, 12(3):131–143.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, H. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *J. Amer. Statist. Assoc.*, 44:335–341.

Miall, R. (2002). *The handbook of brain theory and neural networks - Chapter: Motor Control, Biological and Theoretical*.

Minsky, M. and Papert, S. (1969). *Perceptrons: an introduction to computational geometry*.

Mishkin, M., Ungerleider, L., and Macko, K. (1983). Object vision and spatial vision: Two cortical pathways. *Trends in Neurosci.*, 6:414–417.

Mitzdorf, U. (1985). Current source-density method and application in cat cerebral cortex: investigation of evoked potentials and eeg phenomena. *Physiol Rev.*, 65(1):37–100.

Mitzdorf, U. and Singer, W. (1979). Excitatory synaptic ensemble properties in the visual cortex of the macaque monkey: a current source density analysis of electrically evoked potentials. *J Comp Neurol.*, 187(1):71–83.

Mohseni, P., Najafi, K., Eliades, S., and Wang, X. (2005). Wireless multichannel biopotential recording using an integrated fm telemetry circuit. *Neural Systems and Rehabilitation Engineering, IEEE Transactions*, 13(3):263–271.

Mongillo, G. and Deneve, S. (2007). On-line learning with hidden markov models. *not published yet*.

Moody, J. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Comput.*, 1:281–294.

Moran, D. and Schwartz, A. (1999a). Motor cortical activity during drawing movements: Population representation during spiral tracing. *J. Neurophysiol.*, 82:2693–2704.

Moran, D. and Schwartz, A. (1999b). Motor cortical representation of speed and direction during reaching. *J. Neurophysiol.*, 82:2676–2692.

Moran, J. and Desimone, R. (1985). Selective attention gates visual processing in the extrastriate cortex. *Science*, 229:782–784.

Morizio, J., Irazoqui, P., Go, V., and Parmentier, J. (2005). Wireless headstage for neural prosthetics. *Neural Engineering, 2005*, 2nd:414–417.

Motter, B. (1993). Focal attention produces spatially selective processing in visual cortical areas v1, v2, and v4 in the presence of competing stimuli. *Journal of Neurophysiology*, 70(3):909–919.

Mottet, D. and Bootsma, R. (2001). The dynamics of rhythmical aiming in 2d task space: relation between geometry and kinematics under examination. *Hum Mov Sci.*, 20(3):213–241.

Müller, K., Gopfert, E., and Hartwig, M. (1985). Vep-untersuchungen zur kodierung der geschwindigkeit bewegter streifenmuster im kortex des menschen. *EEG-EMG*, 2(16):75–80.

Mumford, D. (2002). Pattern theory: the mathematics of perception. In Tatsien, L., editor, *International Congress of Mathematicians, Beijing, China*, volume III, pages 401–422. World Scientific.

Murata, N., Kawanabe, M., Ziehe, A., Mueller, K.-R., and Amari, S. (2002). On-line learning in changing environments with applications in supervised and unsupervised learning. *Neural Networks*, 15:743–760.

Murphy, K. (2002). Hidden semi-markov models. Technical report, MIT AI Lab.

Musallam, S., Corneil, B., Greger, B., Scherberger, H., and Andersen, R. (2004). Cognitive control signals for neural prosthetics. *Science*, 305:258–262.

Mushiake, H., Tanatsugu, Y., and Tanji, J. (1997). Neuronal activity in the ventral part of premotor cortex during target-reach movement is modulated by direction of gaze. *J. Neurophysiol.*, 78:567–571.

Mussa-Ivaldi, F. and Bizzi, E. (2000). Motor learning through the combination of primitives. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 355(1404):1755–1769.

Mysore, S., Vogels, R., Raiguel, S., and Orban, G. (2006). Processing of kinetic boundaries in macaque v4. *J Neurophysiol*, 95:1864–1880.

Nakahara, H., Itoh, H., Kawagoe, R., Takikawa, Y., and Hikoska, O. (2004). Dopamine neurons can represent context-dependent prediction error. *Neuron*, 41:269–280.

Natschläger, T., Maass, W., and Markram, H. (2002). The liquid computer: A novel strategy for real-time computing on time series. *Special Issue on Foundations of Information Processing of TELEMATIK*, 8(1):39–43.

Neal, R. (1993). Probabilistic inference using markov chain monte carlo methods. Technical report, University of Toronto.

Neal, R. and Hinton, G. (1999). *Learning in Graphical Model – Chapter: A view of the EM algorithm that justifies incremental, sparse, and other variants.*

Nieuwenhuis, S., Yeung, N., Holroyd, C., Schurger, A., and Cohen, J. (2004). Sensitivity of electrophysiological activity from medial frontal cortex to utilitarian and performance feedback. *Cereb Cortex*, 14(7):741–747.

Niki, H. and Watanabe, M. (1979). Prefrontal and cingulate unit activity during timing behavior in the monkey. *Brain Res.*, 171(2):213–224.

Nobre, A., Sebestyen, G., and Miniussi, C. (2000). The dynamics of shifting visuospatial attention revealed by event-related potentials. *Neuropsychologia.*, 38(7):964–974.

Norris, J. (1997). *Markov Chains.*

Nunez, P. (1995). Neocortical dynamics and human eeg rhythms. *Oxford University Press, New York*, pages 3–67.

Nunez, P., Srinivasan, R., Westdorp, A., Wijesinghe, R., Tucker, D., Silberstein, R., and Cadusch, P. (1997). Eeg coherency i: statistics, reference electrode, volume conduction, laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalogr Clin Neurophysiol*, 103:499–515.

Olshausen, B. and Field, D. (1996). Emergence of simple cells receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609.

O'Regan, J., Rensink, R., and Clark, J. (1999). Change-blindness as a result of 'mudsplashes'. *Nature*, 398:34.

Paninski, L., Fellows, M., Hatsopoulos, N., and Donoghue, J. (2004). Spatiotemporal tuning of motor cortical neurons for hand position and velocity. *J. Neurophysiol*, 91:515–532.

Panzeri, S., Treves, A., Schultz, S., and Rolls, E. (1999). On decoding the responses of a population of neurons from short time epochs. *Neural Computation*, 11:1553–1577.

Parker, D. (1982). Learning logic. Technical report, MIT Center for Computational Research in Economics and Mangement Science.

Passingham, R. (1993). *The frontal lobes and voluntary action.*

Pasupathy, A. and Connor, C. (1999). Responses to contour features in macaque area v4. *J Neurophysiol*, 82:2490–2502.

Pasupathy, A. and Connor, C. (2001). Shape representation in area v4: Position-specific tuning for boundary conformation. *Journal of Neurophysiology*, 86:2505–2519.

Pasupathy, A. and Connor, C. (2002). Population coding of shape in area v4. *Nature Neuroscience*, 5:1332–1338.

Pauli, W. (1933). *Handbuch der Physik*.

Paulus, K., Magnano, I., Piras, M., Solinas, M., Solinas, G., Sau, G., and Aiello, I. (2002). Visual and auditory event-related potentials in sporadic amyotrophic lateral sclerosis. *Clin Neurophysiol.*, 113(6):853–861.

Paus, T. (2001). Primate anterior cingulate cortex: where motor control, drive and cognition interface. *Nat Rev Neurosci.*, 2(6):417–424.

Pawelzik, K., Ernst, U., and Rotermund, D. (2004). Computing spike-by-spike. *Dynamic Perception Workshop*, pages 145–150.

Pawelzik, K., Ernst, U., and Rotermund, D. (2006a). On-line adaptation of neuro-prostheses with neuronal evaluation signals. *Proceedings ESANN'06*, pages 53–58.

Pawelzik, K., Herden, B., and Stieler, W. (2006b). Interview: Das ist nicht der zugang der physik. *Technology Review*, 1.

Pawelzik, K. and Rotermund, D. (2005). filed patent: PCT WO 002005094669 A1.

Pawelzik, K., Rotermund, D., and Ernst, U. (2003). Building representations spike by spike. *Proceedings of the 29th Göttingen Neurobiology Conference*.

Pawelzik, K., Rotermund, D., and Ernst, U. (2006c). Towards on-line adaptation of neuro-prostheses with neuronal evaluation signals. *Neuroscience Meeting Planner 2006*, 13.13.

Pawelzik, K., Rotermund, D., Taylor, K., Ernst, U., and Kreiter, A. (2006d). Attention improves object encoding in monkey area v4. *FENS 2006, 9915010554*.

Pearce, J. and Hall, G. (1980). A model for pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychol Rev.*, 87(6):532–552.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*.

Pearl, J. (2000). Causality: Models, reasoning, and inference.

Pearl, J. and Russell, S. (2003). *The Handbook of Brain Theory and Neural Networks – Chapter: Bayesian Networks*.

Pearlmutter, B. (1990). Dynamic recurrent neural networks. Technical report.

Penfield, W. and Rasmussen, T. (1950). *The cerebral cortex of man*.

Perrin, F., Bertrand, O., and Pernier, J. (1987). Scalp current density mapping: value and estimation from potential data. *IEEE Trans Biomed Eng*, 34:283–288.

Perrinet, L., Samuelides, M., and Thorpe, S. (2004). Coding static natural images using spiking event times: Do neurons cooperate. *IEEE Trans. Neur. Networks*, 15:1164–1175.

Petersen, R., Panzeri, S., and Diamond, M. (2002). Population coding in somatosensory cortex. *Curr Opin Neurobiol.*, 12(4):441–447.

Pfurtscheller, G. and Neuper, C. (2001). Motor imagery and direct brain-computer communication. *Neural engineering: merging engineering and neuroscience*, 89(7):1123–1134.

Picard, N. and Strick, P. (1996). Motor areas of the medial wall: a review of their location and functional activation. *Cereb Cortex.*, 6(3):342–353.

Pinsk, M., Doniger, G., and Kastner, S. (2004). Push-pull mechanism of selective attention in human extrastriate cortex. *J Neurophysiol*, 92:622–629.

Platt, M. and Glimcher, P. (1999). Neural correlates of decision variables in parietal cortex. *Nature*, 400:233–238.

Pollen, D. (1999). On the neural correlates of visual perception. *Cerebral Cortex*, 9(1):4–19.

Pouget, A., Dayan, P., and Zemel, R. S. (2003). Inference and computation with population codes. *Annual Reviews Neuroscience*, 26:381–410.

Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, 435:1102–1107.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rao, C. (1946). Information and the accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37:81–91.

Rao, R. P. (2004). Bayesian computation in recurrent neural circuits. *Neural Computation*, 16:1–38.

Rescorla, R. and Wagner, A. (1972). *Classical conditioning II: Current research and theory Chapter: A theory of classical conditioning: variations in the effectiveness of reinforcement and non-reinforcement*.

Reynolds, J. and Chelazzi, L. (2004). Attentional modulation of visual processing. *Annual Review of Neuroscience*, 27:611–647.

Reynolds, J., Chelazzi, L., and Desimone, R. (1999). Competitive mechanisms subserve attention in macaque areas v2 and v4. *J. Neurosci.*, 19:1736–1753.

Reynolds, J. and Desimone, R. (2003). Interacting roles of attention and visual salience in v4. *Neuron*, 37(5):853–863.

Reynolds, J., Pasternak, T., and Desimone, R. (2000). Attention increases sensitivity of v4 neurons. *Neuron*, 26(3):703–714.

Richardson, W. (1972). Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62:55–59.

Rickert, J., de Oliveira, S. C., Vaadia, E., Aertsen, A., Rotter, S., and Mehring, C. (2005). Encoding of movement direction in different frequency ranges of motor cortical local field potentials. *J Neurosci.*, 25(39):8815–8824.

Ridderinkhof, K., Ullsperger, M., Crone, E., and Nieuwenhuis, S. (2004a). The role of the medial frontal cortex in cognitive control. *Science*, 306:443–447.

Ridderinkhof, K., van den Wildenberg, W., Segalowitz, S., and Carter, C. (2004b). Neurocognitive mechanisms of cognitive control: The role of prefrontal cortex in action selection, response inhibition, performance monitoring, and reward-based learning. *Brain and Cognition*, 56:129–140.

Ringach, D. (2002). Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *J Neurophysiol*, 88:455–463.

Ripley, B. (1987). *Stochastic simulation*.

Rizzolatti, G. and Wolpert, D. (2005). Motor systems. *Curr Opin Neurobiol.*, 15(6):623–625.

Robbe, D., Montgomery, S., Thome, A., Rueda-Orozco, P., McNaughton, B., and Buzsaki, G. (2006). Cannabinoids reveal importance of spike timing coordination in hippocampal function. *Nat Neurosci*, 9:1526–1533.

Rock, I. and Gutman, D. (1981). The effect of inattention on form perception. journal of experimental psychology. *Human perception and performance*, 7:275–285.

Rock, I., Linnett, C., Grant, P., and Mack, A. (1992). Perception without attention: results of a new method. *Cognit. Psychol.*, 24:502–534.

Roesch, M. and Olson, C. (2003). Impact of expected reward on neuronal activity in prefrontal cortex, frontal and supplementary eye fields and premotor cortex. *J. Neurophysiol.*, 90:1766–1789.

Roland, P. and Zilles, K. (1996). Functions and structures of the motor cortices in humans. *Curr Opin Neurobiol.*, 6(6):773–781.

Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Reviews*, 65:386–408.

Rotermund, D., Ernst, U., and Pawelzik, K. (2005). Processing natural images with single spikes. *Proceedings of the 6th Meeting of the German Neuroscience Society/30th Göttingen Neurobiology Conference*, 445A.

Rotermund, D., Ernst, U., and Pawelzik, K. (2006a). Towards on-line adaptation of neuro-prostheses with neuronal evaluation signals. *Biological Cybernetics*, 95:243–257.

Rotermund, D., Taylor, K., Ernst, U., Kreiter, A., and Pawelzik, K. R. (2007a). Attention improves object-specificity of cortical states. *submitted to PLoS Biology*.

Rotermund, D., Taylor, K., Ernst, U., Pawelzik, K., , and Kreiter, A. (2007b). Attention improves object representation in monkey area v4. *Proceedings of the 7th Meeting of the German Neuroscience Society/31th Göttingen Neurobiology Conference*, T32-1A.

Rotermund, D., Taylor, K., Ernst, U., Pawelzik, K., and Kreiter, A. (2006b). Attention improves object discriminability in monkey area v4. *CNS, S2*.

Rotermund, D., Taylor, K., Ernst, U., Pawelzik, K., and Kreiter, A. (2007c). Attention in monkey area v4 is modulated by task demand. *submitted to Neuroscience*.

Rotermunda, D., Taylor, K., Ernst, U., Pawelzik, K., and Kreiter, A. (2005). Attention improves object encoding in monkey area v4. *Society for Neuroscience Abstracts*, 31:591.6.

Rumelhart, D., Hinton, E., and Williams, R. (1986a). Learning representations by back-propogating errors. *Nature*, 323:533–536.

Rumelhart, D., Hinton, G., and Williams, R. (1986b). *Parallel Distributed Processing, Vol. 1 – Chapter: Learning internal representations by error propagation*.

Rushworth, M., Walton, M., Kennerley, S., and Bannerman, D. (2004). Action sets and decisions in the medial frontal cortex. *Trends Cogn Sci.*, 8(9):410–417.

Sahu, S. and Roberts, G. (1999). On convergence of the em algorithmand the gibbs sampler. *Statistics and Computing archive*, 9(1):55–64.

Sakmann, B. and Neher, E. (1984). Patch clamp techniques for studying ionic channels in excitable membranes. *Annual Review of Physiology*, 455–472:46.

Salinas, E. and Abbott, L. (1994). Vector reconstruction from firing rates. *J. Computational Neurosci.*, 1:89–107.

Sanes, J. and Donoghue, J. (2000). Plasticity and primary motor cortex. *Annu Rev Neurosci.*, 23:393–415.

Satoh, T., Nakai, S., and Kimura, M. (2003). Correlated coding of motivation and outcome of decision by dopamine neurons. *The Journal of Neuroscience*, 23(30):9913–9923.

Schaal, S. (2002). *The handbook of brain theory and neural networks - Chapter: Arm and hand movement control*.

Schaal, S. and Sternad, D. (2001). Origins and violations of the 2/3 power law in rhythmic 3d movements. *Experimental Brain Research*, 136:60–72.

Schalk, G., Wolpaw, J., McFarland, D., and Pfurtscheller, G. (2000). Eeg-based communication: presence of an error potential. *Clin Neurophysiol*, 111(12):2138–2144.

Schein, S. and Desimone, R. (1990). Spectral properties of v4 neurons in the macaque. *J Neurosci.*, 3369–3389:10(10).

Scherberger, H., Jarvis, M., and Andersen, R. (2005). Cortical local field potential encodes movement intentions in the posterior parietal cortex. *Neuron*, 46(2):347–354.

Schölkopf, B., Burges, C., and Vapnik, V. (1995). Extracting support data for a given task. *Proceedings, First International Conference on Knowledge Discovery & Data Mining*, 1:252–257.

Schölkopf, B. and Smola, A. (2001). *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond*.

Schölkopf, B., Smola, A., Williamson, R., and Bartelett, P. (2000). New support vector algorithms. *Neural Comp*, 12:1207–1245.

Schultz, W. (1998). Predictive reward signal of dopamine neurons. *J. Neurophysiol.*, 80:1–27.

Schultz, W. (2004). Neural coding of basic reward terms of animal learning theory, game theory, microeconomics and behavioural ecology. *Current Opinion in Neurobiology*, 14:139–147.

Schultz, W., Dayan, P., and Montague, P. (1997). A neural substrate of prediction and reward. *Science*, 275:1593–1599.

Schulzke, E. (2006). *Neuronale Kodierung und Dekodierung von multiplen und dynamischen Reizen*. PhD thesis, University of Bremen.

Schwartz, A. (2004). Cortical neural prosthetics. *Annu. Rev. Neurosci.*, 27:487–507.

Schwartz, A., Taylor, D., and Tillery, S. H. (2001). Extraction algorithms for cortical control of arm prosthetics. *Current Opinion in Neurobiology*, 11:701–707.

Segev, I. and Rall, W. (1998). Excitable dendrites and spines: earlier theoretical insights elucidate recent direct observations. *Trends Neurosci.*, 21:453–460.

Seiple, W., Clemens, C., Greenstein, V., Holopigian, K., and Zhang, X. (2002). The spatial distribution of selective attention assessed using the multifocal visual evoked potential. *Vision Res.*, 42(12):1513–1521.

Sejnowski, T. and Tesauro, G. (1989). The hebb rule for synaptic plasticity: algorithms and implementations. *Neural Models of Plasticity*, Chapter 6:94–103.

Serruya, M. and Donoghue, J. (2004). *Neuroprosthetics: Theory and Practice – Chapter: Design Principles of a Neuromotor Prosthetic Device.*

Seung, H. and Sompolinsky, H. (1993). Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences*, 90:10749–10753.

Shadlen, M. and Newsome, W. (1998). The variable discharge of cortical neurons: implications for connectivity, computation and information coding. *J. Neurosci.*, 18(10):3870–3896.

Shafer, G. and Shenoy, P. (1998). Local computation in hypertrees. Technical report, University of Kansas.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656.

Shidara, M., Aigner, T., and Richmond, B. (1998). Neuronal signals in the monkey ventral striatum related to progress through a predictable series of trials. *The Journal of Neuroscience*, 18(7):2613–2625.

Shima, K. and Tanji, J. (1998). Role for cingulate motor area cells in voluntary movement selection based on reward. *Science*, 282:1335–1338.

Shirow, M. (1991). *Ghost in the Shell.*

Silberberg, G., Bethge, M., Markram, H., Pawelzik, K., and Tsodyks, M. (2004). Dynamics of population rate codes in ensembles of neocortical neurons. *Journal of Neurophysiology*, 91:704–709.

Simons, D. and Rensink, R. (2005). Change blindness: past, present, and future. *Trends Cogn Sci.*, 9(1):16–20.

Singer, W. and Gray, C. (1995). Visual feature integration and the temporal correlation hypothesis. *Ann. Rev. Neurosci.*, 18:555–586.

Skottun, B., Bradley, A., Sclar, G., Ohzawa, I., and Freeman, R. (1987). The effects of contrast on visual orientation and spatial frequency discrimination: A comparison of single cells and behaviour. *J. Neurophysiol.*, 57:773–786.

Smith, J. (2001). Some observations on the concepts of information-theoretic entropy and randomness. *Entropy*, 3:1–11.

Smiyukha, Y., Mandon, S., Galashan, F., Neitzel, S., and Kreiter, A. (2006). Attention-dependent switching of interareal synchronization between v4 neurons and different subpopulations of their v1 afferents. *Soc Neurosci Abstr*, 32:11.2.

Snowden, R., Treue, S., and Andersen, R. (1992). The response of neurons in areas v1 and mt of the alert rhesus monkey to moving random dot patterns. *Experimental Brain Research*, 88:389–400.

Spencer, K., Dien, J., and Donchin, E. (2001). Spatiotemporal analysis of the late erp responses to deviant stimuli. *Psychophysiology.*, 38(2):343–358.

Steinmetz, P., Roy, A., Fitzgerald, P., Hsiao, S., Johnson, K., and Niebur, E. (2000). Attention modulates synchronized neuronal firing in primate somatosensory cortex. *Nature*, 404:187–190.

Sternad, D. and Schaal, S. (1999). Segmentation of endpoint trajectories does not imply segmented control. *Experimental Brain Research*, 124(1):118–136.

Stuphorn, V., Bauswein, E., and Hoffmann, K.-P. (2000). Neurons in the primate superior colliculus coding for arm movements in gaze-related coordinates. *J. Neurophysiol.*, 83:1283–1299.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, S., Braren, M., Zublin, J., and John, E. (1965). Evoked potential correlates of stimulus uncertainty. *Science*, 150:1187–1188.

Taira, M., Boline, J., Smyrnis, N., Georgopoulos, A., and Ashe, J. (1996). On the relations between single cell activity in the motor cortex and the direction and magnitude of three-dimensional static isometric force. *Experimental Brain Research*, 109(2):367–376.

Tanji, J. and Shima, K. (1994). Role for supplementary motor area cells in planning several movements ahead. *Nature*, 371:413–416.

Taylor, D., Tillery, S. H., and Schwartz, A. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, 296:1829–1832.

Taylor, K., Mandon, S., Freiwald, W., and Kreiter, A. (2005). Coherent oscillatory activity in monkey area v4 predicts successful allocation of attention. *Cereb Cortex*, 15:1424–1437.

Teder-Salejarvi, W., Munte, T., Sperlich, F., and Hillyard, S. (1999). Intra-modal and cross-modal spatial attention to auditory and visual stimuli. an event-related brain potential study. *Brain Res Cogn Brain Res.*, 8(3):327–343.

Thoroughman, K. and Shadmehr, R. (2000). Learning of action through adaptive combination of motor primitives. *Nature*, 407:742–747.

Thorpe, S., Delorme, A., and van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks*, 14:715–725.

Thorpe, S. J., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381:520–522.

Tillery, S. H., Taylor, D., and Schwartz, A. (2003). Training in cortical control of neuroprosthetic devices improves signal extraction from small neuronal ensembles. *Rev. Neurosci.*, 14(1-2):107–119.

Tobler, P., Fiorillo, C., and Schultz, W. (2005). Adaptive coding of reward value by dopamine neurons. *Science*, 307:1642–1645.

Todorov, E. (2000). Direct cortical control of muscle activation in voluntary arm movements: a model. *Nat Neurosci.*, 3(4):391–398.

Tolhurst, D., Movshon, J., and Dean, A. (1983). The statistical reliability of signals in single neurons in cat and monkey striate cortex. *Vision Research*, 23:775–785.

Tolias, A., Keliris, G., Smirnakis, S., and Logothetis, N. (2005). Neurons in macaque area v4 acquire directional tuning after adaptation to motion stimuli. *Nature Neuroscience*, 8(5):591–593.

Tomko, G. and Crapper, D. (1974). Neuronal variability: non-stationary responses to identical visual stimuli. *Brain Res.*, 79(3):405–18.

Tootell, R., Hadjikhani, N., Hall, E., Marrett, S., Vanduffel, W., Vaughan, J., and Dale, A. (1998). The retinotopy of visual spatial attention. *Neuron*, 21:1409–1422.

Treisman, A. and Gelade, G. (1980). A feature-integration theory of attention. *Cognit Psychol.*, 12(1):97–136.

Tremblay, L. and Schultz, W. (1999). Relative reward preference in primate orbitofrontal cortex. *Nature*, 398:704–708.

Tremblay, L. and Schultz, W. (2000). Reward-related neuronal activity during Go-Nogo task performance in primate orbitofrontal cortex. *J. Neurophysiol.*, 83:1864–1876.

Tresch, M., Saltiel, P., and Bizzi, E. (1999). The construction of movement by the spinal cord. *Nature Neuroscience*, 2:162–167.

Treue, S. (2001). Neural correlates of attention in primate visual cortex. *TRENDS in Neurosciences*, 24(5):295–300.

Treue, S. and Maunsell, J. (1996). Attentional modulation of visual motion processing in cortical areas mt and mst. *Nature*, 382:539–541.

Treue, S. and Trujillo, J. M. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, 399:575–579.

Tuckwell, H. (1979). Synaptic transmission in a model for stochastic neural activity. *J Theor Biol.*, 77(1):65–81.

Tuckwell, H. (1988). *Introduction to theoretical neurobiology.*

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. minimum torque-change model. *Biol Cybern.*, 61(2):89–101.

Usrey, W., Alonso, J.-M., and Reid, R. (2000). Synaptic interactions between thalamic inputs to simple cells in cat visual cortex. *J. Neurosci.*, 20:561–567.

van Rullen, R. and Koch, C. (2005). *Handbook of clinical neurophysiology Chapter: Visual Attention and Visual Awareness.*

van Rullen, R. and Thorpe, S. J. (2001). Rate coding versus temporal order coding: What the retinal ganglion cell tells the visual cortex. *Neural Computation*, 13:1255–1283.

van Rullen, R. and Thorpe, S. J. (2002). Surfing a spike wave down the ventral stream. *Vision Research*, 42:2593–2615.

van Schie, H., Mars, R., Coles, M., and Bekkering, H. (2004). Modulation of activity in medial frontal and motor cortices during error observation. *Nature Neuroscience*, 7:549–554.

van Veen, V., Holroyd, C., Cohen, J., Stenger, V., and Carter, S. (2004). Errors without conflict: Implications for performance monitoring theories of anterior cingulate cortex. *Brain and Cognition*, 56:267–276.

Vanni, S., Portin, K., Virsu, V., and Hari, R. (1999). Mu rhythm modulation during changes of visual percepts. *Neuroscience.*, 91(1):21–31.

Villringer, A. and Chance, B. (1997). Non-invasive optical spectroscopy and imaging of human brain function. *TINS*, 20(10):435–442.

Viviani, P. and Flash, T. (1995). Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. *J Exp Psychol Hum Percept Perform.*, 21(1):32–53.

von der Malsburg, C. (1985). Nervous structures with dynamical links. *Ber. Bunsenges. Phys. Chem.*, 89:703–710.

von Helmholtz, H. (1860). *Handbuch der Physiologischen Optik - Band II.*

Walsh, B. (2002). Markov chain monte carlo and gibbs sampling. Technical report, Lecture Notes for EEB 596z.

Walton, M., Bannerman, D., Alterescu, K., and Rushworth, M. (2003). Functional specialization within medial frontal cortex of the anterior cingulate for evaluating effort-related decisions. *J Neurosci.*, 23(16):6475–6479.

Warland, D., Reinagel, P., and Meister, M. (1997). Decoding visual information from a population of retinal ganglion cells. *J. Neurophysiol*, 78:2336–2350.

Warzecha, A. and Egelhaaf, M. (2000). Response latency of a motion-sensitive neuron in the fly visual system: dependence on stimulus parameters and physiological conditions. *Vision Res.*, 40(21):2973–2983.

Watanabe, M., Hikosaka, K., Sakagami, M., and Shirakawa, S.-I. (2002a). Coding and monitoring of motivational context in the primate prefrontal cortex. *The Journal of Neuroscience*, 22(6):2391–2400.

Watanabe, M., Tanaka, H., Uka, T., and Fujita, I. (2002b). Disparity-selective neurons in area v4 of macaque monkeys. *The Journal of Neurophysiology*, 87(4):1960–1973.

Weiskopf, N., Mathiak, K., Bock, S., Scharnowski, F., Veit, R., Grodd, W., Goebel, R., and Birbaumer, N. (2004). Principles of a brain-computer interface (bci) based on real-time functional magnetic resonance imaging (fmri). *IEEE Trans Biomed Eng.*, 51(6):966–970.

Welch, G. and Bishop, G. (2004). An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill.

Welling, M. and Hinton, G. (2002). A new learning algorithm for mean field boltzmann machines. *Lecture Notes In Computer Science*, 2415:351–357.

Werbos, P. (1974). *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University.

Wessberg, J., Stambaugh, C., Kralik, J., Beck, P., Laubach, M., Chapin, J., Kim, J., Biggs, S., Srinivasan, M., and Nicolelis, M. (408). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 2000:361–365.

Wilke, S. and Eurich, C. (2002). On the functional role of noise correlations in the nervous system. *Neurocomputing*, 44–46:1023–1028.

Wilkinson, F., James, T., Wilson, H., Gati, J., Menon, R., and Goodale, M. (2000). An fmri study of the selective activation of human extrastriate form vision areas by radial and concentric gratings. *Current Biology*, 10(22):1455–1458.

Wilson, H. R. and Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Kybernetik*, 12:1–24.

Wise, S. (1985). The primate premotor cortex: past, present, and preparatory. *Annu Rev Neurosci.*, 8:1–19.

Wolfe, J. (1998). Attention - chapter: Visual search.

Wolfe, J. and Bennett, S. (1997). Preattentive object files: shapeless bundles of basic features. *Vision Res.*, 37:25–43.

Wolpaw, J., Birbaumer, N., Heetderks, W., McFarland, D., Peckhamand, P., Schalk, G., Donchin, E., Quatrano, L., Robinson, C., and Vaughan, T. (2000). Brain-computer interface technology: a review of the first international meeting. *IEEE Trans. Rehab. Eng.*, 8(2):164–173.

Wolpaw, J., Birbaumer, N., McFarland, D., Pfurtscheller, G., and Vaughan, T. (2002). Brain-computer interfaces for communication and control. *Clin. Neurophys.*, 113:767–791.

Wolpaw, J. and McFarland, D. (2004a). Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proc Natl Acad Sci*, 101(51):17849–17854.

Wolpaw, J. and McFarland, D. (2004b). Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *PNAS*, 101(51):17849–17854.

Wolpert, D. and Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience*, 3:1212–1217.

Womelsdorf, T., Anton-Erxleben, K., Pieper, F., and Treue, S. (2006). Dynamic shifts of visual receptive fields in cortical area mt by spatial attention. *Nat Neurosci*, 9:1156–1160.

Woolsey, C., Settlage, P., Meyer, D., Sencer, W., Hamuy, T. P., and Travis, A. (1952). Patterns of localization in precentral and supplementary motor areas and their relation to the concept of a premotor area. *Res Publ Assoc Res Nerv Ment Dis.*, 30:238–264.

Wörgötter, F. and Porr, B. (2005). Temporal sequence learning, prediction and control - a review of different models and their relation to biological mechanisms. *Neural Comp.*, 17:245–319.

Xie, X. and Seung, H. (2004a). Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, 69:041909.

Xie, X. and Seung, H. S. (2004b). Learning in neural networks by reinforcement of irregular spiking. *Phys Rev E Stat Nonlin Soft Matter Phys.*, 69(4 Pt 1):041909.

Xu, W., Guan, C., Siong, C., Ranganatha, S., Thulasidas, M., and Wu, J. (2004). High accuracy classification of eeg signal. *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)*, 2:391–394.

Xu, X., Ichida, J., Allison, J., Boyd, J., Bonds, A., and Casagrande, V. (2001). A comparison of koniocellular, magnocellular and parvocellular receptive field properties in the lateral geniculate nucleus of the owl monkey (aotus trivirgatus). *The Journal of Physiology*, 531(1):203–218.

Yantis, S. (1998). *Attention - Chapter: Control of visual attention.*

Yeung, N., Botvinick, M., and Cohen, J. (2004a). The neural basis of error detection: Conflict monitoring and the error-related negativity. *Psychological Review*, 111(4):931–959.

Yeung, N., Cohen, J., and Botvinick, M. (2004b). The neural basis of error detection: conflict monitoring and the error-related negativity. *Psychol Rev.*, 111(4):931–959.

Yoo, S., Fairneny, T., Chen, N., Choo, S., Panych, L., Park, H., Lee, S., and Jolesz, F. (2004). Brain-computer interface using fmri: spatial navigation by thoughts. *Neuroreport.*, 15(10):1591–1595.

Young, T. (1802). On the theory of light and colors. *Philosophical Transactions of the Royal Society*, 91:12–49.

Zeki, S. (1993). *A vision of the brain.*

Zhang, N. and Poole, D. (1994). A simple approach to bayesian network computations. *Proceedings of the Tenth Biennial Canadian Artificial Intelligence Conference*, AI-94:171–178.

Zipser, K., Lamme, V., and Schiller, P. (1996). Contextual modulation in primary visual cortex. *The Journal of Neuroscience*, 16(22):7376–7389.

Zrenner, E. (2002). Will retinal implants restore vision? *Science*, 295(5557):1022–1025.

Publications

Articles

- D. Rotermund, K. Taylor, U.A. Ernst, A.K. Kreiter, and K.R. Pawelzik, *Attention improves object-specificity of cortical states*, submitted to PLoS Biology (2007).
- U. Ernst, D. Rotermund, and K. Pawelzik, *Efficient computation based on stochastic spikes*, Neural Computation 19 (5), 1313-1343 (2007).
- D. Rotermund, U. Ernst, and K. Pawelzik, *Towards on-line adaptation of neuroprosthesis with neuronal evaluation signals*, Biological Cybernetics 95 (3), 243-257 (2006).
- M. Bethge, D. Rotermund, and K. Pawelzik, *A second order phase transition in neural rate coding: Binary encoding is optimal for rapid signal transmission*, Phys. Rev. Lett. 90 (8), 088104-1 (2003).
- M. Bethge, D. Rotermund, and K. Pawelzik, *Optimal neural rate coding leads to bimodal firing rate distributions*, Network: Comput. Neural Syst. 14, 303-319 (2003).
- M. Bethge, D. Rotermund, and K. Pawelzik, *Binary tuning is optimal for neural rate coding with high temporal resolution*, Advances in Neural Information Processing Systems 15 15, S. Becker, S. Thrun, K. Obermayer (ed.), MIT Press, 189-196 (2003).
- M. Bethge, D. Rotermund, and K. Pawelzik, *Optimal short-term population coding: When Fisher information fails*, Neural Computation 14(10), 2317-2351 (2002).

Filed Patents

- K. Pawelzik, and D. Rotermund, *System und in ein Gewebe von Lebewesen implantierbare Vorrichtung zur Erfassung und Beinflussung von elektrischer Bio-Aktivität*, in: DE102004014694A1, WO002005094669A1, (2004).

Other publications

- D. Rotermund, K. Taylor, U.A. Ernst, K.R. Pawelzik, and A.K. Kreiter, *Attention in monkey area V4 is modulated by task demand*, in: submitted to Neuroscience (2007).
- D. Rotermund, K. Taylor, U.A. Ernst, K.R. Pawelzik, and A.K. Kreiter, *Attention improves object representation in monkey area V4*, in: Proceedings of the 7th Meeting of the German Neuroscience Society/31th Göttingen Neurobiology Conference, T32-1A (2007).
- K. Pawelzik, U. Ernst, and D. Rotermund, *On-line adaptation of neuro-prostheses with neuronal evaluation signals*, in: Proceedings ESANN'06, Michel Verleysen (ed.), d-side, Evere, Belgium, 53-58 (2006).
- K. Pawelzik, D. Rotermund, K. Taylor, U. Ernst, and A. Kreiter, *Attention improves object encoding in monkey area V4*, in: FENS 2006, 9915010554 (2006).
- K. Pawelzik, D. Rotermund, and U. Ernst, *towards on-line adaptation of neuro-prostheses with neuronal evaluation signals*, in: Neuroscience Meeting Planner 2006, 13.13 (2006).
- D. Rotermund, K. Taylor, U. Ernst, K. Pawelzik, and A. Kreiter, *Attention improves object discriminability in monkey area V4*, in: CNS, S2 (2006).
- D. Rotermund, U. Ernst, and K. Pawelzik,*Processing natural images with single spikes*, in: Proceedings of the 6th Meeting of the German Neuroscience Society/30th Göttingen Neurobiology Conference, 445A (2005).
- D. Rotermund, K. Taylor, U. Ernst, K. Pawelzik, and Andreas Kreiter, *Attention improves object encoding in monkey area V4*, in: Society for Neuroscience Abstracts 31, 591.6 (2005).
- U. Ernst, D. Rotermund, and K. Pawelzik, *An algorithm for fast pattern recognition with random spikes*, in: 26th DAGM Symposium Proceedings, Rasmussen, C.E. et al. (ed.), Springer-Verlag, 399-406 (2004).
- K. Pawelzik, U. Ernst, and D. Rotermund, *Computing spike-by-spike*, in: Dynamic Perception Workshop, Uwe J. Ilg, Heinrich H. Bülthoff, and Hanspeter A. Mallot (eds.), IOS Press, 145-150 (2004).
- K. Pawelzik, D. Rotermund, and U. Ernst, *Building representations spike by spike*, in: Proceedings of the 29th Göttingen Neurobiology Conference, N. Elsner and H. Zimmermann (eds.), Georg Thieme Verlag, Stuttgart, 1041 (2003).
- M. Bethge, D. Rotermund, and K. Pawelzik, *Optimal neural population coding and Fisher information*, in: Verhandlungen der Deutschen Physikalischen Gesellschaft DPG (VI) 37, 1, V. Häselbarth (ed.), Physik-Verlag GmbH, 509 (2002).

- K. R. Pawelzik, U. A. Ernst, D. Trenner, and D. Rotermund, *Building representations spike by spike*, in: Society of Neuroscience Conference 2002, Orlando, 557.12 (2002).

- M. Bethge, D. Rotermund, and K. Pawelzik, *Optimal short-term population coding: When Fisher information fails*, in: Proceedings of the 28th Göttingen Neurobiology Conference, N. Elsner and G. W. Kreutzberg (eds.), Georg Thieme Verlag, Stuttgart, 250 (2001).

Acknowledgment / Danksagung

Am Ende dieses Werkes, mit dem ich mich über 5.5 Jahre beschäftigt habe, möchte ich gerne einigen Menschen danken, die mich bei meiner Arbeit unterstützt und begleitet haben.

Allen voran möchte ich mich bei Klaus Pawelzik bedanken, dass er mir die Möglichkeit bei ihm zu promovieren gegeben und mich stets unterstützt hat. Ohne ihn, als übersprudelnden und nie versiegendem Quell neuer Ideen, wäre meine Arbeit um ein vieles langweiliger. Er hat mir eine enorme Freiheit bei meiner Forschung gelassen und war bei Problemen immer zur Stelle. Auch möchte ich Andreas Kreiter danken, dass er mich an seine kostbaren Datensätze herangelassen hat, bei den Diskussionen über die daraus hervorgegangen Ergebnisse den Finger immer wieder in die Wunde gelegt und trotz extrem engem Zeitplan sich die Zeit genommen hat, dieses doch recht umfangreiche Werk zu begutachten.

Besonders inspirierend war für mich die Zusammenarbeit mit Matthias Bethge und Udo Ernst. Es ist wirklich schade, das es in den letzten Jahren nicht zu einer Weiterführung der gemeinsamen Forschung mit Matthias Bethge gekommen ist. Unsere Zusammenarbeit war immer hochgradig 'optimal' und für mich immer sehr lehrreich. Mit Udo Ernst ist es immer möglich die kompliziertesten Probleme anzugehen und oft auch zu lösen, obwohl uns doch das eine oder andere Adventure genötigt hat, in dem entsprechenden Lösungsbuch nachzuschauen.

Ich möchte auch Udo Ernst und Katja Taylor für das Korrekturlesen meiner Arbeit danken. Ohne eure Hilfe hätte bestimmt niemand verstanden was ich mit der Ansammlung von Wörtern sagen wollte. Agnes Janßens Hilfe bei Verwaltungs- und Koordinationsdingen war und ist absolut unverzichtbar.

Wenn man sich immer wieder durch neue und zusätzliche Projekte von der Fertigstellung der eigenen Doktorarbeit abhalten lässt, bleibt es nicht aus dass man viele Mitglieder in der Arbeitsgruppe kommen und gehen sieht. Ich möchte den folgenden Personen für die schöne Atmosphäre, die interessanten Gespräche und die guten Ratschlägen (die ich oft ignoriert haben, was wiederum die Bearbeitungszeit weiter verlängert hat) danken: Christian Eurich, Rolf Henkel, Onno Böhler, Prof. Helmut Schwegler, Andreas Beuthner, Dieter Gauck, Bailu Si, Nadja Schinkel, Erich Schulzke, David Engelskirchen, Daniel Bartz, Felix Patzelt, Markus Riegel, Ulrich Cherdron, Ste-

fan Liehr, Roland Rothenstein, Stefan Wilke, Klaus Bowe, Axel Etzold, Klaus Franke, Pit Hankel, Andreas Thiel, Dennis Trenner, Ronald Bormann, Frank Emmert-Streib und Aladin Mirmohammedhosseini.

Ich freue mich schon auf die Weiterführung der interessanten und spannenden Kooperation mit Sunita Mandon, Andreas Kreiter, Katja Taylor und Kerstin Schwabe.

Außerdem möchte ich David Rotermund für seinen unermüdlichen Einsatz bei der Installation und Wartung der Arbeitsgruppen-Computer danken, ohne diese zeitraubende Hilfe wäre diese Doktorarbeit nicht möglich gewesen (ist doch wahr!).

Ich hoffe, dass alle die Leute, die ich immer wieder vertrösten musste, weil ich mal wieder keine Zeit hatte, da ich an der Doktorarbeit arbeiten musste, mir irgendwann verzeihen werden.

Und ohne das Herumgekaspere mit Michael Plura hätte ich bestimmt in vielen der nicht so glücklichen Zeiten den letzten Rest meines Verstandes verloren.

Ohne den Beistand meiner Familie wäre es mir nicht möglich gewesen, mein Studium zu beenden oder gar zu promovieren. Ich möchte besonders meiner Mutter danken, die mich auf meinen langen Weg immer unterstützt hat. Wir haben uns gemeinsam durch schwere und unangenehme Zeiten gekämpft. Ohne die Unterstützung durch meinen Großvater, Wilhelm Rotermund, wäre ich bestimmt verzweifelt. Und ich möchte an meinen verstorbenen Vater erinnern, ohne dessen außergewöhnliche Unterstützung ich vermutlich nie studiert hätte.

Lebenslauf

	Name: David Rotermund
14.03.1976	Geboren in Delmenhorst als Sohn von Wilfried Rotermund und Ursula Rotermund.
1983 - 1986	Besuch der Bernard - Rein - Schule, Delmehorst (Grundschule)
1986 - 1988	Besuch des Pestalozzi - Schulzentrums, Delmenhorst (Orientierungsstufe)
1988 - 1992	Besuch der Realschule an der Lilienstrasse, Delmenhorst (Realschule)
1992 - 1994	Ausbildung zum chemisch-technischen Assistenten an der Berufsfachschule für Assistentenberufe, Bremen
1994 - 1995	Fachhochschulreife Biologie/ Chemie an der Fachoberschule, Bremen
1994 - 2004	Gewerbe im Bereich: Erstellung, Wartung und Verkauf von Hard/Software, Dienstleistungen im Bereich Computer, u.a.
1995 - 1996	Studium der Elektrotechnik an der Hochschule Bremen (Vordiplom in Elektrotechnik)
1996 - 2002	Studium der Physik an der Universität Bremen (Diplom in Physik)
2002 - 2007	Doktorand bei Prof. Dr. K. Pawelzik an der Universität Bremen (Mitarbeiter im BMBF Projekt Deutsch-Israelische-Projektkooperation (DIP) - Models and Experiments towards Adaptive Control of Motor Prostheses (METACOMP), Zentrum für Kognitionswissenschaften (ZKW) und Sonderforschungsbereich *Neurokognition* (SFB 517) der DFG)