The University of Tokyo

Thesis

Learning Rules for Data Representation with Dynamical Neural Systems

Michael Gutmann

Spring 2008

Abstract

Neural integration, and the neural representation of sensory input, remains an incompletely understood topic in neuroscience. We show here first that the theoretical approach of modeling neural integration by means of data representation allowed to gain some insight into the principles and mechanisms of neural integration. Data representation is a mathematical method to find a change of basis for the input which matches the structure of the data.

The encoding transforms which are in existing data representation methods used to calculate the new representation of the input with respect to the new basis did however either not include a time structure, or they were acausal. If neural integration is modeled by means of data representation, the input should be encoded into neural activity by means of a valid neuron model. We point then out that the encoding transforms of existing data representation methods correspond to rather abstract neuron models, and that the modeling of neural integration by means of data representation could be improved by using dynamical neural systems for the encoding.

We derive then learning rules for data representation with dynamical neural systems. We focus on neural systems which are composed of integrators. The integrators signal by means of spikes. We use two models for the integrators. In a first model, the neuron model is a formal spiking neuron in the sense that the action potentials are not modeled. In a second model, the neuron model is a real spiking neuron in the sense that the upstroke of the action potential is also modeled. The derived learning rules are for data representation with a single neuron, and in the case of the formal spiking neuron, also for data representation with a population of neurons. In that case, the neurons may have lateral connections.

The application of the learning rules to represent an input signal by means of a population of neurons illustrates the usefulness of the derived learning rules for the study of neural integration and representation: After learning, the input was accurately represented by the population of neurons, and the learning rules led to self-organized neural differentiation.

Acknowledgments

This thesis was prepared at the Aihara-Suzuki-Kohno laboratory of the University of Tokyo. I thank Professor Aihara for hosting me, and all the professors of the laboratory for their support and for providing an excellent infrastructure. The work was financially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) through scholarship No. 040680. Thank goes also to all the members of the laboratory, especially Hiroyasu Andoh, who often kindly helped me over the language barrier. I like further to thank Aapo Hyvärinen for his ongoing support, and my parents for having always encouraged scientific thinking.

Contents

Chapter 1

Introduction

Motivation

The motivation for this thesis lies both in mathematics and in neuroscience.

In mathematics, the same objects, e.g. functions, can be represented in various ways. The identity of the object remains the same but by representing it in a different manner, different characteristics of the object are highlighted. Depending on the calculation that is performed on the object, one representation might be more suitable than another. Not every object has the same characteristics. And not every representation highlights for each object equally well its peculiarities. That is why there is research on methods which calculate from the object a tailored representation that pinpoints its characteristics.

In neuroscience, a big question is how sensory input is represented by neurons. Sensory receptors capture an external stimulus and signal it to the brain. There, other neurons must integrate and transform these incoming signals, and then transmit them to further neurons, such that, finally, an appropriate action might be triggered. Outside the brain, speech is for example represented by the pressure difference in the air. But what happens to sound after it has been captured by the hair cells in the ear? How is the sound represented by the neurons? Research in neural integration, and the neural representation of sensory input, deals with how neurons describe (code) by their activity the sensory input, and what happens to the description as it passes from one area in the brain to another.

A central point of this thesis is to model neural integration by means of data representation. The motivation from mathematics and neuroscience get through that modeling interconnected and give rise to the topic data representation with dynamical neural systems.

Structure

In this thesis, we will develop learning rules for data representation with dynamical neural systems.

- In *Chapter 2*, we will provide background on existing data representation methods. In Section 2.1, we relate data representation to the change of basis. We point out that for a good representation, the basis should be chosen in function of the nature of the data. In Section 2.2, algorithms for data representation that does not include time dependencies of the data are presented. In Section 2.3, data representation with time structure is treated.

In both cases, the algorithms are derived by first formulating the data representation problem as an optimization problem, and then, to iteratively optimize the cost functional.

- In *Chapter 3*, we give background information on neuroscience. We start in Section 3.1 with the modeling of the signaling capacity of neurons. We will review models for neural signal transformation, and we will review also the biophysical processes which absorb heavily energy during neural signaling. In Section 3.2, neural integration is treated. We introduce neural maps and show how they help to visualize neural integration. The development of neural maps leads to Section 3.3 where we treat synaptic plasticity.

- In *Chapter 4*, we explain in Section 4.1 how data representation can be used as a model for neural integration. In Section 4.2, we review previous work which has used the data representation methods that we have discussed in Chapter 2 to get insight into neural integration. The amount of obtained insight shows that data representation is a promising model for neural integration. In Section 4.3, we ask how to learn more about neural integration. Comparing the models for neural signal transformation of Chapter 3 with the signaling transformation of the data representation methods of Chapter 2, we will see that the existing data representation methods work with a rather abstract neuron model. In our opinion, the modeling of neural integration by means of data representation could be improved by using dynamical neural systems for the encoding. This should allow to obtain more insight into neural integration, and the resulting neural representation of sensory input.

- In *Chapter 5*, we develop learning rules for data representation with a single monostable integrator neuron. It is a formal spiking neuron that we have introduced in Chapter 3. Learning rules are obtained by formulating the data representation problem as an optimization problem. Energy consumption of the neuron is also modeled, where we use results that we have reviewed in Chapter 3. Simulations illustrate the applicability of the learning rules, and the influence of the different measures for energy consumption on the learning.

- In *Chapter 6*, we develop learning rules for data representation with a population of neurons of the previous chapter: First, learning rules for a population of neurons without direct synaptic coupling are derived. Then, learning

rules for coupled neurons are derived. Simulations illustrate the applicability of the learning rules, and the influence of the synaptic coupling on the resulting neural representation.

- In *Chapter 7*, we consider again the case of a single neuron. The formal spiking neuron model of Chapter 5 is replaced by a true spiking neuron, the canonical integrator neuron (with another name: quadratic integrate and fire neuron). We develop learning rules for data representation with that neuron model and compare them to the learning rules of Chapter 5.

- *Chapter 8* concludes the thesis.

Chapter 2

Mathematics-Background

2.1 Data representation and the change of basis

In Section 2.1.1, we explain the meaning of the term "data representation" and give examples for data representation with and without time structure. In Section 2.1.2, we establish a link between data representation and the change of basis. We also review the conversion from one representation to another (the coordinate transform). In Section 2.1.3, we then put emphasis on the point that the basis should be chosen in function of the nature of the data. We call the choice of the basis selectivity transform.

2.1.1 Data representation

Data representation is data organization so that meaningful information can be extracted. Data representation is thus a means for knowledge discovery. It should make hidden structure explicit and allow in that way to make predictions about the future.

Figure 2.1 shows an example of input data. The data shows over time an oscillatory behavior with some large excursions. The shown representation is a representation over time where each data point $x(t)$ is plotted for $t = 1, 2, \ldots$. Figure 2.1 indicates also how the data has been created. The Gaussian function $h(t)$, has been shifted by random amounts of time, and the shifted functions were summed up to yield $x(t)$, i.e.

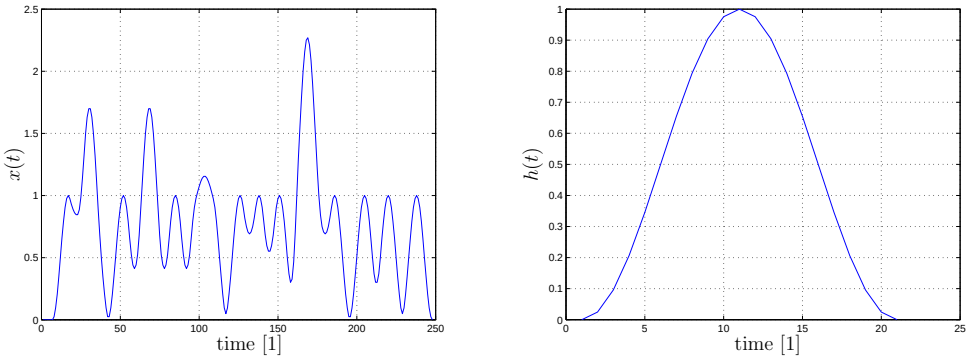
$$x(t) = \sum_n h(t - t^n), \tag{2.1}$$

where $t^n > t^{n-1} > 0$ for all n. An alternative representation of $x(t)$ would thus be given by the plot of $h(t)$ and the numbers $\{t^n\}_n$. This representation yields more insight than the numbers $\{x(t)\}_t$. It shows that there is a basic element $h(t)$ which underlies the data and the knowledge of a single t^{n_o} would allow to predict the data over some later time interval.

This example can be extended to illustrate that re-organization of the data can make structure explicit which remains rather hidden in the representation $\{x(t)\}_t$. Figure 2.2 illustrates the case where

$$x(t) = \sum_{m=1}^{2} \sum_n h_i(t - t_m^n). \tag{2.2}$$

The representation of $x(t)$ in the form of the numbers $\{t_1^n\}_n$ and $\{t_2^n\}_n$, as well as the two functions $h_1(t)$ and $h_2(t)$, allows to grasp the structure of the data



Input data Basic element

Figure 2.1: Input data and basic element. The data $x(t)$ on the left has been artificially created by summing up shifted versions of the function $h(t)$ shown on the right. Given shift time t^{n-1}, the next time t^n was drawn from an uniform distribution with support $(t^{n-1} \quad t^{n-1} + 20]$.

more easily than the pointwise representation of $x(t)$ along the time axis, i.e. the numbers $\{x(t)\}_t$.

Structure in the data means that there are statistical dependencies. Above, we have illustrated the representation of data with dependencies over time. In Figure 2.3, we show two dimensional data $\mathbf{x} = (x_1, x_2)$ with dependencies between x_1 and x_2. Making structure explicit means here finding a representation which makes the sources of the dependencies explicit. The data $\mathbf{x}$ has been created via

$$\mathbf{x} = \sum_{m=1}^{2} \mathbf{h_m} s_m, \tag{2.3}$$

where s_1 and s_2 are two independent random variables. The representation of the data $\mathbf{x}$ in that form would clearly identify the source for the structure in the data and allow to explore its properties by characterization of the sources s_m and the matrix $\mathbf{H} = [\mathbf{h_1} \; \mathbf{h_2}]$.

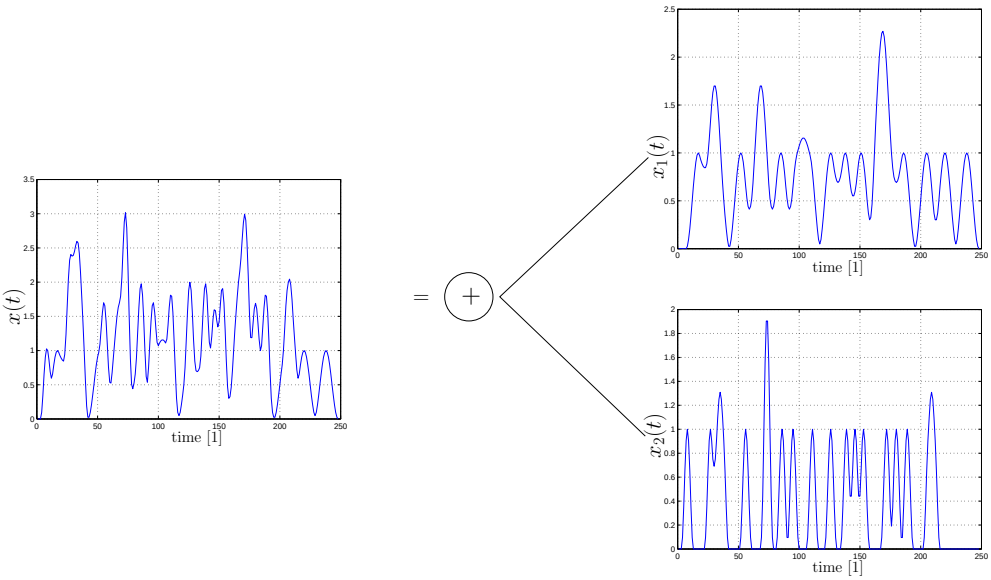


Figure 2.2: The data $x(t)$ on the left has been artificially created by summing up the two curves $x_1(t)$ and $x_2(t)$ on the right. The upper curve $x_1(t)$ is the same as in Figure 2.1. The lower one, $x_2(t)$ was created by the same method but with a narrower Gaussian.

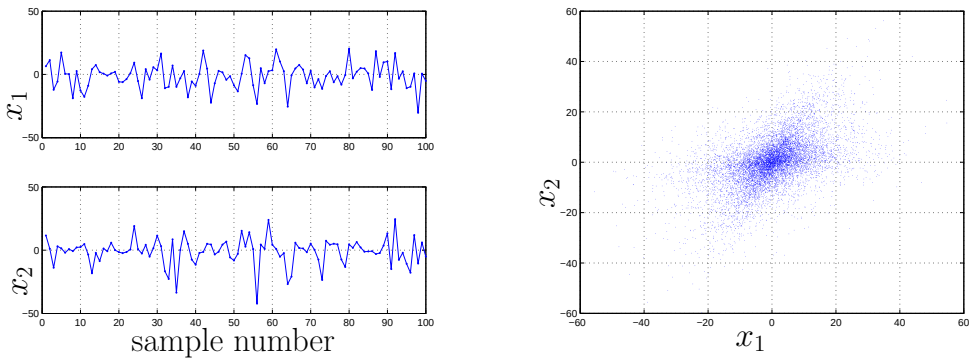


Figure 2.3: Two dimensional data with dependencies. Left: Observed samples of the data x_1 and x_2. Right: Scatter plot. The scatter plot shows that there are statistical dependencies between x_1 and x_2.

2.1.2 Change of basis

In the previous section, data that was harder to understand was created from simpler data by two transforms which had the form

$$x(t) = \sum_m \sum_n s_m^n h_m(t - t_m^n) \tag{2.4}$$

$$\mathbf{x} = \sum_m s_m \mathbf{h_m}. \tag{2.5}$$

The index t went from $t = 1$ to $t = T$ so that $x(t)$ can be written has a vector $\mathbf{x}$. Similar for $h_m(t - t_m^n)$ where each time shift t_m^n defines a new vector $\mathbf{h_m}$. The first equation is a thus a more restrictive version of the second equation because the different vectors $\mathbf{h_m}$ are related by time shift.

For data representation, the inverse transform should be done: given is the data $\mathbf{x}$ which should be decomposed into $\sum_m s_m \mathbf{h_m}$ so that the properties of the s_m and $\mathbf{h_m}$ help to gain insight into the nature of the data. Data $\mathbf{x}$, or $x(t)$, consists of random variables so that the question arises whether any realization of the random variable can be decomposed as in Equations (2.4) and (2.5). Specifically,

1. What properties must the $\mathbf{h_m}$ and $h_i(t)$ have such that their weighted sum can represent any realization of $\mathbf{x}$ and $x(t)$, respectively?

2. Given the $\mathbf{h_m}$ and $h_i(t)$ how are the s_m and s_m^n, respectively, calculated?

3. How are the $\mathbf{h_m}$ and $h_m(t)$, respectively, determined?

What regards the first question, the vector $\mathbf{x}$ is in our case of finite dimension N so that linear algebra shows that linear independence of N vectors $\mathbf{h_m}$ is sufficient, see e.g. [Frazier, 2001]. The same reference gives also conditions on the representation with time shifted $h_m(t)$, where in finite dimension N the time shift is a circular shift. The problem becomes more complicated if larger vector spaces, i.e. infinite dimensional spaces, are involved: the first question is at the heart of Fourier and wavelet theory, see e.g. [Frazier, 2001].

What regards the second question, we first note that if the vector $\mathbf{x} \in \mathbb{R}^N$ is developed with respect to the standard basis $\mathbf{e}^n$, which is a N dimensional vector with all entries being zero but the nth one being one, the representation becomes

$$\mathbf{x} = \sum_{n=1}^{N} x_n \mathbf{e}^n. \tag{2.6}$$

Grouping all the coefficients x_n into the vector $[\mathbf{x}]_\mathbf{e}$ one obtains $\mathbf{x} = [\mathbf{x}]_\mathbf{e}$. The coefficients s_m in Equation (2.5) are in a similar way denoted by $[\mathbf{x}]_\mathbf{h}$. The two coefficient (or coordinate) vectors are related by the coordinate transform

$$[\mathbf{x}]_\mathbf{h} = \mathbf{A}[\mathbf{x}]_\mathbf{e}, \tag{2.7}$$

where the matrix $\mathbf{A}$ satisfies

$$\mathbf{e}^n = \sum_{m=1}^{N} a_{mn}\mathbf{h_m}, \tag{2.8}$$

see e.g. [Frazier, 2001]. If the vectors $\mathbf{h_m}$ are mutually orthogonal then $\mathbf{A} = \mathrm{diag}(1/||\mathbf{h_m}||^2)\mathbf{H^T}$, where the columns of $\mathbf{H}$ are the vectors $\mathbf{h_m}$ so that

$$\mathbf{x} = \sum_{m=1}^{N} \frac{\mathbf{h_m^T x}}{||\mathbf{h_m}||^2}\mathbf{h_m}. \tag{2.9}$$

If the $\mathbf{h_m}$ are related to each other by time shifts, the inner products $\mathbf{h_m^T x}$, $m = 1 \dots N$, can all be calculated together by a convolution.

The third question is about how to actually change the basis. Comparison of Equation (2.8) and Equation (2.9) shows that the coordinate transform becomes simpler if the $\mathbf{h_m}$ are orthogonal. That is why orthogonal bases are practical. For the context of the choice of basis, see [Frazier, 2001] for a discussion about the applicability of wavelets and the Fourier transform.

2.1.3 Adapting the basis to the structure of the data

A lesson from linear algebra is that a proper choice of the basis allows to simplify a problem considerably [Frazier, 2001]. As we have mentioned in the beginning of the previous section, this is also the case for data representation: The s_m and $\mathbf{h_m}$ should be such that maximal insight into the nature of the data is obtained.

Similar to the coordinate transform which maps $[\mathbf{x}]_\mathbf{e}$ to $[\mathbf{x}]_\mathbf{h}$, one should have a map from the $\{\mathbf{e}^n\}_n$ to the $\{\mathbf{h}_m\}_m$. This map should ideally depend on the nature of the data $\mathbf{x}$ which one likes to represent. The basis is thus not fixed in advance but adaptively constructed from the data. The resulting map may be called selectivity transform.

2.2 Data representation without time structure

We have in the previous section seen that the choice of the basis (the selectivity transform) should depend on the nature of the data. We review here methods

which realize this selectivity transform. As a second step, the methods include also the coordinate transform which converts one representation to the other. In Section 2.2.1, the data representation problem is formulated as an optimization problem. We will see that the posed problem has no unique solution. In the framework of principal component analysis (PCA), this under-determinedness manifests itself as a representation which is unique up to an arbitrary rotation. In Section 2.2.2, we review algorithms which iteratively solve the optimization problem and yield a representation of the data which is unique up to the aforementioned rotation. These iterative algorithms are also called learning rules. In Section 2.2.3, we summarize different approaches to resolve the under-determinedness.

2.2.1 Linear representation

We seek a linear decomposition of the data. Data representation is modeled by means of a cost functional as an optimization problem. The decomposition is then obtained at the minimum of the cost functional. We first present the cost functional. Then, we obtain an analytical solution for the minimum. Finally, we present the cost functional which is used in the framework of principal component analysis and its analytical solution. We point out that the optimal solutions are not unique.

The optimization problem

Given is a collection of N zero mean random variables $\{x_n\}_{n=1}^N$ which are defined on the same probability space $(\Omega, \mathcal{B}, P)$, i.e.

$$x_n(\omega) : \Omega \to \mathbb{R}, \quad n = 1 \dots N.$$

The collection of random variables defines a random vector $\mathbf{x}$ with respect to the standard basis $\{\mathbf{e}^{(n)}\}_{n=1}^N$, i.e.

$$\mathbf{x} = \sum_{n=1}^{N} x_n \mathbf{e}^{(n)}. \tag{2.10}$$

The problem is to find a representation of $\mathbf{x}$ in the form of

$$\hat{\mathbf{x}} = \sum_{m=1}^{M} y_m \mathbf{h}_m, \tag{2.11}$$

where $\mathbf{h}_m \in \mathbb{R}^N$, so that $\mathbf{x}$ is well approximated for $M \leq N$. Accuracy of the approximation is measured by means of the mean squared error, i.e. the representation should minimize

$$J = \frac{1}{2} E\left(||\mathbf{x} - \hat{\mathbf{x}}||_2^2\right). \tag{2.12}$$

This provides a formulation of the data representation problem in terms of an optimization problem. The M dimensional coordinate vector $\mathbf{y}$ of the coordinates with respect to the vectors $\{\mathbf{h}\}_m$ is searched as

$$\mathbf{y} = \mathbf{W^T x}, \tag{2.13}$$

so that the cost functional, which is to be minimized, becomes

$$J(\mathbf{H}, \mathbf{W}) = \frac{1}{2} E\left(||\mathbf{x} - \mathbf{HW^T x}||_2^2\right), \tag{2.14}$$

where $\mathbf{H}$ is the matrix with $\mathbf{h}_m$ as its m-th column. An equivalent problem is given by the replacement of the expectation by a sample mean, i.e.

$$J'_t(\mathbf{H}, \mathbf{W}) = \frac{1}{2} \sum_{i=1}^{t} \beta^{t-i} \left(||\mathbf{x}(i) - \mathbf{HW^T x}(i)||_2^2\right), \tag{2.15}$$

for a given constant β and t samples. Or

$$J''_t(\mathbf{H}, \mathbf{W}) = \frac{1}{2t} \sum_{i=1}^{t} \left(||\mathbf{x}(i) - \mathbf{HW^T x}(i)||_2^2\right) \tag{2.16}$$

for uniform weighting.

Solution

This problem has been studied by [Baldi and Hornik, 1995, 1989]. The problem of minimizing J has not a unique solution, as it can be seen from the cost functional. Only the product $\mathbf{HW^T}$ is uniquely defined. They show that the global minimum of J is attained at

$$\mathbf{H}^* = \mathbf{U_M C}^{-1} \tag{2.17}$$

$$\mathbf{W^{T*}} = \mathbf{C U_M}^T, \tag{2.18}$$

where $\mathbf{C}$ is an arbitrary invertible $M \times M$ matrix and the columns of $\mathbf{U_M}$ are the eigenvectors with the largest M eigenvalues of $\mathbf{R_{xx}}$ (the eigenvalues are assumed

to be different from each other). The matrix $\mathbf{R_{xx}}$ is given by

$$\mathbf{R_{xx}} = E(\mathbf{x}\mathbf{x}^T) \quad \text{forJ} \tag{2.19}$$

$$\mathbf{R_{xx}} = \sum_{i=1}^{t} \beta^{t-i}\mathbf{x}\mathbf{x}^T \quad \text{forJ}' \tag{2.20}$$

$$\mathbf{R_{xx}} = \sum_{i=1}^{t} \frac{1}{t}\mathbf{x}\mathbf{x}^T \quad \text{forJ}''. \tag{2.21}$$

The cost at the optimum values $J(\mathbf{H}^*, \mathbf{W}^*) = \text{trace } \mathbf{R_{xx}} - \sum_{i=1}^{M} \lambda_i$, where λ_i is the eigenvalue associated with the vector which forms the i-th column of $\mathbf{U_M}$.

The PCA setting

Principal component analysis (PCA) works with the assumption $\mathbf{H} = \mathbf{W}$. One obtains as cost functional

$$J_{\text{PCA,r}}(\mathbf{W}) = \frac{1}{2} E\left(||\mathbf{x} - \mathbf{W}\mathbf{W}^\mathbf{T}\mathbf{x}||_2^2\right), \tag{2.22}$$

with the optimal solution $\mathbf{W}^*$ being given by

$$\mathbf{W}^* = \mathbf{U_M}\mathbf{C_o}^T, \tag{2.23}$$

where $\mathbf{C_o}$ is an arbitrary orthogonal $M \times M$ matrix. The minimization of $J_{\text{PCA,r}}$ results thus in a solution $\mathbf{W}^*$ with orthonormal columns, see also [Yang, 1995].

In the standard PCA setting, this orthonormality of the columns is included as a constraint in the optimization problem. In that case, the cost functional can be written as $J_{\text{PCA,r}}(\mathbf{W}) = 1/2\left(\text{trace } \mathbf{R_{xx}} - \text{trace } \mathbf{W}^\mathbf{T}\mathbf{E_x}\mathbf{W}\right)$. The second term on the right is the expectation of the squared norm of $\mathbf{y}$, i.e. the variance for the zero mean assumption of $\mathbf{x}$. Hence, in case of an orthonormality constraint, the minimization of $J_{\text{PCA,r}}$ is equivalent to the maximization of the output variance

$$J_{\text{PCA,v}}(\mathbf{W}) = \text{trace } \mathbf{W}^\mathbf{T}\mathbf{E_x}\mathbf{W}. \tag{2.24}$$

Including the orthonormality constraint allows for an additional interpretation of the cost functional. The solution, however, is still given by Equation (2.23).

2.2.2 Iterative optimization

If the correlation matrix $\mathbf{R_{xx}}$ of the input data $\mathbf{x}$ is known, algorithms which were developed to solve the eigenvalue problem can be used to solve the data representation problem above. Here, we review algorithms which can be used if $\mathbf{R_{xx}}$ is not known, see also [Baldi and Hornik, 1995; Diamantaras and Kung, 1996]. The solutions are determined up to an orthogonal transform as in Equation (2.23).

Key point of the iterative optimization algorithms

One part of the data representation problem is to find the approximation $\hat{\mathbf{x}}$ from the output $\mathbf{y}$. This is a linear estimation problem, see e.g. [Gray and Davisson, 2004] with the optimal solution $\mathbf{W^o}$ satisfying the equation

$$\mathbf{W^o R_{yy}} = \mathbf{R_{xy}}, \tag{2.25}$$

where $\mathbf{R_{xy}} = E(\mathbf{x}\mathbf{y}^T)$. It can be verified that $\mathbf{W}^*$ from Equation (2.23) is a solution to that equation for $\mathbf{y} = \mathbf{W}^{*\mathbf{T}}\mathbf{x}$. If $\mathbf{R_{yy}}$ is positive definite, the solution is unique so that $\mathbf{W}^* = \mathbf{W^o}$.

The iterative optimization algorithms presented below use that property. In each update step k, they create the output data $\mathbf{y}(k)$ from input $\mathbf{x}(k)$ via the matrix $\mathbf{W}(k-1)$, which is the matrix which resulted from the update step $k-1$. The update rule uses then this $\mathbf{y}(k)$ to calculate the new $\mathbf{W}(k)$.

Iterative optimization based on the expectation

The presented algorithm is an online gradient descent algorithm. The algorithm is known in the literature as Symmetric Error Correction (SEC) algorithm or subspace learning algorithm, see the review [Baldi and Hornik, 1995]. The algorithms calculate iteratively $\mathbf{W}^*$ of Equation (2.23).

Considering $\mathbf{y} = \mathbf{W^T x}$ fixed, the cost functional $J_{\mathrm{PCA,r}}$ of Equation (2.22) becomes

$$\tilde{J}_{\mathrm{PCA,r}}(\mathbf{W}) = \frac{1}{2} E\left(||\mathbf{x} - \mathbf{W}\mathbf{y}||_2^2\right). \tag{2.26}$$

Calculation of the gradient with respect to the matrix $\mathbf{W}$ and dropping the expectation yields the following gradient descent update rule

$$\triangle \mathbf{W} \;=\; \mu\left[(\mathbf{x} - \mathbf{W}\mathbf{y})\mathbf{y^T}\right], \tag{2.27}$$

where $\mu > 0$ is the step size and the term $(\mathbf{x} - \mathbf{W}\mathbf{y})$ the reconstruction error. This is the also the update rule of the least-mean-square algorithm (LMS), see e.g [Haykin, 2001]. Note that here, however, $\mathbf{y}$ is calculated from $\mathbf{x}$ as $\mathbf{y} = \mathbf{W^T x}$.

Iterative optimization based on the sample average

Algorithms which use the sample average are based on the cost functional of Equation (2.15) in the PCA setting, i.e. with $\mathbf{H}$ replaced by $\mathbf{W}$. They also

consider $\mathbf{y} = \mathbf{W^T x}$ as being fixed so that the effective cost functional becomes

$$\tilde{J}'_{\mathrm{PCA,r}}(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^{t} \beta^{t-i} \left(||\mathbf{x}(i) - \mathbf{W}\mathbf{y}(i)||_2^2 \right). \tag{2.28}$$

The iterative minimization of such a cost functional is effectively done by means of a recursive-least-squares (RLS) algorithm, see e.g. [Haykin, 2001]. Direct application of the RLS algorithm yields the update rule

$$\triangle \mathbf{W} = \mu(\mathbf{x} - \mathbf{W}\mathbf{y})\mathbf{y^T}\mathbf{R_{yy}^{-1}}, \tag{2.29}$$

where μ is given in each iteration by

$$\mu = \frac{1}{\beta + \mathbf{y}^T \mathbf{R_{yy}^{-1}} \mathbf{y}} \tag{2.30}$$

and $\mathbf{y}$ is calculated as $\mathbf{y} = \mathbf{W^T x}$. The matrix $\mathbf{R_{yy}^{-1}}$ is updated by

$$\mathbf{R_{yy}^{-1}} \leftarrow \frac{1}{\beta} \left(\mathbf{R_{yy}^{-1}} - \mu \mathbf{R_{yy}^{-1}} \mathbf{y} (\mathbf{R_{yy}^{-1}} \mathbf{y})^T \right), \tag{2.31}$$

and the iteration can be started with the identity matrix. It converges to $\mathbf{W}^*$ of Equation (2.23). This algorithm is called the PAST algorithm [Yang, 1995]. Here, all the columns of $\mathbf{W}$ are determined at the same time. Other approaches calculate the columns one after another (deflationary mode). The PAST algorithm forms the basis for such an algorithm, the PASTd [Yang, 1995], or equivalently, the APEX algorithm [Diamantaras and Kung, 1996]. The two algorithms yield a solution which corresponds up to the sign to the first M eigenvectors of $\mathbf{R_{xx}}$, i.e. $\mathbf{U_M}$.

Interpretation

We have seen that the minimization of $J_{\mathrm{PCA,r}}$ in Equation (2.22) can be considered as maximization of the variance of $\mathbf{y}$ under the constraint of orthonormality of the columns of $\mathbf{W}$. Variance maximization and orthonormality constraint is also visible in the update rules above. Both update rules have the common term $(\mathbf{x} - \mathbf{W}\mathbf{y})\mathbf{y^T} = \mathbf{x}\mathbf{y^T} - \mathbf{W}\mathbf{y}\mathbf{y^T}$. The first term comes from the derivative of $J_{\mathrm{PCA,v}}$ in Equation (2.24) with respect to $\mathbf{W}$. i.e.

$$\frac{\delta J_{\mathrm{PCA,v}}}{\delta \mathbf{W}} = E\left(\mathbf{x}\mathbf{y^T}\right). \tag{2.32}$$

The second term comes from the orthonormality constraint. After the update, $\mathbf{W} + \triangle\mathbf{W}$ should still have orthonormal columns so that

$$(\mathbf{W} + \triangle\mathbf{W})^T(\mathbf{W} + \triangle\mathbf{W}) \stackrel{!}{=} \mathbf{1}_M, \tag{2.33}$$

where $\mathbf{1}_M$ is the M dimensional identity matrix. With the ansatz

$$\triangle \mathbf{W} = \alpha(\mathbf{x}\mathbf{y}^\mathbf{T} + \mathbf{W}\mathbf{S}), \tag{2.34}$$

with α being small, one obtains the following condition for $\mathbf{S}$

$$\mathbf{S} + \mathbf{S}^\mathbf{T} \stackrel{!}{=} -2\mathbf{y}\mathbf{y}^\mathbf{T}. \tag{2.35}$$

The condition is fulfilled for $\mathbf{S} = -\mathbf{y}\mathbf{y}^\mathbf{T}$, which yields the above algorithms. Another possibility would be

$$S = -\text{diag}\left(\mathbf{y}\mathbf{y}^\mathbf{T}\right) - 2\text{UpperTr}\left(\mathbf{y}\mathbf{y}^\mathbf{T}\right), \tag{2.36}$$

which yields the Stochastic Gradient Ascent (SGA) algorithm, see the review in [Baldi and Hornik, 1995]. The term UpperTr means the upper triangular part of the matrix without the diagonal. Closely related to the former choice is

$$S = -\text{diag}\left(\mathbf{y}\mathbf{y}^\mathbf{T}\right) - \text{UpperTr}\left(\mathbf{y}\mathbf{y}^\mathbf{T}\right), \tag{2.37}$$

which yields the Generalized Hebbian Algorithm (GHA) of [Sanger, 1989]. This algorithm converges to the first dominant M eigenvectors of $\mathrm{R_{xx}}$.

2.2.3 Resolving the under-determinedness of the problem

We summarize here different approaches to resolve the under-determinedness of the data representation problem, i.e. approaches to fix the orthogonal matrix $\mathbf{C_o}$ or the invertible matrix $\mathbf{C}$. Table 2.1 presents an overview.

Resolution through inhomogeneities

Among the algorithms, that we have presented above, only those which were based on a deflationary approach (PASTd, APEX) and the GHA algorithm converge to a solution with $\mathbf{C_o} = \text{diag}(\pm 1, \pm 1, \ldots, \pm 1)$. They share the common point that they are asymmetric algorithms in the sense that the update rules for each column of $\mathbf{W}$ are not exactly the same. Such an inhomogeneity can also be introduced into the cost functional. In [Xu, 1993], among others, the following cost functional was considered

$$J(\mathbf{W}) = \frac{1}{2} E\left(||\mathbf{x} - \mathbf{W}(\mathbf{D}\mathbf{W}^\mathbf{T}\mathbf{x})||_2^2\right). \tag{2.38}$$

Matrix $\mathbf{D}$ is a diagonal matrix with unequal, strictly positive elements. It amplifies each element of the output vector $\mathbf{y} = \mathbf{W}^\mathbf{T}\mathbf{x}$. From this criterion, the update rule

$$\triangle \mathbf{W} = \mu\left[\mathbf{x}\mathbf{y}^T\mathbf{D} - \mathbf{W}\mathbf{D}\mathbf{y}\mathbf{y}^T\right] \tag{2.39}$$

Resolution approach	Min. reconstruction error
1. Deflationary iterative optimization	Yes: $J_{\mathrm{PCA,r}}$
2. Add inhomogeneities to the cost functional	Yes
3. Add norm-constraints on the columns $\mathbf{w_m}$	Yes: $J_{\mathrm{PCA,r}}$
4. Punish norm of the columns $\mathbf{w_m}$	No
5. Punish norm of $\mathbf{y}$	No
6. Maximization of independence of $\mathbf{y}$	Yes: $J(\mathbf{H}, \mathbf{W})$

Table 2.1: Summary of the approaches to resolve the under-determinedness of the data representation problem. The resolution can happen through additional constraints on $\mathbf{W}$, or through modification of the original cost functional. For approach 1 and 3 the cost functional $J_{\mathrm{PCA,r}}$ is minimized and the orthogonal $\mathbf{C_o}$ is fixed. For approach 6, the more general $J(\mathbf{H}, \mathbf{W})$ is minimized, and the invertible $\mathbf{C}$ is fixed. Approach 2 also minimizes a reconstruction error (as defined in the text body). Approach 4 and 5, however, yield a trade-off between minimization of the reconstruction error and the additional punishment. In these approaches, the dimension M of $\mathbf{y}$ can be larger than the dimension of the input $\mathbf{x}$.

was deduced. The iteration converges to $\mathbf{W}^*$ which has, up to the sign, the first dominant eigenvectors of $\mathbf{R_{xx}}$ as its columns.

Resolution through additional constraints on W

In order to determine the free orthogonal matrix $\mathbf{C_o}$, in [Oja et al., 1992a,b], hard norm constraints on $\mathbf{W}$ were considered. They prove that under the constraint

$$\mathbf{W^T W} \stackrel{!}{=} \mathrm{diag}(\omega_1^2, \ldots, \omega_M^2) \tag{2.40}$$

the unique solution to the minimization of $J_{\mathrm{PCA,r}}$ in Equation (2.22), or the maximization of $J_{\mathrm{PCA,v}}$ in Equation(2.24) is given by $\mathbf{W}^* = \mathbf{U_M \Omega}$, where $\mathbf{\Omega} = \mathrm{diag}(\omega_1, \ldots, \omega_M)$.

In the same work, the iteration

$$\triangle \mathbf{W} = \mu \left[\mathbf{x y^T} - \mathbf{W y y^T \Omega^{-2}} \right] \tag{2.41}$$

is shown to converge to that solution. Calculating $(\mathbf{W} + \triangle \mathbf{W})^T (\mathbf{W} + \triangle \mathbf{W})$, we observe that the norm constraint for each column of $\mathbf{W}$ is included in this iteration, the orthogonality constraint, however, is not well taken into account.

The iteration

$$\triangle \mathbf{W} = \mu \left[\mathbf{x}\mathbf{y}^{\mathbf{T}} - \mathbf{W}\mathbf{\Omega}^{-\mathbf{2}}\mathbf{y}\mathbf{y}^{\mathbf{T}} \right] \tag{2.42}$$

does respect orthogonality and norm of the columns of $\mathbf{W}$.

Resolution through punishment of the norm of the columns of W

In contrast to the hard constraints on $\mathbf{W}$, soft constraints on the norm of the columns of $\mathbf{W}$ were considered in [Vincent et al., 2005; Vincent and Baddeley, 2003]. In [Vincent and Baddeley, 2003], they add to the cost functional $J_{\mathrm{PCA,r}}$ in Equation (2.22) the term

$$J_1(\mathbf{W}) = \alpha \sum_{m=1}^{M} f\left(\sum_{n=1}^{N} |w_{nm}|^p - \beta \right), \tag{2.43}$$

where $\alpha > 0$ and $\beta > 0$ are fixed constants, the function $f(x) = 1$ for $x \geq 0$ and zero for $x < 0$, and the parameter p was varied between 0.5 and 1.5. This additional term in the cost functional leads to an additional term in the iterative optimization rules of Section 2.2.2. Calculation of the gradient of J_1 shows that each column $\mathbf{w_m}$ of $\mathbf{W}$ needs an additional update if $\sum_{n=1}^{N} |w_{nm}|^p \geq \beta$. This additional term is

$$\triangle \mathbf{w_m} = -\mu_1 \left[\mathrm{sign}(\mathbf{w_m}) |\mathbf{w_m}|^{p-1} \right], \tag{2.44}$$

where both the norm and the sign is taken element-wise. Parameter μ_1 indicates the step size for this additional, secondary update. A variant of this scheme is considered in [Vincent et al., 2005]. There, the term

$$J_2(\mathbf{W}) = \alpha \sum_{m=1}^{M} \sum_{n=1}^{N} |w_{nm}| \tag{2.45}$$

is added to the cost functional $J_{\mathrm{PCA,r}}$. This leads to the following modification of the learning rules of Section 2.2.2. In each iteration, the additional update

$$\triangle \mathbf{w_m} = -\mu_2 \mathrm{sign}(\mathbf{w_m}), \tag{2.46}$$

takes places for each column $\mathbf{w_m}$, $m = 1, \ldots M$, of $\mathbf{W}$.

The hard constraints on $\mathbf{W}$, as also the inhomogeneities, provided a means to fix the orthogonal $\mathbf{C_o}$ of the under-determined $\mathbf{W}^*$ of Equation (2.23). The soft constraints, however, do not determine $\mathbf{C_o}$ but $\mathbf{W}$ by changing the optimization problem. The obtained matrix $\mathbf{W}$ does not minimize $J_{\mathrm{PCA,r}}$ but the sum $J_{\mathrm{PCA,r}} + J_1$ or $J_{\mathrm{PCA,r}} + J_2$.

Resolution through punishment of the norm of y

In a similar esprit, the cost functional $J_{\mathrm{PCA,r}}$ in Equation (2.22) can be augmented by

$$J_3(\mathbf{W}) = \alpha \sum_{m=1}^{M} |\mathbf{w_m^T x}|, \tag{2.47}$$

see [Vincent et al., 2005]. $J_{\mathrm{PCA,r}}$ requires reconstructability and J_3 punishes large output values y_m. Parameter α weights the contribution of each cost functional. Calculation of the gradient of J_3 shows that the learning rules of Section 2.2.2, which minimize $J_{\mathrm{PCA,r}}$, are modified in the sense that their updates are followed by

$$\triangle \mathbf{w_m} = -\mu_3 \mathrm{sign}(y_m)\mathbf{x}, \tag{2.48}$$

where μ_3 is the step size.

Resolution through maximization of statistical independence of the y_m

For the above methods where $\mathbf{C_o} = \mathrm{diag}(\pm 1, \pm 1, \ldots, \pm 1)$, the outputs $\mathbf{y}$ become uncorrelated. If the assumption $\mathbf{H} = \mathbf{W}$ of the PCA setting is dropped, the underdeterminedness of the data representation problem can also be resolved by making the outputs more statistically independent than being uncorrelated.

Statistical independence is commonly measured by mutual information, see e.g. [Hyvärinen et al., 2001a],

$$I(\mathbf{y}) = \sum_{m=1}^{M} E\left(\log_2 \frac{1}{p_{y_m}(y_m)}\right) - E\left(\log_2 \frac{1}{p_{\mathbf{y}}(\mathbf{y})}\right). \tag{2.49}$$

The term $p_{\cdot}(.)$ is here the probability density. Mutual information is invariant to scaling of the involved random variables. The variances of the outputs y_m are thus undefined and can be fixed to unity. If $\mathbf{C} = \mathrm{diag}(\lambda_i^{-1/2})$, $\mathbf{W}^*$ of Equation (2.23) leads to uncorrelated, unit variance (white) outputs y_m. Since this $\mathbf{C}$ is not orthogonal any more, we re-consider here the original cost functional of Equation (2.14) with the optimal solution given in Equations (2.17) and (2.18). We can write the undetermined invertible matrix $\mathbf{C}$ of these equations as

$$\mathbf{C} = \mathbf{B^T} \mathrm{diag}(\lambda_i^{-1/2}), \tag{2.50}$$

with unknown matrix $\mathbf{B}$. This matrix operates on

$$z := \mathrm{diag}(\lambda_i^{-1/2}) \mathbf{U_M^T x}, \tag{2.51}$$

which is white and yields $\mathbf{y} = \mathbf{B^T z}$, which is by assumption white as well. The output $\mathbf{y}$ is white if the unknown matrix $\mathbf{B}$ is orthogonal. We summarize here a method of [Hyvärinen et al., 2001a] to find the unknown orthogonal matrix $\mathbf{B}$. An iterative algorithm which yields the white $\mathbf{z}$ from the input $\mathbf{x}$ can be found in [Cardoso and Laheld, 1996].

Cost functional for B Orthogonal matrix $\mathbf{B}$ should be such that the statistical dependence between the elements y_m, $m = 1, \ldots, M$ of $\mathbf{y}$ is minimized. This is called independent component analysis (ICA). Since $\mathbf{y} = \mathbf{B^T z}$, the probability density $p_{\mathbf{y}}(\mathbf{y})$ in the definition for mutual information is $p_{\mathbf{z}}(\mathbf{z})/|\det\mathbf{B}|$, which equals $p_{\mathbf{z}}(\mathbf{z})$ by the orthogonality of $\mathbf{B}$. Hence mutual information $I(\mathbf{y})$ becomes

$$I(\mathbf{y}) = \sum_{m=1}^{M} E\left(\log_2 \frac{1}{p_{y_m}(y_m)}\right) - E\left(\log_2 \frac{1}{p_{\mathbf{z}}(\mathbf{z})}\right). \tag{2.52}$$

The second term involving $\mathbf{z}$ is constant so that the minimization of $I(\mathbf{y})$ is thus equivalent to the minimization of

$$\tilde{J}_{\mathrm{ICA}}(\mathbf{B}) = \sum_{m=1}^{M} E\left(\log_2 \frac{1}{p_{\mathbf{b_m^T z}}(\mathbf{b_m^T z})}\right), \tag{2.53}$$

where $y_m = \mathbf{b_m^T z}$. This is a sum of the output entropies $H(y_m)$. As the Gaussian distribution maximizes the entropy of an unit variance variable, the minimization of $\tilde{J}_{\mathrm{ICA}}$ can be interpreted as the maximization of the sum of non-Gaussianities.

The minimization of $\tilde{J}_{\mathrm{ICA}}$ is difficult because the probability densities $p_{\mathbf{b_m^T z}}(\mathbf{b_m^T z})$ are unknown. The problem should thus be formulated as the minimization of

$$J_{\mathrm{ICA}}(\mathbf{B}) = \sum_{m=1}^{M} E\left(G_m(\mathbf{b_m^T z})\right), \tag{2.54}$$

where $G_m(.)$ is an unknown function. The functions should be such that the minimization of J_{ICA} leads to a matrix $\mathbf{B}$ which makes the outputs y_m as independent from each other as possible. Because we use a rotation to achieve that goal, it is not reasonable to assume that full statistical independence of the system outputs can be achieved. What can be demanded, however, is that the functions G_m are such that if $\mathbf{z}$ had been created by a linear mixture of independent components s_m, i.e.

$$\mathbf{z} = \tilde{\mathbf{A}}\mathbf{s}, \tag{2.55}$$

for a $M \times M$ mixture matrix $\tilde{\mathbf{A}}$, that then, a matrix $\mathbf{B}^*$ is obtained at the minimum with $\mathbf{B}^* = \tilde{\mathbf{A}}^{-1}$. The mixture model can be written as

$$\mathbf{z} = \sum_{m=1}^{M} \tilde{\mathbf{a}}_\mathbf{m} \frac{1}{\alpha_m} \alpha_m s_m, \tag{2.56}$$

for some α_m so that both the sign and the variance of the s_i are undetermined. In the following, the variance of the s_m are assumed to be one. This makes the mixture matrix $\tilde{\mathbf{A}}$ orthogonal and $\mathbf{B}^* = \tilde{\mathbf{A}}^T$.

It is shown in [Hyvärinen and Oja, 1998] that any even function G divides the independent components s_m, and hence the columns of $\tilde{\mathbf{A}}^T$, into two groups. The value

$$\beta(s_m) = E\left(s_m g(s_m) - g'(s_m)\right), \tag{2.57}$$

where g is the derivative of G, can be used to classify the s_m into those which have $\beta(s_m) > 0$ and those which have $\beta(s_m) < 0$. The minimization of $E\left(G(\mathbf{b}^T z)\right)$ under the constraint $||\mathbf{b}||_2 = 1$ allows to find the columns of $\tilde{\mathbf{A}}^T$ at local minima if the corresponding s_m have a value $\beta(s_m) < 0$. The remaining columns, associated with the s_m that have a value $\beta(s_m) > 0$, can be found at local maxima.

The minimization of J_{ICA} of Equation (2.54) can thus be based on a single function G through the determination of the local extrema of

$$J_G(\mathbf{b}) = E(G(\mathbf{b}^T z)) \tag{2.58}$$

under the unity norm constraint on $\mathbf{b}$. Statistical considerations can guide the choice of G. In practice, the expectation is calculated as the sample average so that columns $\hat{\mathbf{b}}$, which are obtained through the use of sample averages, differ from the optimal columns of $\mathbf{B}^*$. The function G can be chosen such that the mean squared error $E(||\hat{\mathbf{b}} - \mathbf{b}^*||^2)$ is minimized. This is done in [Hyvärinen, 1997], and the optimal $G^o(y)$ is

$$G^o(y) = c_1 \log p(y) + c_2 y^2 + c_3, \tag{2.59}$$

where $p(.)$ is the probability density of an independent component s. If this relation is applied to a density function of the form $p(s) = C_1 \exp\left(C_2 |s|^\alpha\right)$, one obtains

$$G^o(y) \propto |y|^\alpha \tag{2.60}$$

when the constants c_2 and c_3 are set to zero. For $0 < \alpha < 2$, the probability density is called supergaussian, for $\alpha = 2$, it is a Gaussian, and for $\alpha > 2$, it is called subgaussian. For supergaussian independent components s_m, which occur in the

context of sound and image analysis, variants of G^o which are more robust to outliers are shown in Table 2.2. Random variables with a supergaussian distribution are also called sparse random variables because they take very small absolute values or very large values more often than a Gaussian random variable. For compensation, values in between are relatively rare. For a sparse random variable, the maximization of non-Gaussianity becomes then maximization of sparseness.

Iterative optimization We review numerical methods to find the local maxima and minima of J_G of Equation (2.58) under the constraint that the obtained column vectors $\mathbf{b}$ are orthogonal to each other and of unit norm. We distinguish between gradient and quasi-Newton based iteration algorithms.

The gradient of J_G is

$$\frac{\delta J_G(\mathbf{b})}{\delta \mathbf{b}} = E\left[\mathbf{z}g(\mathbf{b}^\mathbf{T}\mathbf{z})\right]. \tag{2.61}$$

For $\beta > 0$ in Equation (2.57), J_G must be maximized so that the increment $\triangle\mathbf{b}$ follows the direction of the gradient. For $\beta < 0$, J_G must be minimized so that the increment should happen along the opposite direction. Dropping the expectation, we obtain the stochastic gradient rule with stepsize μ

$$\triangle\mathbf{b} = \mu \text{sign}(\beta)\mathbf{z}g(\mathbf{b}^\mathbf{T}\mathbf{z}). \tag{2.62}$$

If the sign of β is known, it can be substituted in the above equation. If for example the function $g(.) = \tanh(.)$, the sign of β is < 0 for a supergaussian independent component s while it is > 0 for a subgaussian independent component. If the sign is not known, β can be estimated via

$$\triangle\beta = \mathbf{b}^\mathbf{T}\mathbf{z}g(\mathbf{b}^\mathbf{T}\mathbf{z}) - g'(\mathbf{b}^\mathbf{T}\mathbf{z}) - \beta. \tag{2.63}$$

The learning rule can be applied for each column of the matrix $\mathbf{B}$. Orthonormality of the columns can be assured by the inclusion of additional terms in the learning rule, which iteratively orthonormalize the columns, by explicit orthonormalization after each iteration, or by means of a deflationary approach. In case of orthonormalization after each iteration, the gradient based learning rule becomes

$$\triangle\mathbf{B} = \mu\left[\mathbf{z}g(\mathbf{B}^\mathbf{T}\mathbf{z})^T \text{diag}(\text{sign}(\beta_m))\right], \tag{2.64}$$

$$\mathbf{B} \leftarrow \mathbf{B}(\mathbf{B}^\mathbf{T}\mathbf{B})^{-1/2}, \tag{2.65}$$

where μ is the step size. For iterative orthonormalization, the rule becomes [Hyvärinen and Oja, 1998]

$$\triangle\mathbf{B} = \mu\left[\mathbf{z}g(\mathbf{B}^\mathbf{T}\mathbf{z})^T \text{diag}(\text{sign}(\beta_m)) + 0.5(\mathbf{B} - \mathbf{B}\mathbf{B}^\mathbf{T}\mathbf{B})\right], \tag{2.66}$$

where the function $g(.)$ operates element-wise on the vector $\mathbf{B^T z}$ and each β_m is obtained from an iteration as in Equation (2.63). For the deflationary approach, and a recurrent least squares (RLS) based iterative optimization which leads to an automatic iteration dependent choice of the step size μ, see [Hyvärinen et al., 2001a].

The quasi-Newton based iteration scheme uses an approximation of the Hessian matrix of $J_G(\mathbf{b})$ in Equation (2.58). The update rule is

$$\mathbf{b_m} \leftarrow \mathbf{b_m} - \left[\frac{\delta^2 J_G}{\delta \mathbf{b}_m^2}\right]^{-1} \frac{\delta J_G}{\delta \mathbf{b_m}} \quad m = 1 \dots M. \tag{2.67}$$

The Hessian matrix is given by

$$\left[\frac{\delta^2 J_G}{\delta \mathbf{b}_m^2}\right] = E\left[\mathbf{z}\mathbf{z}^{\mathbf{T}} g'(\mathbf{b_m^T z})\right], \tag{2.68}$$

which is in [Hyvärinen, 1999] approximated as $E(\mathbf{z}\mathbf{z}^{\mathbf{T}})E(g'(\mathbf{b_m^T z}))$. Since $\mathbf{z}$ is white, the approximation of the Hessian becomes

$$\left[\frac{\delta^2 J_G}{\delta \mathbf{b}_m^2}\right] \approx E\left[g'(\mathbf{b_m^T z})\right] \mathbf{1}_M \tag{2.69}$$

which can easily be inverted. From that, the following approximative Newton iteration is obtained

$$\mathbf{b_m} \leftarrow E\left[\mathbf{z} g(\mathbf{b_m^T z})\right] - E(\left[g'(\mathbf{b_m^T z})\right] \mathbf{b_m} \quad m = 1 \dots M \tag{2.70}$$

$$\mathbf{B} \leftarrow \mathbf{B}(\mathbf{B^T B})^{-1/2}, \tag{2.71}$$

see [Hyvärinen, 1999] for a proof of convergence. Here, orthogonality of $\mathbf{B}$ is assured by the explicit orthogonalization after each update of the columns of $\mathbf{B}$. This is a symmetric approach where all the columns are learned at the same time. The columns could also be orthonormalized by means of a deflationary approach where each columns is learned one after another. Note that the gradient based iterative optimization is an online algorithm while the Newton iteration works in batch-mode.

2.3 Data representation with time structure

We discuss here the representation of data $\mathbf{x}$ which has a time structure. Since the representation has time structure, we assume that the data is also time dependent so that $\mathbf{x}(t)$ is a random process. In Section 2.3.1, we formulate the data

$G(y)$	$\frac{1}{a}\log\cosh(ay)$	$-\exp(-y^2/2)$	$\lvert y\rvert^\alpha$
$g(y)$	$\tanh(ay)$	$y\exp(-y^2/2)$	$\alpha\text{sign}(y)\lvert y\rvert^{\alpha-1}$

Table 2.2: Summary of functions G which are used as cost functional in ICA in case of supergaussian independent components s_m. The function g is the derivative of G. Constant a is in the interval [1 2], and $\alpha \in$ (0 2). The first function G is a good general purpose function. The second function is used for the highly supergaussian case ($\alpha < 1$). The last one may lead to numerical instabilities if $\alpha < 1$. See [Hyvärinen, 1997].

representation problem as an optimization problem. We review then two different approaches to solve the optimization problem. In Section 2.3.2, we present a method which is based on the decomposition of a function into a linear expansion of elements which belong to a certain fixed function-dictionary. The optimization problem is then solved by iteratively optimizing the form of the dictionary. The method of Section 2.3.3 makes more assumptions on the statistical properties of the data but does not need such a nested loop for the optimization.

2.3.1 Linear representation with time dependent kernels

As in Section 2.2, the data representation problem is formulated as an optimization problem. Given is a collection of T zero mean N dimensional random vectors $\{\mathbf{x}(t)\}_{t=1}^T$, where each random vector is of the same form as in Section 2.2. This collection of random vectors defines a vector valued random process $\mathbf{x}(t)$. It can be represented with respect to the standard basis $\{\mathbf{e}^{(n)}\}_{n=1}^N$ of $\mathbb{R}^N$ as

$$\mathbf{x}(t) = \sum_{n=1}^{N} x_n(t)\mathbf{e}^{(n)}. \tag{2.72}$$

The problem is to find a representation of $\mathbf{x}(t)$ in the form of

$$\hat{\mathbf{x}}(t) = \sum_{\gamma} s_\gamma \mathbf{h}_\gamma(t), \tag{2.73}$$

where $\mathbf{h}_\gamma \in \mathbb{R}^N$, so that the expected sum of the errors

$$e(t) = ||\mathbf{x}(\mathbf{t}) - \hat{\mathbf{x}}(\mathbf{t})||^2 \tag{2.74}$$

is minimized. In short, the representation should be such that

$$J = E\left(\sum_t e(t)\right) \tag{2.75}$$

is minimal. In the following, we review the case where $\gamma = (m, \tau)$ and $\mathbf{h}_\gamma(t)$ has the form $\mathbf{h}_m(t - \tau)$. Then, the approximation $\hat{\mathbf{x}}$ becomes

$$\hat{\mathbf{x}} = \sum_{m=1}^{M} \sum_{\tau} s_m(\tau) \mathbf{h}_m(t - \tau). \tag{2.76}$$

The coefficients $s_m(\tau)$ and the elements $\mathbf{h}_m(t)$ are unknown and have to be determined such that J in Equation (2.75) is minimized. The coefficients $s_m(\tau)$ are stochastic while the $\mathbf{h}_m(t)$ are the same for each realization of the random process $\mathbf{x}(t)$. The time dependent representation includes the time-independent case, which we have considered in Section 2.2. Its methods can therefore also be applied to the time-independent case.

2.3.2 Iterative optimization based on matching pursuit

We present here a method which uses in an inner loop an algorithm called matching pursuit [Mallat and Zhang, 1993] to find for a given set of $\{\mathbf{h}_m(t)\}_{m=1}^M$ coefficient pairs $\{(s_m^n, \tau_m^n), n = 1 \dots N_m\}_{m=1}^M$ which lead to an approximation of any instance of $\mathbf{x}(t)$. These atoms are then updated in an outer loop by a steepest gradient algorithm on J of Equation (2.75). In this outer loop, the coefficient pairs are held fixed and are not subject to a modification.

This idea has been used in [Smith and Lewicki, 2005, 2006] for the representation of sound. For sound, the dimension N equals one so that $\mathbf{h}_m(t)$ becomes a scalar $h_m(t)$. It has also been used for the representation of images, i.e. time-independent data [Perrinet, 2004]. In that case $T = 1$, and the index $\gamma = m$ and the atom $\mathbf{h}_m(t)$ becomes a time independent vector $\mathbf{h}$.

Matching pursuit loop

Matching pursuit decomposes a realization of the random process $\mathbf{x}(t)$ in a recursive fashion. Mathematical properties and convergence issues can be found [Mallat and Zhang, 1993]. We present here the application of matching pursuit to sound, as in [Smith and Lewicki, 2005, 2006]. The algorithm starts with $R^0 x(t) = x(t)$. In the k-th iteration, the inner product between $R^k x(t)$ and each of the $h_m(t - \tau)$ is calculated,

$$y_m(\tau) = \sum_t R^k x(t) h_m(t - \tau) \quad m = 1 \dots M. \tag{2.77}$$

The $h_m(t - \tau)$ are assumed to be normalized to unit norm. Parameter τ is here a positive integer. The inner product which is parameterized with τ becomes thus

the convolution between $R^k x(t)$ and $\check{h}_m(t) = h_m(-t)$. The next step in the k-th iteration consists then of finding

$$(m^*, \tau^*) = \text{argmax}_{m,\tau} |y_m(\tau)|. \tag{2.78}$$

Assuming that during the iterations till now, i.e. from 0 till k-1, m^* was already obtained $n-1$ times, one sets $m = m^*$ and defines

$$s_m^n = y_{m^*}(\tau^*) \tag{2.79}$$
$$\tau_m^n = \tau^*. \tag{2.80}$$

The maximal projection of $R^k x(t)$ on the shifted atoms is then subtracted from $R^k x(t)$ to obtain $R^{k+1} x(t)$

$$R^{k+1} x(t) = R^k x(t) - s_m^n h_m(t - \tau_m^n). \tag{2.81}$$

Then, iteration $k+1$ starts. The recursion stops in iteration k_s if $y_{m^*}(\tau^*)$ is smaller than a fixed threshold. As a result of the recursion, one obtains a decomposition of $x(t)$ into

$$x(t) = \sum_{m=1}^{M} \sum_{n=1}^{N_m} s_m^n h_m(t - \tau_m^n) + R^{k_s} x(t). \tag{2.82}$$

In virtue of the orthogonal projection and the recursive decomposition, the approximation error $||R^{k_s} x||^2 = \sum_t |R^{k_s} x(t)|^2$ equals

$$||R^{k_s} x||^2 = ||x||^2 - \sum_{m=1}^{M} \sum_{n=1}^{N_m} (s_m^n)^2 \tag{2.83}$$

Steepest gradient loop

Direct calculation of the gradient of J in Equation (2.75) yields

$$\frac{\delta J}{\delta h_m(t)} = -E \left(\sum_{n=1}^{N_m} s_m^n e(t - \tau_m^n) \right). \tag{2.84}$$

In [Smith and Lewicki, 2006], a stochastic gradient based minimization of J is performed so that after the decomposition of a realization of $x(t)$, i.e. after each sample, the update

$$\triangle h_m(t) = \mu \left[\sum_{n=1}^{N_m} s_m^n e(t - \tau_m^n) \right] \quad m = 1 \ldots M \tag{2.85}$$

takes place.

The approximation error $||R^{k_s}x||^2$ of Equation (2.83) is actually equal to $\sum_t e(t)$. One can see from Equation (2.83) that the minimization of J in Equation (2.75) is equal to the maximization of $E\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m}(s_m^n)^2\right)$. This is done in [Perrinet, 2004] by means of a stochastic gradient algorithm.

2.3.3 Iterative optimization based on activity bubbles

Here, we discuss the decomposition of the N dimensional random process $\mathbf{x}(t)$ when the number of atoms is N. The presented method is due to [Hyvärinen et al., 2003]. It works with input $\mathbf{x}(t)$ being spatially decorrelated and normalized to unit variance for each time step t. Special emphasis is on the case where the decomposition elements $\mathbf{h}_m(t)$ are time independent vectors. We limit the review to that case and denote them by $\mathbf{b}_m$. The approximation $\hat{\mathbf{x}}(t)$ becomes then

$$\hat{\mathbf{x}}(t) = \sum_{m=1}^{N} s_m(t)\mathbf{b}_m \tag{2.86}$$

$$= \mathbf{B}\mathbf{s}(t), \tag{2.87}$$

where the matrix $\mathbf{B}$ contains the vectors $\mathbf{b}_m$ as columns. The cost functional J of Equation (2.75) can be minimized by demanding that $\mathbf{B}$ is invertible and by setting $\mathbf{s}(t) = \mathbf{B}^{-1}\mathbf{x}(t)$.

Cost functional for B

The unknown matrix $\mathbf{B}$ is found by means of maximum likelihood estimation making the following assumptions on the $s_m(t)$

1. The $s_m(t)$ are sparse, uncorrelated and of unit variance for all t and m. Sparseness has been defined in Section 2.2 where we discuss independent component analysis (ICA).

2. The squares $(s_m(t))^2$ are temporally correlated for all m. In words, the $s_m(t)$ are changing smoothly over time and are said to be temporally coherent.

3. The squares $(s_m(t))^2$ are spatially, i.e. over the index m, correlated for all t. In words, the $s_m(t)$ are changing smoothly as the index m varies (spatial coherence).

Sparseness, together with temporal and spatial smoothness (coherence), lead then to activity "bubbles" of the coefficients $s_m(t)$. For more details on temporal coherence see [Hurri and Hyvärinen, 2003a,b]. Spatial energy correlation has also

been treated in [Hyvärinen and Hoyer, 2000; Hyvärinen et al., 2001b] as an extension of ICA which is presented in Section 2.2. The three assumptions are in mathematical form formulated as

$$s_m(t) = z_m(t) f\left(\sum_{j=1}^{N} \gamma(m,j) \sum_{\tau} \phi(\tau) u_j(t-\tau)\right), \tag{2.88}$$

where $z_m(t)$ are identically, independently distributed (iid) zero mean, unit variance Gaussian random variables. The $u_j(t)$ are iid nonnegative, sparse random variables with marginal density p_u. The function $\phi(t)$ is a low pass filter, and $\gamma(m,j)$ is a neighborhood function taking on the values zero or one. The function f is $f(y) = y^{-1/2}$. The whiteness assumption on $\mathbf{x}(t)$ for all t and spatial decorrelation of the $s_m(t)$, together with the normalization to unit variance, restricts the set of possible matrices $\mathbf{B}$ to orthogonal matrices. Its inverse becomes then $\mathbf{B^T}$ and $s_m(t) = \mathbf{b_m^T x}(t)$.

The likelihood $p_{\mathbf{x}(t)}$ of a realization of $\mathbf{x}(t)$ is

$$p_{\mathbf{x}(t)} = |\det \mathbf{B}^{-1}| p_{\mathbf{s}(t)}, \tag{2.89}$$

which equals $p_{\mathbf{s}(t)}$ due to the orthogonality of $\mathbf{B}$. In [Hyvärinen et al., 2003] a lower bound for the log likelihood $\log p_{\mathbf{s}(t)}$ is then derived from Equation (2.88). It is

$$\log p_{\mathbf{s}(t)} \geq \sum_{t=1}^{T} \sum_{m=1}^{N} G(y_m(t)), \tag{2.90}$$

where

$$y_m(t) = \sum_{j} \gamma(m,j) \sum_{\tau} \check{\phi}(\tau) \left(s_j(t-\tau)\right)^2, \tag{2.91}$$

$$G(y) = \log \int \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}uy\right) p_u(u) \sqrt{\phi(0)u} \mathrm{d}u. \tag{2.92}$$

Function G depends on the probability distribution p_u. It may be analytically evaluated in some special cases, but in [Hyvärinen et al., 2003] it was taken as $G(y) = -\sqrt{y}$. A more stable version is $G(y) = -\sqrt{y+\epsilon}$. where ϵ is a small constant. The function $\check{\phi}$ is the time inverted ϕ, i.e. $\check{\phi}(t) = \phi(-t)$. As $s_j(t-\tau) = \mathbf{b_j^T x}(t-\tau)$, where the vector $\mathbf{b_j}$ is the j-th column of $\mathbf{B}$, one obtains the cost functional J_b which is to be maximized:

$$J_b(\mathbf{B}) = \left\langle \sum_{t=1}^{T} \sum_{m=1}^{N} G(y_m(t)) \right\rangle, \tag{2.93}$$

where the symbols $<>$ indicate average over the samples of $\mathbf{x}(t)$.

Comparison to ICA cost functional

In Equation (2.54) and (2.58) of Section 2.2, we have seen that for the maximization of sparseness of the output, one had to find column vectors $\mathbf{b}_m$ such that

$$J_{\mathrm{ICA}}(\mathbf{B}) = \sum_{m=1}^{M} E\left(G(y_m)\right), \tag{2.94}$$

where $y_m = \mathbf{b_m^T z}$ and $G(y) = |y|^\alpha$, is minimized. The cost functional J_b generalizes this equation. If ϕ and $\gamma(j,m)$ in Equation (2.91) are such that $y_m(t) = s_m(t)^2$, i.e. without pooling over space and integration over time, one obtains $G(y_m(t)) = -\sqrt{s_m(t)^2} = -|s_m(t)|$. If we choose in J_{ICA} $\alpha = 1$, and note that maximization of J_b with $G(y) = -|y|$ is the same as minimization with $G(y) = |y|$, we see that the optimization problem is the same.

Gradient ascent algorithm

The cost functional J_b is maximized by a gradient ascent algorithm. The matrix $\mathbf{B}$ is held orthogonal by explicit orthonormalization of its columns in each iteration. Calculation of the gradient of J_b with respect to $\mathbf{b}_m$ gives

$$\frac{\delta J_b}{\delta \mathbf{b}_m} = 2\left\langle \sum_t \left(\sum_{i=1}^{N} g(y_i(t))\gamma(m,i) \right) \sum_\tau \check{\phi}(\tau) s_m(t-\tau)\mathbf{x}(t-\tau) \right\rangle, \tag{2.95}$$

where g is the derivative of G. The product of input and output $s_m(t)\mathbf{x}(t)$ is smoothed out over time by the convolution with $\check{\phi}$. The smoothed product is then weighted by a term which reflects the differential activity in the neighborhood and integrated over time. The update rule becomes then

$$\triangle \mathbf{b}_m = \mu \frac{\delta J_b}{\delta \mathbf{b}_m} \quad m = 1 \ldots N \tag{2.96}$$

$$\mathbf{B} \leftarrow \mathbf{B}(\mathbf{B^T B})^{-1/2}. \tag{2.97}$$

The rule can be simplified if in J_b, the summation over t is replaced by the fixed time $T/2$. This simplification is done in [Hyvärinen et al., 2003]. Calculation of the gradient with respect to this cost functional, called $\tilde{J}_b$, leads to

$$\frac{\delta \tilde{J}_b}{\delta \mathbf{b}_m} = 2\left\langle \sum_t \left(\sum_{i=1}^{N} g(y_i)\gamma(m,i) \right) s_m(t)\mathbf{x}(t) \right\rangle, \tag{2.98}$$

where $y_i = y_i(T/2)$. Differences to Equation (2.95) are that there is no smoothing with $\check{\phi}$ and that the differential activity of the neighborhood is time independent. The product $s_m(t)\mathbf{x}(t)$ is thus equally weighted in the sum over t.

2.4 Summary

We provide here a summary of the most important points of this chapter. We start with the connection between data representation and the change of basis (Section 2.1). Then, we summarize the idea of obtaining a change of basis which is adapted to the data by the modeling of the data representation problem as an optimization problem (Section 2.2.1 and Section 2.3.1). Finally, we summarize the different algorithms for data representation without time structure (Section 2.2.2 and Section 2.2.3) and with time structure (Section 2.3.2 and Section 2.3.3).

Data representation through an appropriate change of basis

1. Data representation is data organization so that meaningful information can be extracted. The representation should make structuredness of the data explicit. This allows for knowledge discovery.

2. The re-organization of the data $\mathbf{x}$ can be considered as the representation of $\mathbf{x}$ with respect to another basis than the standard basis.

3. The choice of the basis should be based on the structure of the data. There are thus two maps: The first map, the selectivity transform, maps the old basis to the new one. The second map, the coordinate transform, maps the coordinates of $\mathbf{x}$ with respect to the old basis to the coordinates with respect to the new basis.

Change of basis as the solution of an optimization problem

1. The adaptation of the basis to the structure of the data is achieved through the formulation of the data representation problem as an optimization problem.

2. The optimization parameters are the selectivity and the coordinate transform.

3. If the representation has no time structure, and if the cost functional is limited to the measurement of the accuracy of the representation, the solution for the selectivity and the coordinate transform is not unique. Additional constraints have to be added to obtain a unique solution.

Iterative optimization for data representation without time structure

1. Without additional constraints, the solution for the selectivity transform and the coordinate transform are defined up to an invertible matrix $\mathbf{C}$. In the framework of PCA, this matrix is constrained to be orthogonal.

2. There are algorithms which leave the orthogonal $\mathbf{C_o}$ undetermined. Examples are the Symmetric Error Correction (SEC) or subspace learning rule, the PAST algorithm, or the Stochastic Gradient Ascent (SGA) algorithm. Others, e.g. the PASTd, APEX, or Generalized Hebbian Algorithm (GHA) lead to a solution where $\mathbf{C_o}$ is a diagonal matrix with ± 1 as its diagonal elements.

3. Examples for additional constraints which make the solution unique are punishment of the norm of the columns of $\mathbf{W}$, or of the output coordinate vector $\mathbf{y}$. Another method to resolve the under-determinedness of the problem is to require statistical independence of the y_m (independent component analysis (ICA)). This method fixes the undetermined invertible matrix $\mathbf{C}$.

Iterative optimization for data representation with time structure

1. We have reviewed two methods for the iterative optimization. The first is based on matching pursuit and makes no assumptions about the statistical properties of the data. The second makes more statistical assumptions but the optimization has a simpler form.

2. Matching pursuit decomposes a function into a linear expansion of elements which belong to a certain fixed function-dictionary. The optimization, and thus the data representation problem is then solved by iteratively optimizing the form of the dictionary. There are thus two nested loops in this kind of optimization.

3. The second method makes the assumption that in the decomposition of $\mathbf{x}(t)$ into $\sum_m s_m(t)\mathbf{h_m}$, the coefficients $s_m(t)$ show spatio-temporal activity bubbles. The label "spatial" relates here to the index m. Activity bubble is related to spatial and temporal smoothness. Most of the time, the $s_m(t)$ are thought to be zero but if they are nonzero, they stay so for some time, and the neighbors behave in a similar fashion. With this assumption, the new data representation is learned by means of maximum likelihood estimation.

Chapter 3

Neuroscience-Background

3.1 Biophysics and modeling of neurons

The neuron is a cell. As a biological unit it has a wealth of properties which are of interest to science. In Section 3.1.1, we present biophysical properties which are related to the signaling capacity of the neurons. In Section 3.1.2, we review their energy consumption at rest and during signaling. In Section 3.1.3, we present models which describe neurons in mathematical form.

3.1.1 Biophysics of signaling

Neurons are important to brain science because they allow for signaling. We review here what biophysical quantity is used for signaling and what kind of biophysical mechanisms are used to transform the signal.

Biophysical origin of the signal

Charge separation The interior of a cell is protected from the extracellular environment by a membrane. The membrane is essentially impermeable, but it features pores which allow to access vital products in the extracellular space and to dispose of waste products.

There are two types of pores: ion channels and active transporters. Active transporters regulate the ion concentration (measured as molarity M, i.e. in moles per liter or $0.001m^3$) inside the cell. Table 3.1 shows that they produce concentration gradients across the cell membrane. Ion channels are pores which allow ions to cross the cell membrane down the concentration gradient, that has been established by the active transporters. There are different kinds of ion channels and they are selective for different kinds of ion types and have different opening-closing properties. The diffusion of ions through open channels leads to a charge separation. The membrane acts like a capacitor and as charge accumulates on either side of the membrane, an electrical potential difference (voltage) is built up. The ions, which diffuse down the concentration gradient, meet an opposing electrical force, and eventually, the diffusional force is balanced by the electrical drift force. The equilibrium potential for each ion type can be calculated by the Nernst equation

$$E = \frac{RT}{zF} \ln \frac{[\text{Ion}]_o}{[\text{Ion}]_i}, \tag{3.1}$$

with R the gas constant, F the Faraday's constant (total charge in 1 mole of monovalent ions), T the temperature in Kelvin, and z the valence of the ion.

When the difference between the intracellular and extracellular electrical potential (voltage V) equals the equilibrium potential E of a certain ion channel, no net current is flowing through it. That is why the equilibrium potential is also called reversal potential. Table 3.1 shows the equilibrium potentials for the major ions in the nerve cell. Figure 3.1 summarizes the process of charge separation, diffusion and building up of the equilibrium potential.

Current through the channels If for a given channel type i the voltage V exceeds the reversal potential, electrical drift current is flowing through it. This is modeled by Ohm's law as

$$I_i = g_i(V - E_i), \tag{3.2}$$

where g_i is the conductance of the channel. The gating and intrinsic properties of the channel type i can be incorporated into the conductance g_i so that it may depend on voltage V, intracellular quantities like the calcium (Ca^{2+}) concentration or extracellular quantities and time. Figure 3.2 illustrates that for $V > E_i$, a positive channel current I_i moves ions from inside the cell into the extracellular space which effectively discharges the capacitor. The voltage V decreases towards E_i.

Equivalent circuit There are several kinds of channels in a membrane patch. The ion currents through the different channels change the amount of charge stored on the capacitor. This changes the voltage V across the membrane and the reversal potentials of the ion channels. A change in the voltage V may affect in turn the ion currents if they are voltage gated, and the value of the reversal potential affects the ion currents directly. Figure 3.3 shows the correspondent electrical equivalent circuit for the case of Na^+ and K+ ions. Using the principles of charge and mass conservation, the following differential equations can be derived from Figure 3.3

$$\dot{V} = -\frac{1}{C}\big(g_{Na}(V - E_{Na}) + g_K(V - E_K) + I_p\big), \tag{3.3}$$

$$[\dot{Na}]_i = \frac{S}{FVol}\big(-g_{Na}(V - E_{Na}) - 3I_p\big), \tag{3.4}$$

$$[\dot{K}]_i = \frac{S}{FVol}\big(-g_K(V - E_K) + 2I_p\big), \tag{3.5}$$

where S is the surface of the membrane patch, Vol the volume of the patch and F is the Faraday constant. Reversal potentials E_{Na} and E_K depend in virtue of the Nernst equation on $[Na]_i$ and $[K]_i$, respectively. The pump current I_p depends

also on the ion concentrations [Gerster et al., 1997]. The conductances g_{Na} and g_K depend both on V and through their intrinsic dynamics on time t, and may be on extracellular quantities as well. Voltage V is the biophysical quantity which is used as the signal. The ion concentrations $[Na]_i$ and $[K]_i$ are to used to store the energy which is needed for the signal transformation.

Biophysical origin of the signal transformation

Signal transformation means shaping the membrane voltage V. This is done by opening and closing ion channels, i.e. increasing or decreasing the conductances g_i with the right timing. What kind of signal transformation results, depends on the factors which cause the opening and closing of the ion channels.

If the surface to volume ratio S/Vol is small, the time scales in the differential equation (3.3) for V and the differential equations (3.4) and (3.5) for the ion concentrations are different. The ion concentrations can be considered as constant for the analysis of the behavior of the signaling variable V in Equation (3.3). At the equilibrium with respect to the fast time scale, V equals

$$V_{eq} = \frac{g_{Na}E_{Na} + g_K E_K - I_p}{g_{Na} + g_K}. \tag{3.6}$$

If for example g_K dominates g_{Na} the equation can be approximated as

$$V_{eq} \approx E_K - \frac{g_{Na}}{g_K}(E_K - E_{Na}), \tag{3.7}$$

for g_{Na}/g_K small.

Graded signals If, given an extracellular stimulus, the conductance of an ion channel increases with the strength of the stimulus, a graded signal results. The simplest example are the mechanosensitive channels in the "swimming neuron", the paramecium [Greenspan, 2007]. Figure 3.4 shows that mechanosensitive K^+ channels in the posterior part of the paramecium lead to a graded decrease of the membrane voltage V (hyperpolarization) upon stimulation. Mechanical stimulation increases the K^+ conductance so that K^+ flows out of the cell and the membrane voltage V goes towards $E_K < 0$ (Equation (3.7)). In the anterior part, however, mechanosensitive calcium channels are present. Increasing their conductance by mechanical stimulation drives Ca^{2+} into the cell and V goes towards E_{Ca} (Analogue to Equation (3.7)). Since E_{Ca} is larger than 100mV an increase of V (depolarization) results.

Action potentials For graded signals, if the stimulation ceases, the conductance and voltage return to their original state. There is no amplification. In many cases, however, the conductance of the ion channels depends on the voltage V. The "gating variables" of the channel either increase with increasing V (activation), which leads to larger conductance g, or they decrease (inactivation), which leads to smaller g. Table 3.2 shows that depending on the direction of the ion current, i.e. inward or outward, a self-exciting positive feedback loop or a self-limiting negative feedback loop between g and V results.

The voltage-sensitive calcium channel of the paramecium in Figure 3.4 leads to a positive feedback loop. Under normal conditions, the intracellular concentration of Ca^{2+} is much less than the extracellular concentration so that the Ca^{2+} current is inward. As the channel conductance increases with increasing voltage V the conditions for a self-exciting positive feedback loop are given. The membrane voltage V increases till the reversal potential of Ca^{2+}. With the inflow of Ca^{2+}, the channel inactivates [Brehm and Eckert, 1978] and the voltage returns to its rest state. This trajectory which is initialized by a self-amplifying positive feedback loop is called action potential or spike. In the paramecium, a new spike can only be triggered when the Ca^{2+} has been pumped out of the cell.

Ion	Extracellular concentration (mM)	Intracellular concentration (mM)	$\frac{[\text{Ion}]_o}{[\text{Ion}]_i}$	Equilibrium potential (mV)
Na^+	145	5-15	29-9.7	90-61
K^+	5	140	0.03	-90
Cl^-	110	4	27.5	-89
Ca^{2+}	2.5-5	10^{-4}	2.5-5 10^4	136-146
A^-	25	147	0.5	cannot diffuse

Table 3.1: Charge separation and equilibrium potentials in a typical mammalian neuron. Temperature $T = 37\,^\circ\text{C}$. Modified from [Izhikevich, 2006]

3.1.2 Energy consumption

In the differential equations (3.3) to (3.5) for the signaling variable V and the ion concentrations, we have analyzed the fast time scale for the evolution of V. While the pump current can be neglected for the analysis of the behavior of the signal in this short time interval, during which the signal changes abruptly, its action

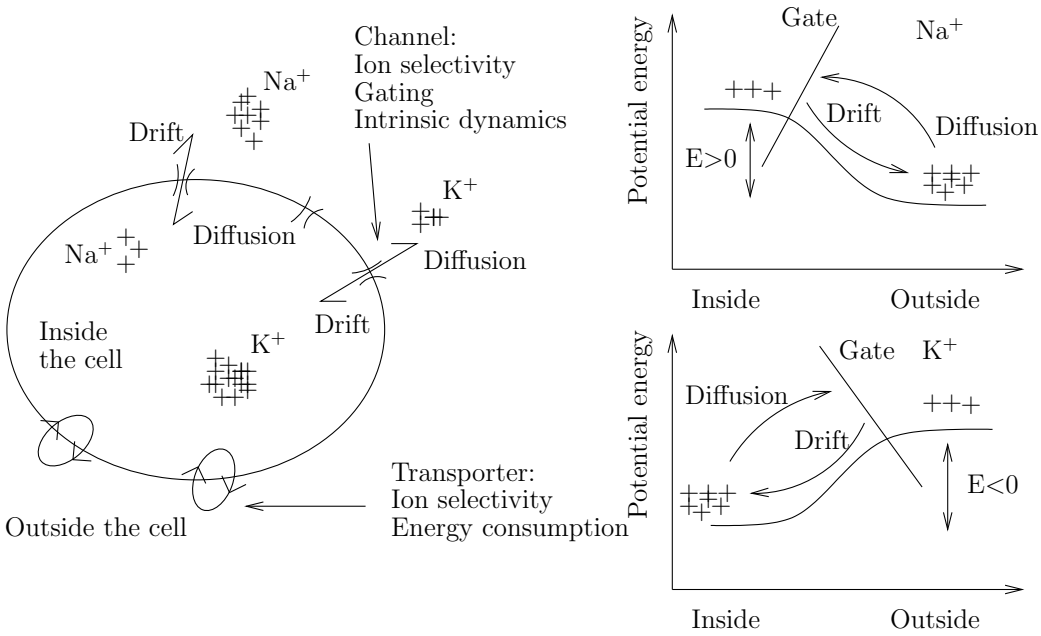


Figure 3.1: Transporters create a concentration gradient of the ions across the membrane. Diffusion of the ions across channels leads to charge separation and building of a potential difference. The resulting electrical drift current balances the diffusion current when the potential difference (voltage) V, measured from inside the cell to outside the cell, equals the equilibrium potential E. The net current is zero at the equilibrium potential. For clarity of the figure, only the sodium (Na^+) and potassium (K^+) ions are shown. For each ion type there is an associated transporter system. All transporters consume either directly or indirectly energy (ATP). An example is the sodium-potassium pump which moves in every cycle 3 sodium ions out of the cell and imports 2 potassium ions. For every different kind of channel, there is a different equilibrium potential according to the types of ions it lets pass. Several channels let through multiple kinds of ions. In that case the equilibrium potential E is not determined by the Nernst equation but by the voltage V where the net current through the channel is zero. An important aspect of a channel is its gating property. A channel might only let ions pass if a certain gating condition is fulfilled. Gating conditions are the magnitude of the membrane voltage V, the concentration of intracellular quantities like calcium (Ca^{2+}), or the extracellular presence of certain ligands, or mechanical stretch of the cell membrane.

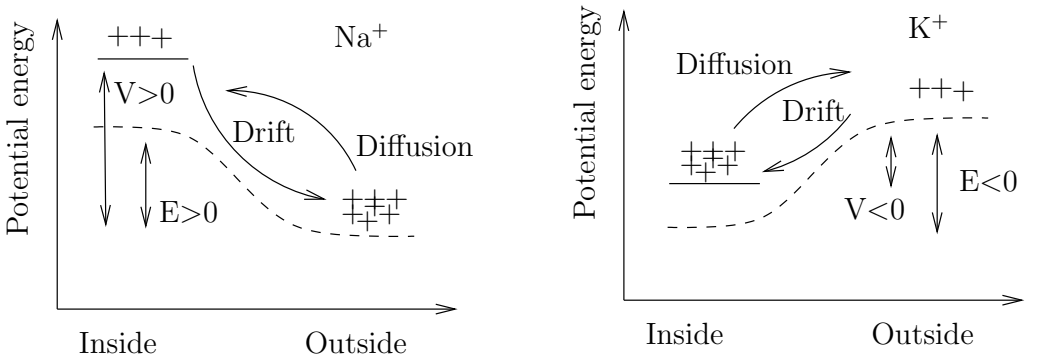


Figure 3.2: When the membrane voltage V exceeds the reversal potential E_i an outward current I_i results. Left: If the intracellular potential is increased to V, the drift current becomes stronger than the diffusion current so that Na^+ ions flow out of the cell. Right: If the intracellular potential is increased to V, the drift current is reduced so that K^+ ions diffuse out of the cell. In both cases an outward current results which removes positive ions from the intracellular space. This decreases the intracellular potential V towards E_i.

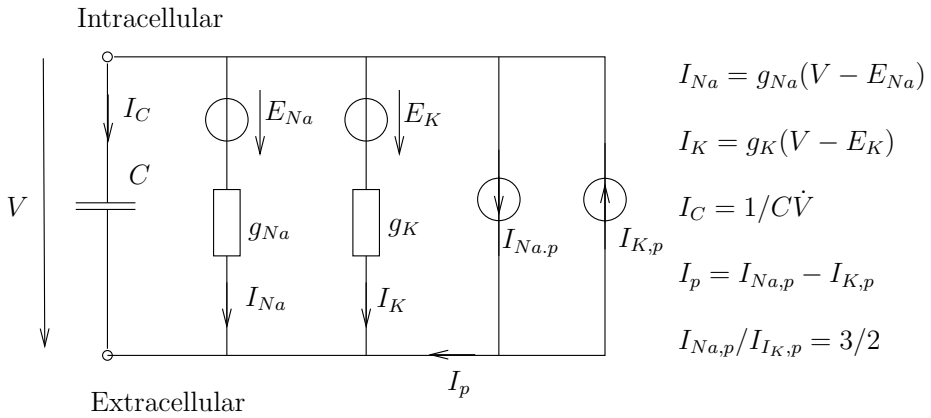


Figure 3.3: Equivalent circuit for a patch of membrane with Na^+ and K^+ channels and a Na^+ / K^+ pump. The pump extrudes in one cycle 3 Na^+ ions and imports 2 K^+. In each cycle one unit of energy (ATP) is consumed.

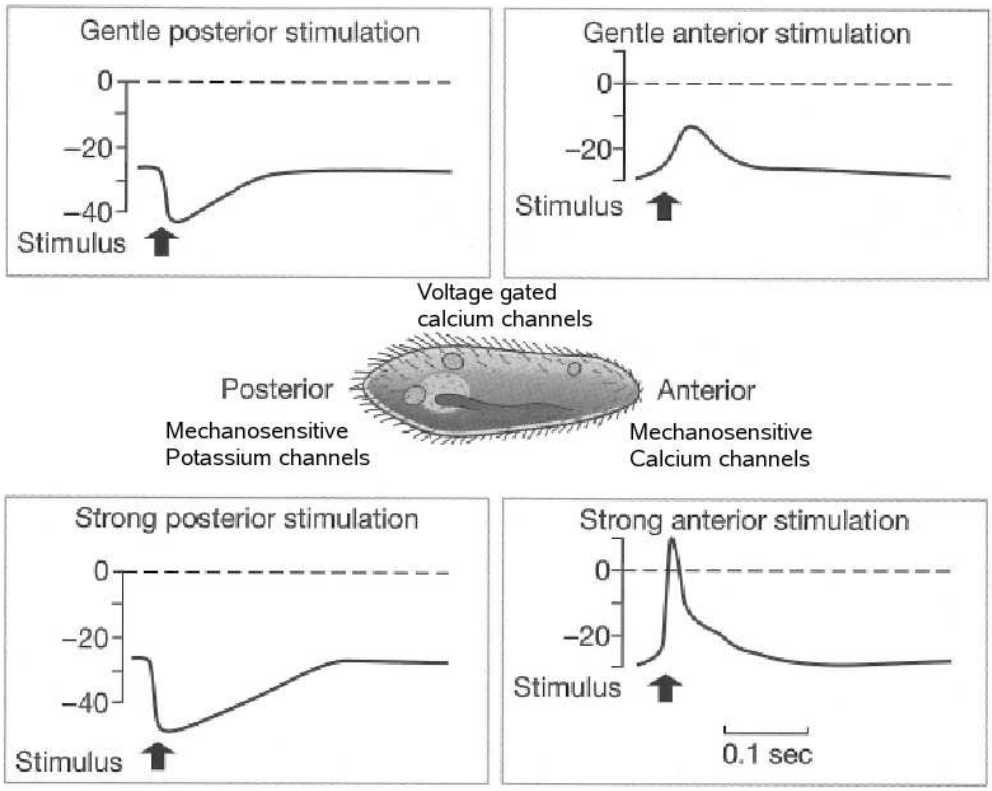


Figure 3.4: Ion channels in the paramecium and graded signaling with mechanosensitive K^+ and Ca^{2+} channels. The membrane of the paramecium contains in an uniformly distributed fashion voltage-gated Ca^{2+} channels. In the anterior part, mechanosensitive Ca^{2+} can also be found. In the posterior part, they are lacking but mechanosensitive K^+ channels are present. Adapted from [Greenspan, 2007].

	Inward current	Outward current
Activating gating	Positive feedback	Negative feedback
Inactivating gating	Negative feedback	Positive feedback

Table 3.2: Voltage dependent ion channels may lead to positive or negative feedback loop between voltage and gating. Gates which increase the conductance g_i of the channel of type i with increasing voltage V are called activating. Gates which decrease the conductance are called inactivating. Inward currents charge the capacitor and increase thus the membrane voltage V while outward currents decrease the voltage V. If for example an inward current flows through a channel with a gate which activates, a positive feedback loop results. For positive feedback loops, the voltage V is pushed towards the reversal potential E_i of the channel. For negative feedback loops, on the other hand, the voltage V returns to its equilibrium value.

cannot be neglected in the long term. First, we emphasize the difference between transporters and ion channels. Then we show that the pump current allows to calculate the energetic cost of the rest state and the signaling. Finally, we present a short cost balance.

Transporters vs. ion channels

Transporters and ion channels are complementary. Transporters store energy in the form of ion concentration gradients while the currents through the ion channels, which lead to signal transformation, dissipate the stored energy. The consumed energy becomes visible in the degradation of the concentration gradient of the ions involved in the signaling. The slow time scale describes the evolution of the ion concentrations and their maintenance through the pump current. We consider here the Na^+/K^+ pump which consumes in each pump cycle one unit of energy (ATP), see Figure 3.3.

Cost of the rest state

Rest state means that no signaling event has happened for a long time so that the sodium and potassium concentrations have reached their equilibrium value $[\bar{Na}]_i$ and $[\bar{K}]_i$, respectively. These concentration values determine the reversal potentials $E_{Na}([\bar{Na}]_i) = \bar{E}_{Na}$ and $E_{Na}([\bar{Na}]_i) = \bar{E}_{Na}$. From Equation (3.4), it

can be seen that the pump current I_p verifies at the equilibrium

$$\bar{I}_p = -\frac{g_{Na}}{3}(\bar{V}_{eq} - \bar{E}_{Na}), \tag{3.8}$$

where $\bar{V}_{eq}$ is the voltage at rest (long term equilibrium). From Equation (3.3), it follows that $\bar{V}_{eq}$ verifies

$$\bar{V}_{eq} = \frac{g_{Na}\bar{E}_{Na} + g_K\bar{E}_K - \bar{I}_p}{g_{Na} + g_K}. \tag{3.9}$$

The two equations can be solved to obtain $\bar{I}_p$ and $\bar{V}_{eq}$ in terms of the conductances and reversal potentials:

$$\bar{V}_{eq} = \frac{2g_{Na}\bar{E}_{Na} + 3g_K\bar{E}_K}{2g_{Na} + 3g_K} \tag{3.10}$$

$$\bar{I}_p = \frac{g_{Na}g_K}{2g_{Na} + 3g_K}(\bar{E}_{Na} - \bar{E}_K). \tag{3.11}$$

The individual channel conductances are more difficult to measure than the input resistance $R_{in} = 1/(g_{Na}+g_K)$. That is why in [Attwell and Laughlin, 2001], where the energy consumption of the rest state is determined from experimental data, the conductances g_K and g_{Na} are expressed by R_{in} and $\bar{V}_{eq}$ so that $\bar{I}_p$ becomes

$$\bar{I}_p = \frac{(\bar{V}_{eq} - \bar{E}_K)(\bar{E}_{Na} - \bar{V}_{eq})}{R_{in}(2\bar{E}_{Na} - 3\bar{E}_K + \bar{V}_{eq})}. \tag{3.12}$$

The energy consumption of the neuron per unit time and surface area is then given by $\bar{I}_p/F$ with F being the Faraday constant.

Cost of reaching the rest state

After a graded or global signaling event, the ion concentrations and membrane voltage have not yet reached their rest states $[\bar{Na}]_i$, $[\bar{K}]_i$ and $\bar{V}_{eq}$, respectively. Differential equations (3.3) to (3.5) allow to obtain differential equations for the changes $\triangle$ from the resting values. Since the membrane voltage operates at a faster time scale than the ion concentrations, $\triangle V$ reaches quickly the value

$$\triangle V = \frac{g_{Na}\triangle E_{Na} + g_K\triangle E_K - \triangle I_p}{g_{gNa} + g_K}. \tag{3.13}$$

The calculations in [Attwell and Laughlin, 2001] show then that $\triangle[Na]_i = -\triangle[K]_i$ at all times. Making the assumption that $\triangle I_p$ varies linearly with $\triangle[Na]_i$ the differential equation for $\triangle[Na]_i$ is solved in [Attwell and Laughlin, 2001]. The energy which is consumed to reach the rest state, given a certain amount of excess Na^+ ions or lack of K^+ ions, is then approximated by

$$\int_0^\infty \triangle I_p/F \mathrm{d}t \approx \frac{\text{Ion load}}{3}. \tag{3.14}$$

Energy efficient signaling

The Na^+/K^+ pump consumes 50% of all energy in the brain [Lennie, 2003]. The Na^+/K^+ pump is needed to maintain the ion gradients when the neuron is in the rest state, and to restore the ion gradients after a global or graded signaling event. Figure 3.5 illustrates the associated costs. For energy efficient signaling, the use of the Na^+/K^+ pump should be minimized. Any factor which reduces the ion fluxes without affecting the signaling quality improves energy efficiency. Postsynaptic currents should be small, and every action potential is valuable. Estimates for the human cortex show that the sustainable firing rate of a neuron is less than 1Hz (1 spike/s) [Lennie, 2003].

3.1.3 Modeling the signaling of neurons

We review here models which describe the signaling behavior of neurons. The action of transporters are normally excluded from these models because their current is normally dominated by the currents through the ion channels. This implies that the modeled signaling is free of energetic costs. We further limit the discussion to single compartment models, i.e. the spatial distribution of the neuron is ignored and the neuron is considered to be iso-potential. We treat first the biophysically more detailed models and then simplified models.

Minimal models

Minimal models are models which can create with the least variables both non self-amplified signals (graded signals) and self-amplified signals (action potentials). Minimal models are discussed in more detail in [Izhikevich, 2006].

Activation and inactivation of channels The modeling of the signaling variable V happens via Equation (3.3). As mentioned in Section 3.1.1, it is the feedback loop between channel activation and membrane voltage which accounts for action potentials. Channel activation is modeled by means of activation and inactivation gates m and h, respectively. The conductance of a channel is given by $g = \bar{g}mh$, and the dynamics of activation and inactivation by $\dot{m} = (m_\infty(V) - m)/\tau$ and $\dot{h} = (h_\infty(V) - h)/\tau$, respectively. The steady-state activation and inactivation functions m_∞ and h_∞ are shown in Figure 3.6. Recall from Table 3.2 that the direction of the current through the channels decides whether the gating variable forms with V a positive or negative feedback loop.

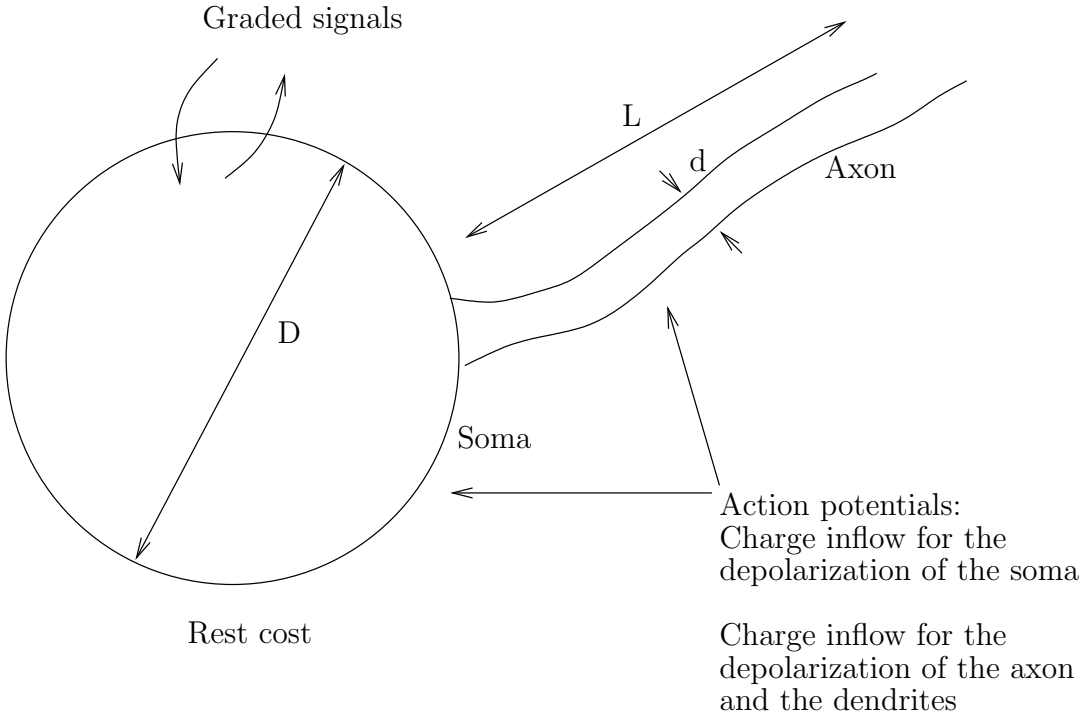


Figure 3.5: Energy consumption of the Na^+/K^+ pump. The rate of the maintenance cost of the ion gradients at rest is given by Equation (3.12). It scales proportionally to the surface of the neuron (πD^2 for a sphere approximation). The signaling related costs depend on the inflow (outflow) of Na^+ (K^+). While the inflow and outflow is variable for graded signals, for global signals, the inflow of Na^+ is fixed. The cost of the depolarization of the soma scales again with its surface. The cost of the propagation of the action potential down the axon and dendrites scales proportionally to dL. For typical rodent data, the propagation down the axon and dendrites dominates the soma related cost (with a ratio of 5:1) [Attwell and Laughlin, 2001]. If the graded signal is due to a typical excitory synapse, 85% of the total cost of the graded signal accrues to the postsynaptic neuron, both in the rodent [Attwell and Laughlin, 2001] or the human [Lennie, 2003]. The cost of a single action potential can be broken down into two dominant parts: First, the cost associated with the graded signals that are caused by the action potential (postsynaptic ion load). Second, the cost to propagate the action potential to the postsynaptic neurons. For the human cortex [Lennie, 2003], the first cost amounts to $1.2 \cdot 10^9$ ATP, while the second cost amounts to $9.2 \cdot 10^8$ ATP. The postsynaptic ion load costs thus 30% more than the propagation.

Structure of minimal models Minimal models consist of a differential equation for the voltage V and one for the gating variable which forms a negative feedback loop with V. The amplifying gating variable is assumed to reach quickly its steady state value.

A neuron model with an activating Na^+ channel and an activating K^+ channel is an example of a minimal model. The equations are

$$\dot{V} = -\frac{1}{C}\left(\bar{g}_{Na} m_\infty(V)(V - E_{Na}) + \bar{g}_K n(V - E_K) + g_L(V - E_L)\right) \quad (3.15)$$

$$\dot{n} = \frac{1}{\tau}\left(n_\infty(V) - n\right). \quad (3.16)$$

The pump current of Equation (3.3) is modeled here as leak current.

Excitability Depending on the parameters of the above differential equation, the membrane voltage V shows characteristic differences in its response upon stimulation. Figure 3.7 shows trajectories when the stimulation does not activate the self-amplifying positive feedback circle. Figure 3.8 shows the case where self-amplification takes place. The used parameters are given in Table 3.3. The term excitability describes the behavior of the neuron upon stimulation. The four cases shown in Figures 3.7 and 3.8 describe for a two dimensional system qualitatively all the possible types of excitability [Izhikevich, 2006]. Figure 3.9 summarizes the classification of neurons by their excitability, and Table 3.4 shows characteristic properties of the different types of excitability.

Simplified models for integrators

Two of the four excitability-types in Figure 3.9 show no oscillations upon weak stimulation. The neuron models are said to be in integrator mode. The minimal model models both the onset of the spike as well as its termination. It can be simplified if the modeling of the stop of the action potential is foregone. When the membrane voltage reaches are certain value there is a reset. The signaling variable is in that case not the membrane voltage itself but the time when it reaches the value after which the reset happens ("spike timing").

Canonical integrator (quadratic integrate and fire neuron) If a minimal model close to a saddle-node or saddle-node on invariant circle bifurcation (see Figure 3.9) is developed into Taylor series around its equilibrium point until the squared terms, the membrane voltage V can locally be described by

$$\dot{u} = a_1(u - u_R)(u - u_T) + a_2 I, \quad (3.17)$$

for $u_R < u_T$ and constants $a_1, a_2 > 0$. If $I = 0$, $u = u_R$ is a stable equilibrium point while $u = u_T$ is an unstable one. If one adds the reset condition: $u \leftarrow c$ if $u > u_p$ for a peak voltage u_p and reset voltage c, one obtains the equations for the canonical integrator or, with another name, the quadratic integrate and fire neuron, see e.g. [Izhikevich, 2006]. The time t^f when the voltage u passes u_p is called the firing time. An equivalent equation is

$$\dot{u} = a_1(u - u_{sn})^2 + a_2(I - I_{sn}), \tag{3.18}$$

where

$$u_{sn} = \frac{u_R + u_T}{2}, \tag{3.19}$$

$$I_{sn} = \frac{a_1}{a_2}\left(\frac{u_T - u_R}{2}\right)^2. \tag{3.20}$$

Figure 3.10 shows that by varying I the system undergoes a SNIC or a SN bifurcations at $I = I_{sn}$ in function of parameter c.

Integrate and fire neuron If only linear terms are retained in Equation (3.17) one obtains an equation of the form [Izhikevich, 2006]

$$\dot{u} = -a_1 u + a_2 I, \tag{3.21}$$

for constants $a_1, a_2 > 0$. If a reset condition such as $u \leftarrow c$ if $u > \theta$ is added, a neuron model is obtained which foregoes the modeling of the positive feedback loop that causes the action potential. While in the canonical integrator model the reset is not modeled, in this neuron model, the modeling is stopped with the onset of the action potential. The idea would be that the action potential (onset and reset) happens infinitely fast. The firing time t^f is given by the time where u passes the threshold θ.

Spike response model The solution of the integrate and fire neuron can be written in closed form as

$$u(t) = c\exp[-a_1(t - \hat{t})] + a_2 \int_0^{t-\hat{t}} \exp[-a_1 s] I(t - s)\mathrm{d}s, \tag{3.22}$$

where $\hat{t}$ is the time where u has crossed the threshold θ for the last time, i.e. the time of the last spike. If the exponential kernels in the above equation are named as

$$\eta(t) = c\exp[-a_1 t], \tag{3.23}$$

$$\kappa(t, s) = a_2 \exp[-a_1 s] H(t - s), \tag{3.24}$$

with $H()$ being the Heaviside function, one obtains

$$u(t) = \eta(t - \hat{t}) + \int_0^\infty \kappa(t - \hat{t}, s) I(t - s) \mathrm{d}s. \tag{3.25}$$

The spike response model [Gerstner and Kistler, 2002] generalizes this equation to arbitrary kernels η and κ.

Simplified models for resonators

Two of the four excitability-types in Figure 3.9 show oscillations upon weak stimulation. The neuron model is said to be in resonator mode. As in case of the integrators, the minimal model can be simplified if the termination of the action potential is not modeled, i.e. if it is replaced by a reset. The signaling variable becomes then also the time after which the reset happens ("spike timing").

Canonical resonator The behavior of a two dimensional dynamical system close to a AH bifurcation can locally be described by the complex-valued system

$$\dot{z} = (a_1 + ia_2)z + (a_2 + ia_3)z|z|^2 + I, \tag{3.26}$$

where the imaginary part of z is the voltage like variable and the real part is the current like variable, see e.g. [Izhikevich, 2006]. The addition of reset condition yields a neuron model.

Resonate and fire neuron The resonant and fire neuron [Izhikevich, 2001] is obtained by linearization of the canonical resonator, i.e.

$$\dot{z} = (a + i\omega)z + I. \tag{3.27}$$

The constant a is < 0 and the constant $\omega > 0$. Reset happens if the imaginary part of z passes the threshold θ, i.e. $z \leftarrow z_r$. With the choice of the reset value z_r, a monostable or bistable system can be created [Izhikevich, 2001], see Figure 3.11.

Firing-rate based models

Definition The firing rate of a neuron is not a well defined quantity. It can be defined as temporal average, as trial average or as population average, see e.g. [Gerstner and Kistler, 2002]. In the temporal average, the number of spikes $n_{sp}(T)$ happening in a time window of length T is divided by T, i.e. rate ν equals

$$\nu = \frac{n_{sp}(T)}{T}. \tag{3.28}$$

In the trial average, K trials are performed, and in each trial, the number of spikes $n_k(t; t+\triangle t)$ happening during the time window $[t \quad t+\triangle t]$ are counted. The firing rate ν is then defined as the spike density ρ,

$$\rho(t) = \frac{1}{\triangle t}\frac{n_k(t; t+\triangle t)}{K}. \tag{3.29}$$

In the population average $A(t)$, the number of neurons $n_a(t; t+\triangle t)$ which spike during the interval $[t \quad t+\triangle t]$ are determined and related to the total number N of neurons in the population, i.e.

$$A(t) = \frac{1}{\triangle t}\frac{n_a(t; t+\triangle t)}{N}. \tag{3.30}$$

Input-output relation The following input-output relation of firing rates can be applied to all of the three definitions above. Assume a neuron or, in case of the population average, a cell assembly is "connected" to neurons (or cell assemblies) indexed by j. Assume further that the total input current I of that neuron can be described by the equation

$$I = \sum_j w_j r_j, \tag{3.31}$$

where w_j is the strength of the connection from neuron (cell assembly) j to that neuron. The variable r_j denotes the firing rate of the neuron (cell assembly) j. The input-output relation is then obtained by relating input current I to firing rate r. This is done by means of the activation function f,

$$r = f(I). \tag{3.32}$$

Typical activation functions are the linear function, the threshold linear function, or compressing functions such as the sigmoidal function, the tangens hyperbolicus or the logarithmic function, see e.g. [Rolls and Deco, 2002].

The above input-output relations are instantaneous in time. It is possible to consider temporal delays by writing for the input current I

$$\tau_I \dot{I} = -I + \sum_j w_j r_j, \tag{3.33}$$

and for the relation for the firing rate

$$\tau_r \dot{r} = -r + f(I), \tag{3.34}$$

see e.g. [Dayan and Abbott, 2001].

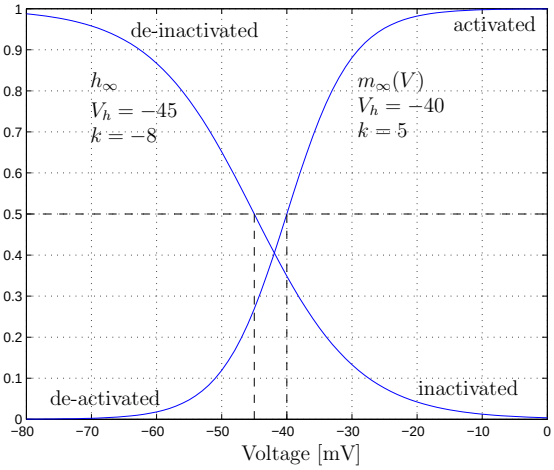


Figure 3.6: The activation and inactivation functions have the form $1/(1 + \exp((V_h - V)/k)$. The parameter V_h indicates the voltage where the activation or inactivation reaches the value 0.5. The parameter k indicates the sensitivity to the voltage and its sign decides if the gate is activating or inactivating. The gate is activating for $k > 0$, and inactivating for $k < 0$.

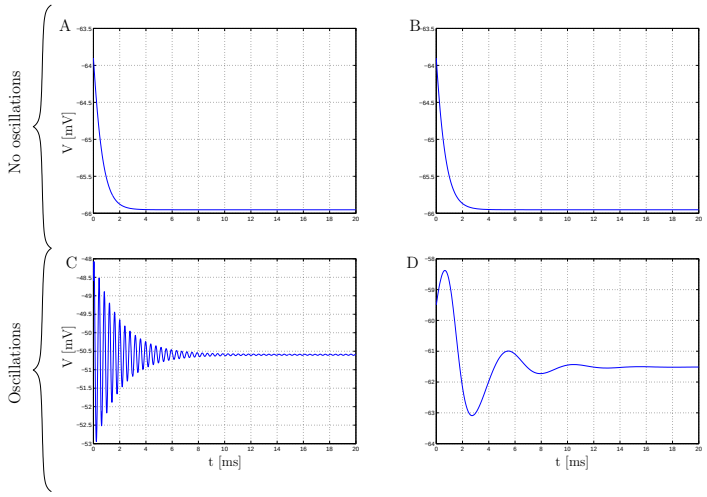


Figure 3.7: Responses of the neuron to shock input which moves the voltage from its rest state V_{eq} to $V_{eq}+2mV$. The perturbation is so small that the self-amplifying positive feedback loop is not activated. The four configurations of Table 3.3 can be qualitatively divided into those which feature subthreshold oscillations and those which do not.

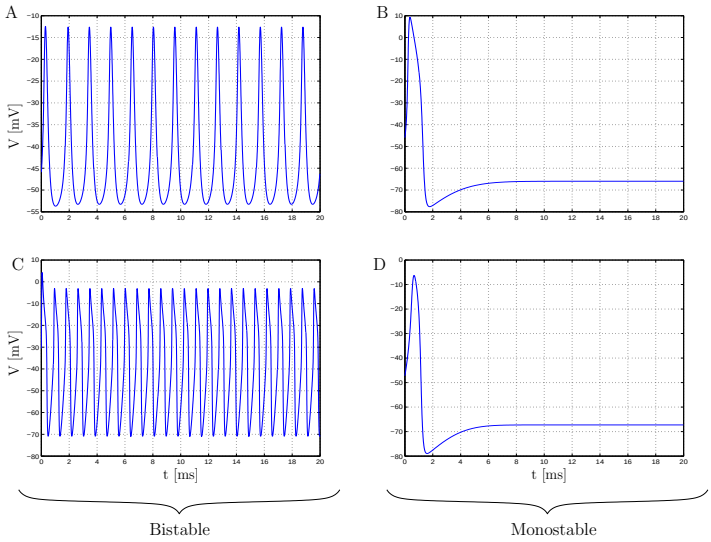


Figure 3.8: Responses of the neuron to shock input which moves the voltage from its rest state V_{eq} to $V_{eq}+20mV$. The perturbation is such that the self-amplifying positive feedback loop is activated. The four configurations of Table 3.3 can be qualitatively divided into those which are bistable and those which are monostable.

Type	V_{hk} (mV)	k_k (mV)	g_k (mS/cm^2)	V_{hNa} (mV)	k_{Na} (mV)	g_{Na} (mS/cm^2)	τ (ms)	I (μA/cm^2)
A	-25	5	10	-20	15	20	0.15	0
B	-25	5	10	-20	15	20	1	0
C	-45	5	30	-30	7	30	0.15	360
D	-45	5	10	-20	15	20	1	0

Table 3.3: Different parameter sets for the differential equations (3.15) and (3.16) which lead to different kinds of excitability. Other parameters are $C = 1\mu F/cm^2$ and $E_L = -80mV$.

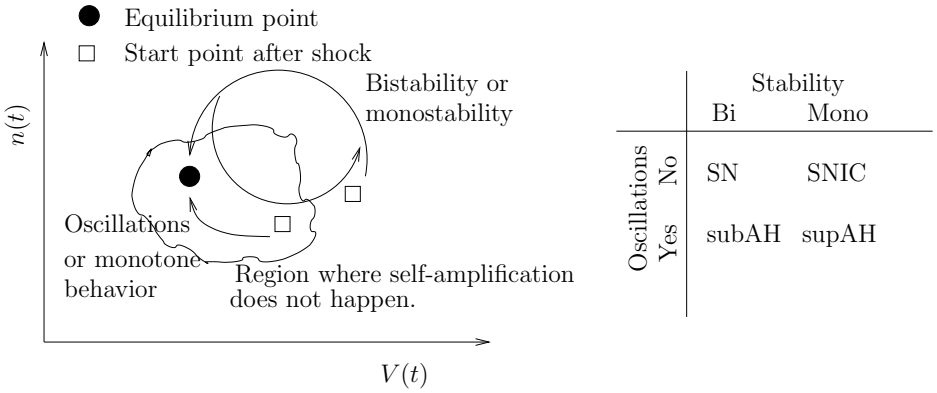


Figure 3.9: A two dimensional dynamical system can loose stability either by Saddle-Node (SN) or Andronov Hopf (AH) bifurcation (see e.g. [Izhikevich, 2006]). Systems close to a SN bifurcation do not feature oscillations upon a small shock while systems close to a AH bifurcation do. Small shocks which do not activate the positive-feedback loop can therefore be used to determine whether the neuron is close to a SN or AH bifurcation (see Figure 3.7). Shocks which activate the self-amplification mechanism are used to determine if the neuron is monostable or bistable (see Figure 3.8). This allows to determine subtypes of the SN and H bifurcation, i.e. saddle node on invariant circle (SNIC) for SN bifurcation and supercritical (supAH) or subcritical (subAH) for the AH bifurcation. Configuration A of Table 3.3 corresponds to the SNIC bifurcation, B to the SN, C to the subAH, and D to the supAH bifurcation.

Property	Excitability type			
	SNIC	SN	subAH	supAH
Oscillatory potentials	No	No	Yes	Yes
Bistability	Yes	No	Yes	No
Arbitrarily low spike freq.	Possible	No	No	No
Spike latency	Large	Large	Small	Small
Freq. preference	No	No	Yes	Yes
Well defined threshold	Yes	Yes	Possible	No
All or none AP	Yes	Yes	No	No
Inhibition induced spiking	No	No	possible	possible
Post-inhibitory spike	No	No	possible	possible
Intuitive behavior	Integrator		Resonator	

Table 3.4: For the different excitability types (SNIC, SN, subAH, supAH) different stimuli are needed to elicit a action potential. The property "arbitrary low spike frequency" means that the neuron can emit spikes at any frequency. "Spike latency" describes the time which passes till the neuron emits a spike upon stimulation. "Frequency preference" means that an input within a certain frequency band can more easily elicit an action potential than an input outside that preferred frequency band. The property "well defined threshold" characterizes the finding that in some neurons the self-amplifying positive feedback loop is started when the membrane voltage V has passed a certain threshold. "All or none action potentials (AP)" describe whether the action potential has a stereotyped form or if it can be influenced by the strength of the input. Property "Inhibition induced spiking" means that an inhibitory input can elicit a spike. "Post-inhibitory spike" describes whether the neuron can fire a spike when it is released from inhibition. After [Izhikevich, 2006].

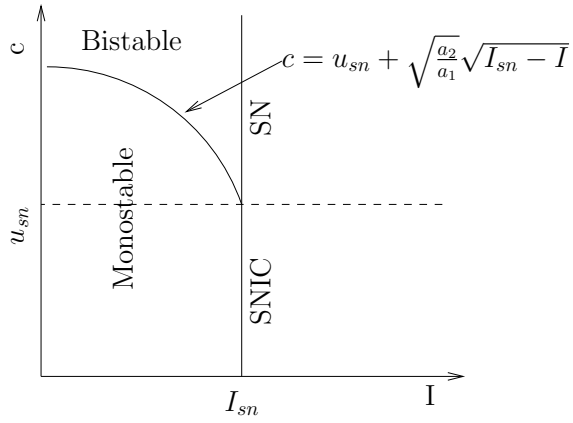


Figure 3.10: Bifurcation diagram for the canonical integrator neuron model of Equation (3.17). In function of the reset voltage c, the system undergoes a SNIC or a SN bifurcation when I passes I_{sn}. The quantities u_{sn} and I_{sn} are defined in Equations (3.19) and (3.20), respectively.

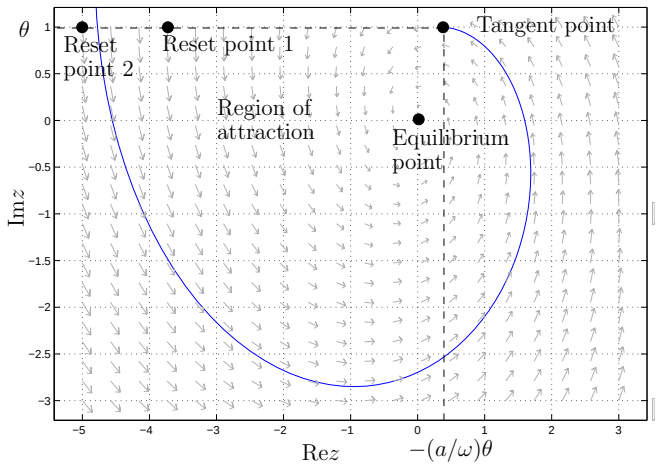


Figure 3.11: The resonate and fire model of Equation (3.27) can be mono or bistable. For the threshold θ, the tangent point is defined as the point where the trajectory enters horizontally into the threshold line $z = \theta$. Integration backwards in time yields the boundary of the region of attraction of the equilibrium point. If the reset point z_r is inside that boundary, the system is monostable (e.g. reset point 1), if it is outside, the system is bistable (e.g. reset point 2).

3.2 Organization of sensory systems

A sensory system starts with the receptors. An object in the environment may emit different forms of energy. The emitted energy is captured by the receptors and converted into an electrical signal. Distinct receptors are specialized for the transduction of particular forms of energy. In Section 3.2.1, we point out that this receptor specificity leads to a fragmentation of the input space. We further point out that the transduction may lead to ambiguous signals. For a proper understanding of "what object is out there", and hence an adequate action, the fragments must at least partially be recombined. During the process of recombination, the fragmented input is glued together and the ambiguity in the signals is reduced. This recombination of receptor signals to form a neural representation of the sensory stimulus is called neural integration. Section 3.2.2 introduces neural maps and explains that they are indicators of how this integration could work along the pathway through the brain. The development of the neural maps is treated in Section 3.2.3.

3.2.1 Input fragmentation into ambiguous signals

First, we give details on how the input from the environment is fragmented. Then, examples for the sensory modalities of touch and, in greater detail, vision are presented. Other modalities are proprioception, temperature, pain, taste, smell, balance, and hearing.

An ambiguous labeled line code

Receptors are selective for a particular type of energy and a particular area of stimulation. Each class of receptors, which specializes for a certain energy form, is not homogeneous. Instead, it contains a variety of receptors which sense submodalities. Activity in a given receptor neuron signals only the presence of a given energy-submodality at a given location. Each axon of the receptor neurons can thus be labeled with the energy-submodality and the location of the receptor. The time course of the receptor-activity reflects the time course of the energy-submodality at the area of stimulation. Complex input stimulates several kinds of receptors at the same time. The input is decomposed into different submodality-lines, which correspond to different kinds of receptors. Each receptor signal could however be the result of a multitude of different input signals. This makes the signal of an individual receptor ambiguous. It is the total population of receptors

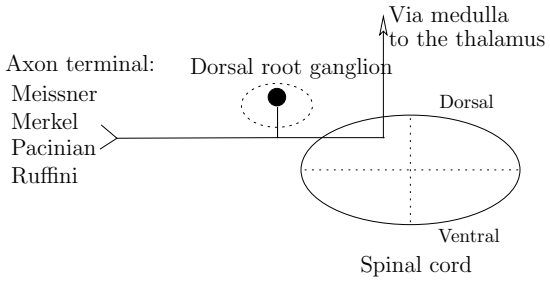


Figure 3.12: Input fragmentation for touch. The receptor neuron is located in the dorsal root ganglion. Its axon terminals specialize into four different morphologies. The Meissner's corpuscles and Merkel disk receptors are located in superficial layers underneath the skin. The Pacinian corpuscles and Ruffini endings are in deep layers. They sense different kinds of submodalities, see Table 3.5 Adapted from [Kandel et al., 2000; Purves, 2004].

which encode together in their activity the time course of the input.

The case of touch

The receptor neuron for the modality of touch is located in the dorsal root ganglion close to the spinal cord. Specialization of the nerve terminals allows these neurons to sense mechanical energy. The nerve terminals show four different kinds of morphologies, see Figure 3.12. Meissner's corpuscles are selective for stroking touch (flutter). Merkel disk receptors are specialized to pressure and texture. Pacinian corpuscles sense vibration, and Ruffini endings stretch of the skin. A single touch is thus decomposed into different submodalities, and the ganglion cell transmits each submodality as a separate line via the medulla to the thalamus.

The receptive field of a receptor cell is the area where it is responsive. Receptor neurons have together with the morphological specialization of their nerve terminals different dynamical properties and receptive field sizes. Table 3.5 shows that receptor neurons whose nerve terminals are in the deep layers in the skin have large receptive fields (Ruffini and Pacinian) while those with nerve terminals in the superficial layers (Merkel and Meissner) have small receptive fields. There are rapidly adapting cells (Meissner and Pacinian) and slowly adapting cells (Merkel and Ruffini) both in superficial and deep layers.

Receptor	Receptive field	Adaptation	Submodality
Meissner	Small	Rapid	Flutter
Merkel	Small	Slow	Pressure
Pacinian	Large	Rapid	Vibration
Ruffini	Large	Slow	Stretch

Table 3.5: Receptor neurons with different kinds of nerve terminals show different dynamical properties and receptive field sizes. Receptor neurons with small receptive fields have terminals that are located in the superficial layers of the skin (Merkel and Meissner). Neurons with large receptive fields have terminals in deeper layers (Ruffini and Pacinian).

The case of vision

Fragmentation by the photoreceptor-horizontal cell system There are two general forms of photoreceptor cells for vertebrates. Rod cells, which are highly sensitive to light and slowly adapting, are responsible for night vision. They respond best to wavelength in the green range (500nm). Cone cells, which are less sensitive to light and rapidly adapting, are responsible for day vision. In primate retinas, there are three types of cones. One has peak sensitivity for long wavelengths (red, L cones), one for medium wavelength (green, M cones), and one for short wavelength (blue, S cones). The peaks are not evenly distributed in the wavelength interval. L and M cones have similar sensitivities. In other mammals, these cones are lumped together into a single LM cone type.

Rod and cone receptor cells are non-uniformly distributed in the retina, see Figure 3.13. L and M cones are densely distributed in the fovea while rods and S cones are not present. Cones trade therefore spatial resolution for light sensitivity while rods do the opposite. Each cone receptor type (L, M, or S) does further not form a regular mosaic but the receptors are in clusters randomly distributed across the retina [Solomon and Lennie, 2007].

Horizontal cells provide a negative feedback system for the photoreceptors. Functionally, there are different systems for rods and cones. For the cones, there is an additional one which is exclusively for L and M cones. Coupling between horizontal cells allows them to sum light responses across a broad region. Their negative feedback onto the photoreceptor cells allows for adaptation to light conditions. Equivalently, the horizontal cells cause a center-surround organization

where an average photoreceptor signal is subtracted from the local photoreceptor signal to enhance contrast.

Local processing by the bipolar and amacrine cells There are thus four unevenly distributed photoreceptor types which fragment the incoming light into spectral and luminance submodalities. We limit the discussion here to the three submodalities for day vision, i.e. the cone-system. A cone of a given spectral preference and location can catch and transduce photons of any frequency and spatial origin. Each photon contributes also equally to electrical cone activity. That is why the absolute value of the individual cone activity is by itself an ambiguous signal. Electrical cone activities need to be compared among each other to obtain useful information. This starts in the neural circuit of the retina which is shown in Figure 3.14.

The first step in the processing happens in the OPL (see Figure 3.14) where the photoreceptors synapse onto the bipolar cells. There is a large variety of bipolar cell types. Important differences between them are the number and types of cones which contact them, and the properties of the synaptic connection (receptor type and kinetics). Table 3.6 shows that in the mammalian retina, there are about 12 different bipolar cell types with different properties. The cone signals are channeled into 12 parallel pathways which indicate color, light increment and decrement, and temporal frequency. The bipolar cells contact in the IPL (see Figure 3.14) retinal ganglion cells and amacrine cells. Amacrine cells feed back to bipolar cells and synapse also onto ganglion cells. This second processing step adds complexity to the retinal circuitry. There are about 30-40 different types of amacrine cells [Masland, 2001a,b; Rodieck, 1998] and their role remains not well understood.

Outcome of local processing is transmitted by the retinal ganglion cells The outcome of the local processing of the cone activities is visible on the level of the retinal ganglion cells. The ganglion cells project from the retina to higher areas of the brain. Some 15-20 different types of ganglion cells exist in the mammalian retina [Masland, 2001a,b; Rodieck, 1998; Wassle, 2004], as determined by morphology [Rockhill et al., 2002] or physiology [Devries and Baylor, 1997], may be assisted by clustering methods [Carcieri et al., 2003]. The latter classification method is based on the receptive field of the ganglion cell, as illustrated in Figure 3.15. Each of the cell types seems to provide complete coverage of the retina so that each class of ganglion cells processes the entire visual scene [Devries and

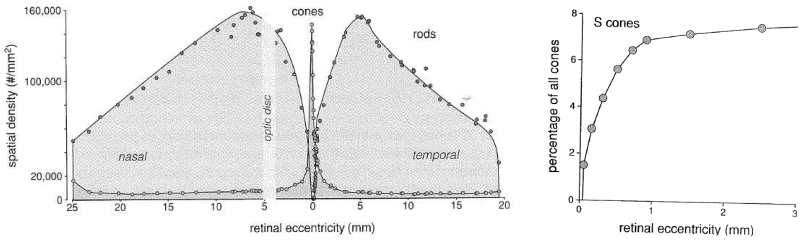


Figure 3.13: Nonuniform sampling of the visual space by the photoreceptors in the retina. Left: For cones, there is high resolution at the center of gaze and sampling is coarse for the periphery. This nonuniform sampling allows for high resolution and a wide field-of-view with fewer cones than uniform sampling. Rods are only present in the periphery. Right: Distribution of S-cones. S-cones are not present in the fovea. Else, they account for about 8% of all cones. Adapted from [Rodieck, 1998].

Baylor, 1997].

Table 3.7 summarizes the different types of retinal ganglion cells. A majority of retinal ganglion cells and cell types, namely 12-15, project to the lateral geniculate nucleus (LGN, see Figure 3.17) of the thalamus [Dacey et al., 2003]. It is important to note that there are in total about four million cones, but only about 1.6 million ganglion cells [Rodieck, 1998].

Response properties of the ganglion cells are a product of the local processing happening in the IPL by interaction with bipolar and amacrine cells. The degree of interaction depends also on the stimulation of the retina. The characterization of ganglion cells by means of the receptive field, i.e. the spike-triggered-average given noise input, does not describe the local processing completely. There are reports that OFF-ganglion cells can switch dynamically for a short period of about 100 milliseconds to ON-type given the right stimulation in the periphery [Geffen et al., 2007]. This switch is thought to be due to the activation of a certain amacrine cell type. Changes in the structure of the receptive field after a few seconds in a statistically new environment has also been reported [Hosoya et al., 2005]. This is pattern adaptation which adds to other observed plasticity in the retina, namely light and contrast adaptation [Baccus and Meister, 2004; Solomon et al., 2004].

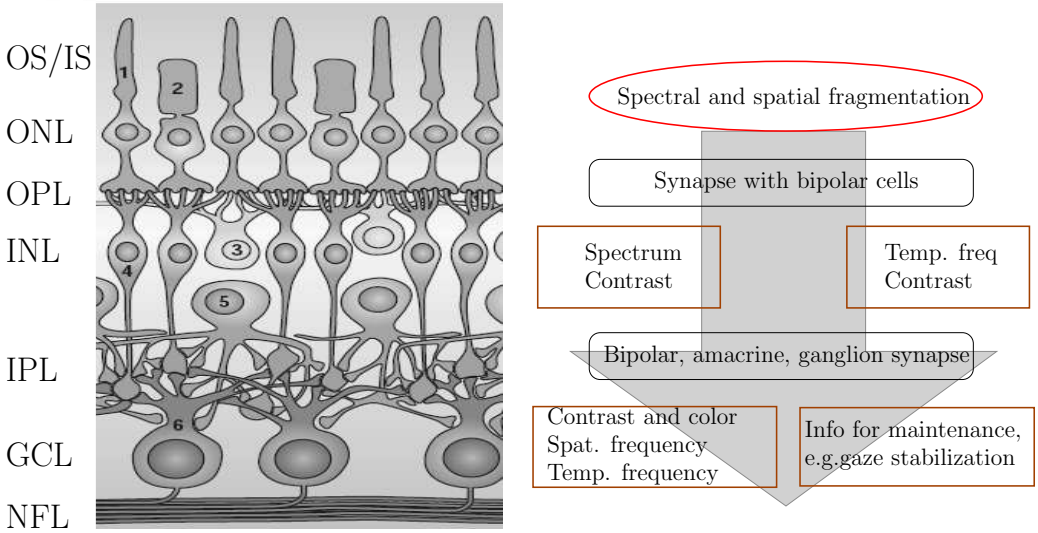


Figure 3.14: Left: Schematic of mammalian retina. Right: Local processing of the cone signals. The local circuit consists of six classes of neurons: rods (1), cones (2), horizontal cells (3), bipolar cells (4), amacrine cells (5) and retinal ganglion cells (RGC) (6). The cells are clearly organized into different layers: Outer and inner segments of rods and cones (OS/IS), outer nuclear layer (ONL), outer plexiform layer (OPL), inner nuclear layer (INL), inner plexiform layer (IPL), ganglion cell layer (GCL), optic nerve fiber layer (NFL). The cones provide a signal which is spectrally and spatially fragmented. The signal is processed in two steps. After the OPL, the bipolar cells form two functional groups, see Table 3.6. One group represents spectral and contrast information. The other group represents temporal frequency and contrast information. The next step of the processing happens with the IPL. The cone signal is represented on the GCL by two functional groups of ganglion cells, see Table 3.7. One group extracts information for maintenance functions, and the other one projects to the lateral geniculate nucleus (LGN, see Figure 3.17) of the thalamus and later to the cortex. The cells which project to the LGN are a heterogeneous group with different tuning behaviors with respect to spectral as well as temporal, and coupled to the temporal frequency tuning, spatial frequency.

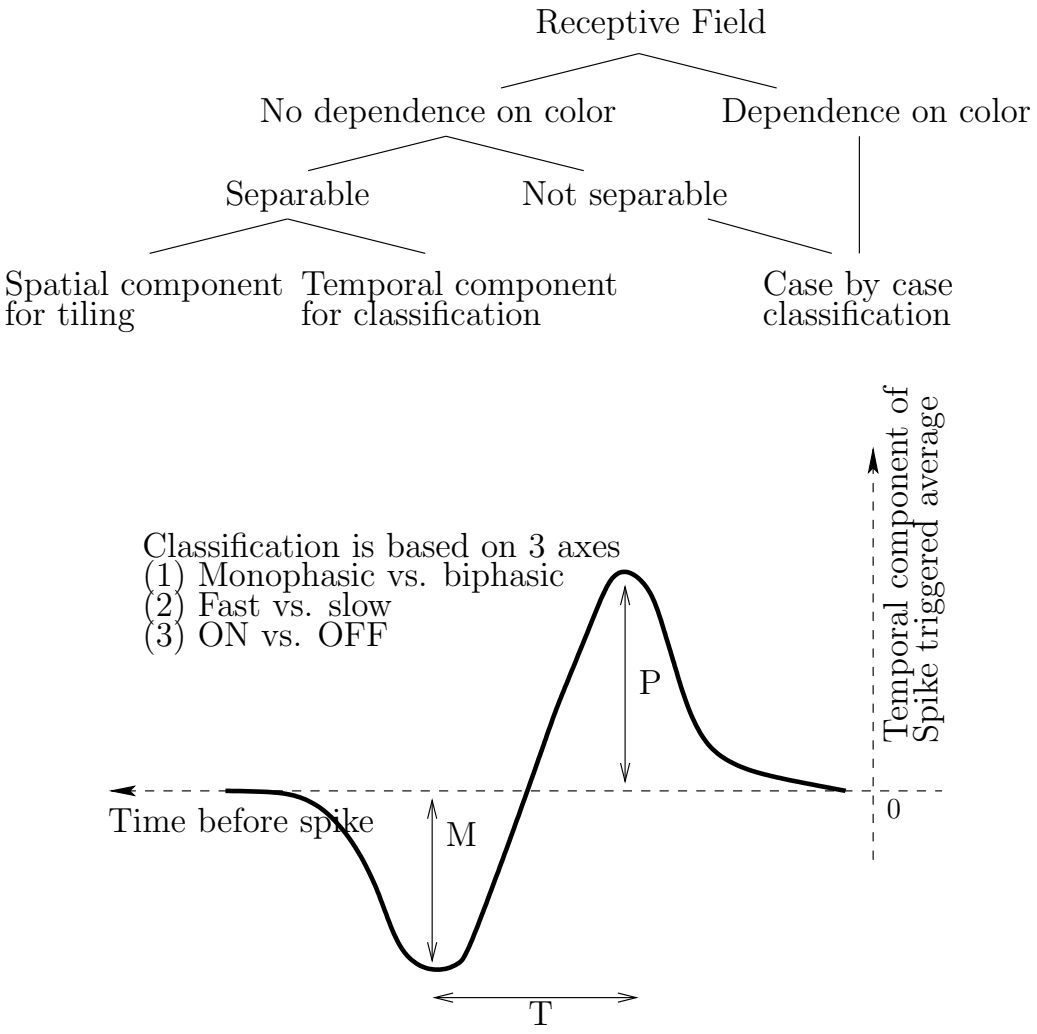


Figure 3.15: Top: The receptive field is used to classify ganglion cells and to determine the tiling of the retina by the different classes of neurons. Bottom: Properties of the temporal component of the receptive field are used for the classification of ganglion cells, e.g. as in [Devries and Baylor, 1997; Segev et al., 2006]. They are useful for the comparison between species and lend themselves to clustering methods. There are no strict conventions but if P and M are roughly of the same magnitude the cell is called biphasic. If either M or P is larger, the cell is called monophasic. Biphasic cells calculate the difference between the current and the past input. As in [Devries and Baylor, 1997], they often show transient response dynamics. Monophasic cells, on the other hand, have sustained response dynamics. The value of T is used to classify the cell to be slow or fast. If the positive peak is closer to the spike onset, the cell is called ON cell, else OFF cell. The depicted cell would be a biphasic ON cell, fast or slow depending on T.

Type	Number and types of cones	Synaptic connection	Property
Midget bipolars (5 types)	Single L-cone	+	L-OFF
	Single L-cone	−	L-ON
	Single M-cone	+	M-OFF
	Single M-cone	−	M-ON
	Single S-cone	+	S-OFF
S-cone bipolar	Fovea: single cone, Periphery: 2-3 cones	−	S-ON
Diffuse bipolar (6 types)	5-10 cones	+, quick recovery	OFF-transient
		+, slow recovery	OFF-sustained
		+, slow recovery	OFF-sustained
		−, quick recovery	ON-transient
		−, slow recovery	OFF-transient
		−, slow recovery	OFF-transient
Giant (wide field)	50-100 cones	?	?

Table 3.6: Types of cone bipolar cells in the mammalian retina (after [MacNeil et al., 2004; Rodieck, 1998; Wassle, 2004]).The type of a ganglion cell is determined by size, morphology and stratification in the IPL. + means a sign-conserving and − a sign-inverting synapse, respectively. The label ON means that the cell increases firing for light increment (positive contrast). OFF means increased firing for light decrement (negative contrast). The label transient means that the cell acts as high-pass filter while a sustained cell acts as a low-pass filter. OFF cells terminate in the outer half of the IPL while ON cells terminate in the inner half. Transient cells are localized more to the center of the IPL while sustained cells are found more in the peripheral IPL.

Type	Input	Response dynamics	Receptive field structure	Projection	"Role"
Parasol GC ON and OFF type (brisk transient, α, Y, M)	5-6 diffuse BC Coupled to parasol and amacrine cells.	Transient cells Short response latency.	Center-surround Large Non-linear summation	LGN by fast axons (magnocellular layers)	Contrast vision achromatic
Midget GC ON and OFF type (brisk sustained, β, X, P)	Single midget BC No coupling.	Sustained cells	Center-surround Small Linear summation. Red-green opponency	LGN (parvocellular layers)	Contrast and color vision.
Small bistratified GC	OFF diffuse BC and S-cone BC Coupled to amacrine cells	Blue ON-yellow OFF center-surround		LGN (koniocellular layers)	Color vision
ON direction selective GC (3 types)	Coupled to amacrine cells	Slow movement, one axis per cell type		Accessory optic system	Stabilization of gaze during head movement.
ON-OFF direction selective GC (4 types)	Coupled to amacrine cells	Fast movement, one axis per cell type		Superior collicus and LGN	Smooth tracking
Biplexiform GC	Bipolar cells and directly from rods	Long response latency, little adaptation	uniform	Suprachiasmatic nucleus	Luminance detector, circadian rhythm.
Other GC	Cells related to the S-cone pathway, edge detector cell, P giant cell, ϵ cell, γ cell, ...				

Table 3.7: Types of ganglion cells in the mammalian retina (after [Devries and Baylor, 1997; Masland, 2001a,b; Rockhill et al., 2002; Rodieck, 1998; Solomon and Lennie, 2007; Wassle, 2004]). ON cells increase their firing for light increment, OFF cells increase their firing for light decrement. The labels transient and sustained describe the firing behavior upon stimulation. Transient cells act as temporal high pass and sustained cells as low pass filters, respectively. The small size of the receptive field of midget cell makes them selective for higher spatial frequencies while parasol cells are thought to be selective for lower spatial frequencies. About 12 to 15 GC types are estimated to project to the LGN [Dacey and Packer, 2003; Dacey et al., 2003]. The parasol (2 types, 8% of all GC), midget (4 types, 70% of all GC) and small bistratified cell together account only for 7 types. Cells which project to the LGN (see Figure 3.17) are responsible for conscious vision. The other cells help to stabilize the image on the retina and to provide cues for the circadian rhythm ("maintenance functions").

3.2.2 Neural maps along the sensory pathway

Neural maps are well arranged and organized representations of the sensory input space by means of the neural tissue. First, we show examples of neural maps for both the modality of touch and of vision. Then we point out that neural maps can reveal principles of neural integration.

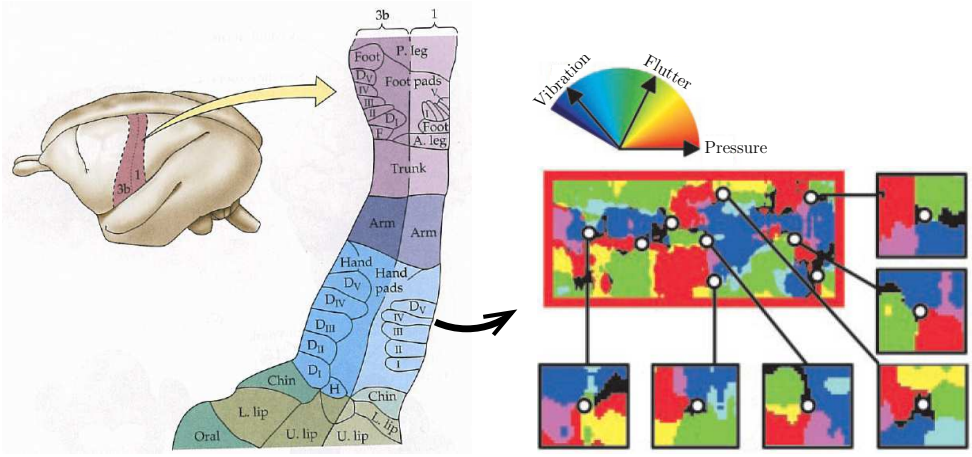
The case of touch

In Figure 3.12, we have shown that for the modality of touch, the dorsal root ganglion projects to the thalamus. For the four different touch-submodalities (flutter, pressure, vibration, stretch, see Table 3.5), the pathways to the thalamus and from there to the primary somatosensory cortex (S1) remain segregated. S1 contains four areas, area 3a, 3b, 1 and 2, where area 3b is the primary input stage from the thalamus.

Each of the four regions of the primary somatic sensory cortex contains a complete representation of the body surface [Kandel et al., 2000]. The representation is based on the responsiveness of the cortex to peripheral stimulation at a certain location of the body. This representation is said to implement a map from the somatosensory input to the cortex. Figure 3.16 shows this body map for the areas 3b and 1.

There is also organization with respect to the submodality of the input. In area 3b, there are segregated domains for input provided by Merkel and Meissner cells [Kandel et al., 2000]. For area 1, segregation has been established for cells which respond preferably to pressure (as Merkel receptor cells), flutter (as Meissner receptor cells) and vibration (as Pacinian receptor cells) [Friedman et al., 2004]. This is also shown in Figure 3.16. Each representation of a body part is thus further divided into the submodalities. From Figure 3.16 we see that pressure, flutter, or vibratory stimulation of a digit is represented by neighboring cells in area 1.

Along the ventral-dorsal axis, there are several layers in the cortex. However, the properties of the cells are such that, within a vertical column, the tuning properties towards a spatial location on the body surface and touch-submodality remains constant. This organization is called columnar organization.



Clustering with respect to stimulus location Clustering with respect to submodalities

Figure 3.16: Organization of the primary somatosensory cortex. Cells which show preference for the same location and submodality of stimulation are clustered together. Adapted from [Friedman et al., 2004; Purves, 2004].

The case of vision

In Table 3.7, we have shown that about 12-15 different types of retinal ganglion cells project to the lateral geniculate nucleus of the thalamus (LGN). We have further pointed out that not all of them are well characterized. What is known is that they differ in their color preferences, as well as their spatial and temporal frequency tuning. We review here that cells which show similar tuning properties are in the LGN and in the visual cortex clustered together. We will see that the clustering happens in a highly organized way, as in the case of touch. This organization is said to provide a map from the visual input onto the neural tissue.

Neural maps in the LGN Cells of the LGN can be characterized along the same axes as the retinal ganglion cells, i.e. spectral, spatial and temporal frequency tuning, and receptive field center. Since the LGN receives input from both eyes, ocular dominance of a cell adds to the above criteria. Figure 3.18 shows the organization of the LGN in a schematic way. It can be seen that cells with similar tuning properties are clustered together. There is organization with respect to receptive field center, ocular dominance, spectral sensitivity, and as parasol and midget ganglion cells project to separate layers, with respect to temporal and spatial frequency tuning. Note, however, that information about spatial detail

and red-green opponency is jointly coded by means of the P cells.

Neural maps in V1 Properties of neurons in the first cortical stage (V1) are often characterized by their responses to small patches of gratings. The characterization works over the properties of the stimuli and the strength of the response of the neuron. Stimuli are shown to either the left or right eye, they have a particular location in visual space, a particular orientation, a particular spatial frequency, a direction of movement (motion), and chromatic properties. The firing rate of the neuron defines then the tuning curves of the cells: ocular dominance, receptive field size, orientation preference, spatial frequency tuning, temporal frequency tuning and direction preference, and spectral tuning, see e.g. [Lennie and Movshon, 2005].

For the spectral tuning, it is interesting to note that while chromatic stimuli are a good driver for the cells of the LGN, only 5-10 % of all cells in the cortex respond robustly to chromatic stimuli. These cells are insensitive to precise spatial form (e.g. orientation). Many cells are weakly color opponent but the chromatic properties are not stable, i.e. they do not remain the same if other attributes of the stimulus are varied. These color sensitivity may result from the random clustering of the cone receptors and has been called "accidental" [Lennie and Movshon, 2005]. The neural correlates of color perception, i.e. the binding of chromatic properties of objects to their spatial properties, remains not well understood [Solomon and Lennie, 2007].

Analysis of the distribution of the properties of neurons over the cortex shows that neighboring points in V1 have neighboring receptive field centers, but they are also tuned to nearby values for ocular dominance, orientation preference, spatial frequency tuning, and direction preference, see Figure 3.19. The stimulus space, formed by the above characteristics, is said to be mapped onto the visual cortex.

Cells across the cortical layers are invariant with respect to the above characteristics, but with respect to color, cells in the cortical layers show differences. Figure 3.20 shows that cells with similar chromatic properties are clustered together in the same cortical layer. Cells in the cortex are therefore also clustered together according to their tuning properties: laminar clustering for color and columnar clustering for the other tuning properties.

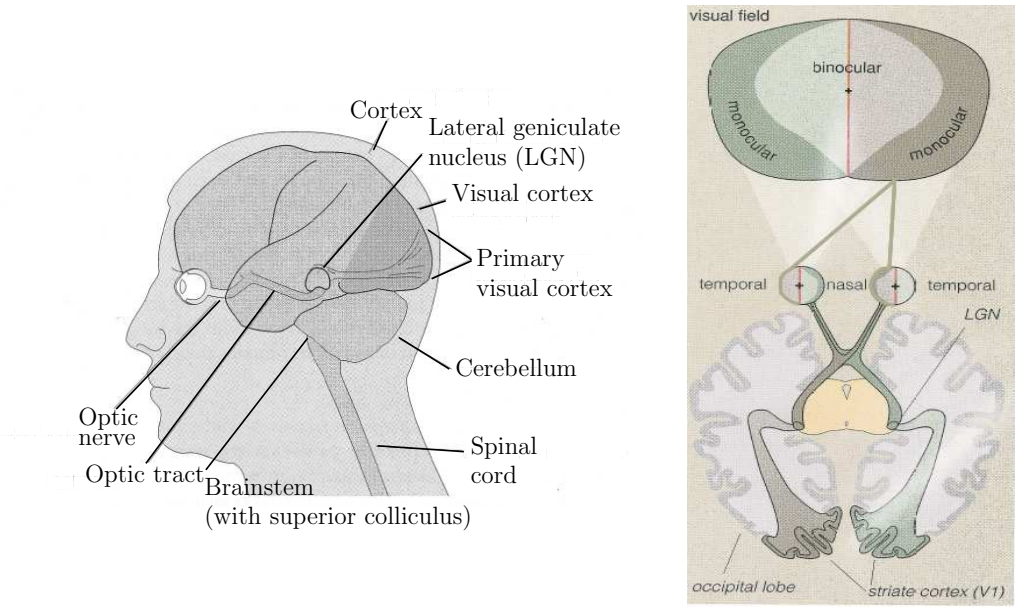


Figure 3.17: Visual pathway. The ganglion cells of the retina project to the lateral geniculate nucleus (LGN), which is connected to the primary visual cortex (V1). Adapted from [Rodieck, 1998].

Neural maps and principles of neural integration

Properties of neural maps The neural maps in S1, LGN, and V1 have the common property that across the surface of the neural tissue, the preferred stimulus location changes smoothly ("smoothness property"). In short, cortical neighbors have neighboring stimulus locations. They further share the common property that a cell assembly that is defined by its preference for a stimulus location contains neurons tuned to the remaining parameters ("Coverage property"). For touch, the cell assembly in Figure 3.16 which responds to stimulation of a single digit contains neurons tuned to pressure, flutter and vibration. In case of the LGN in Figure 3.18, the vertical line across the LGN layers contains cells tuned both to the ipsilateral and contralateral eye, different colors as well as spatial and temporal frequency. In case of V1, the analogue holds. The two properties above, i.e. smoothness and coverage, can also be formulated with respect to other stimulus parameters than stimulus location.

The properties of neural maps have quantitatively been investigated for the case of V1 [Das, 2005; Hubener et al., 1997; Yu et al., 2005]. The direction pref-

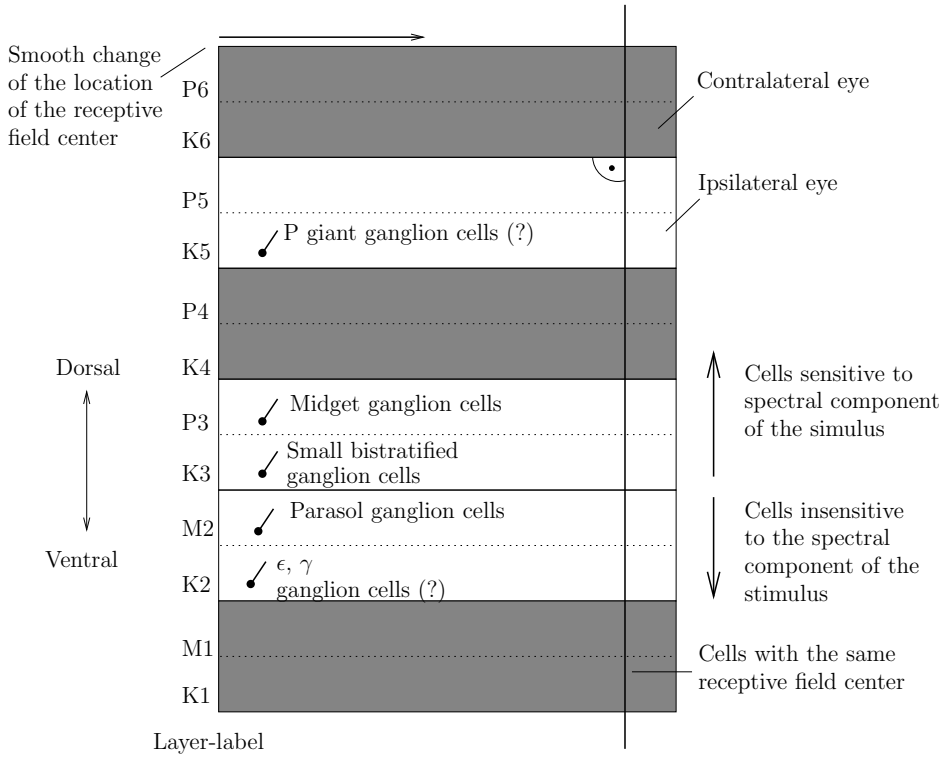


Figure 3.18: Schematic of the organization of the LGN (coronal section). The layers of the LGN have been classified into the P, M, and K layers, see e.g. [Hendry and Reid, 2000; Rodieck, 1998]. The 3 types of layers receive inputs from different retinal ganglion cells. The axons of the ganglion cells terminate in a ordered manner in the LGN. Ganglion cells which have neighboring receptive field centers end close-by in each sublayer (e.g. P5) of the LGN (retinotopic map). LGN-cells which are aligned along the ventral-dorsal axis have all the same receptive field center. The axons of the ganglion cells are sorted according to the eye from where they originate. This leads to ocular dominance regions. Layers in gray are associated with the contralateral eye and the layers in white with the ipsilateral eye, respectively. The P layers receive input from midget ganglion cells, and the M layers from parasol ganglion cells. Although the synaptic input from the retina form for these layers only a tiny fraction of the total synaptic input (which comes mostly from V1), the synaptic amplification is strong so that retinal input drives the cells in the P and M layer [Hendry and Reid, 2000]. The cells which project to the K layers are more heterogeneous and it is not clear whether retinal, cortical or brainstem input dominates. The layers K3 and K4 receive input from small bistratified ganglion cells. Layers K5 and K6 are thought to be innervated by P giant ganglion cells, and K1 and K2 by ϵ and γ ganglion cells [Hendry and Reid, 2000]. The cells in the P, M, and K level inherit the properties from the ganglion cells. Hence a separation into color-sensitive cells and color-insensitive cells occurs. The P layers have together about 10^6 cells, the M and K layers 10^5. The divergence ratio LGN:V1 cells is for P cells 1:50, for M cells 1:300, and for K cells 1:50. That is why the P and M cells are assumed to dominate the K cells in their influence on V1.

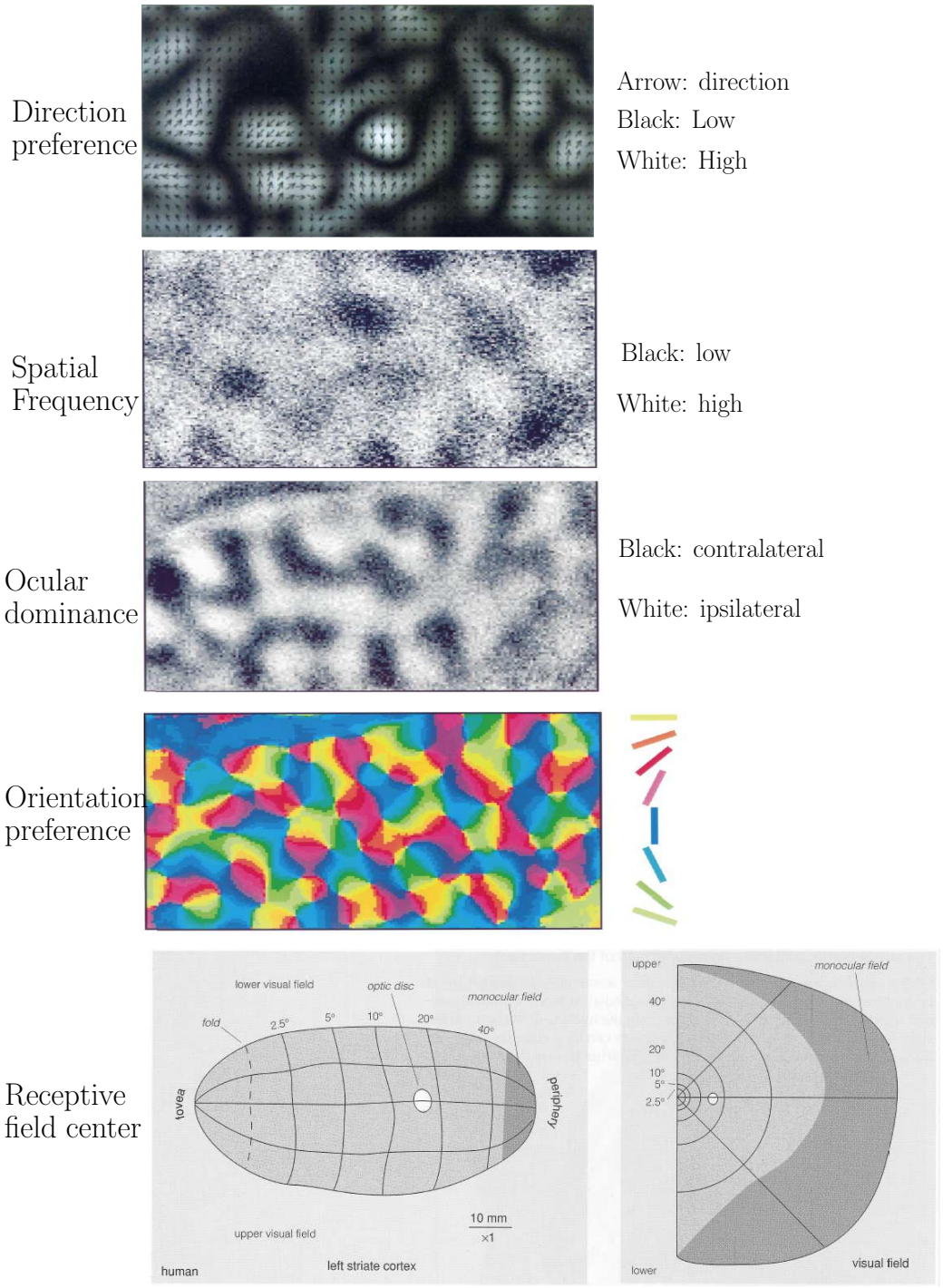


Figure 3.19: Maps of spatial location of the receptive field, orientation preference, ocular dominance, spatial frequency, and direction preference. The maps show how the tuning properties of the neurons are distributed over the surface of the primary visual cortex. While these tuning properties change across the surface of the cortex, along the ventral-dorsal axis, they do not change (columnar organization). From [Hubener et al., 1997; Rodieck, 1998; Weliky et al., 1996].

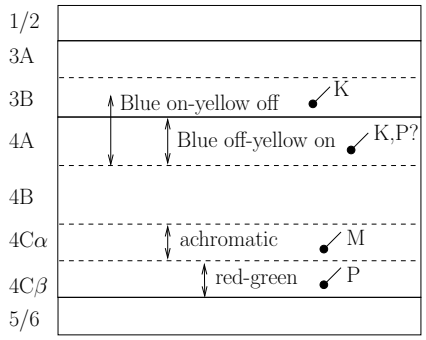


Figure 3.20: Neurons in different cortical layers show different color tunings. This laminar clustering provides a partition of the columns of Figure 3.19 into smaller cell assemblies where each cell has well defined tuning properties. From [Chatterjee and Callaway, 2003].

erence map and color properties have not been included. The finding is that the smoothness property, i.e. that each stimulus property should change smoothly across the cortex, and the coverage property, i.e. that the cortex responds to all possible combinations of stimulus properties at every point in visual space, are robust properties. A further property, the separability of the maps, has also been established. This property means that the orientation preference map is for example the same whether obtained at high or low spatial frequencies.

The same investigations [Das, 2005; Yu et al., 2005] have also shown that assuming separability, the coverage property and smoothness property are enough to account for many more properties of the neural maps in V1. They applied a dimension reduction algorithm [Obermayer et al., 1992] which maximizes coverage and smoothness to a six dimensional space of stimulus parameters (receptive field center (2 dim), orientation preference and sharpness of tuning (2 dim), ocular dominance, and spatial frequency preference) in order to find a map onto the simulated two dimensional cortical surface. This work also shows that as a consequence of coverage and smoothness, the individual maps are arranged in such a way that regions where rapid change in the tuning preference occurs avoid each other. At a location on the cortical surface where the tuning preference with respect to a certain stimulus parameter changes rapidly does the tuning to the other parameters change only slowly. This property can also be seen in the LGN, where the ocular dominance changes rapidly across the ventral-dorsal axis but the receptive field center of the cells remains the same.

Consequences for neural integration Neural integration means the combination of fragmented input along the sensory pathway. Neural maps visualize that the integration is such that neurons with similar tuning properties are clustered together (smoothness property). We have in Section 3.1.2 seen that the cost of action potential propagation increases with the length of the axonal connection. The interaction between neurons which are situated close by in the cortex is thus cheaper than interaction between neurons far apart. That is why the smoothness property is sometimes also called minimal-wiring property. The neural integration is therefore such that the interaction between neurons which have similar tuning properties becomes inexpensive.

The coverage property of the neural maps means that there is no blind spot in the cortical representation of the stimulus. Although the L and M cone photoreceptors are for example randomly distributed in clusters, and although there are no S cone photoreceptors in the fovea (see Section 3.2.1) we can normally perceive color equally well over the visual space. The coverage property implies thus that neural integration is such that an accurate representation of the input is possible.

The separability property of the neural maps means that a certain tuning property of a neuron is invariant to the change of other parameters of the input. The neural integration is thus such that activity of a neuron is a robust signal with a meaning associated to it which reduces the ambiguity of the signal.

3.2.3 Development of neural maps

The development of neural maps can be divided into a prenatal phase, an early postnatal phase and into the period where the brain is mature. We review here in parallel the mechanisms of development and the instructive case of the ocular dominance columns in V1 for each of the three development phases. Ocular dominance regions in the LGN are reviewed in [Huberman, 2007].

Initial construction of neural circuits

Developmental mechanisms First, different brain regions are established and neurons are generated. These clusters of neurons need to be connected by axonal pathways, and synapses between individual neurons need to be formed. This happens during prenatal time. Neural activity is internally created and is not input dependent, i.e. it is so called spontaneous neural activity.

Axon guidance works via cell adhesion and chemical attraction or repulsion forces. It has been found that topographic maps are formed from the beginning on

with the outgrowth of the axons [Polleux, 2005]. The sensitivity of the axons to the gradients of the molecular cues gives dynamically rise to the ordered distribution of the axons which form the topographic maps [Dickson, 2002; Wilkinson, 2001].

After initial contact of an axon with its target site rapid synapse assembly occurs. The synapse assembly has in turn influence on the pre- and postsynaptic neurons so that synapse formation goes together with the remodeling of the axon arbor structure. Excess synaptic inputs are eliminated by a competitive process where synapse disassembly precedes the withdrawal of the innervating axonal branch. The competitive process is a competition among the axonal input branches for the ownership of the target cell. The synapses of coactive input axonal branches are stabilized while those of axons with uncorrelated neural activities are removed. The winning axons sprout then to establish more synapses with the target cell [Cohen-Cory, 2002].

The case of ocular dominance columns It has been found that ocular dominance columns are present prior to any visual experience [Horton and Hocking, 1996]. This initial establishment of ocular dominance columns is suggested to be strongly driven by molecular cues [Crowley and Katz, 1999, 2000]. Spontaneous neural activity works then on the pre-patterned connections to support ocular segregation or destroy it [Cohen-Cory, 2002; Katz and Shatz, 1996; Katz and Crowley, 2002; Penn et al., 1998].

Early postnatal development

Developmental mechanisms Postnatal development acts thus not on a tabula rasa but on a pre-structured neural system. In contrast to the neural development up to this point, in the postnatal period, however, the environment can actively influence the neural system. Electrical activity relates at least partially to the input.

Critical periods are defined as the temporal window during which the environment is most influential. A given skill needs during that temporal window specific environmental stimuli for normal development. Abnormal stimuli can compromise it. It is thought that the main role of experience during the critical period is to reinforce and augment an already appropriately formed set of basic neural connections [Katz and Crowley, 2002]. Section 3.3 discusses how the input can influence the neural system by means of patterned electrical activity by causing modifications in the synaptic connections.

The case of ocular dominance columns There is evidence that abnormal postnatal neural activity interferes with the further development of ocular dominance columns. Normal neural activity can be changed by modifying the visual input. In two classic studies [Purves, 2004] visual input to one eye was blocked (monocular deprivation experiment) and the visual input to each eye was decoupled from the other (strabistic animal experiment), respectively.

In the monocular deprivation experiment, monocular dominance in the primary visual cortex shifted to the open eye. Eye closure had only an effect on the animal during the first three month of life in the cat, in primates during the six first month.

In strabismus, the two eyes cannot be aligned so that points on the retina are no longer stimulated by objects in the same location. The visual inputs of the eyes are decoupled from each other. Under these conditions, the number of binocular cells decreases, and cells are driven exclusively by one eye or the other.

The mature brain

Developmental mechanisms Neuronal properties in the mature brain can be reorganized to an appreciable extent, either in response to lesions or by experience [Donoghue, 1995; Weinberger, 1995].

The change on the neuronal level depends on the region and on the type of influence. It can happen by spread of axons, formation of new synapses, and most often by the regulation of existing synapses. This regulation is mediated by electrical activity as discussed in Section 3.3. The time scale of synaptic change goes from milliseconds and minutes (synaptic facilitation and depression) to weeks and month (long term potentiation and depression).

The case of ocular dominance columns Monocular deprivation leads to a shift in ocular dominance in the primary visual cortex when the manipulation happens during the critical period, but not when it happens in adult animals. This does not imply, however, that the adult cortex is not plastic. As shown for example in [Chino et al., 1995], bilateral symmetric retinal lesions which produces in the primary visual cortex a zone of binocularly deprivation induces cortical reorganization. The deafferented neurons in the cortex become sensitive to new portions of the visual field with nearly normal response properties.

3.3 The Hebbian synapse

In Section 3.2.3, we have seen that patterned neural activity can influence the neural system. This can happen through changes in the strength of the synaptic connections between the neurons. The "Hebbian synapse" is a theoretical construct which models properties of synaptic plasticity. In Section 3.3.1, we give a summary of what the term Hebbian synapse all implies. In Section 3.3.2, we highlight fundamental differences between the Hebbian synapse in firing-rate based modeling and in spike-timing based modeling. Section 3.3.3 reviews that Hebbian synaptic plasticity is a model for the neural mechanisms which cause the formation of neural maps.

3.3.1 Characterization

Synaptic plasticity is here characterized along the axes activity dependency, spatial specificity, associativity, and stability.

Activity dependency

Patterned activity of the pre- and postsynaptic cells was found to induce modifications in the synapse. Depending on how activity is measured, the modifications in the synapse are modeled differently. Activity is measured by means of the firing rate or by means of the spike timings. The firing rate is a macroscopic quantity, which is obtained after averaging, see Section 3.1.3. Firing rate based modification rules are therefore considered to be macroscopic learning rules while spike timing based rules are referred to as microscopic rules.

Firing-rate based rules for synaptic plasticity are summarized in Table 3.8 according to [Rauschecker and Singer, 1981] : It is the concurrent firing of the pre- and postsynaptic cell which leads to a strengthening of the synaptic efficiency to fire the postsynaptic cell.

In the setting where activity is measured by means of spike-timings, the order of the spiking of the pre- and postsynaptic neuron becomes important. If the presynaptic neuron spikes before the postsynaptic neuron (pre-post), the synaptic efficiency in firing the post-synaptic cell gets stronger. If the postsynaptic neuron fires first (post-pre), the synaptic efficiency is weakened. This is illustrated in Figure 3.21. The consequence of this rule for synaptic plasticity, where the precise timing of single spikes are important, is that the rule can serve to detect causality. The temporal order of firing does not enter into the update rule in case of firing-

rate dependent plasticity. Even if the presynaptic neuron fires by chance after the postsynaptic neuron and has hence not caused its firing, the efficiency of the synaptic connection is increased.

Post \ Pre	Active	Silent
Active	Synaptic efficiency ↑	Synaptic efficiency ↓
Silent	Synaptic efficiency ↓	Synaptic efficiency ↓

Table 3.8: Synaptic modification for the firing rate setting [Rauschecker and Singer, 1981]. Synaptic efficiency denotes the capacity of the presynaptic cell to make the postsynaptic cell fire. Concurrent firing of the pre- and postsynaptic cell leads to an increase of the synaptic efficiency. The other cases to a decline of the synaptic efficiency. This decline is sometimes omitted, i.e. the synaptic strength is not altered.

Spatial specificity

A longstanding dogma was that only synapses between active pre- and postsynaptic cells become modified. This means that, with reference to Figure 3.22, the synapse S_1 connecting the presynaptic neuron to neuron N_1 would not become modified if N_1 is silent. This property is called locality. There is however evidence for pre- and postsynaptic lateral spread, as well as back-propagation [Bi and Poo, 2001]. We illustrate this behavior in Figure 3.22.

Associativity

The characteristics associativity is related to spatial specificity and activity dependency. Consider the case where the postsynaptic neuron is active, but neuron N_2 in Figure 3.22 is only so weakly active that no modification occurs in the connecting synapse S_2. Associativity means that if the presynaptic neuron is active as well, the synapse S_2 is modified together with the synapse S.

Stability

Stability of a synaptic connection is conceptually the opposite requirement to plasticity. Several mechanisms are proposed to reach a balance in this trade-off [Cooper, 2005]. In the setting of spike-timing dependent plasticity in Figure 3.21, and also in the setting of the firing rate based plasticity of Table 3.8, the

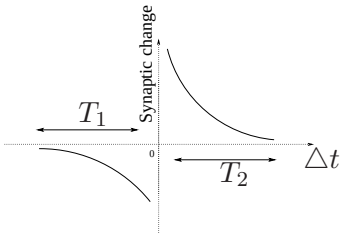


Figure 3.21: Synaptic modification in the spike timing setting: Spike timing dependent plasticity (STDP) and critical windows. The time $\triangle t$ between the pre- and postsynaptic spike is experimentally measured as the time between the onset of the excitory postsynaptic potential (EPSP) and the postsynaptic spike. Negative intervals mean that the postsynaptic cell emitted a spike before the EPSP was triggered by a presynaptic spike. The critical windows T_1 and T_2 differ from brain region to brain region. In case of spike trains, the STDP rule was not found to add up linearly if either the presynaptic neuron or the postsynaptic neuron fired consecutively. The efficacy of each spike in synaptic modification was suppressed by the preceding spike in the same neuron if they were closely spaced together [Froemke and Dan, 2002]. Adapted from [Bi and Poo, 2001].

modification rule itself brings relative stability. Another proposition is synaptic redistribution where the synaptic strength is only modified around a fixed working point to increase the efficiency to transmit transient signals. Synaptic scaling has also been proposed. The idea is that the relative strength of synapses is preserved while their absolute value is globally adjusted.

3.3.2 Spike-timing vs. firing rate

We have mentioned in the previous section that the temporal asymmetry of spike-timing dependent plasticity (STDP) respects causality while the firing-rate based plasticity does not. We discuss here consequences of the temporal asymmetry in case of a network of neurons. The presented results were established in [Song and Abbott, 2001; Song et al., 2000].

Emergence of competition

Figure 3.23 shows three neurons N_1 to N_3 which converge on the single neuron N_4. Assuming that the four firing times are arranged as shown in the figure, the efficiency of synapse S_{41} and S_{42} is according to Figure 3.21 increased by

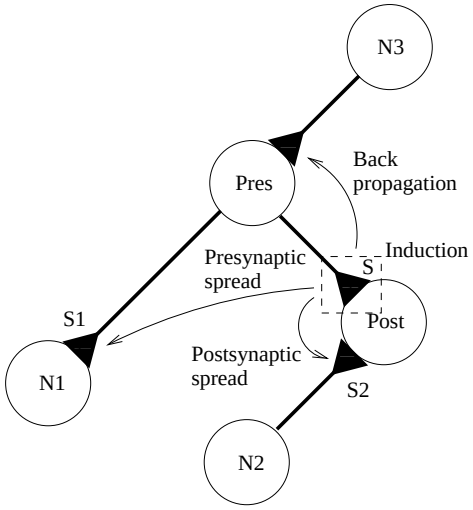


Figure 3.22: Illustration of spatial specificity. Given that synapse S is modified, the dogma of locality is saying that the synapse S_1 is not modified if neuron N_1 is silent. There are cell-specific findings where this dogma does not hold. Synapses which are on the same postsynaptic neuron as S were found to be modified as well if synapse S is modified (postsynaptic spread). Similarly, synapses on the same presynaptic axons as S can be altered too (presynaptic spread). Finally, back-propagation of the synaptic modification to synapses on the immediate input side was also observed. Adapted from [Bi and Poo, 2001].

STDP while the efficiency of synapse S_{43} is reduced. Since $\triangle t_{41} > \triangle t_{42}$, the strengthening of synapse S_{42} is stronger than the one of S_{41}. This means that short latency input is more effective. Since N_1 and N_2 fire in a correlated fashion they can more easily elicit a postsynaptic spike than N_3. This means that synapses which receive correlated inputs are more likely to be strengthened. Since the change of synaptic efficiency depends on the timing of the post-synaptic spike, competition for the control of the post-synaptic spike timing emerges. Neuron N_3 is not good at controlling the spike timings of N_4 and synapse S_{43} is therefore loosing synaptic efficiency.

In case of firing-rate dependent plasticity, the above scheme does not happen. Neuron N_3 shows about the same firing-rate overlap with neuron N_4 as N_2 so that the efficiency of S_{43} and S_{24} is equally increased. Synapse S_{41} is also increased. But since neuron N_1 shows a firing-rate pattern with less overlap, the increase is also less pronounced.

Emergence of driver and teacher neurons

The temporal asymmetry in STDP leads for the recurrent connection and firing pattern shown in Figure 3.24 to a strengthening of S_{21} and a weakening of S_{12} so that neuron N_1 can more strongly drive neuron N_2 but the influence of N_2 on N_1 is reduced. In case of firing rate based plasticity, the efficiency of both S_{21} and S_{12} is increased. This leads to a reinforcing loop that may cause stability problems.

Assume that N_1 in Figure 3.24 fires prior to N_2 because it is more strongly driven by an input neuron N_0 than N_2, as shown in the initial configuration in Figure 3.25. With the strengthening of S_{21} due to STDP, N_1 drives N_2 more strongly so that the latency of the spike of N_2, given a spike of input neuron N_0, decreases. This reduced latency causes an increase in efficiency of synapse S_{20}. Eventually, N_2 fires at about the same time as N_1 (sometimes later, sometimes earlier) so that the synaptic efficiency of S_{21} is decreased by STDP. On the other hand is $\triangle t_{20} > 0$, so that S_{20} is strengthened. This process converts the initial network configuration into the end configuration of Figure 3.25. Neuron N_2 behaves similarly as N_1 and is equally driven by N_0. Neuron N_1 is said to act as teacher for N_2, and the teaching activity ceases as the pupil N_2 begins to behave as N_1.

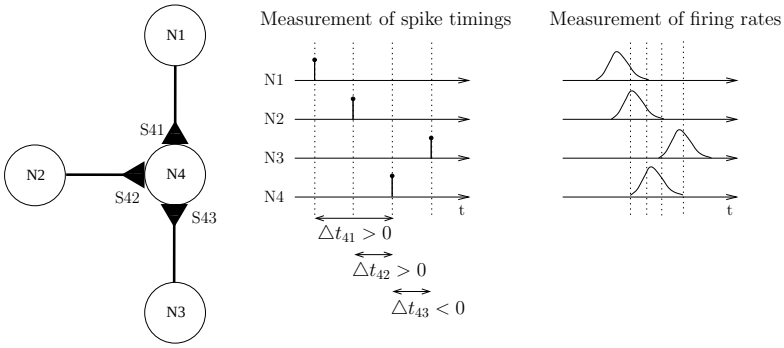


Figure 3.23: For STDP, competition for the control of the timing of the postsynaptic spike emerges. Synaptic efficiency of S_{41} and S_{42} is increased while the one of S_{43} is reduced. For firing-rate based plasticity, the efficiency of all synapses is here increased.

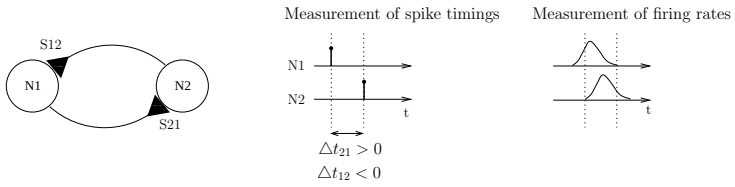


Figure 3.24: The situation of recurrent connections. If neuron N_1 fires consistently before neuron N_2, in case of STDP, the efficiency of the synaptic connection S_{21} is increased but that of S_{12} is reduced. In case of firing-rate based plasticity, the efficiency of both connections is increased. As the temporal asymmetry in STDP leads to strengthening of S_{21} and weakening of S_{12}, neuron N_1 can more strongly drive neuron N_2 but the influence of N_2 on N_1 is reduced.

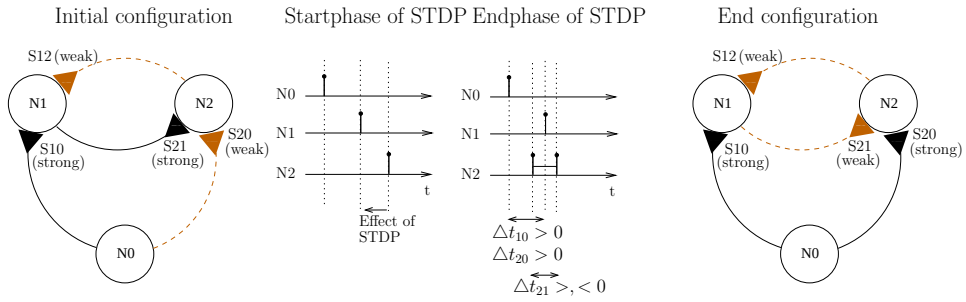


Figure 3.25: Assume that both N_1 and N_2 of Figure 3.24 receive input from neuron N_0, but with S_{10} being strong and S_{20} being weak. Neuron N_1 drives N_2 so that the latency of the spike of N_2 is reduced (middle left). As $\triangle t_{20}$ becomes smaller, the efficiency of S_{20} is increased so that also N_0 drives N_2. Neuron N_2 fires finally at about the same time as N_1 (sometimes earlier, sometimes later: middle right). This decreases the efficiency of S_{21} so that the shown end configuration results.

3.3.3 Modeling the development of neural maps

We review here that Hebbian synaptic plasticity provides a model for the neural mechanisms which lead to the formation of neural maps. We treat first firing-rate and then spike-timing dependent plasticity (STDP).

Firing-rate based plasticity

The application of firing-rate based plasticity rules to the study of neural maps has a long tradition, see e.g. [Rauschecker and Singer, 1981]. Extensive reviews are [Miller, 1996; Swindale, 1996]. We review here the approach to ocular dominance columns of [Miller et al., 1989].

Figure 3.26 shows the network architecture which was used in the study. Simulated contra- and ipsilateral eye layers of the LGN project to a simulated V1 layer (see also Figure 3.18). The N dimensional activity vectors $\mathbf{a_1}$ and $\mathbf{a_2}$ of the contra- and ipsilateral eye-layer are fixed and are described by the correlation matrices $R = E[\mathbf{a_i}\mathbf{a_i}^T]$ $(i = 1, 2)$ and $P = E[\mathbf{a_1}\mathbf{a_2}^T]$. The cortical interaction function $b(\beta_0, \beta_1)$ between two cortical neurons at location β_0 and β_1 has also been fixed. The study investigates the development of the feedforward connections s_i $(i = 1, 2)$ from the two LGN layers to V1.

The M dimensional cortical activity vector $\mathbf{a_c}$ was obtained via

$$\begin{aligned} \mathbf{a_c} &= \mathbf{Sa} + \mathbf{Ba_c} && (3.35) \\ &= \mathbf{ISa}, && (3.36) \end{aligned}$$

where $\mathbf{S} = [\mathbf{S_1} \quad \mathbf{S_2}]$, $\mathbf{a}^T = [\mathbf{a_1}^T \quad \mathbf{a_2}^T]$, and $(\mathbf{S_i})_{\beta\alpha} = s_i(\beta, \alpha)$ for $i = 1, 2$, $(\mathbf{B})_{\beta_0,\beta_1} = b(\beta_0, \beta_1)$, and $\mathbf{I} = (\mathbf{1} - \mathbf{B})^{-1}$. This corresponds to the firing-rate based modeling approach presented in Section 3.1.3. Plasticity of the feedforward connections $\mathbf{S}$ was modeled as

$$\begin{aligned} \dot{\mathbf{S}} &= \mathbf{a_c a}^T - \gamma \mathbf{S} && (3.37) \\ &= \mathbf{ISaa}^T - \gamma \mathbf{S} && (3.38) \end{aligned}$$

The first term on the right models the increase in synaptic efficiency if the pre- and postsynaptic cell are concurrently active ($\mathbf{a} > 0$, and assuming $\mathbf{a}_c > 0$) while the second term models the decrease of synaptic efficiency, see also Table 3.8. The analysis in [Miller et al., 1989] proceeds by taking the average (with the assumption $E(\mathbf{Saa}^T) = E(\mathbf{S})E(\mathbf{aa}^T)$), and introducing $\mathbf{S^D} = E(\mathbf{S_1} - \mathbf{S_2})$. If $(\mathbf{S}^D)_{\beta,\alpha} > 0$ the contribution of the contralateral input from the LGN cell at location α to the cortical cell at β dominates. Else the ipsilateral input dominates. This matrix of differences of synaptic strength verifies

$$\dot{\mathbf{S}}^\mathbf{D} = \mathbf{IS^D C^D} - \gamma \mathbf{S^D}, \tag{3.39}$$

with $\mathbf{C^D} = \mathbf{R} - \mathbf{P}$, the difference between autocorrelation and crosscorrelation matrix of the LGN activity. This matrix equation, with initial condition $\mathbf{S^D} = \mathbf{0}$ can be solved analytically when $\mathbf{C^D}$ and I are assumed to be circular matrices. With the ansatz

$$\mathbf{S^D}(t) = \sum_{\beta,\alpha=1}^{M,N} M(k, l, t) \exp(i2\pi k\beta/M) \exp(i2\pi l\alpha/N) \mathbf{e}^{(\beta)} \mathbf{e}^{[\alpha]}, \tag{3.40}$$

where $\mathbf{e}^{(\beta)}$ is a column unit vector while $\mathbf{e}^{[\alpha]}$ is a row unit vector, one obtains a differential equation for $M(k, l, t)$, which is

$$\frac{dM(k, l, t)}{dt} = (\tilde{I}(k)\tilde{C}^D(l) - \gamma) M(k, l, t). \tag{3.41}$$

The tilde terms in this equation denote the Fourier transforms of the first column of the respective matrix. Solving this equation, one obtains for $s^D(\beta, \alpha, t) =$

$\mathbf{e}^{[\beta]}\mathbf{S}^{\mathbf{D}}(t)\mathbf{e}^{(\alpha)}$ the following solution

$$s^D(\beta,\alpha,t) = \sum_{k,l} M(k,l,0)\exp\left[i2\pi\left(\frac{l\alpha}{N}+\frac{k\beta}{M}\right)\right]\exp\left[(\tilde{C}^D(l)\tilde{I}(k)-\gamma)t\right]. \tag{3.42}$$

If there are some (l,k) such that $\tilde{C}^D(l)\tilde{I}(k)-\gamma>0$, the initial feedforward connectivity pattern is unstable and ocular dominance maps may emerge. Each pair (l,k) defines a characteristic pattern, and the fastest growing pattern will determine the resulting ocular dominance structure. We denote the dominant pattern by (l^*,k^*).

The relative contribution of ipsi- and contralateral input from the LGN, as measured by s^D oscillates for a fixed cortical location β as one moves over the LGN-space, i.e. as α varies. The degree of oscillation is given by l^*. This implies that the degree of monoocularity depends on the input correlations $\mathbf{C}^{\mathbf{D}}$. In order to have cortical mono-ocular cells, $l^*=0$.

The dominant k^* determines the oscillations as one moves across V1. As $s^D(\beta,\alpha)>0$ means that the contralateral LGN cell at α dominates the ipsilateral LGN with respect to the input to cortical cell β, oscillations of s^D imply that there are alternating regions of contra- and ipsilateral ocular dominance. In short, ocular dominance maps emerge if $k^*\neq 0$. The width of the columns is determined by the value of k^*, i.e. by the intracortical connections b. If for example no such lateral connections exist no ocular dominance map can be formed. If further $k^*=0$, the whole simulated V1 is dominated by either ipsi- or contralateral input (assuming $l^*=0$), i.e. there is only one selectivity column.

STDP

The setup of Figure 3.26 was also used by [Song and Abbott, 2001] to study the formation of neural maps by means of spike-timing dependent plasticity (STDP). Integrate and fire neuron models (see Section 3.1.3) were used to simulate neural activity. As in the case of firing rate based plasticity, the activity of the LGN layer was decided by the modeler and the feedforward connections were subject to STDP. In Figure 3.24, we have illustrated that recurrent connections are problematic in the framework of firing rate based plasticity, but not so in the case of STDP. That is why in [Song and Abbott, 2001], in contrast to [Miller et al., 1989] where the intracortical connections were fixed, the recurrent connections of the simulated V1 were subject to STDP as well.

Neural maps also emerge after the learning with STDP. As in the firing-rate

based framework, it is the cortical connections which decided whether neural maps, selectivity columns, or no columns at all emerge. For the formation of maps, inhibitory connections were needed additionally to the excitory connections which were subject to STDP (and constrained to a fixed neighborhood). Excitory connections alone led to the formation of a single selectivity column. If the modeling of lateral connections was omitted, no selectivity column emerged.

The study shows that firing-rate based plasticity and STDP lead essentially to the same outcomes although STDP works with a sparser set of assumptions and a more realistic neuron model. There has been research if this equivalence does also hold for the modeling of the reorganization of V1 after retinal lesions [Young et al., 2007]. Receptive fields of neurons in V1 that are inactivated by retinal damage were found to shift to a non-damaged retinal location. The firing-rate based learning rule in this study could not account for this experimental finding. STDP, however, reproduced the results.

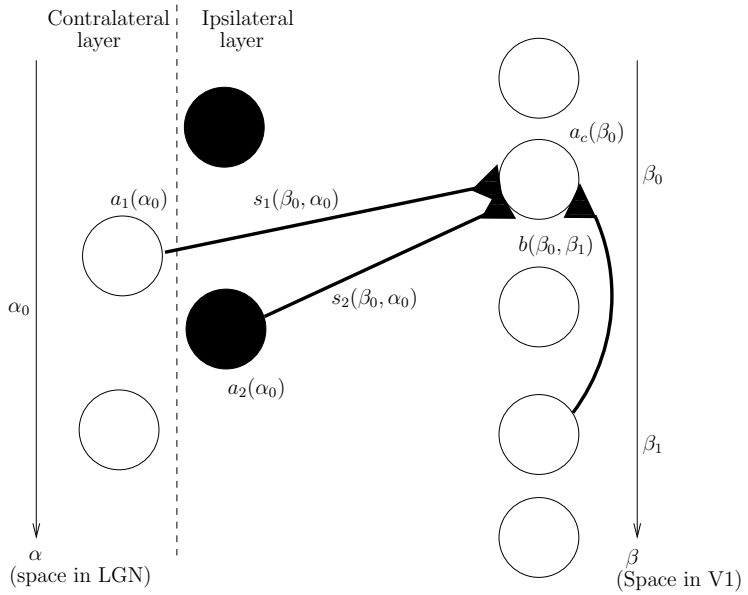


Figure 3.26: Network architecture for the study of cortical map formation by means of Hebbian synaptic plasticity. In [Miller et al., 1989], firing rate based plasticity is used. The activity $a_i(\alpha_0)$ of a LGN neuron at location α_0 in contralateral layer ($i = 1$) or ipsilateral layer ($i = 2$) is given by the firing rate. Cortical firing rate $a_c(\beta_0)$ of a cortical cell at location β_0 is given by a weighted linear sum of the firing rates of the convergent neurons. The weighting is given by the scalar $s_i(\beta_0, \alpha_0)$ for a cell of the LGN located at α_0, and the scalar $b(\beta_0, \beta_1)$ for a cortical cell located at β_1. In [Song and Abbott, 2001], spike-timing dependent plasticity is applied to this kind of network. The neuron were however modeled by means of integrate and fire neurons (see Section 3.1.3. Another difference is that in [Miller et al., 1989] the cortical connections were fixed while in [Song and Abbott, 2001], their synaptic efficiencies were subject to STDP.

3.4 Summary

We provide here a summary of the most important points of this chapter. We start with biophysics of signaling and energy consumption (Section 3.1.1 and 3.1.2). We summarize then neuron models (Section 3.1.3). After that, the topic of input fragmentation and signal ambiguity is treated (Section 3.2.1). Then, the relation between neural maps and neural integration is summarized (Section 3.2.2). This is followed by possible principles (Section 3.2.2) and mechanisms behind neural integration (Section 3.2.3 and 3.3).

Biophysics of signaling and energy consumption

1. The membrane voltage acts as the basic signaling quantity in neurons. The energy which is needed to perform signal transformations is stored in the gradient of the ion concentrations across the cell membrane.

2. Maintenance of this gradient is needed to assure the signaling-capacity of the neuron. The neuron absorbs energy for this maintenance.

3. Energy uptake can be divided into a base-line consumption and a consumption which is related to the creation and propagation of action potentials, and to incoming postsynaptic currents.

Neuron models

1. Neurons show different excitability types. A basic classification distinguishes between those which feature oscillations upon weak, shock-wise stimulation (resonators), and those which do not (integrators). These two types can be further divided into those which return to their rest state after a strong shock input (monostable) and those which continue emitting action potentials (bistable).

2. Simple models for these four excitability types forego the modeling of the whole voltage trajectory during an action potential. In these models, the time of the onset of an action potential becomes the signaling variable.

3. Another framework uses the firing rate to describe the behavior of neurons. In this case, differential equations may not be needed to model neural behavior. Firing rates are averaged quantities so that the resolution is not

as high as in the case of, for example, the simple models for the different excitability types.

Input fragmentation and signal ambiguity

1. Receptor cells transduce the sensory stimulus into electrical signals. This transduction process leads to a fragmentation of the stimulus and there is some ambiguity in the resulting electrical signal in the sense that different stimuli can lead to the same response.

2. For a proper understanding of "what is out there", ambiguity must be resolved and fragmented information must be glued together by the recombination of receptor signals. We call this recombination of receptor signals to form a neural representation of the sensory stimulus neural integration.

3. For the special case of daylight vision, the transduction happens by means of the cone photoreceptors. The arrangement of cones on the retina determines the spatial fragmentation, and the spectral properties of the cones determine the spectral fragmentation of visual input. A cone can catch and transduce photons of any frequency and spatial origin. Each photon contributes also equally to cone activity. This is the origin of the signal ambiguity of the cone receptors.

4. In the retina, the signals from 4 million cones converge onto about 1.6 million ganglion cells. Parts of these ganglion cells reflect the neural integration happening on the level of the retina. They show clear spatial and temporal frequency preferences, color preferences and contrast sensitivities. The cells have often joint tuning properties such as for example the pair red-green sensitivity and spatial resolution for the midget cells.

Neural maps and neural integration

1. Cells with similar tuning properties are often clustered together to form cell assemblies. These cell assemblies are well arranged among each other so that they are thought to form a representation of the sensory input by means of the neural tissue. This organization of cells with clear tuning properties is called a neural map.

2. For vision, the LGN shows organization with respect to at least receptive field center, ocular dominance, and spectral sensitivity. In V1, organization

with respect to direction, spatial frequency and orientation preference is added.

3. This additional degree of organization of V1 reflects the neural integration which happens from LGN to V1. Neural maps are useful because their comparison across the sensory pathway allows to get hold of the action of neural integration.

Possible principles of neural integration

1. The principles of smoothness (minimal wiring to assure energy efficiency), coverage (representation accuracy), and response invariance can account for many properties of neural maps.

2. As neural maps visualize the effects of neural integration, these principles can be considered to be also important for neural integration.

Possible mechanisms of neural integration

1. Understanding the neural mechanisms which lead to the formation of neural maps helps to understand the mechanisms of neural integration.

2. Hebbian synaptic plasticity can cause the formation of neural maps both in the framework of firing rate based plasticity and spike-timing dependent plasticity. Hebbian plasticity may therefore also be a neural mechanism which is important for neural integration.

3. Correlations in the synaptic input influence strongly the development of synapses which are subject to Hebbian plasticity (both for the case of firing rate and spike-timings). They influence in that way the selectivity of the postsynaptic neuron. The structure of the cortical connections determine the clustering of the neurons and therefore the structure of the neural map.

4. Spike-timing dependent plasticity was shown to be able to account for cortical reorganization after lesions which firing-rate dependent plasticity could not explain.

Chapter 4

Previous Work and Research Questions

4.1 Data representation as a model for neural integration

In Section 4.1.1, we model neural integration by means of data representation. This modeling approach allows to draw several analogies between the two research fields. Both emphasize the role of the input space. In Section 4.1.2, we introduce for that reason the use of natural stimuli and discuss statistical properties of natural scenes.

4.1.1 Analogies

In Section 3.2, we have presented the neuroscience topic "neural integration". If an input signal is transduced by sensory receptors, the input is fragmented and each receptor signal shows some ambiguity. Neural integration is about the recombination of receptor signals to reduce the ambiguity and to invert the fragmentation, which leads to a neural representation of the sensory stimulus that was sensed by the receptors. We illustrate neural integration in Figure 4.1.

In Section 2.1, we have presented the mathematics topic "data representation". If a random vector is presented in the standard basis, its properties may not be easily visible. Data representation is about the conversion of the data in a representation so that the structure and peculiarities of the data are made explicit. We illustrate data representation in Figure 4.2.

If we take the representation of the input by means of the standard basis as a model for input fragmentation, we can draw a series of analogies between data representation and neural integration. The representation of N dimensional input data $\mathbf{x}(t)$ with respect to the standard basis is as in Equation (2.72)

$$\mathbf{x}(t) = \sum_m x_m(t)\mathbf{e^{(m)}}, \tag{4.1}$$

which can be rewritten as

$$\mathbf{x}(t) = \sum_m \sum_n x_m(n)\mathbf{e^{(m)}}(t-n), \tag{4.2}$$

where $\mathbf{e^{(m)}}(t-n)$ is a N dimensional unit vector which is always zero but at $t=n$ where its m-th component values one. The representation of the data in an adapted basis, i.e.

$$\mathbf{x}(t) = \sum_m \sum_\tau s_m(\tau)\mathbf{h_m}(t-\tau), \tag{4.3}$$

models then the effect of neural integration. Basis vector $\mathbf{h_m}$ models the selectivity of a neuron m after the neural integration, and the coefficient $s_m(\tau)$ stands for the activity of that neuron. Time shift τ shows the time of the activity of the neuron. In other words, the change of selectivities from neurons on neural level A to neurons on neural level B is modeled by means of the selectivity transform. The change of activities is modeled by the coordinate transform. Neural encoding of the input $\mathbf{x}(t)$ means then the calculation of the coefficients $x_m(n)$ or $s_m(\tau)$. Decoding of $\mathbf{x}(t)$ from the neural activity means the reconstruction of $\mathbf{x}$ from the coefficients $x_m(n)$ or $s_m(\tau)$ as in Equation (4.2) or Equation (4.3), respectively.

In Section 3.2.2, we have pointed out some important principles for neural integration. These were smoothness (minimal wiring to assure energy efficiency), coverage (representation accuracy), and response invariance. These principles have their analogues in the cost functionals which were used in Section 2.2 and 2.3 to obtain the selectivity and coordinate transform. Maximizing coverage has its analogue in the minimization of the reconstruction error. Maximal smoothness or minimal wiring is in Section 2.3.3 used explicitely, and also in form of additional constraints for the optimization. The decoding approach of Equation (4.3) relates to the invariance property. Each neuron has its tuning property, which is modeled by the basis vector $\mathbf{h_m}$, and the activity of the neuron is an indicator for the presence of that stimulus characteristics in the input.

4.1.2 The input data

We discuss the role of the input space where the signals are drawn from which get by the sensory system first fragmented and then integrated. We introduce the idea of using natural stimuli to reduce the number of tuning parameters which are under control of the researcher. Statistical properties of natural scenes, both snapshots and movies, are discussed.

Choice of the input space

In Section 3.3.3, we have presented a study [Miller et al., 1989] where Hebbian synaptic plasticity was used to learn neural maps. The study showed that correlations in the activity of the LGN neurons, which form the lower layer, affect the emergent selectivity of the V1 neurons, which form the next layer. The statistical properties of the input space are thus of importance. They were specified by the researcher, and it is not clear what choice is the most appropriate. In Section 3.2.2, we have reviewed work [Das, 2005; Yu et al., 2005] which established

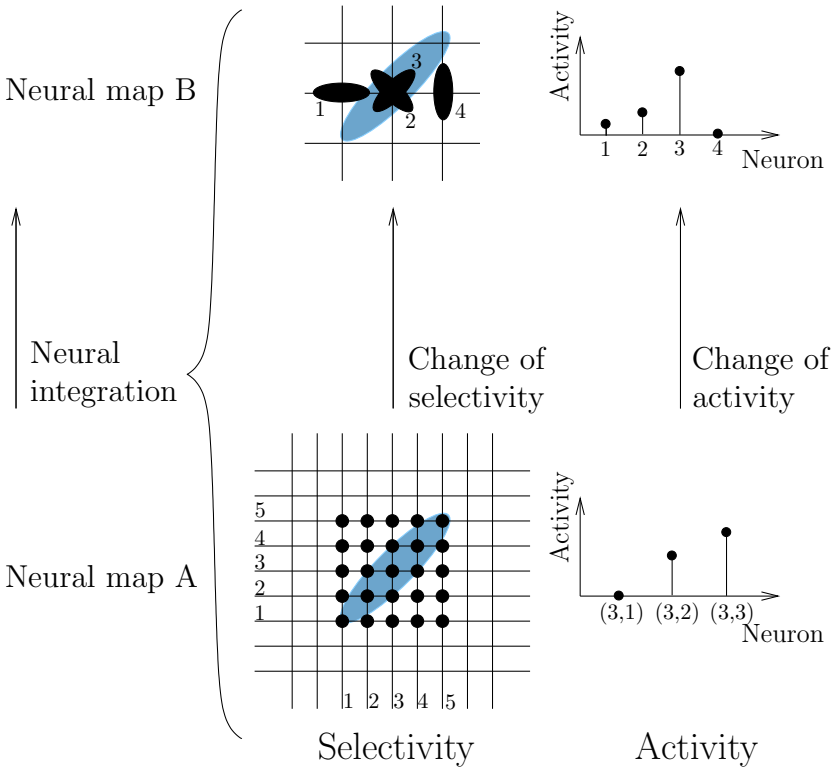


Figure 4.1: Neural integration. Moving along the sensory pathway, the properties of the neurons and the way they are grouped together to form cell assemblies changes. In this schematic, the selectivities of neurons at the level A are rather unspecific. They react whenever a dotshaped stimulus is present. This leads to neural activities as shown on the right. Neurons at the level B show orientation selectivity. The representation of the stimulus through the neural activities changes from level A to level B: For the given stimulus in blue is neuron 3, which has the same preferred orientation as the stimulus, strongly active. Neuron 2, however, less so. Neuron 4 is selective for stimuli at another spatial location and does not respond at all. The dotshaped selectivities at level A lead to a fragmentation of the stimulus. The change of selectivity of the neurons shows that these fragments are combined as one moves from level A to level B, and the neural representation of the stimulus changes accordingly.

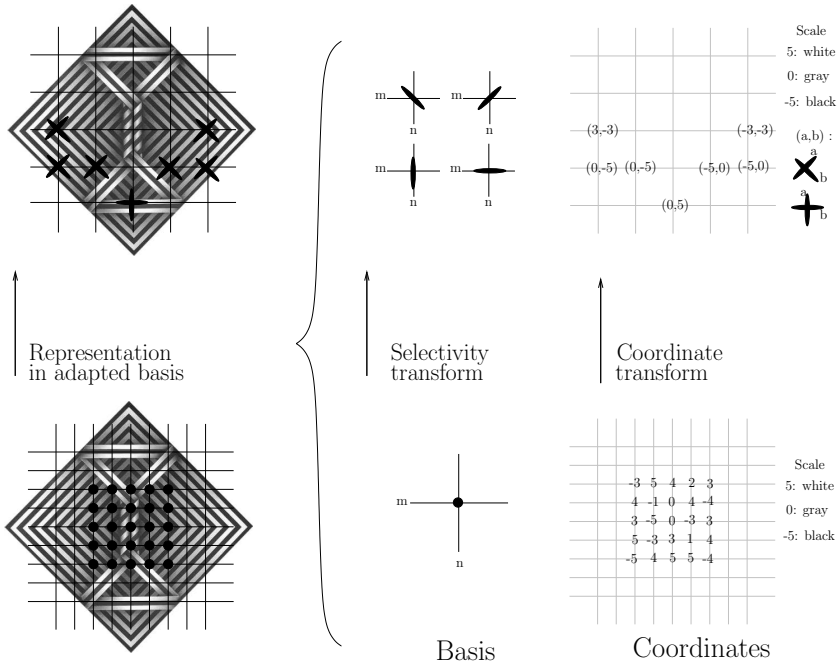


Figure 4.2: Data representation. The image can be represented in the standard basis or also in a basis which is more adapted to the structure of the image. The standard basis leads to a pixelization of the image where the structure of the image is hidden in the activity pattern of the coordinates. If an adapted basis is used, the oriented bandlike structure of the image can more easily be read out from values of the coordinates.

the importance of the principals smoothness (or minimal wiring), coverage, and response invariance for the formation of neural maps. The distribution of the signal parameters which were mapped to the simulated neural tissue was also fixed by the researcher.

Having the statistical properties determined by the researcher has the advantage that the interpretation of the results can happen in controlled manner. On the other hand, it is a tuning parameter which is under control of the researcher and thus due to some arbitrariness. Another approach consists in using natural stimuli in the research. If the interest is in vision, the input data corresponds to snapshots or movies which were taken in a natural environment. If the interest is in auditory processing, natural sounds (speech, environmental sounds) is taken as input data. The use of this kind of ecologically valid input eliminates the aforementioned freedom in the choice of the input data. On the other hand, properties of natural stimuli are not well understood so that the interpretation of the results becomes difficult. The use of data representation as a model for neural integration shows that the difficulty in the interpretation of the results might be offset by the possible benefit of gaining insight into the structure of natural stimuli.

Statistical properties of natural scenes

Natural scenes (after sampling) are modeled as a random process $x(\mathbf{n}, t)$, where the index $\mathbf{n} = (n_x, n_y)$ labels spatial location and the index t is a label for time. All indices are supposed to run over a finite range. The double index for the spatial location can be converted in a single index by stacking the columns of each image upon each other. Natural scenes are then modeled by the time dependent random vector $\mathbf{x}(t)$. Stationarity is assumed in both space and time since no point should be anyhow special. We review first statistical properties when the temporal dimension is ignored (snapshots). Then we deal with the time dependent case (movies).

Autocorrelation of snapshots The second order moments of natural snapshots have been characterized in several studies by calculation of the autocorrelation $R_x(\mathbf{n_2} - \mathbf{n_1}) = E(x(\mathbf{n_1})x(\mathbf{n_2}))$. Note that in the different studies the preprocessing of the images, e.g. calibration or correction for the sampling process, was always done in a different way.

The Discrete Time Fourier Transform (DTFT) of the autocorrelation function gives the power spectrum $S_x(\mathbf{k}) = \sum_{\mathbf{n}} R_x(\mathbf{n}) \exp(-i\mathbf{kx})$. The energy in a certain

spatial frequency band $[k_1\ k_2]$ is given by

$$E(k_1, k_2) = \int_{k_1}^{k_2} \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(k, \phi) \mathrm{d}\phi}_{S_x(k)}\ k \mathrm{d}k, \tag{4.4}$$

where $S_x(k)$ denotes the power spectrum averaged over different orientations. This averaged power spectrum was found to follow a power law [Balboa and Grzywacz, 2003; Field, 1987; Ruderman and Bialek, 1994; Ruderman, 1994, 1997; van der Schaaf and van Hateren, 1996],

$$S_x(k) \propto \frac{1}{k^{2-\eta}}, \tag{4.5}$$

where η takes values around 0.2, depending on the study. Evaluation of the integral shows that each octave of spatial frequency contains the same amount of energy, i.e. $E(k_1, nk_1) \propto \ln(n)$ for a given positive integer n.

Using the property of the 2-D Fourier Transform

$$\mathrm{DTFT}^{-1}\left[g(a\mathbf{k})\right](\mathbf{n}) = \frac{1}{a^2}\mathrm{DTFT}^{-1}\left[g(\mathbf{k})\right](\mathbf{n}/a), \tag{4.6}$$

we obtain with $g(a\mathbf{k}) = S_x(ak)$ and $S_x(ak) = 1/(a^{2-\eta})S_x(k)$ that

$$\frac{1}{a^2} R_x\left(\frac{r}{a}\right) = \frac{1}{a^{2-\eta}} R_x(r). \tag{4.7}$$

Hence, the autocorrelation $R_x(r/a)$ is given by

$$R_x\left(\frac{r}{a}\right) = \left(\frac{1}{a}\right)^{-\eta} R_x(r), \tag{4.8}$$

with radius r. The second order statistical properties of images are thus, up to a normalization constant, invariant to zooming (upsampling or downsampling), see Figure 4.3. A powerlaw relation is also found for contrast $\Phi = \ln[x/x_0]$, where x_0 is a scaling constant[Ruderman and Bialek, 1994], and is a robust phenomenon which was observed for images taken from a variety of different habitats, for the example of underwater images, see [Balboa and Grzywacz, 2003].

Reasons for the power law of the power spectrum have been proposed to relate to the probability distribution of the sizes of objects, and their mutual occlusions when occuring in a natural image [Ruderman, 1997]. This explains also the robustness of the occurrence of the power law distribution for all kinds of calibrations and preprocessings of images: as long as the preprocessing is not such that it erases the possibility to locate an object in space, the resulting power distribution should follow a power law.

Self similarity of snapshots Natural snapshots show further properties related to scaling. They are thought to be self-similar. A random process is said to be self-similar of index H if $\{x(an)\}_n$ is equal to $\{a^H x(n)\}_n$. From that definition, it follows that the marginal distributions $f_{x(an)}$ of a self-similar random process verify

$$f_{x(an)}(\beta) = \frac{1}{a^H} f_{x(n)}\left(\frac{\beta}{a^H}\right). \tag{4.9}$$

This implies that $\mathrm{Var}(x(an)) = a^{2H}\mathrm{Var}(x(n))$. The scaling constant a^H is given by the standard deviation σ of a scaled random process which has been of unit variance before scaling. Assuming $\mathrm{Var}(x(n)) = 1$, we have by the definition of a self-similar process

$$f_{x(an)/\sigma_a}(\beta) = f_{x(n)}(\beta). \tag{4.10}$$

The quantity H here values in terms of η in Equation (4.8) $H = -\eta/2$.

In [Ruderman and Bialek, 1994; Ruderman, 1994], self-similarity has been tested in natural snapshots by comparison of the histograms of contrast $\Phi(\mathbf{n})$ with the histograms of contrast $\Phi_N(\mathbf{n})$ which were obtained by the construction of $N \times N$ block pixels and their subsequent renormalization to unit variance:

$$\bar{\Phi}_N(\mathbf{n}) = \frac{1}{N^2} \sum_{\mathbf{m}\in\mathcal{N}(\mathbf{n})} \Phi(\mathbf{m}), \tag{4.11}$$

where $\mathcal{N}(\mathbf{n})$ is a N pixel neighborhood around $\mathbf{n}$, so that

$$\Phi_N(\mathbf{n}) = \frac{\bar{\Phi}_N(\mathbf{n})}{\sqrt{V}ar(\bar{\Phi}_N(\mathbf{n}))}. \tag{4.12}$$

Note that if marginals of natural snapshots were self-similar, this distribution had by equation (4.10) to be the same for all kinds of scalings N. Figure 4.4 shows that it is so for the tested scales.

Autocorrelation of movies Dependencies along the temporal dimension of natural scenes have also been investigated. Spatiotemporal correlations have been measured in [Dong and Atick, 1995]. The measurements showed that the orientation averaged power spectrum $S_x(k,\omega)$ is given by

$$S_x(k,\omega) \propto \frac{1}{k^{3-\eta}} F\left(\frac{\omega}{k}\right). \tag{4.13}$$

The proposed explanation is that the power law emerges from objects with a static power spectrum $\propto 1/k^{2-\eta}$ appearing at a wide range of depths and moving with a distribution of velocities relative to the observer.

First the conditional autocorrelation function $R_x(\mathbf{n}, t|\mathbf{v}, r)$ is calculated, i.e. the autocorrelation given that the objects in the movie move at velocity $\mathbf{v}$ at a distance r from the observer. $R_x(\mathbf{n}, t|\mathbf{v}, r)$ is then given by $R_x(\mathbf{n}, t|\mathbf{v}, r) = R_0(\mathbf{n} - t\mathbf{v}/r)$, where R_0 indicates the static autocorrelation function obtained from the snapshots. Note that we have to use $\mathbf{v}/r$, the velocity of the objects as it appears at location of the observer. The Fourier Transform gives then $S_x(\mathbf{k}, \omega|\mathbf{v}, r) = S_0(\mathbf{k})\delta(\omega - \mathbf{k}\mathbf{v}/r)$. Assuming rotation invariance or averaging over different orientations, we have then

$$S_x(k, \omega) \propto \frac{1}{k^{2-\eta}} \iint \delta(\omega - kv/r) P(v|r) P(r) \mathrm{d}v \mathrm{d}r, \tag{4.14}$$

where the power law behavior of the static spectrum is used. The distances are assumed to be uniformly distributed on $[r_1\ r_2]$ so that

$$\begin{aligned} S_x(k, \omega) &\propto \frac{1}{k^{2-\eta}} \int_{r_1}^{r_2} \mathrm{d}r \int \delta(\omega - kv/r) P(v|r) \mathrm{d}v & (4.15)\\ &= \frac{1}{k^{2-\eta}} \int_{r_1}^{r_2} \mathrm{d}r \int \frac{r}{k} \delta(v - \omega r/k) P(v|r) \mathrm{d}v & (4.16)\\ &= \frac{1}{k^{3-\eta}} \int_{r_1}^{r_2} r P\left(\frac{\omega}{k} r | r\right) \mathrm{d}r & (4.17)\\ &= \frac{1}{k^{3-\eta}} F\left(\frac{\omega}{k}\right), & (4.18) \end{aligned}$$

which is the above relation. In case of high temporal frequencies ω, but low spatial frequencies k, $F(\omega/k)$ becomes proportional to k^2/ω^2 so that

$$S_x(k, \omega) \propto \frac{1}{k^{1-\eta}} \frac{1}{\omega^2} \tag{4.19}$$

is obtained. This means that the power spectrum is in that case separable in space and time.

4.2 Previous work: Application of data representation methods to neuroscience

In Section 4.2.1, we give a neural interpretation of the data representation methods of Section 2.2 and 2.3. We will see that the data representation methods allow to link the principles with the mechanisms of neural integration. In Section 4.2.2, we show that these methods can reproduce neural maps when applied on natural stimuli.

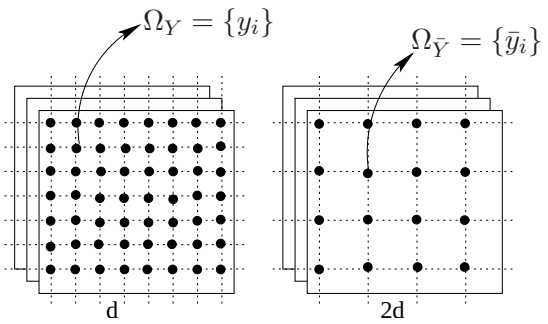


Figure 4.3: Given a set of images, we may sample them at two different rates. The set of images Ω_x is obtained by sampling at the rate $1/d$, and the set $\Omega_{\bar{x}}$ by sampling at $1/(2d)$. Scale invariance with respect to correlation means then that both datasets provide the same statistical information as measured by correlation. In formula, we have $R_{\bar{x}}(r) = R_x(r/a) = (1/a)^{-\eta} R_x(r)$, with $a = 1/2$ and $(1/a)^{-\eta} \approx 0.87$ in the example here. Note that when we look at an object from a farer distance, the object is sampled at a coarser scale at the retina. Scale invariance means then that the statistics contained in the environment as seen from farer away is the same as the statistics contained in the environment as seen at a closer distance.

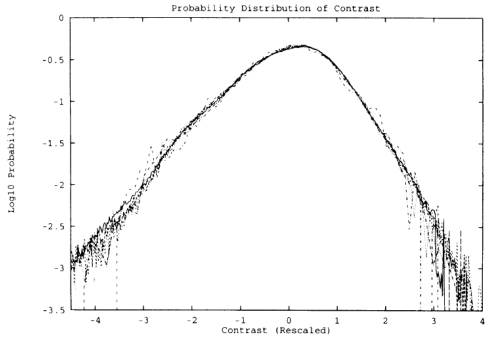


Figure 4.4: Scaling of log-contrast histograms over scales N=1,2,4,8,16, and 32. From [Ruderman and Bialek, 1994; Ruderman, 1994].The figure shows also that the distribution of contrast Φ is far from a Gaussian with nearly exponential tails. This puts automatically a constraint on the usefulness of the power spectrum. Since a Gaussian ensemble of images with the observed power spectrum $S_x(k)$ is the maximum entropy ensemble of images consistent with the power spectrum of natural images, a visual system which views this Gaussian ensemble collects more visual information that one which views the real world.

4.2.1 Linking principles with mechanisms of neural integration

In Section 3.2.2 and Section 3.3, we have reviewed principles and mechanisms, respectively, which can well explain neural maps. The principles, however, are abstract and there is no link to the mechanisms. Here, we will show how modeling neural integration by data representation is able to make a connection between the principles and the mechanisms. Figure 4.5 provides an overview. The principles correspond to the cost functional (see Section 4.1.1) and the iterative optimization leads to learning rules which feature Hebbian mechanisms (reviewed in Section 3.3). We start with the neural interpretation of data representation methods without time structure, then we deal with the time dependent case.

Data representation without time structure

In Section 2.2, we began the review of data representation methods without time structure with the discussion of the cost functional

$$J(\mathbf{H}, \mathbf{W}) = \frac{1}{2} E\left(||\mathbf{x} - \mathbf{H}\mathbf{W}^{\mathbf{T}}\mathbf{x}||_2^2\right). \tag{4.20}$$

With the nomenclature of Section 4.1.1, the input $\mathbf{x}$ is encoded by means of the activity transform

$$\mathbf{y} = \mathbf{W}^{\mathbf{T}}\mathbf{x}, \tag{4.21}$$

so that the activity of each neuron m is given by

$$y_m = \sum_{i=1}^{N} w_{im} x_i, \tag{4.22}$$

where w_{im} indicates the element at the i-th row and m-th column of the matrix $\mathbf{W}$. This element of the matrix models the synaptic strength from input x_i to the neuron m. The above equation corresponds to the input-output relation for linear firing-rate based modeling, which we have introduced in Section 3.1.3. The activity y_m models thus the firing rate of neuron m. Decoding of the neural activities $\mathbf{y}$ happens via

$$\hat{\mathbf{x}} = \mathbf{H}\mathbf{y} \tag{4.23}$$

$$= \sum_m y_m \mathbf{h_m}. \tag{4.24}$$

We have reviewed in Sections 2.2.2 and 2.2.3 iterative optimization schemes to find the optimal matrices $\mathbf{H}$ and $\mathbf{W}$, where in some cases the assumption $\mathbf{H} = \mathbf{W}$

was made. Comparison with Section 3.3 shows that iterative minimization of the reconstruction error leads to the emergence of Hebbian synaptic plasticity. The subspace learning algorithm in Equation (2.27) (see the review [Baldi and Hornik, 1995]) was

$$\triangle \mathbf{W} = \mu \left[(\mathbf{x} - \mathbf{W}\mathbf{y})\mathbf{y}^{\mathbf{T}} \right], \quad \text{or elementwise} \tag{4.25}$$

$$\triangle w_{im} = \mu \left(x_i - \sum_{p=1}^{N} w_{ip} y_p \right) y_m. \tag{4.26}$$

The change $\triangle w_{im}$ is thus proportional to the product of the error

$$e_i = x_i - \sum_{p=1}^{N} w_{ip} y_p \tag{4.27}$$

and the firing rate y_m. The product $x_i y_m$ is the basic element of firing-rate based Hebbian plasticity. While this term leads to an increase in synaptic efficiency, the remaining terms have the opposite effect and lead to a decrease of synaptic efficiency, compare to Table 3.8. The subspace learning algorithm can also be obtained by maximizing output variance under orthogonality constraint of the columns $\mathbf{w_m}$, see Section 2.2.2. Variance maximization leads to the product term $x_i y_m$ while the orthogonality constraint leads to the remaining terms. The other algorithms presented in Section 2.2.2, such as the GHA of [Sanger, 1989] show the same emergence of Hebbian plasticity.

The algorithms in Section 2.2.3 have compared to the subspace learning algorithm additional terms in order to resolve the under-determinedness of the data representation problem, which we have discussed in Section 2.2.1. Punishment of the norm of the columns of $\mathbf{W}$, as proposed in [Vincent et al., 2005; Vincent and Baddeley, 2003], was motivated by the idea of punishing the synaptic strength. Punishment of the norm of the output $\mathbf{y}$, as in [Vincent et al., 2005] means punishment of high firing rates. In Section 3.1.2, we have reviewed that the cost of an action potential increases with the length of the axon. Facilitation of low firing rates shares thus the same goal as the minimal wiring principle of neural integration (Section 3.2.2). Equation (2.48) shows that punishment of high firing rates leads to the additional term

$$\triangle w_{im} = -\mu_3 \text{sign}(y_m) x_i \tag{4.28}$$

in the update rule for the synaptic weights. This is an instance of nonlinear firing-rate based Hebbian plasticity. Nonlinear Hebbian plasticity results also

from the idea of maximizing the statistical independence of the firing rates y_m, see Equation (2.62) and [Hyvärinen and Oja, 1998]. Here, the assumption $\mathbf{H} = \mathbf{W}$ is not made. From the nonlinearities in Table 2.2, which are used to measure statistical independence, we see that the cost functional to maximize independence can also be interpreted as punishment of high output firing rates.

Data representation with time structure

Section 2.3 dealt with data representation methods which feature time structure. We presented methods which minimize the cost functional

$$J = E\left(\sum_t e(t)\right), \tag{4.29}$$

which we have introduced in Equation (2.75). The error $e(t)$ equals $||\mathbf{x}(\mathbf{t}) - \hat{\mathbf{x}}(\mathbf{t})||^2$ where $\hat{\mathbf{x}}$ is the approximation. In Section 2.3.2, a matching pursuit based iterative optimization algorithm was presented [Mallat and Zhang, 1993; Perrinet, 2004; Smith and Lewicki, 2005, 2006]. With the notation of [Smith and Lewicki, 2005, 2006], where $\mathbf{x}(t)$ is a scalar $x(t)$, the reconstruction was

$$\hat{x}(t) = \sum_{m=1}^{M}\sum_{n=1}^{N_m} s_m^n h_m(t - \tau_m^n). \tag{4.30}$$

The number of basis vectors M is fixed, and the algorithm finds $h(t)$, N_m, and the s_m^n from the structure of the input data. In Section 2.3.3, a method due to [Hyvärinen et al., 2003] which works with

$$\hat{\mathbf{x}}(t) = \sum_{m=1}^{N} s_m(t)\mathbf{b_m} \tag{4.31}$$

was presented. It can be extended to include time dependency of the basis vectors, i.e. $\mathbf{b_m}(t)$. The methods works on spatially whitened input $\mathbf{x}(t)$ and the number N equals the dimension of $\mathbf{x}(t)$. The algorithm finds the $s_m(t)$ and the vectors $\mathbf{b_m}$ from the structure of the data.

The neural interpretation of the coefficients $s_m(t)$ and s_m^n of the two algorithms is not the same. Neural encoding for the latter algorithm happens via

$$s_m(t) = \mathbf{b_m^T x}(t), \tag{4.32}$$

i.e. the inner product between the basis vector and the input is taken. As in the case of time independent data representation, $s_m(t)$ is the firing rate and $\mathbf{b_m}$ is

a vector of synaptic weights. Neural encoding for the algorithm of Section 2.3.2 works also by means of the calculation of the inner product with $h_m(t-\tau)$. It differs however in two points. First, the inner product $y_m(\tau)$ is calculated between $h_m(t-\tau)$ and the input $x(t)$ from which the actual estimate $\hat{x}(t)$ has been subtracted out and not the total input $x(t)$. Second, the coefficient s_m^n is obtained after a winner-take-all operation on the $y_m(\tau)$, i.e.

$$s_m^n = y_{m^*}(\tau^*), \quad \text{where} \tag{4.33}$$

$$(m^*, \tau^*) = \text{argmax}_{m,\tau}|y_m(\tau)|. \tag{4.34}$$

Index $m = m^*$, and index n indicates how often $h_m(t)$ was the winning element. The winning time τ^* becomes the τ_m^n in the reconstruction. It indicates the time of the activity of the neuron. The coefficients s_m^n model the activity of the neuron. They are in $\mathbb{R}$, i.e they have graded values. According to the discussion of Section 3.1.1, graded signals do not well correspond to action potentials. Hence the s_m^n should be interpreted as firing rate around the time τ_m^n. The coefficient s_m^n could model the intraburst frequency of a burst happening at time τ_m^n. Nevertheless, in [Smith and Lewicki, 2005, 2006], they are said to be "analogue spikes", and τ_m^n is referred to as spike timing. Note that this kind of encoding is acausal and does not correspond to any of the dynamical neural systems which we have introduced in Section 3.1.3.

Additionally to the minimization of the reconstruction error, which corresponds to the coverage principal of neural integration, the algorithm of Section 2.3.3 includes the principles of sparseness and smoothness of the response over space and time. Spatial smoothness was another principle of neural integration, see Section 3.2.2. There, we discussed that it relates to minimal wiring, which is important for energy-efficient signaling. Maximization of sparseness is in Section 2.3.3 equivalent to the maximization of independence of the output firing rates. In the previous paragraph, where we have discussed the neural interpretation of time independent data representation, maximization of independence was in turn related to the punishment of high output firing rates. Hence, sparseness and smoothness contributes to energy efficiency.

The learning rules for the algorithms of Section 2.3.2 and Section 2.3.3 show Hebbian mechanisms. Equation (2.85) for the update of $h_m(t)$ was

$$\triangle h_m(t) = \mu \left[\sum_{n=1}^{N_m} s_m^n (x(t-\tau_m^n) - \hat{x}(t-\tau_m^n)) \right] \quad m = 1 \ldots M, \tag{4.35}$$

which is the summed product between the neural activity and the reconstruction error, as for the subspace learning algorithm for time independent data representation in the previous paragraph. Again, the product $s_m^n x(t-\tau_m^n)$ is the basic Hebbian plasticity term. The update rule for the $\mathbf{b_m}$ was given with Equation (2.95) and its simplification in Equation (2.98) by

$$\triangle \mathbf{b_m} = \mu \left\langle \sum_t \left(\sum_{i=1}^{N} g(y_i)\gamma(m,i) \right) s_m(t)\mathbf{x}(t) \right\rangle, \tag{4.36}$$

followed by orthonormalization. It shows also the Hebbian product term of neural activity times input $s_m(t)\mathbf{x}(t)$, weighted by a term that reflects the differential activity of neuron m and those of its neighborhood.

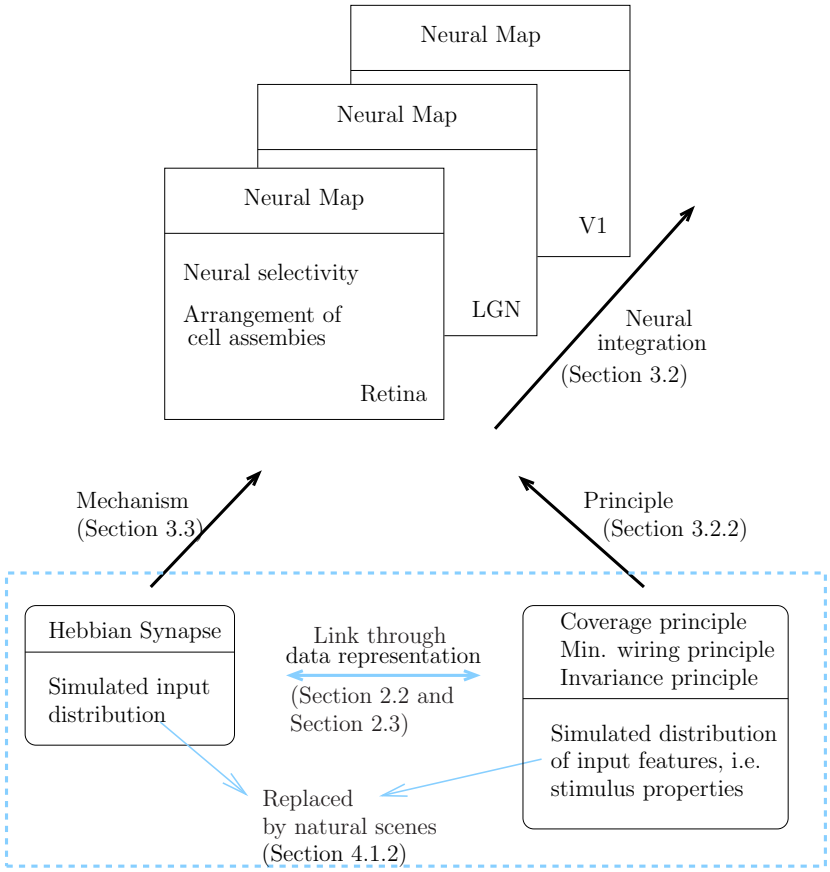


Figure 4.5: Data representation methods allow to connect the principles with the mechanisms of neural integration (depicted diagram shows the case of vision). In Section 3.2.2 and Section 3.3, we have reviewed principles and mechanisms, respectively, which can well explain neural maps. The principles, however, are abstract and there is no link to the mechanisms. Neural maps are connected to neural integration as discussed in Section 3.2. The idea of modeling neural integration by means of data representation, see Section 4.1.1, allows to deduce the mechanisms from the principles. Data representation has been reviewed in Section 2.2 for a representation without time structure and in Section 2.3 for a representation with time structure, respectively. Data representation methods use natural scenes as input data, see Section 4.1.2.

4.2.2 Neural integration and the representation of natural stimuli

Here, we show representations of natural stimuli which are obtained by application of the data representation methods of Section 2.2 and Section 2.3. The change of selectivity which is obtained by the data representation methods matches the change of selectivity which happens in the process of neural integration. The arrangement of the simulated neurons with respect to each other matches further the arrangement of cell assemblies in the cortex, i.e. neural maps are reproduced by application of data representation methods to natural stimuli. First, we discuss the application of data representation methods without time structure, then we deal with methods with time structure. We organize the algorithms with respect to their cost functional, interpreted in neuroscientific terms as in Section 4.2.1.

Data representation without time structure

Representation accuracy and decorrelation In Section 2.2.2, we have reviewed the Generalized Hebbian Algorithm (GHA) of [Sanger, 1989], see Equations (2.34) and (2.37), which learns a representation based on the minimization of the reconstruction error and decorrelation of the output. This algorithm has been applied to natural snapshots in [Hancock et al., 1992; Heidemann, 2006]. Figure 4.6 shows the learned filters $\mathbf{w_m}$. Their properties do however not correspond to known physiological properties. They have been interpreted as a basis for steerable oriented bandpass filters, see [Perona, 1995]. Sensory output decorrelation was postulated by [Barlow, 1961], and put into concrete form by [Atick and Redlich, 1992; van Hateren, 1992c]. The latter work is a theoretical analysis based on properties of the powerspectrum of natural snapshots, see Section 4.1.2, and includes a trade-off between noise suppression and whitening. Additionally to the lack of physiological findings which correspond to properties of the obtained filters, direct test of the decorrelation hypothesis showed that the postulate does not seem to hold [Puchalla et al., 2005; Segev et al., 2006]. Other experimental work [van Hateren, 1992a,b] did not test the hypothesis as directly since they could not record from a population of cells, but only from a single one. In contrast to spatial decorrelation, the idea of temporal decorrelation is supported by experimental evidence from the LGN [Dan et al., 1996; Dong and Atick, 1995].

Representation accuracy, punishment of synaptic strength, and synaptic convergence Application of the algorithm of [Vincent et al., 2005; Vincent and Baddeley, 2003], which we have reviewed in Section 2.2.3 (Equations (2.44)

and (2.46)), to natural snapshots leads to filters with strong similarities to receptive fields of X and Y ganglion cells of the cat (see Table 3.7). The receptive fields are shown in Figure 4.6. Learning of the representation under different noise conditions showed also that the filters changed their characteristics from bandpass to low pass, which has also been observed in experiments. The work samples the snapshots in a space variant manner which models to arrangement of the photoreceptors on the retina (see Figure 3.13). This allowed them to investigate also the changes of the receptive field properties as one moves towards the periphery.

Representation accuracy, punishment of output firing rate, and synaptic divergence Application of the algorithm of [Vincent et al., 2005], which we have reviewed in Section 2.2.3 (Equation (2.48)), to natural snapshots together with synaptic divergence, i.e. more outputs than inputs, leads to receptive fields which show properties of simple cells in V1. Simple cells are cells which can be modeled by linear, oriented, localized bandpass filters. The filters which emerge from [Vincent et al., 2005] show these properties. Again, the changes in the receptive field properties as one moves towards the periphery could be accounted for by the algorithm of [Vincent et al., 2005].

Representation accuracy and statistical independence (sparseness) of the outputs The algorithm of [Hyvärinen, 1999], which we have reviewed in Section 2.2.3 (Equation (2.70)), was applied to natural snapshots and movies [van Hateren and van der Schaaf, 1998; van Hateren and Ruderman, 1998], see Figure 4.6. The data representation algorithm does not model the time structure of the input. Therefore, the time frames were stacked onto each other to yield the time independent data vector $\mathbf{x}$. Learning the representation of the snapshots yielded filters that match well the spatial frequency bandwidth, orientation tuning bandwidth, aspect ratio and length of the receptive fields of simple cells in the monkey (see [van Hateren and Ruderman, 1998] for the exact definitions). For snapshot input, the distribution of the peak of the spatial frequency response does not match well the experimental findings. Learning the representation of movies alleviates this [van Hateren and van der Schaaf, 1998].

Data representation with time structure

Representation accuracy with matching pursuit encoding The algorithm of [Mallat and Zhang, 1993; Smith and Lewicki, 2005], which we have reviewed

in Section 2.3.2, has been applied to natural sounds [Smith and Lewicki, 2006]. The learned $h_m(t)$ show similarities to cochlear filters and match the temporal bandwidth-center frequency distribution of physiological data.

Representation accuracy, sparseness, temporal and spatial smoothness The algorithm of [Hyvärinen et al., 2003], which we have reviewed in Section 2.3.3, has been applied to natural movies. The results include the previously seen match of the properties of the learned filters to properties of simple cells of V1. Sparseness together with spatial smoothness leads also the emergence of complex cell properties and neural maps [Hyvärinen and Hoyer, 2001], see Figure 4.6 and Figure 3.19. The addition of temporal smoothness allowed for learning of motion selective filters and the simulated neurons are arranged with respect to their preferred direction of motion, i.e. they form a neural map.

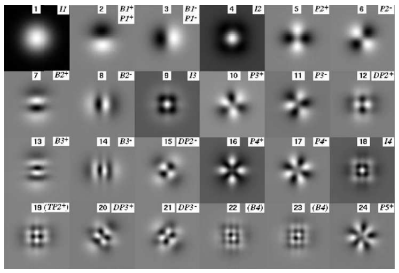
4.3 Research questions

4.3.1 Directions to learn more about neural integration

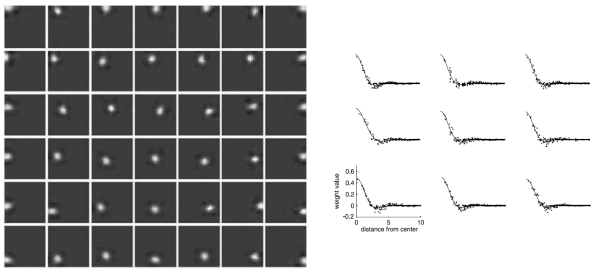
Table 4.1 provides a summary of the insights into neural integration which were obtained through the use of data representation methods. The table shows that the decoding scheme, which reflects the invariance principle, and the maximization of the *representation accuracy* (minimal reconstruction error, coverage) are important ingredients. It shows also that the algorithms which include a form of *energy minimization* (case 3 to 7 in Table 4.1) lead to representations with connection to physiology. Temporal *smoothness and sparseness*, which are at the origin of bubble coding (case 7), can also be interpreted into the decomposition which is obtained by matching pursuit (case 6). Time τ_m^n indicates the time where the activity bubble occurs. The value of s_m^n indicates its strength. The activity bubbles could be thought to model a burst code. Representation of natural scenes by activity bubbles leads to the emergence of *neural maps*.

Direction A

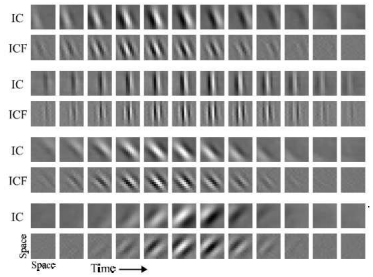
Cells in the primary visual cortex show clear tuning towards orientation, spatial frequency, and location. If this tuning property is considered to lead to a fragmentation of the input, the question arises how the spatial frequency channels, the orientation channels, and the different locations are recombined. This kind of research question has been addressed in [Gutmann et al., 2005; Hyvärinen et al.,



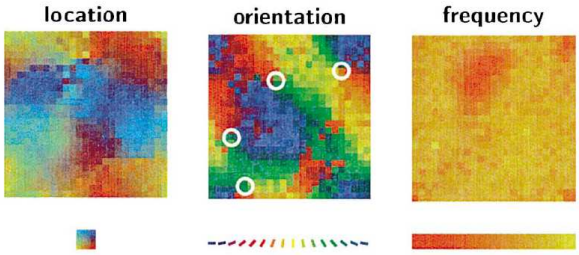
Filters from maximization of representation accuracy and decorrelation



Filters from maximization of representation accuracy punishment of synaptic strength, and synaptic convergence



Filters from maximization of representation accuracy and statistical independence



Neural maps via bubble coding

Figure 4.6: Application of data representation methods to natural scenes. The principle of representation accuracy (minimal reconstruction error) and decorrelation does not lead to filters with known physiological properties. Representation accuracy together with punishment of synaptic strength, which limits the postsynaptic ion flux, leads to filters which match well the difference of Gaussian model for retinal Ganglion cells. Maximization of representation accuracy and statistical independence (sparseness) of the outputs leads to filters which match well the properties of simple cells in cortex. Sparseness is related to the punishment of the output firing rate. Methods which maximize representation accuracy and punish the output firing rate yield also such filters. The inclusion of spatial smoothness, which can be extended to temporal smoothness, leads to filters with not only similar neural selectivities as in V1 but also similar spatial arrangement of the simulated neurons with respect to each other (neural map).

2005; Hyvärinen et al., 2005] and [Nuding and Zetzsche, 2007]. This work extends the list of Table 4.1. The interest is that it allows to learn properties of cells beyond V1 which remain far less well understood than V1: It allows thus to make theory-driven predictions, see [Felsen and Dan, 2005].

Direction B

In Section 3.1.3, we have reviewed a range of neuron models. In Table 4.1, firing-rate based models, with linear activation function, are mostly used for encoding. Matching pursuit based encoding does not correspond to any presented (or known) neuron model. The Hebbian plasticity rules, which emerge from the data representation methods, are thus also firing-rate based. In Section 3.3, we have however seen that spike-timing dependent plasticity is a richer model for synaptic plasticity.

An alternative direction of research would be to use for the encoding part in the data representation scheme the neuron models of Section 3.1.3 which are obtained by simplification of the minimal spiking neuron model. In other words, the activity transform would happen by means of dynamical neural systems. This would allow to obtain data representation methods that are tailored to neuroscience.

The simplified models were characterized along the axes resonator-integrator and monostability-bistability (see Table 3.4). It would be interesting to investigate how the use of the different simplified models in the encoding affect the data representation, and hence the neural integration. The simplified models use spike timings as signaling variable. As data representation with firing-rate based neuron models led to the emergence of firing-rate dependent Hebbian plasticity, it would be worthy to see whether spike timing dependent plasticity emerges for data representation with simplified minimal models. In spike timing based plasticity, recurrent connections can be plastic (Section 3.3.2). An example of a recurrent connection was seen in Section 3.2.1 at the bipolar-amacrine-ganglion cell synapse in the retina. Data representation with simplified minimal models would allow to investigate such configurations and other neural microcircuits from the perspective of neural integration.

4.3.2 Formulation of three research questions

This thesis aims at finding learning rules for data representation where the encoding is performed by a dynamical neural system. We have a system in mind which is composed of the simplified spiking neuron models of Section 3.1.3.

Figure 4.7 relates this aim to previous work and shows the basic approach: Input data x is transformed by a dynamical neural system into spike timings. This encoding process should be such that the input can be obtained by decoding of the output. The energy consumption during the encoding process should also be minimized. This defines an optimization problem. The encoder and decoder are learned by iterative minimization. Their joint learning provides representation learning.

In this thesis, we treat the following three research questions:

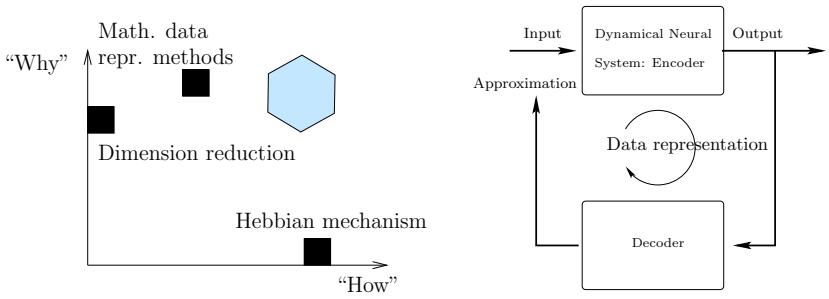
Question 1 The encoding is done by a single neuron that is modeled by the spike response model of Section 3.1.3, due to [Gerstner and Kistler, 2002]. Find learning rules for the decoder and the free parameters of the encoder.

Question 2 Extend the single neuron case to multiple neurons. Each neuron is modeled by the spike response model. The extension contains two steps. As a first step, the encoding is done by multiple neurons, but the neurons receive no lateral inputs. The second step consists then in the inclusion of lateral connections. Find learning rules for the decoder, and the free parameters of the encoder, i.e. for the feedforward path from the input to the neurons and the lateral connections between the neurons.

Question 3 Like the first question, but the encoding is done by a single canonical integrator neuron (quadratic integrate and fire neuron, reviewed in [Izhikevich, 2006]). The main difference between the spike response model and the canonical integrator model is that in the latter model, the upstroke of the action potential is also simulated (see Section 3.1.3).

Data representation method		Neural interpretation	Link to principles/mechanisms of neural integration				Application to
Cost	Constraint		Coverage	Smoothness	Invariance	Hebb	natural stimuli
1.$E(\|\|\mathbf{x}-\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{x}\|\|^2)$ $\mathbf{W}: N\times M, M\leq N$		Linear-firing rate based encoding Encoding such that accurate decoding is possible	√	–	√	√	
2.$E(\|\|\mathbf{x}-\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{x}\|\|^2)$ $\mathbf{W}: N\times M, M\leq N$	Decorrelation	1. and "decorrelation hypothesis"	√	–	√	√	Basis for steerable oriented bandpass filters.
3.$E(\|\|\mathbf{x}-\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{x}\|\|^2)$ $\mathbf{W}: N\times M, M\leq N$	$\sum\|w_{nm}\|$	1. and punishment of synaptic strength, convergence	√	–	√	√ (nonlinear)	⇝ Retinal ganglion cells
4.$E(\|\|\mathbf{x}-\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{x}\|\|^2)$ $\mathbf{W}: N\times M, M\geq N$	$\sum\|\mathbf{w_m^T x}\|$	1. and punishment of output firing rate, divergence	√	–	√	√ (nonlinear)	⇝ V1 cells
5.$E(\|\|\mathbf{x}-\mathbf{H}\mathbf{W}^{\mathbf{T}}\mathbf{x}\|\|^2)$ $E(G(\mathbf{w_m^T x}))$		1. and maximal statistical independence (sparseness)	√	–	√	√ (nonlinear)	⇝ V1 cells
6.$E(\sum(x(t)-\hat{x}(t))^2)$, $\hat{x}(t)=\sum_{m=1}^{M}\sum_{n=1}^{N_m}$ $s_m^n h_m(t-\tau_m^n)$		Winner take all style acausal encoding, τ_m^n: time of neural activity, s_m^n: "analogue spike".	√	–	√	√	⇝ cochlear filters
7. $E(\sum\|\|\mathbf{x}(t)-\hat{\mathbf{x}}(t)\|\|^2)$ $E(\sum\sum G(y_m(t)))$, $\hat{\mathbf{x}}(t)=\sum_{m=1}^{N}[\mathbf{b_m^T x}(t)]\mathbf{b_m}$	Orthonormal filters	1. and max. sparseness, spatio-temporal smoothness, Instantaneous encoding	√	√	√	√ (nonlinear)	⇝ neural maps of V1

Table 4.1: Overview of data representation methods used in previous work. The upper part of the table shows data representation methods without time structure and the lower part data representation methods with time structure, respectively. *1. Algorithm:* see Equation (2.27) *2.* Equations (2.34) and (2.37) *3.* Equations (2.44) and (2.46) *4.* Equation (2.48) *5.* Equations (2.62) and (2.70) *6.* Equation (2.85) *7.* Equation (2.98). Punishment of synaptic strength means consideration of energy costs that are related to postsynaptic ion flux. Convergence means that there are less output neurons than input neurons. Note that a neuron at rest consumes energy. Convergence means less fixed energy costs. Vice versa for divergence. Punishment of output firing rate means consideration of energy costs related to the action potential. See Section 3.1.2 for a discussion of neural energy consumption. See Section 4.2.1 for a neural interpretation of the above data representation methods. See Sections 3.2.2 and 3.3 for principles and mechanisms of neural integration. Section 4.2.2 deals with the representation of natural stimuli.



(a) Goal (b) Approach

Figure 4.7: (a) Summary of previous work and proposed research direction. As discussed in the text body, previous work accounts much for the "why" and "how", i.e. principles and mechanisms, of neural integration. The use of data representation methods helped to connect them. The shaded area depicts the goal of research direction B of Section 4.3.1. (b) The central idea is to use (standard) dynamical neural systems for the encoding part of the data representation loop. The representation of the input is learned by maximization of approximation accuracy and minimization of energy consumption during encoding.

Chapter 5

Data Representation with a Single Integrator Neuron

5.1 Cost functional for data representation

In Section 5.1.1, we recapitulate the first research question of Section 4.3.2. In Section 5.1.2, we obtain part one of the cost functional which measures the accuracy of the representation. In Section 5.1.3, we present part two of the cost functional which measures the energy consumption during the encoding process.

5.1.1 Recapitulation of the first research question

The goal of this chapter is to learn a representation of simple sensory input where the encoding happens with the spike response model, see Section 3.1.3 and [Gerstner and Kistler, 2002]. Figure 5.1 visualizes the research task.

Encoding of the input $x(t)$ is done by firing single spikes. The spike timings t^f constitute the output signal of the encoder. For a proper representation of the input by means of the spike timings, the input should be reconstructible from the neural activity. Learning the representation is based on the minimization of the reconstruction error, with an added penalty to minimize energy consumption during the encoding process. The research task is to formulate the optimization problem and to find and test an iterative optimization rule.

5.1.2 Part one of the cost functional: reconstruction error

The assumed neuron model is closely related to the SMR_0 model [Gerstner and Kistler, 2002]. The equation for the membrane voltage u is

$$u(t) = \eta_0 \exp\left[-\frac{t-\hat{t}}{\tau}\right] + \int_0^{\min\{t,T_w\}} x(t-s)w(s)\mathrm{d}s + I_n(t), \tag{5.1}$$

where $I_n(t)$ is a sufficiently smooth noise current, $\hat{t}$ the last spike time before time t, and w the unknown encoding filter, to be learned, of length T_w. The convolution of input x with w produces the input current I. Spike timings $\{t^f; f = 1, \ldots\}$ are defined by $u(t^f) = \theta$, where $\theta > 0$ is a fixed threshold. The remaining constants are the recovery time constant τ of the recovery current I_r and the reset amount $\eta_0 < 0$.

The reconstruction $\hat{x}$ is sought under the form

$$\hat{x}(t) = \sum_{f: t-T_p < t^f < t+T_d} h(t - t^f), \tag{5.2}$$

which introduces a delay T_d in the estimate. For the reconstruction at time t, spikes happening prior to $t - T_p$ are not considered. The decoding filter h is

unknown and to be learned. The arguments for h are in the range $[-T_d \quad T_p]$. For a good decoder $h(-T_d) = h(T_p) = 0$ should hold. The role of $h(s)$ is different for $s > 0$ and $s < 0$. For $s > 0$, the input at t is predicted from a spike event at $t^f < t$. On the other hand, for $s < 0$, the input is reconstructed from a later spike event at $t^f > t$.

The first part of the cost functional which is due to the reconstruction error is

$$J_e(h, w) = \frac{1}{2T} \int_0^T (\hat{x}(t) - x(t))^2 \mathrm{d}t, \tag{5.3}$$

where T is a fixed time horizon.

The neuron model in Equation (5.1) has been related to detailed biophysical quantities [Gerstner and Kistler, 2002]. The encoding filter w can be considered to model a physical time-invariant system. The decoding filter h, however, is of more abstract nature. It is a hypothetical quantity and assigns meaning to each spike. It implements the homunculus, which generates a running commentary on the spike train. The idea of the homunculus has been advocated in e.g. [Rieke et al., 1997]. Biophysically, it could be implemented in various ways, but we want to leave that question open.

5.1.3 Part two of the cost functional: energy consumption

We introduce two ways to measure the energy consumption, from where we deduce the cost functionals that we use in the optimization problem. First, we use the average power P_a,

$$P_a = \frac{1}{T} \int_0^T I(t)^2 \mathrm{d}t. \tag{5.4}$$

In signal processing, this is the average power of the signal $I(t)$ [Reinhard, 1997]. It is the electrical power that is consumed in a unit resistance through which the current $I(t)$ flows. A second measure for energy consumption stems from the idea that $I(t)\mathrm{d}t$ is proportional to the ion load that must be pumped out, or in, to restore the ion gradients in the neuron (Section 3.1.2). The total postsynaptic ion load per time T is

$$P_p = \frac{1}{T} \int_0^T |I(t)| \mathrm{d}t. \tag{5.5}$$

The output $I(t)$ of the convolution between x and w satisfies $I(t)^2 \leq ||w||_2^2 ||x||_2^2$, see e.g. [Frazier, 2001]. Hence, $P_a \leq ||w||_2^2 ||x||_2^2$, and in order to minimize the average power consumption P_a during the encoding, we seek to minimize additionally

to J_e

$$J_2(w) = \int_0^{T_w} w(t)^2 \mathrm{d}t. \tag{5.6}$$

Alternatively, we can use the relation $P_a \leq ||w||_1^2 ||x||_2^2/T$, see also e.g. [Frazier, 2001], where the squared L_1 norm of w indicates the amplification gain, i.e. the ratio between output and input power. Hence, in order to punish amplification, we could minimize additionally to J_e

$$J_{1s}(w) = \left(\int_0^{T_w} |w(t)| \mathrm{d}t \right)^2. \tag{5.7}$$

For the second measure of energy consumption P_p, we have due to properties of the convolution, see e.g. [Frazier, 2001], $P_p \leq ||w||_1 ||x||_1/T$. Hence, in order to punish postsynaptic ion load, we minimize additionally to J_e

$$J_1(w) = \int_0^{T_w} |w(t)| \mathrm{d}t. \tag{5.8}$$

Alternatively, we could also take P_p directly as additional quantity to be minimized, i.e.

$$J_p(w) = \frac{1}{T} \int_0^{T} |I(t)| \mathrm{d}t. \tag{5.9}$$

In the following, we denote the energy cost by J_E , which can be J_2, J_{1s}, J_1, or J_p.

5.2 Learning rule

The online rule for encoding filter w is based on a steepest descent method on the total cost J, which is the sum of the reconstruction error J_e, defined in Equation (5.3), and the energy cost J_E, defined in Equations (5.6) to (5.9). The update rule for decoding filter h is based on a least squares algorithm (see e.g. [Haykin, 2001]).

5.2.1 Encoding filter w

Key to the update rule for w is the calculation of the functional derivative $\delta J/\delta w(s)$. In the Appendix Section 5.5, we calculate it for J_e, see [Gutmann and Aihara, 2008], and the energy cost J_E. The decoding filter h is held fixed. The functional derivative of J_e is

$$\frac{\delta J_e}{\delta w(s)} = -\frac{1}{T} \sum_f \bar{e}(t^f) y_f(s), \tag{5.10}$$

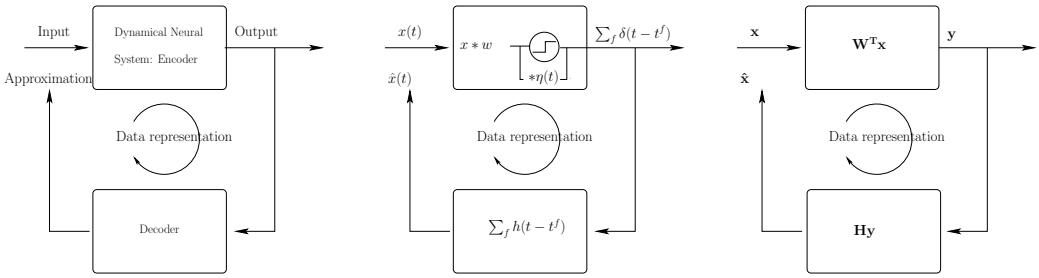


Figure 5.1: (a) Input data x is transformed by a dynamical neural system. This encoding process should be such that the input can be decoded from the output. The energy consumption during the encoding process should also be minimized. The encoder and decoder together provide data representation. (b) Here, we encode by means of the spike response neuron model (see Section 3.1.3 and [Gerstner and Kistler, 2002]), and decoding is done from the spike timings t^f. Learning rules for the unknown encoding filter w and decoding filter h are derived from the minimization of the mean squared reconstruction error, with an added penalty to minimize energy consumption during the encoding process. (c) In PCA and ICA, the decoder and encoder are unknown linear transforms. The transform is found by minimization of the the expected value of $||\mathbf{x} - \hat{\mathbf{x}}||_2$, and for ICA, by additional maximization of the statistical independence of the components of the output $\mathbf{y}$, see Section 2.2.

where

$$\bar{e}(t^f) = \int_{t^f-T_d}^{t^f+Tp} (\hat{x}(t) - x(t))\dot{h}(t-t^f)\mathrm{d}t, \tag{5.11}$$

$$y_f(s) = -x(t^f - s)\frac{1}{\dot{u}(t^f)} + \Gamma(t^f, t^{f-1})y_{f-1}(s), \tag{5.12}$$

$$\Gamma(t^f, t^{f-1}) = \frac{-\eta_0}{\tau\dot{u}(t^f)} \exp\left[-\frac{t^f - t^{f-1}}{\tau}\right]. \tag{5.13}$$

The functional derivatives of J_E are, depending on the measurement of the energy consumption,

$$\frac{\delta J_E}{\delta w(s)} = 2w(s) \qquad \text{for} \qquad J_E = J_2 \tag{5.14}$$

$$\frac{\delta J_E}{\delta w(s)} = 2||w||_1 \mathrm{sign}(w(s)) \qquad \text{for} \qquad J_E = J_{1s} \tag{5.15}$$

$$\frac{\delta J_E}{\delta w(s)} = \mathrm{sign}(w(s)) \qquad \text{for} \qquad J_E = J_1 \tag{5.16}$$

$$\frac{\delta J_E}{\delta w(s)} = \frac{1}{T}\int_0^T \mathrm{sign}(I(t))x(t-s)\mathrm{d}t \qquad \text{for} \qquad J_E = J_p. \tag{5.17}$$

We propose the following online rule: After spike k at t^k, update the encoder by

$$w_k(s) = w_{k-1}(s) + \mu\left(\bar{e}(t^k)y_k(s) - \alpha\frac{\delta J_E}{\delta w(s)}\right), \tag{5.18}$$

where $\mu > 0$ is the step size and α weights the influence of the energy constraint. The algorithm is initialized with $y_0 = 0$ and e.g. $w_0 = 0$. If $J_E = J_p$, we integrate in the update rule not from zero till T but from t^{k-1} till t^k.

5.2.2 Decoding filter h

For the learning of h, we form from the spike timings a binary $\rho(n)$ by binning the spike timings into containers of size $\triangle t$. If there is a spike in the bin centered at $t = n\triangle t$, then $\rho(n)$ equals one, otherwise zero. We discretize $h(s)$ with the same bin size. The reconstruction in Equation (5.2) at $t = n\triangle t$ becomes then

$$\hat{x}(n) = \mathbf{h}\boldsymbol{\rho}(n), \tag{5.19}$$

where

$$\mathbf{h} = [h(-N_d)\dots h(N_p)] \tag{5.20}$$

$$\boldsymbol{\rho}(n) = [\rho(n+N_d)\dots\rho(n-N_p)]^T. \tag{5.21}$$

We may further search for $h(n)$ in the form of

$$h(n) = \sum_k c_k \Psi_k(n), \tag{5.22}$$

for some given Ψ_k and unknown c_k. The Ψ_k can for example be the Daubechies's D6 wavelet basis on Z (see e.g. [Frazier, 2001]). Omitting wavelets located in the highest frequency bands in that representation allows for a reduction in the parameters to be learned for the decoder h. With reference to [Frazier, 2001], we are looking in that case for $h(n)$ in V_{-j}, with e.g. $j = 2$.

Denoting by $\mathbf{\Psi}$ the matrix with row k given by $[\Psi_k(-N_d) \dots \Psi_k(N_p)]$, we obtain for the reconstruction

$$\begin{aligned} \hat{x}(n) &= \mathbf{c}\mathbf{\Psi}\boldsymbol{\rho}(n), & (5.23)\\ &= \mathbf{c}\mathbf{y}(n), & (5.24) \end{aligned}$$

where $\mathbf{y} = \mathbf{\Psi}\rho$. The row vector $\mathbf{c}$ is to be determined such that J_e is minimized. Calculation of the derivative of J_e with respect to the row vector $\mathbf{c}$, i.e. after discretization, gives

$$\frac{\delta J_e}{\delta \mathbf{c}} = \frac{1}{N} \sum_{n=1}^{N} (\hat{x}(n) - x(n)) \mathbf{y}^{\mathbf{T}}(n). \tag{5.25}$$

Using the stochastic gradient, the following least mean square (LMS) like learning rule is obtained

$$\triangle \mathbf{c}(n) = -\alpha_h \left[\hat{x}(n) - x(n)\right] \mathbf{y}^{\mathbf{T}}(n), \tag{5.26}$$

where α_h is the step size. The step size can be chosen optimally in each update step by using a recursive least squares (RLS) algorithm for the learning of $\mathbf{c}$ (see e.g. [Haykin, 2001] for a discussion of the LMS and RLS algorithm).

5.2.3 Interpretation

We discuss here key elements in the update rule (5.18) for the encoder w: $\bar{e}$ in Equation (5.11), and Γ in Equation (5.12). Then we discuss the role of noise, and the influence of w on the learning of h.

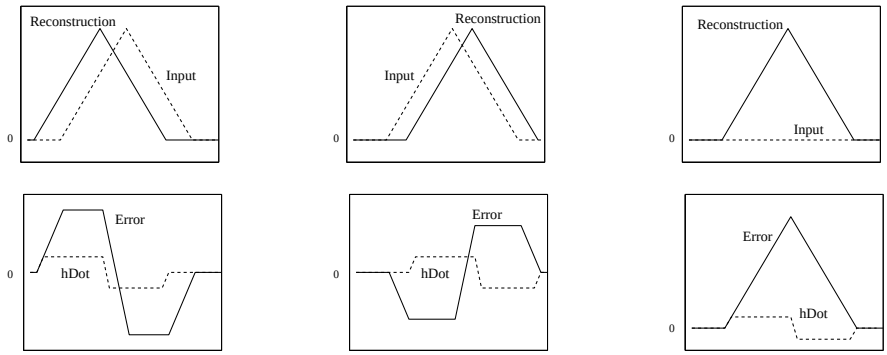
Role of $\bar{e}$ Let us assume that the input x is positive valued. The parameter Γ in Equation (5.12) is also positive so that $y_f(s) < 0$. The quantity $\bar{e}$ can be positive or negative. In the regime where the cost associated with the norm of w does not matter, it is the sign of $\bar{e}$ which decides whether w increases ($\bar{e} < 0$) or decreases

($\bar{e} > 0$). Figures 5.2a and 5.2b illustrate that for a perfect h, $\bar{e} > 0$ corresponds to the case where the estimate $\hat{x}$ leads the input x because the spike is triggered too early, while $\bar{e} < 0$ happens in case $\hat{x}$ lags behind. Figure 5.2c shows the case of a noise triggered spike. Here, $\bar{e} \approx 0$ so that the input x is in the update to a large extent not considered. In that sense is $\bar{e}$ an indicator for the reliability of the spike.

Role of Γ and $\dot{u}$ Above, it has been pointed out that $y_f(s)$ is negative and Γ is positive. If the two spike timings t^k and t^{k-1} are close together then is $\Gamma(t^k, t^{k-1})$ not negligible. The term y_k is thus more negative than for negligible values of Γ. If two input features are closely spaced together, it is for example likely that the refractory current I_r causes the second spike to be triggered late. The estimate for the second input event will lag behind the true input so that $\bar{e} < 0$ (Figure 5.2b). The large y_k causes then a more significant increase in w than usually. In Equation (5.13), the term $1/\dot{u}(t^k)$ appears in the denominator of Γ. Hence both terms in Equation (5.12) are weighted by that factor, and it can be interpreted to lead to a variable step size.

Role of noise If w is initialized as zero, the noise current I_n is essential in the beginning of the learning. It drives the neuron and triggers the spikes. Later, when w is large enough, spikes will be mostly triggered by the input current I and the resulting recovery current I_r essentially suppresses the influence of the noise current I_n: The neuron is driven by the input current I. The noise current is optional in the modeling. It could be omitted when the algorithm is initialized with $w_0 \neq 0$.

Influence of encoder w on decoder h We explained above how the decoder h influences over $\bar{e}$ the learning of the encoder w. The encoder exerts also influence on the learning of h. Since the reconstruction is sought in the form of Equation (5.2), the spiking timings produced by w select the input data for the update of h. This can be seen in Equation (5.26) where the spike timings define the vector $\mathbf{y}$. Noise triggered spikes provide thus bad learning data for h while spikes which were triggered over I by a characteristic feature in the input provide good learning data.



(a) $\bar{e} > 0 : w \downarrow$ (b) $\bar{e} < 0 : w \uparrow$ (c) $\bar{e} \approx 0 : w \leftrightarrow$

Figure 5.2: The role of $\bar{e}$ in case of perfect decoder h. (a) The reconstruction $\hat{x}$ leads the input x. The resulting error $e = \hat{x} - x$ is weighted with $\dot{h}$ and integrated to yield $\bar{e}$ (Equation (5.11)). As discussed in the text body, $\bar{e} > 0$ means w decreases. The decrease causes the next spike to happen later and the reconstruction matches the input better. (b) The opposite case where the reconstruction lags behind the input (c) The spike is noise triggered: $\bar{e}$ is small so that w is not affected by x.

5.3 Simulations

We apply the online rule to an input x consisting of a randomized hat-shaped signal part and noise. In Section 5.3.1, we present the case where only the reconstruction error J_e, defined in Equation (5.3), is minimized, and the energy consumption is not punished. In Section 5.3.2, simulations are shown where the energy constraint is measured by the average power consumption P_a, defined in Equation (5.4). In Section 5.3.3, simulations are shown where the energy constraint is given by the postsynaptic ion load P_p, defined in Equation (5.5). In all cases, an encoder w and decoder h have emerged at the end of the learning process which have the property that the encoding is such that reconstruction from the spike timings yields a good approximation of x in the form of stereotyped hat-like features.

5.3.1 Without punishment of energy consumption

The online rule was initialized with $w = 0$ and $h = 0$. Only the reconstruction error J_e was minimized. Figure 5.3 shows the development of the encoding filter w and decoding filter h. From the figure, we see that after a period of strong growth the development slows down. This is further investigated in Figure 5.4. As the reconstruction becomes more accurate, $\bar{e}$ becomes smaller so that the development

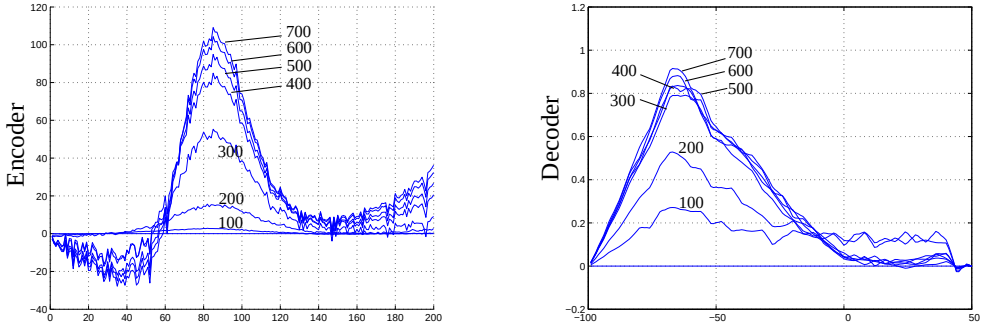


Figure 5.3: Development of encoding filter w (left) and decoding filter h (right) without punishment of energy consumption. Decoding filter h develops from an unstructured shape in round 100 to its almost final shape in round 300. For the encoder, the strongest development happens from round 200 to 400, after which the development slows down.

slows down more and more with increasing learning rounds. Figure 5.5 shows that after many learning rounds, the encoding filter w and decoding filter h do not change much any more over hundreds of further learning rounds. We then stopped the learning.

The resulting encoding filter w has a characteristic shape with a large main peak, a negative primary sidelobe and a secondary positive sidelobe. We discuss the shape in more detail in Figure 5.6.

Simulation parameters For memory reasons, we split the input x into several parts. We run several rounds where after discretization x had length $25 \cdot 1024$. Discretization and integration step size was 10^{-3} time units. Integration was done using a Simpson scheme. The noise current I_n was obtained by convolution of an i.i.d. Gaussian random process (mean 11, standard deviation 8) with an exponential kernel with time constant $\tau_m = 0.05$ time units. Recovery time constant τ was set to 0.1 time units, step size μ to 1. The decoder h was searched in V_{-2} so that 69 wavelet coefficients needed to be learned. We used the RLS based update rule. For the encoder w, 200 points were learned.

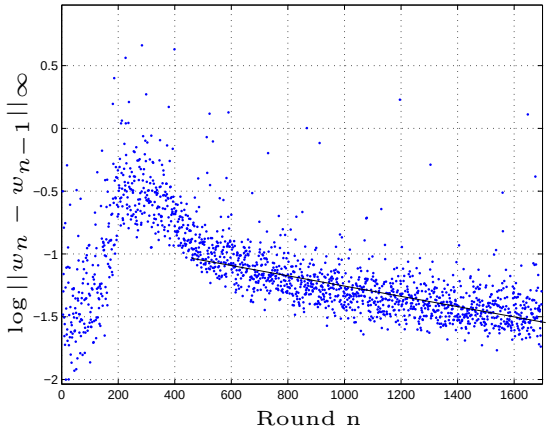


Figure 5.4: Characterization of the growth of the encoding filter (without punishment of energy consumption). After each round n, the logarithm of the sup norm of the change of the encoding filter, i.e. $\log \max_s |w_n(s) - w_{n-1}(s)|$, is calculated. The plot shows that growth in the beginning is slow. Between round 200 and 400, there is strong growth. The growth slows then down more and more with increasing learning rounds.

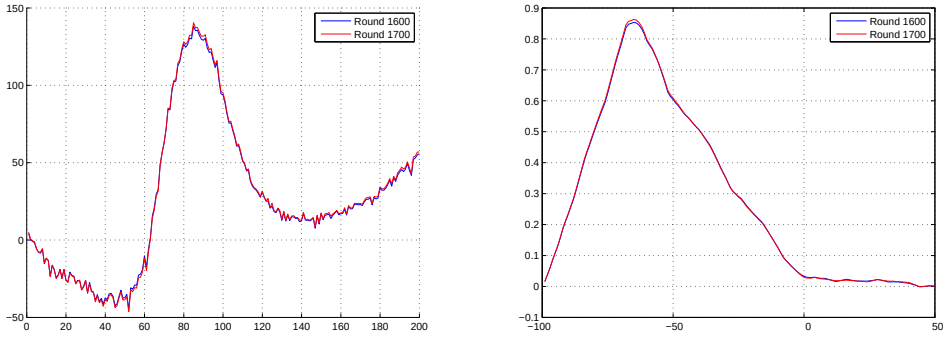


Figure 5.5: Comparison of the encoding filter w (left) and decoding filter h (right) after round 1600 with those after round 1700 (learning without punishment of energy consumption). There is minimal change. The resulting encoding filter has a large main peak, a negative primary sidelobe, and a positive secondary sidelobe. The primary sidelobe prevents noise triggered spikes and the secondary sidelobe helps to overcome the refractory current (see discussion of Figure 5.6).

5.3.2 With punishment of average power consumption

Punishment of L_2 norm, i.e. $J_E = J_2$

The online rule was initialized with $w = 0$ and $h = 0$. Figure 5.6a shows an overview of the simulation result. Figure 5.6b shows characteristic stages in the learning process of w and h. Figure 5.6c shows the associated currents and reconstructions.

In *stage 1*, w is so small that the spikes are purely noise triggered. As explained in Section 5.2.3, $\bar{e}$ is in that case small and w grows consequently slowly. Noise triggered spikes provide bad learning data for h so that the growth of the decoder goes slowly as well. In *stage 2*, the decoder h shows some structure which supports the learning of w. Encoder w is still small but the input current can have influence on the spike timings by providing, given the right amount of noise, just the amount of current needed to reach the threshold. *Stage 3* shows the situation where w has strongly developed but h is still in an early state of development. The encoder w is noisy and there is some overshoot compared to the final stage. But w is large enough to drive the neuron and to provide good training data for h. Therefore, in *stage 4*, h has reached a good decoding performance. From stage 3 on, the energy constraint on w takes effect and the encoder converges to the attractor shown as *final stage*. The final form of w reflects the trade-offs in the learning. The main peak needs be large enough to provide spikes in case of an input feature, but it cannot be too large due to the energy constraint mediated by α.

In case of closely spaced input features, the recovery current I_r makes it difficult to reach a good balance in that trade-off. The secondary lobe of the encoder w in Figure 5.6b provides at a low energy overhead additional input current I to overcome the recovery current I_r. The amplitude of w, and also h becomes smaller for values of α larger than in the present case. For $\alpha = 10^{-3}$, for example, we have $w < 10$ and $h < 0.5$. The general shape of both kernels is however related.

Simulation parameters are mostly as in the case without punishment of energy consumption. Differences are that we run in total 300 rounds where after discretization x had length $50 \cdot 1024$. The punishment of the energy consumption was weighted by $\alpha = 10^{-4}$.

Punishment of squared L_1 norm, i.e. $J_E = J_{1s}$

Figure 5.7 shows results of the same simulation when the squared L_1 norm is punished (case $J_E = J_{1s}$). The resulting encoding filter w shows a large main peak,

a negative primary sidelobe and a small secondary positive sidelobe. A characteristic feature are the clear zeros after the main peak and before the primary sidelobe.

Simulation parameters are the same as for the simulation with the punishment of J_2.

5.3.3 With punishment of postsynaptic ion load

Punishment of L_1 norm, i.e. $J_E = J_1$

In Figure 5.8, we contrast the development of the encoding filter w and the decoding filter h for different weightings of the energy cost $J_E = J_1$. Increasing $\alpha = 10^{-4}$ to $\alpha = 5 \cdot 10^{-4}$) reduces the main peak of w by one half (from a peak value of 80 in the first row to 40 in the second row). The inhibitory sidelobe becomes weaker and the secondary sidelobe vanishes. Increasing α further to 10^{-3} (third row) reduces the main peak drastically. The encoding filter w has a clear form but its amplitude is low. It can not provide clear learning data for the decoder so that h remains unstructured. For $\alpha = 5 \cdot 10^{-3}$, the encoding filter fluctuates around zero. The decoding filter h is consequently not well developed either.

The energy constraint for $\alpha = 5 \cdot 10^{-3}$ is too strong to permit further development. Figure 5.9 shows that for $\alpha = 10^{-3}$, however, further growth takes place.

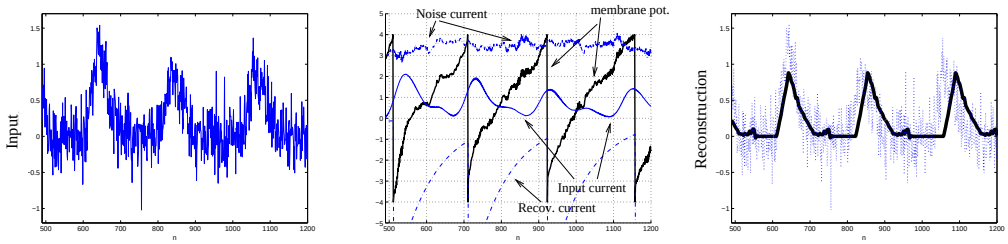
The encoding filters w after learning are shown in Figure 5.10. While the main peaks of the filters are comparable for the different values of α, the secondary lobes are clearly reduced, and the width of the main peak becomes narrower for increasing values of α, i.e. increasing weighting of $J_E = J_1$.

Simulation parameters are as in Section 5.3.2, but the length of input $\mathbf{x}$ in each round was $25 \cdot 1024$. No mathematical convergence criterion was used for Figure 5.10. If the filters were well developed and the state did not change over at least hundred rounds, we considered to have reached the attractor.

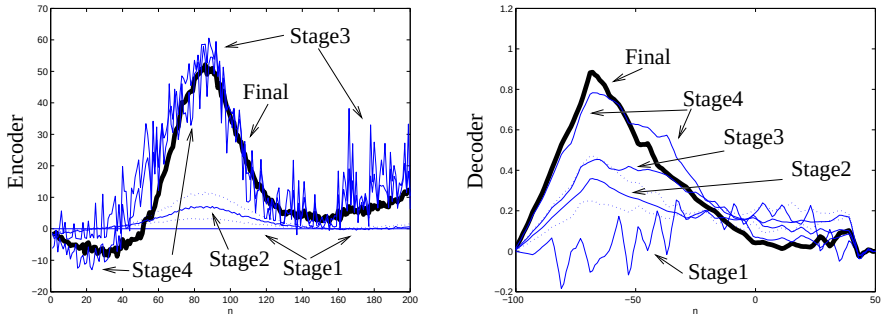
Punishment of $J_E = J_p$

Figure 5.11 shows encoding filter w and decoding filter h after learning where the energy constraint is given by $J_E = J_p$. The filters w do not differ much for different J_p-weightings α, but for larger α a secondary negative sidelobe emerges after the main peak.

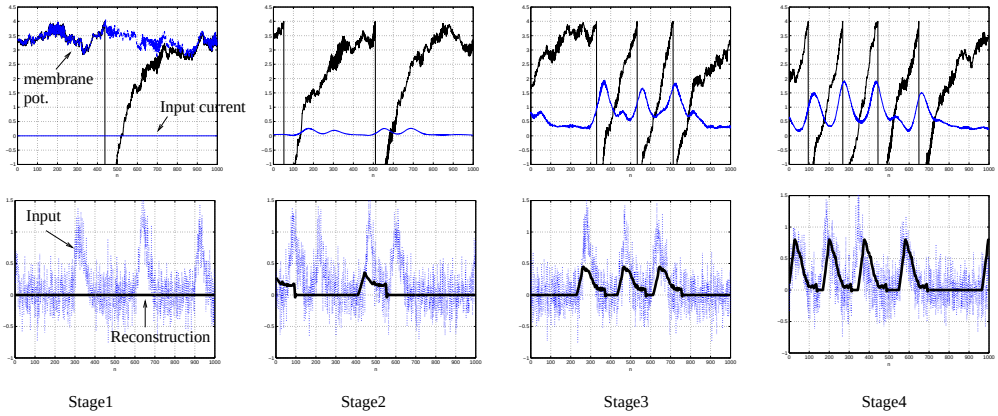
Simulation parameters are as for the above case where $J_E = J_1$. The step size



(a) Reconstruction after learning (for stage "final" below).



(b) Characteristic stages in the development of the encoder w and decoder h.



(c) Currents and reconstructions.

Figure 5.6: Feature extraction example for $J_E = J_2$. (a) The input is encoded with a formal spiking neuron such that decoding of the neural response yields the essential parts of the input: The input x is convolved with the encoder w to form the input current I. Together with the noise current I_n and the recovery current I_r, they form the membrane potential u. If u reaches the threshold $\theta = 4$ at $t = t^f$, the spike timing t^f is recorded for the reconstruction and the voltage reset to $-\theta$. The spike causes a recovery current I_r. (b) The learning process can qualitatively be separated into different stages. Learning happening during stage 1 and 2 is noise driven. Then w develops strongly providing from stage 3 on good training data for the decoder h which develops nearly to the final form (see stage 4 curve). The difference between w shown as stage 3 and 4 is that a negative primary sidelobe develops and the secondary sidelobe is reduced. The encoder attains then a smooth form (final stage). (c) In stage 1, the input current I due to w and the decoder h are close to zero. In stage 2, I becomes visible and in stage 3, it drives the neuron. The role of the negative primary sidelobe of w is to prevent too early spiking. The secondary sidelobe helps to overcome the refractory current for closely space input features.

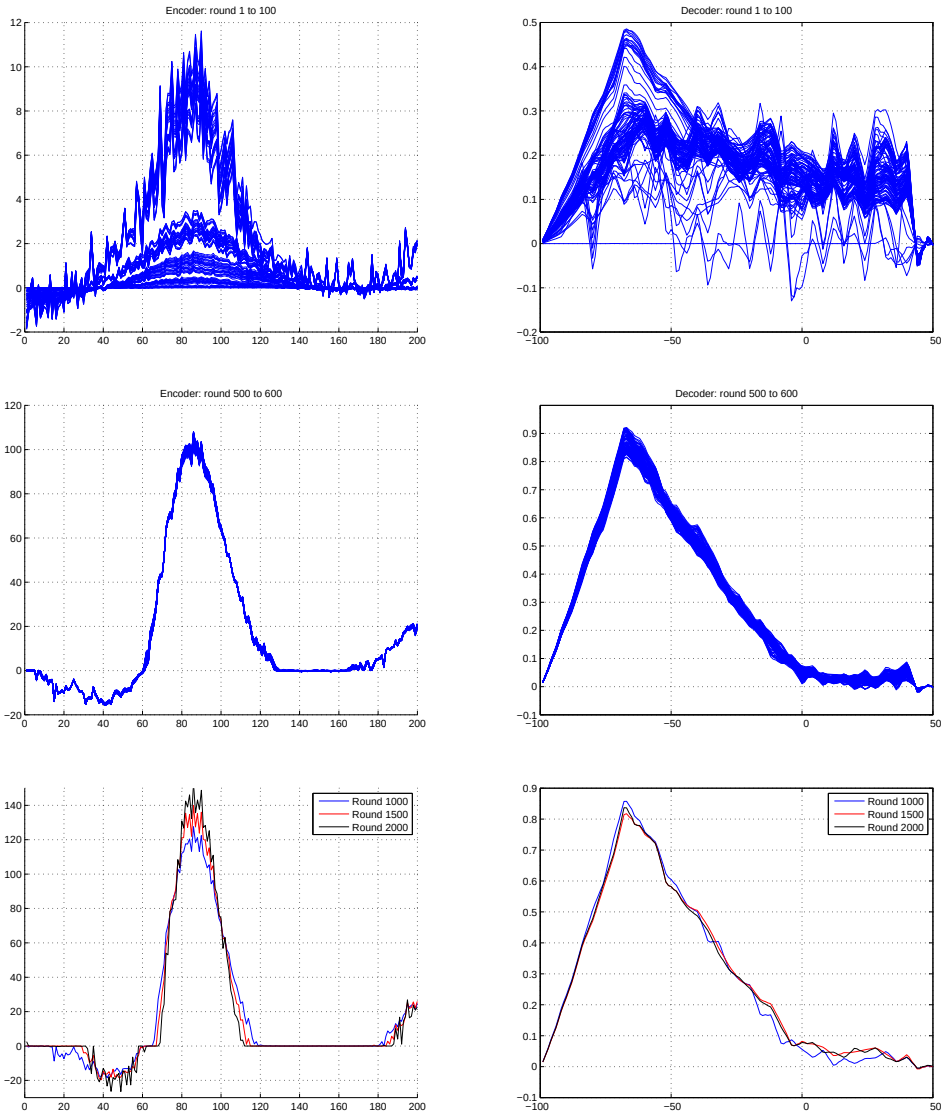


Figure 5.7: Development of encoding filter w (left) and decoding filter h (right) for the punishment of the squared L_1 norm, $J_E = J_{1s}$. The resulting encoding filter w shows a large main peak, a negative primary sidelobe and a small secondary positive sidelobe. A characteristic feature are the clear zeros after the main peak and before the primary sidelobe. Note: Different scales.

μ was however set to a larger value: $\mu = 5$ for the simulations with $\alpha = 0.005$ and $\alpha = 0.0075$. For the simulation with $\alpha = 0.001$ was $\mu = 10$ for the first two hundred rounds, then it was set to $\mu = 1$.

5.4 Discussion

We have have dealt here with the first research question of Section 4.3.2. We formulated the data representation problem as an optimization problem in which reconstruction error and an energy cost are minimized. This enables learning of the encoder and decoder. We used variational calculus to derive an online rule for the learning of encoder and decoder. Simulations showed that the online rule can learn the general shape of the input distribution: The selectivity of the decoding filter h matches the input.

Learning rule

The input before a spike is fired enters together with the history of previous spike events, weighted by the Γ factor, in a normalized form into the update of the encoder (Equation (5.12) and (5.18)). This update is however weighted by $\bar{e}$, an index for the reliability of the spike. It could be interpreted as an instance of experience-dependent plasticity, see e.g. [Yao et al., 2007]. The calculation of that reliability index is done by means of the decoder. The learning rules for the encoder and decoder are synergistic in the sense that the encoder influences the update of the decoder too. This happens through the selection of its learning data by triggering spikes at the right time.

The obtained learning rule does not relate much to the learning rules which we have presented in Section 2.3. In the learning rule of [Smith and Lewicki, 2006] in Equation (2.85), the reconstruction error enters into the update weighted by the neural activity. Since we work here with all-or-none spikes, there is no such weighting by neural activity. Instead of the error, the input before a spike is used for the update, weighted by reliability of the spike (as measured by $\bar{e}$.) In the learning rule of [Hyvärinen et al., 2003] in Equation (2.98), it is also the input and not the error which enters into the update rule. This time however, weighted by its neural activity.

A further difference to the data representation methods of Section 2.3 is that here, the encoding is causal. The learning rules for the encoding filter w and decoding filter h are causal as well. Note further that the update of w in Equa-

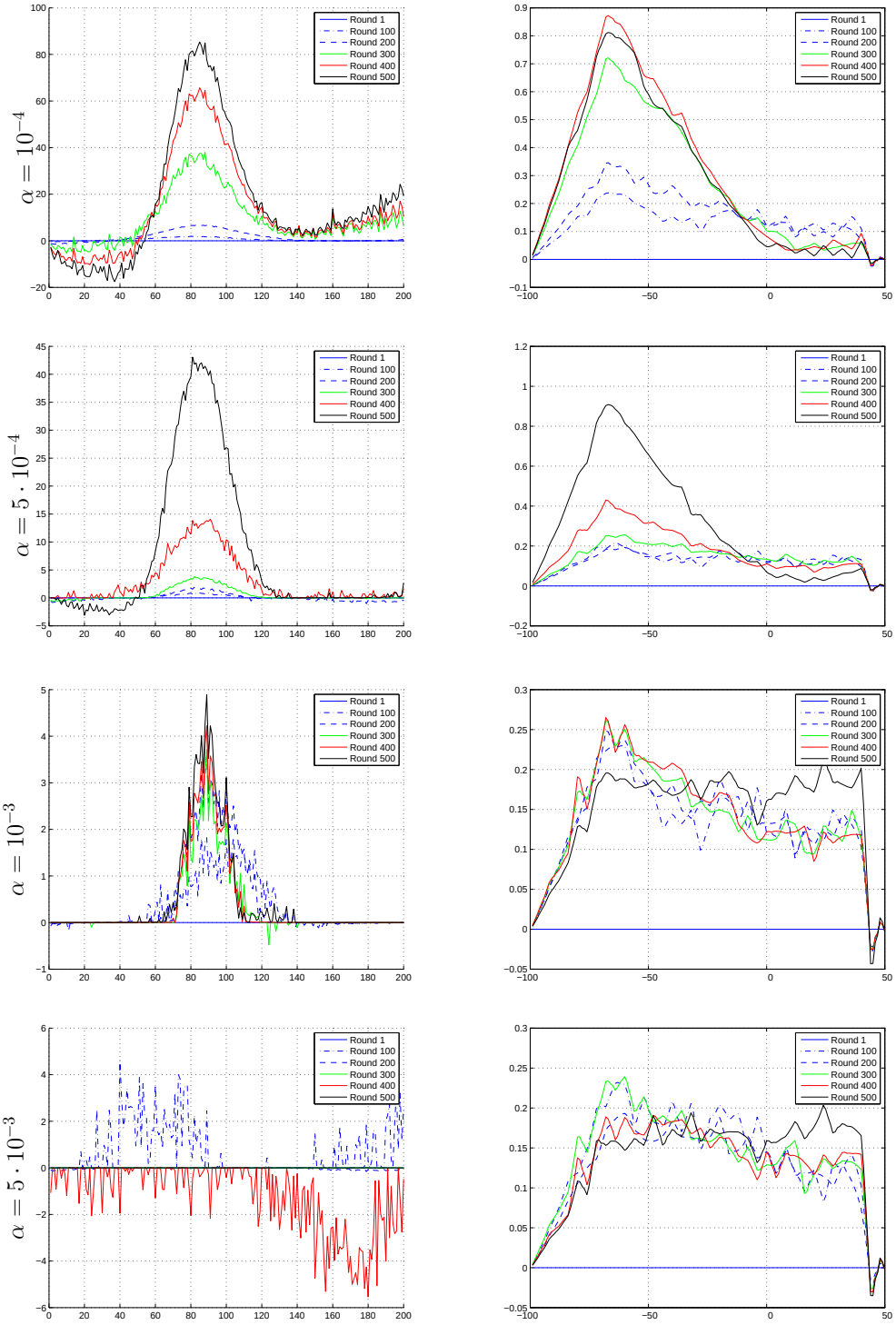


Figure 5.8: Development of encoding filter w (left) and decoding filter h (right) for various weightings of the L_1 norm, $J_E = J_1$. Note the different scales.

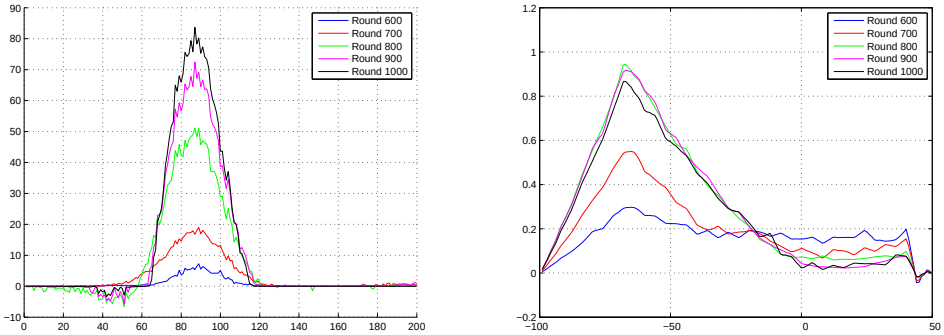


Figure 5.9: Further development of encoding filter w (left) and decoding filter h (right) for the weighting $\alpha = 10^{-3}$ (with punishment of the L_1 norm, i.e. $J_E = J_1$).

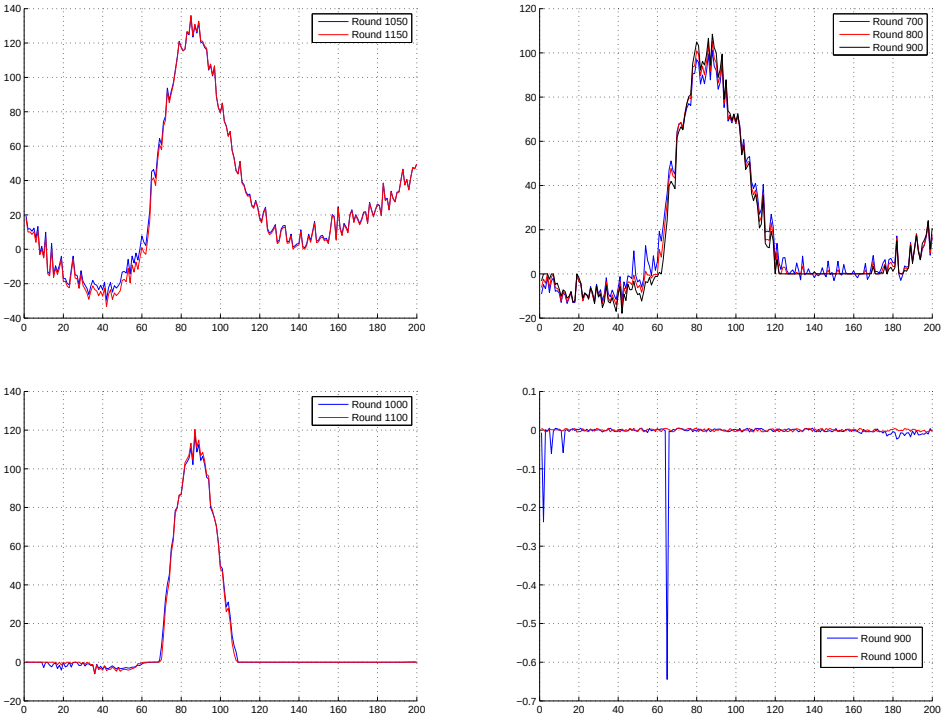


Figure 5.10: Encoding filter w after learning. First row shows the case of $\alpha = 10^{-4}$ (left) and $\alpha = 5 \cdot 10^{-4}$ (right). The second row shows $\alpha = 10^{-3}$ (left) and $\alpha = 5 \cdot 10^{-3}$ (right). Learning was done with punishment of the L_1 norm, i.e. $J_E = J_1$.

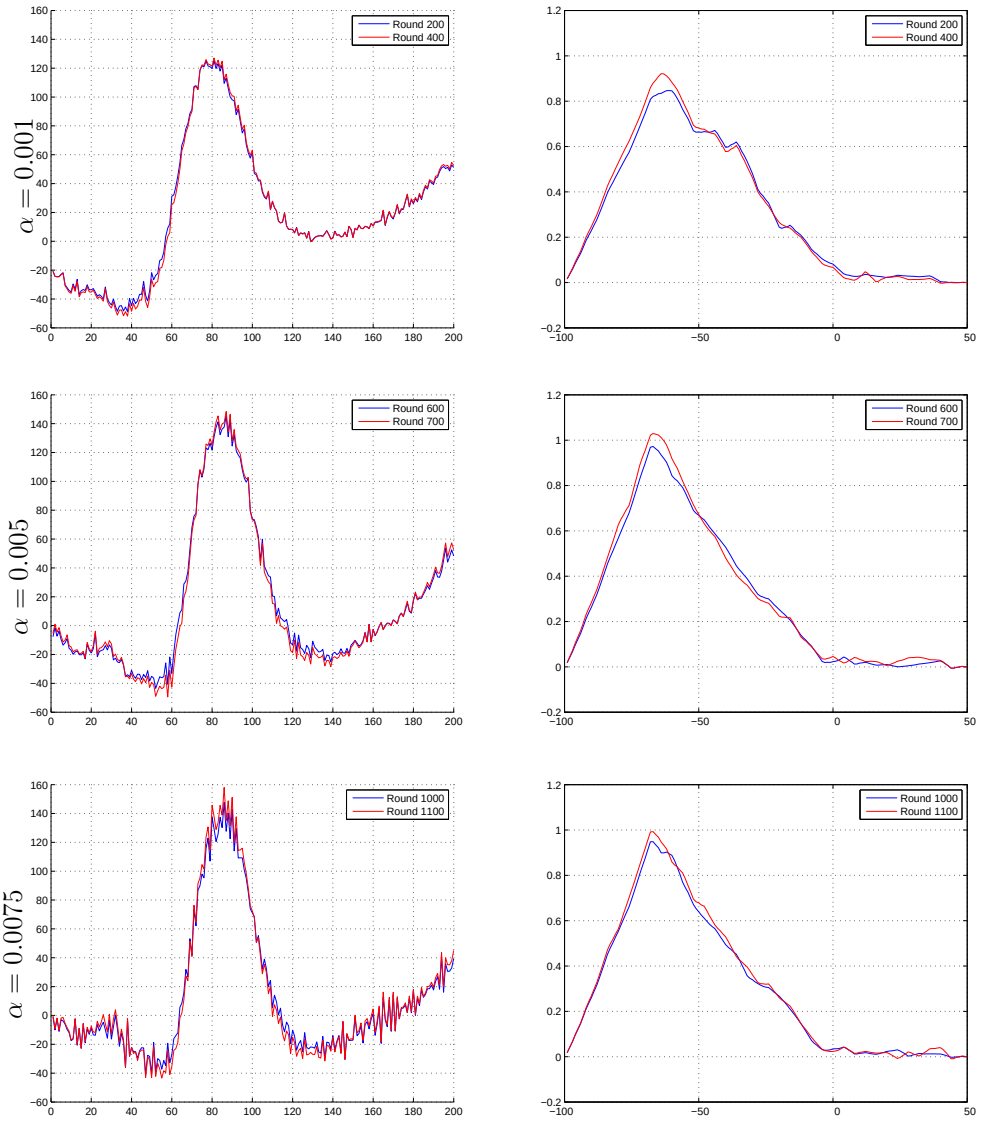


Figure 5.11: Encoding filter w (left) and decoding filter h (right) after learning for various weightings α of the energy cost J_p. A secondary negative sidelobe emerges after the main peak for $\alpha = 0.005$ and $\alpha = 0.0075$.

tion (5.18) cannot take place immediately after the spike event at t^k. For the calculation of $\bar{e}(t^k)$, one needs the approximation $\hat{x}(t)$ at $t = t^k + T_p$. And this approximation $\hat{x}(t^k + T_p)$ is only obtained at time $t^k + T_p + T_d$ (see Equation (5.2)). Hence, the update of w can only take place at time $t^k + T_p + T_d$.

Simulations

We have shown simulations for all types of punishment of energy consumption that we have introduced in Section 5.1.3. The learning rules were given in Section 5.2.1. Here, we compare and comment on the resulting encoding filters.

- Figure 5.5 shows that energy constraints are not needed for a stable development. But the punishment of energy consumption can optionally be included in order to investigate the energy efficiency of the representation.

- The punishment of the energy consumption in the form of J_2 leads in each update (Equation (5.18)) to a subtraction of the scaled vector $\delta J_2/\delta w(s)$ which is proportional to w. The encoding filter in Figure 5.6, which results from the learning, has the same graded shape as the one obtained without energy punishment. The peak and the sidelobes are however smaller.

- The inclusion of J_{1s} leads in each update to the subtraction of the scaled vector $\delta J_{1s}/\delta w(s)$ which has a direction given by $\text{sign}(w)$. It is further scaled by the L_1 norm of w. The encoding filter in Figure 5.7, which results from the learning, has no graded shape but attains clear zeros. It still features the sidelobes.

- The inclusion of J_1 leads to similar results (Figure 5.10). The important difference is that a vector which is not related to the scale of w is subtracted in each update. This suppresses the development in the early stages of the learning process.

- For J_p, the subtraction vector is input-dependent. This leads to a new characteristic in the learned encoding filter. The shape is similar to the case where there is no punishment but a new negative sidelobe after the main peak and before the secondary positive sidelobe emerges (Figure 5.11).

The different forms of energy punishment lead thus to different encoding filters. The subtraction of a vector which is proportional to w itself has been called multiplicative enforcement of the constraint [Miller and MacKay, 1994]. This

is the case for J_2. The subtraction of a vector with another direction than w was called subtractive enforcement. This is the case for J_1. The punishment in the form of J_{1s} is in between, and J_p can not be classified in that manner.

5.5 Appendix: Calculation of the functional derivatives

Functional derivative $\delta J_e/\delta w(s)$

We start by perturbating $w(s)$ to $w(s)+\delta w(s)$ where

$$\delta w(s) = \epsilon\varphi(s), \tag{5.27}$$

for a small constant $\epsilon > 0$ and a sufficiently smooth, but else arbitrary function $\varphi(s)$. The perturbation δw causes a perturbation δt^f of the spike timings which in turn cause a perturbation $\delta\hat{x}(t)$ of the reconstruction $\hat{x}$. The resulting perturbation δJ_e of the cost functional J_e is

$$\delta J_e = \frac{1}{T}\int_0^T (\hat{x}(t)-x(t))\delta\hat{x}(t)\mathrm{d}t + \frac{1}{2T}\int_0^T (\delta\hat{x}(t))^2\mathrm{d}t \tag{5.28}$$

We use the chain rule to obtain

$$\delta\hat{x}(t) = \sum_f \frac{\partial\hat{x}(t)}{\partial t^f}\delta t^f. \tag{5.29}$$

For a fixed spike index f, Equation (5.2) leads to

$$\frac{\partial\hat{x}(t)}{\partial t^f} = \begin{cases} -\dot{h}(t-t^f) & t^f - T_d < t < t^f + T_p \\ 0 & \text{else} \end{cases} \tag{5.30}$$

In order to evaluate Equation (5.29) and thus Equation (5.28), we must know the perturbation δt^f.

The perturbation δt^f The spike timing $t^f > T_w$ is defined by $u(t^f) = \theta$, i.e.

$$\begin{aligned}\theta &= \eta_0\exp\left[-\frac{t^f-t^{f-1}}{\tau}\right] + \int_0^{T_w} x(t^f-s)w(s)\mathrm{d}s \\ &\quad + I_n(t^f).\end{aligned} \tag{5.31}$$

After perturbation of $w(s)$, me make for δt^f the following ansatz

$$\delta t^f = \epsilon a_f + O(\epsilon^2), \tag{5.32}$$

where a_f needs to be determined. The implicit equation for a_f is given by

$$\begin{aligned}\theta &= \eta_0\exp\left[-\frac{t^f-t^{f-1}}{\tau}\right]\exp\left[-\frac{\epsilon(a_f-a_{f-1})+O(\epsilon^2)}{\tau}\right] \\ &\quad + \int_0^{T_w} x(t^f+\epsilon a_f+O(\epsilon^2)-s)(w(s)+\epsilon\varphi(s))\mathrm{d}s \\ &\quad + I_n(t^f+\epsilon a_f+O(\epsilon^2)).\end{aligned} \tag{5.33}$$

The constant $\epsilon > 0$ can be made arbitrarily small, so that via Taylor series[1] and Equation (5.31) we obtain

$$\begin{aligned} 0 &= \epsilon\Bigg\{ a_f \Bigg(\frac{-\eta_0}{\tau} \exp\left[-\frac{t^f - t^{f-1}}{\tau} \right] + \dot{I}_n(t^f) + \\ &\int_0^{T_w} \dot{x}(t^f - s) w(s) \mathrm{d}s \Bigg) + \frac{a_{f-1}\eta_0}{\tau} \exp\left[-\frac{t^f - t^{f-1}}{\tau} \right] \\ &+ \int_0^{T_w} x(t^f - s)\varphi(s)\mathrm{d}s \Bigg\} + O(\epsilon^2). \end{aligned} \tag{5.34}$$

From Equation (5.34), we see that a_f must satisfy

$$a_f = -\frac{\int_0^{T_w} x(t^f - s)\varphi(s)\mathrm{d}s}{\dot{u}(t^f)} + \Gamma(t^f, t^{f-1}) a_{f-1} \tag{5.35}$$

where

$$\Gamma(t^f, t^{f-1}) = \frac{-\eta_0}{\tau \dot{u}(t^f)} \exp\left[-\frac{t^f - t^{f-1}}{\tau} \right], \tag{5.36}$$

$$\begin{aligned} \dot{u}(t^f) &= \frac{-\eta_0}{\tau} \exp\left[-\frac{t^f - t^{f-1}}{\tau} \right] + \dot{I}_n(t^f) \\ &\quad + \int_0^{T_w} \dot{x}(t^f - s) w(s) \mathrm{d}s. \end{aligned} \tag{5.37}$$

Finally, we see from Equation (5.35) that a_f has the form

$$a_f = \int_0^{T_w} y_f(s)\varphi(s)\mathrm{d}s, \tag{5.38}$$

so that we obtain the update rule for y_f

$$y_f(s) = \frac{-x(t^f - s)}{\dot{u}(t^f)} + \Gamma(t^f, t^{f-1}) y_{f-1}(s), \tag{5.39}$$

and δt^f is given by

$$\delta t^f = \epsilon \int_0^{T_w} y_f(s)\varphi(s)\mathrm{d}s + O(\epsilon^2). \tag{5.40}$$

Calculation of the functional derivative $\delta J_e / \delta w$ The derived Equation (5.40) allows with Equation (5.30) to evaluate Equation (5.29), and hence to obtain δJ_e via Equation (5.28).

[1] We assume that the optional noise current is smooth enough.

Equation (5.29) for $\delta\hat{x}(t)$ introduces a sum over the spike timings t^f into the integral

$$M = \int_0^T (\hat{x}(t) - x(t))\delta\hat{x}(t)\mathrm{d}t \tag{5.41}$$

of Equation (5.28). Since there are only finitely many spikes during the time interval $[0 \quad T]$, the sum can only have finitely many terms. We interchange thus the summation and the integration to obtain with Equation (5.30)

$$\begin{aligned} M &= -\sum_f \underbrace{\int_{t^f - Td}^{t^f + T_p} (\hat{x}(t) - x(t))\dot{h}(t - t^f)\mathrm{d}t}_{\bar{e}(t^f)} \\ &\quad \cdot \int_0^{T_w} \epsilon y_f(s)\varphi(s)\mathrm{d}s + O(\epsilon^2). \end{aligned} \tag{5.42}$$

The quadratic terms in Equation (5.28) yield terms in the order of ϵ^2 so that we obtain with the previous equation for M

$$\delta J_e = \epsilon\left(-\frac{1}{T}\sum_f \bar{e}(t^f) \int_0^{T_w} y_f(s)\varphi(s)\mathrm{d}s \right) + O(\epsilon^2). \tag{5.43}$$

Taking the principal linear part [Gelfand and Fomin, 2000], we obtain the final expression for the functional derivative of J_e with respect to w

$$\frac{\delta J_e}{\delta w(s)} = -\frac{1}{T}\sum_f \bar{e}(t^f) y_f(s) \tag{5.44}$$

Functional derivative $\delta J_E/\delta w(s)$

Straightforward calculation gives

$$\frac{\delta J_2}{\delta w(s)} = 2w(s) \tag{5.45}$$

$$\frac{\delta J_{1s}}{\delta w(s)} = 2||w||_1 \mathrm{sign}(w(s)) \tag{5.46}$$

$$\frac{\delta J_1}{\delta w(s)} = \mathrm{sign}(w(s)) \tag{5.47}$$

$$\frac{\delta J_p}{\delta w(s)} = \int_0^T \mathrm{sign}(I(t))x(t-s)\mathrm{d}t \tag{5.48}$$

Chapter 6

Data Representation with Multiple Integrator Neurons

6.1 Cost functional for data representation

In Section 6.1.1, we recapitulate the second research question of Section 4.3.2. In Section 6.1.2, we obtain part one of the cost functional which measures the accuracy of the representation. In Section 6.1.3, we present part two of the cost functional which measures the energy consumption during the encoding process.

6.1.1 Recapitulation of the second research question

The goal of this chapter is to learn a representation of sensory input where the encoding happens with multiple neurons of the spike response type, see Section 3.1.3 and [Gerstner and Kistler, 2002]. This is the extension of Chapter 5 to the case of multiple neurons.

There are two different cases to distinguish. In the first case, the neurons have no lateral connections. In the second case, they have lateral connections. Figure 6.1 visualizes the research task. Learning the representation is based on the minimization of the reconstruction error, with an added penalty to minimize energy consumption during the encoding process, as in Chapter 5. The research task is to formulate the optimization problem and to find and test an iterative optimization rule.

6.1.2 Part one of the cost functional: reconstruction error

Each neuron is modeled as in Chapter 5. The equation for the membrane voltage u_m of neuron m is

$$\begin{aligned} u_m(t) &= \eta_0 \exp\left[-\frac{t-\hat{t}}{\tau}\right] + \int_0^{\min\{t,T_w\}} x(t-s)w_m(s)\mathrm{d}s \\ &\quad + \sum_{j\neq m} \sum_{f:t-T_v<t_j^f<t} v_{mj}(t-t_j^f) + In_m(t), \end{aligned} \tag{6.1}$$

where $In_m(t)$ is a sufficiently smooth noise current, $\hat{t}$ the last spike time before time t, and w_m the unknown causal encoding filter, to be learned, of length T_w. The lateral filters, converging from neuron j onto neuron m are given by v_{mj}. They are causal and of length T_v. They are to be learned. The lateral input current due to neuron j is denoted by I_{mj}. The convolution of input x with w_m produces the feedforward input current I_m. Spike timings $\{t_j^f; f = 1, \ldots\}$ are defined by $u_j(t^f) = \theta$, where $\theta > 0$ is a fixed threshold. The remaining constants are the recovery time constant τ of the recovery current I_r and the reset amount $\eta_0 < 0$.

As in Chapter 5, the reconstruction $\hat{x}$ is sought under the form

$$\hat{x}(t) = \sum_m \sum_{f: t-T_p < t_m^f < t+T_d} h_m(t - t_m^f), \tag{6.2}$$

which introduces a delay T_d in the estimate. For the reconstruction at time t, spikes happening prior to $t - T_p$ are not considered.

The first part of the cost functional which is due to the reconstruction error is then given by

$$J_e(h_m, w_m, v_{mj}) = \frac{1}{2T} \int_0^T (\hat{x}(t) - x(t))^2 \mathrm{d}t, \tag{6.3}$$

where T is a fixed time horizon.

6.1.3 Part two of the cost functional: energy consumption

We measure here the energy consumption by means of the ion load per unit time that must be pumped out, or in, to restore the ion gradients in the neuron, see Section 3.1.2. We have introduced that idea already in Section 5.1.3, where the L_1 norm of the feedforward filter w was used as punishment. For neuron m, the energy consumption due to the feedforward current is measured by $||w_m||_1$.

Neuron m receives additionally to the feedforward current I_m lateral input current I_{mj} from neuron j, see Equation (6.1). We measure the ion load from the lateral current in the same way as for the feedforward current, i.e. by

$$P_p(m, j) = \frac{1}{T} \int_0^T |I_{mj}(t)| \mathrm{d}t. \tag{6.4}$$

With the definition of I_{mj} through Equation (6.1), we obtain

$$P_p(m, j) \leq \frac{1}{T} \sum_f \int_0^T |v_{mj}(t - t_j^f)| \mathrm{d}t, \tag{6.5}$$

which equals $\bar{f}_j ||v_{mj}||_1$, where $\bar{f}_j$ is the number of spikes of neuron j during time T (average firing rate). For each neuron m, we have to sum over all convergent neurons. The measurement of the energy consumption due to lateral input onto neuron m is then $\sum_j \bar{f}_j ||v_{mj}||_1$.

The total energy consumption by the network is due to feedforward and lateral input onto each neuron m. According to the above discussion, it becomes

$$J_E(w_m, v_{mj}) = \sum_m \left(||w_m||_1 + \sum_{j \neq m} \bar{f}_j ||v_{mj}||_1 \right). \tag{6.6}$$

This is the cost functional for the punishment of energy consumption for this chapter.

In Section 3.1.2, we have seen that the propagation of action potentials is about equally costly as the postsynaptic ion load. This energy cost depends on the length of the axonal connection. Denoting the length of the connection from neuron m to neuron j by l_{jm}, the propagation cost of the spikes of neuron m could be quantified by

$$P_s(m) = \bar{f}_m \sum_j l_{jm}. \tag{6.7}$$

Here, we do not introduce a topology, i.e. the lengths l_{jm}. We forego the punishment of the propagation of the spikes. Equally, the cost for the maintenance of the steady state of each neuron (see Section 3.1.2) is not modeled.

6.2 Learning rule

The online rule for the feedforward filters w_m and lateral filters v_{mj} is based on a steepest descent method on the total cost J, which is the sum of the reconstruction error J_e, defined in Equation (6.3), and the energy cost J_E, defined in Equation (6.6). The update rule for the decoding filters h_m is based on a recursive least squares (RLS) or a simpler least mean square (LMS) algorithm (see e.g. [Haykin, 2001]).

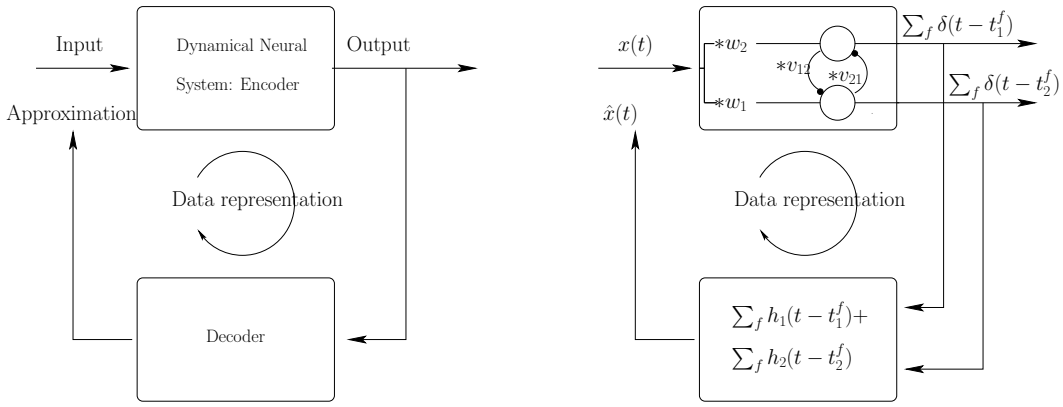
6.2.1 Feedforward encoding filter w_m

Key to the update rule for the feedforward filters w_m is the calculation of the functional derivative $\delta J/\delta w_m(s)$. In the Appendix Section 6.5, we calculate it for J_e and the energy cost J_E. The lateral filters v_{mj} and the decoding filters h_m are held fixed.

Functional derivative of J_e

The functional derivative of J_e with respect to w_m is

$$\frac{\delta J_e}{\delta w_m(s)} = -\frac{1}{T} \sum_i \sum_f \bar{e}(t_i^f) y_i^m(s, f), \tag{6.8}$$



(a) General scheme

(b) Scheme in this chapter
(Picture shows the special case of two neurons)

Figure 6.1: (a) Input data x is transformed by a dynamical neural system. This encoding process should be such that the input can be decoded from the output. The energy consumption during the encoding process should also be minimized. The encoder and decoder together provide data representation. (b) Here, we encode by means of multiple neurons which are modeled by the spike response model (see Section 3.1.3 and [Gerstner and Kistler, 2002]), and decoding is done from the spike timings t_m^f. The picture shows the special case of two neurons. We distinguish between the case where there are no lateral connections between the neurons from the case where a neuron has convergent lateral inputs. In that case, the lateral input at time t onto neuron m is modeled by $\sum_j \sum_f v_{mj}(t-t_j^f)$. Learning rules for the unknown feedforward encoding filters w_m, the lateral filters v_{mj}, and decoding filters h_m are derived from the minimization of the mean squared reconstruction error, with an added penalty to minimize energy consumption during the encoding process.

where

$$\bar{e}(t_i^f) = \int_{t_i^f - T_d}^{t_i^f + Tp} (\hat{x}(t) - x(t))\dot{h}_i(t - t_i^f)\mathrm{d}t, \tag{6.9}$$

$$y_m^m(s,f) = \frac{1}{\dot{u}_m(t_m^f)}\Big[- x(t_m^f - s) + \Gamma(t_m^f, t_m^{f-1})y_m^m(s,f-1) + \sum_{n\neq m}\sum_{\nu}\dot{v}_{mn}(t_m^f - t_n^\nu)y_n^m(s,\nu)\Big], \tag{6.10}$$

$$y_i^m(s,f) = \frac{1}{\dot{u}_i(t_i^f)}\Big[\Gamma(t_i^f, t_i^{f-1})y_i^m(s,f-1) + \sum_{n\neq i}\sum_{\nu}\dot{v}_{in}(t_i^f - t_n^\nu)y_n^m(s,\nu)\Big], \quad i \neq m, \tag{6.11}$$

$$\Gamma(t^2,t^1) = \frac{-\eta_0}{\tau}\exp\left[-\frac{t^2 - t^1}{\tau}\right]. \tag{6.12}$$

The initial values in the recursion are $y_m^m(s,0) = 0$ and $y_i^m(s,0) = 0$. The indices in the above equations, e.g. for $y_i^m(s,f)$, have the following meaning: Upper index m indicates the filter $w_m(s)$ with respect to which the derivative is taken. The variable s refers to the variable s in $w_m(s)$. Index f is used to enumerate the spikes of neuron i.

Functional derivative of J_E

We assume that small changes in a feedforward filter $w_m(s)$ may change the timings of the spikes, but that they do not change the mean firing rate $\bar{f}$ in Equation (6.6). The functional derivative of J_E with respect to $w_m(s)$ is then

$$\frac{\delta J_E}{\delta w_m(s)} = \mathrm{sign}(w_m(s)). \tag{6.13}$$

This is the same as in Equation (5.16), where we have dealt with the single neuron case.

Online rule

For the feedforward encoding filter $w_m(s)$, we propose the following online rule: If neuron $i \neq m$ emits at t_i^f its f-th spike, update $w_m(s)$ by

$$w_m(s) \leftarrow w_m(s) + \mu\left[\bar{e}(t_i^f)y_i^m(s,f)\right]. \tag{6.14}$$

If neuron m emits at t_m^f its f-th spike, update $w_m(s)$ by

$$w_m(s) \leftarrow w_m(s) + \mu\left[\bar{e}(t_i^f)y_i^m(s,f) - \alpha\mathrm{sign}(w_m(s))\right], \tag{6.15}$$

where $\mu > 0$ is the step size and α weights the influence of the energy constraint.

6.2.2 Lateral encoding filter v_{mj}

We assume $v_{mj}(s)$ can be written as

$$v_{mj}(s) = \sum_{n=1}^{N_{mj}} c_n^{mj} \Upsilon_n(s). \tag{6.16}$$

Each lateral filter v_{mj} is thus characterized by the coefficients c_n^{mj} which can be collected into the column vector $\mathbf{c^{mj}}$. The functions $\Upsilon_n(s)$ at instant s can also be collected into the column vector $\boldsymbol{\Upsilon}(s)$.

Key to the update rule for the lateral filters v_{mj} is the calculation of the functional derivative $\delta J/\delta v_{mj}(s)$, i.e. the derivative $\delta J/\delta \mathbf{c^{mj}}$. In the Appendix Section 6.5, we calculate it for J_e and the energy cost J_E. The feedforward filters w_m and the decoding filters h_m are held fixed.

Functional derivative of J_e

The functional derivative of J_e with respect to $v_{mj}(s)$, i.e. with respect to the vector $\mathbf{c^{mj}}$, is

$$\frac{\delta J_e}{\delta \mathbf{c^{mj}}} = -\frac{1}{T} \sum_i \sum_f \bar{e}(t_i^f) \mathbf{z_i^{mj}}(f), \tag{6.17}$$

where $\bar{e}(t_i^f)$ is given by Equation (6.9), and

$$\begin{aligned} \mathbf{z_m^{mj}}(f) &= \frac{1}{\dot{u}_m(t_m^f)} \Big[-\sum_\nu \boldsymbol{\Upsilon}(t_m^f - t_j^\nu) + \Gamma(t_m^f, t_m^{f-1}) \mathbf{z_m^{mj}}(f-1) + \\ &\quad \sum_{n \neq m} \sum_\nu \dot{v}_{mn}(t_m^f - t_n^\nu) \mathbf{z_n^{mj}}(\nu) \Big], \end{aligned} \tag{6.18}$$

$$\begin{aligned} \mathbf{z_i^{mj}}(f) &= \frac{1}{\dot{u}_i(t_i^f)} \Big[\Gamma(t_i^f, t_i^{f-1}) \mathbf{z_i^{mj}}(f-1) + \\ &\quad \sum_{n \neq i} \sum_\nu \dot{v}_{in}(t_i^f - t_n^\nu) \mathbf{z_n^{mj}}(\nu) \Big], \quad i \neq m. \end{aligned} \tag{6.19}$$

The term $\Gamma(t^2, t^1)$ was defined in Equation (6.12). The vectors $\mathbf{z_m^{mj}}$ and $\mathbf{z_i^{mj}}$ are in the recursion initialized with zero, i.e. $\mathbf{z_m^{mj}}(0) = 0$ and $\mathbf{z_i^{mj}}(0) = 0$.

Functional derivative of J_E

We assume that small changes in a lateral filter $v_{mj}(s)$ may change the timings of the spikes, but that they do not change the mean firing rate $\bar{f}$ in Equation (6.6).

Under that assumption, we obtain

$$\frac{\delta J_E}{\delta \mathbf{c^{mj}}} = \bar{f}_j \int_0^{T_v} \text{sign}(v_{mj}(t))\boldsymbol{\Upsilon}(t)\mathrm{d}t \tag{6.20}$$

Online rule

For the lateral filter $v_{mj}(s)$, we propose the following online rule: If neuron $i \neq j$ emits at t_i^f its f-th spike, update the coefficients $\mathbf{c^{mj}}$ of $v_{mj}(s)$ by

$$\mathbf{c^{mj}} \leftarrow \mathbf{c^{mj}} + \mu \left[\bar{e}(t_i^f)\mathbf{z_i^{mj}}(f)\right]. \tag{6.21}$$

If neuron j emits at t_j^f its f-th spike, update $\mathbf{c^{mj}}$ by

$$\mathbf{c^{mj}} \leftarrow \mathbf{c^{mj}} + \mu \left[\bar{e}(t_j^f)\mathbf{z_j^{mj}}(f) - \alpha \int_0^{T_v} \text{sign}(v_{mj}(t))\boldsymbol{\Upsilon}(t)\mathrm{d}t\right], \tag{6.22}$$

where $\mu > 0$ is the step size and α weights the influence of the energy constraint.

6.2.3 Decoding filter h_m

The approach of Section 5.2.2 for the case of a single output channel can also be used for multiple output channels. We obtain then for the reconstruction $\hat{x}$

$$\begin{aligned} \hat{x}(n) &= [\mathbf{c_1} \ \ldots \mathbf{c_M}][(\boldsymbol{\Psi}\rho_\mathbf{1})^\mathbf{T} \ \ldots (\boldsymbol{\Psi}\rho_\mathbf{M})^\mathbf{T}]^T & (6.23)\\ &= \mathbf{cy}(n) & (6.24)\end{aligned}$$

with column vector $\mathbf{y} = [(\boldsymbol{\Psi}\rho_\mathbf{1})^\mathbf{T} \ \ldots (\boldsymbol{\Psi}\rho_\mathbf{M})^\mathbf{T}]^T$ and unknown row vector $\mathbf{c} = [\mathbf{c_1} \ \ldots \mathbf{c_M}]$ which is learned by a LMS or RLS algorithm as in Section 5.2.2.

6.3 Simulations

6.3.1 Without lateral connections: Two channel input

Without lateral connections, the learning rules for the feedforward encoding filter remain the same as in Section 5.2.1. We use here those with a punishment of energy consumption via $J_E = J_2$. The learning rule for the decoder is given in Section 6.2.3. We use here the wavelet-based RLS algorithm to learn the decoding filters.

We simulate the condition where a signed input x was split into two channels. The plus-channel x_+ features the positive part of x while the minus-channel x_- the sign-reversed negative part of x. There are thus statistical dependencies between

the two channels. Each channel is filtered with a corresponding encoding filter $w_\pm$ to produce spike timings $\{t_\pm^f\}$. The input x is then reconstructed via

$$\hat{x}(t) = \sum_{f:t-T_p<t_+^f<t+T_d} h_+(t-t_+^f) + \sum_{f:t-T_p<t_-^f<t+T_d} h_-(t-t_-^f), \tag{6.25}$$

where $h_\pm(s)$ is not constrained to positive or negative values.

In Figure 6.2, we show results for the case where x consists of randomly placed, one period long sinus curve segments with added noise: The learned encoding filters $w_\pm$ and decoding filters $h_\pm$ have the property that w_+ and w_- transform the input x in such a way into spike timings $\{t_\pm^f\}$ that the reconstruction $\hat{x}$ features the sinusoidal segments. Decoder h_+ develops into the positive part, and h_- into the negative one. The spike timings t_+^f and t_-^f are such that the two half-waves are neatly joined together to yield the sine segment in the reconstruction.

Simulation parameters For the integration scheme, see Section 5.3.1. We used $\alpha = 10^{-4}$, and step size $\mu = 0.5$. The decoders were searched in the wavelet subspace V_{-1} (see [Frazier, 2001]). They had 109 unknown coefficients. For each encoding filter 200 points were learned.

6.3.2 Without lateral connections: Divergent connections

Without lateral connections, the learning rules for the feedforward encoding filter remain the same as in Section 5.2.1. We use here those with no punishment of energy consumption, or via $J_E = J_1$. The learning rule for the decoder is given in Section 6.2.3. We use here the LMS based learning rule in the standard basis.

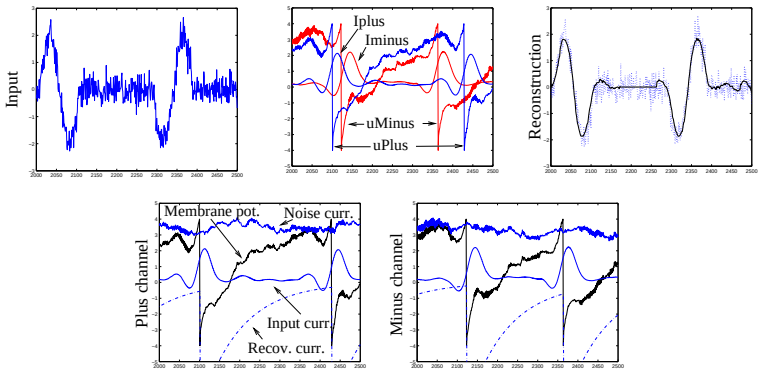
Figure 6.3 shows an excerpt of the input $x(t)$ that we seek to represent by means of the spike timings of integrator neurons. We use here five integrators to represent $x(t)$. A single input channel is thus split into five output channels (divergent configuration). We need to learn the feedforward encoding filters w_1 to w_5 and the decoding filters h_1 to h_5.

Figure 6.4 shows the learning results when energy consumption is not punished ($\alpha = 0$). Figure 6.5 and Figure 6.6 depict the case $\alpha = 0.05$ and $\alpha = 0.1$, respectively. In all cases, the feedforward encoding filters w_m are initialized to $w_m = 0.2$. The neurons receive however different noise currents In_m. From the figures, we see that some feedforward filters are suppressed with learning. This happens for all values of α. The difference for larger values of α is that the encoders loose their "graded" shape, which assists the suppression.

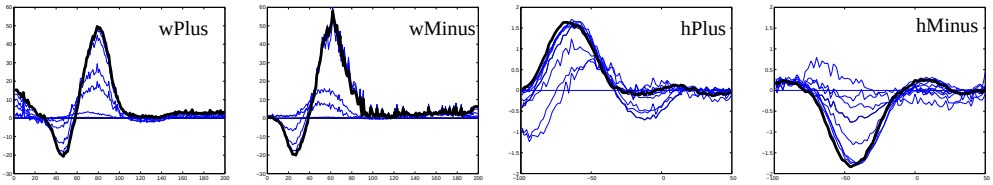
We show for the case $\alpha = 0$ the representation of $x(t)$ by means of the population of neurons in Figure 6.7. The figure shows that the population activity

represents the input accurately. It further shows that the individual neurons have different roles. Neuron 2 codes for the mean input. Neuron 1 codes for small details while neuron 5 codes for an intermediate resolution. Figure 6.3 showed that the input consists of elements of two different resolutions. Neuron 1 and 5 pinpoint this property of the input.

The filters shown in the above figures have rather large delays. Figure 6.8 illustrates that the initial conditions, together with chance in virtue of the noise current, can influence the time lag of the filters.

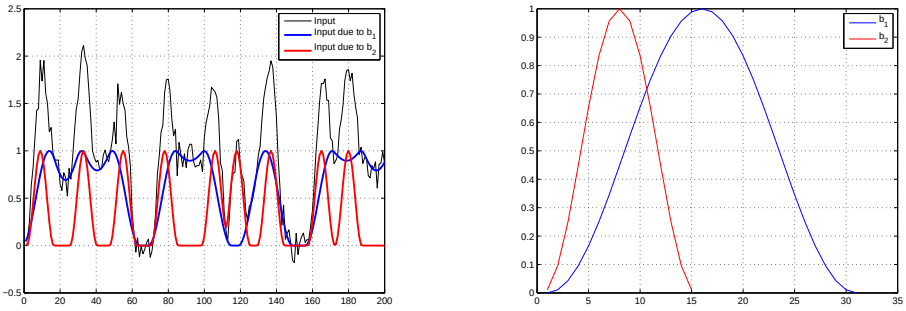


(a) Reconstruction and the involved currents after learning.



(b) Development of the encoding filters $w_\pm$ and decoding filters $h_\pm$

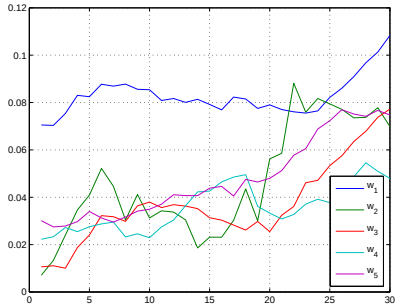
Figure 6.2: Two channel input (a) The positive and negative parts of the input x are encoded in two distinct channels. Refer to Figure 5.6 for details of the encoding mechanism. In the involved currents, we note that the input current $I_\pm$ reduces the membrane potential $u_\pm$ shortly before a spike event. This reduces the possibility of a noise triggered spike allowing for an accurately timed reconstruction. (b) The curves show some development stages in the learning process. The thick curves correspond to the case shown in (a) The decoding filters $h_\pm$ develop such that each reconstructs only one half of the sine wave. Hence, the spike timings in Figure 6.2a must be such that the two half-waves fit neatly together to yield the reconstruction. The two encoding filters $w_\pm$ develop into similar shape. Encoding filter w_- is more noisy and more shifted to the left leading to a faster response.



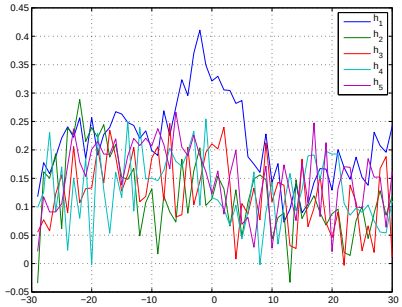
(a) Input (b) Basic elements

Figure 6.3: (a) The input is artificially created via $x(t) = \sum_f b_1(t-t_1^f) + \sum_f b_2(t-t_2^f) + n(t)$. The shift amounts t_i^f are random variables, and $n(t)$ is a small amount of white Gaussian noise. Given t_i^{f-1}, the next shift time t_i^f is drawn from an uniform distribution with support $(t_i^{f-1}+10 \quad t_i^{f-1}+40]$. Shown is an excerpt of $x(t)$ (black). In blue and red are the contributions of b_1 and b_2, respectively. (b) The basic elements $b_1(t)$ and $b_2(t)$. They have two different scales. This is the same setup as in Figure 2.1 and 2.2 of Section 2.1.1.

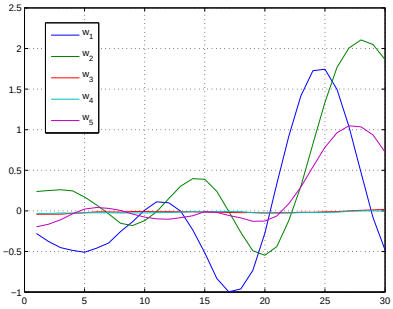
Simulation parameters The feedforward filters w_m were discretized to have 30 sample points. The decoding filters h_m had length 60 after discretization. The allowed delay was $N_d = 30$, and N_p was set to 30 as well. The matrix $\mathbf{\Psi}$ was the unity matrix. We were thus searching for h in the standard basis, and 60 points had to be determined. The noise current was obtained from $In(t) = \sum_{f:t_n^f<t} A\exp((t-t_n^f)/\tau)$ where the times t_n^f were obtained from a Poisson process with rate 0.35. The amplitude $A = 0.6$ and $\tau = 8$. The stepsize μ for the learning of the encoding filters was 0.001. The stepsize for the decoding filters $\mu_h = 0.005$.



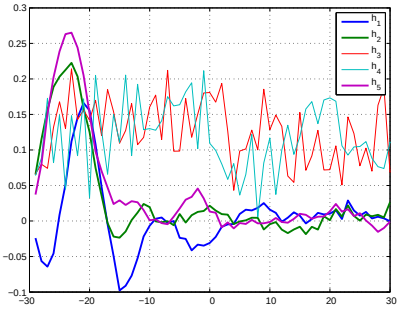
(a) Encoding filter, beginning



(b) Decoding filter, beginning

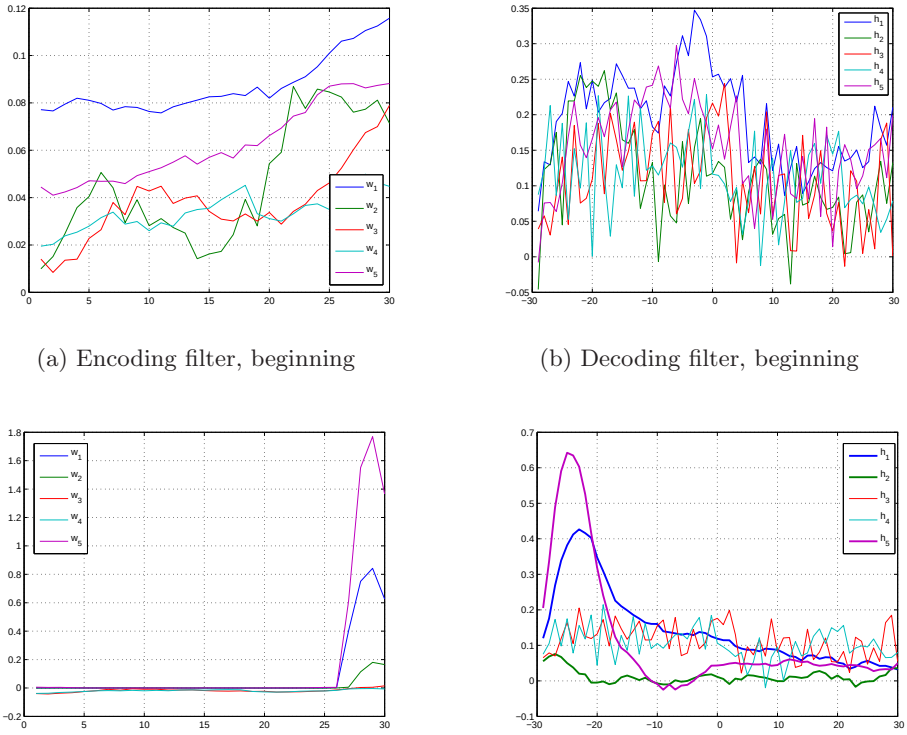


(c) Encoding filter, end



(d) Decoding filter, end

Figure 6.4: Learning of five encoding and decoding filters for input as defined in Figure 6.3, energy consumption is not punished. Panel (a) in the first row shows the encoding filters w_1 to w_5 towards the beginning of the learning process. They were all initialized to the constant $w_m = 0.2$. Panel (b) shows the corresponding decoding filters h_1 to h_5. For initialization, the sample points of h_m were drawn independently from a normal distribution with standard deviation 0.1. The second row shows the filters after learning. Panel (c) shows that three filters have attained a clear shape while the other two are close to zero. The corresponding neurons cannot emit spikes. The filters have a similar shape but are shifted with respect to each other. Panel (d) depicts the corresponding decoding filters. The thicker lines indicate the decoding filters which are relevant in the sense that the corresponding neurons emit spikes.



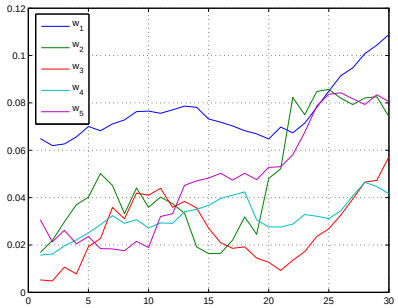
(a) Encoding filter, beginning

(b) Decoding filter, beginning

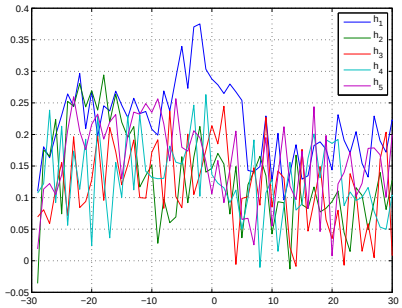
(c) Encoding filter, end

(d) Decoding filter, end

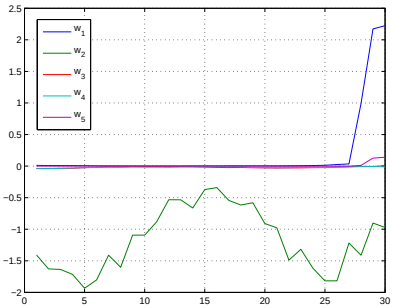
Figure 6.5: Learning of five encoding and decoding filters for input as defined in Figure 6.3, the energy punishment J_1 is weighted by $\alpha = 0.05$. See Figure 6.4 for an explanation of the panels (a) to (d).



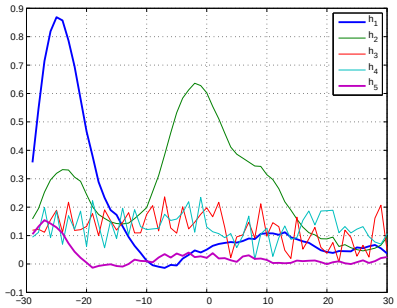
(a) Encoding filter, beginning



(b) Decoding filter, beginning

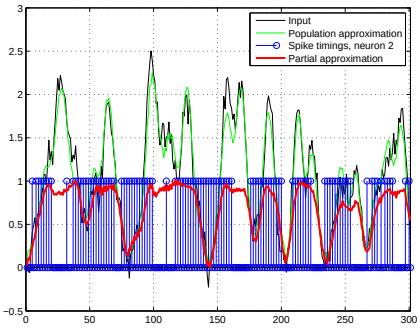


(c) Encoding filter, end

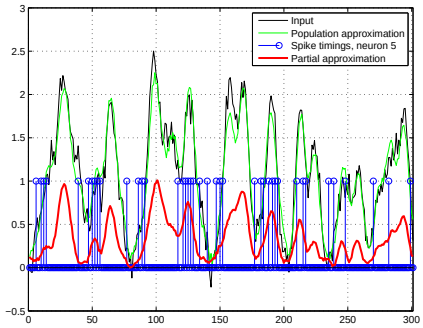


(d) Decoding filter, end

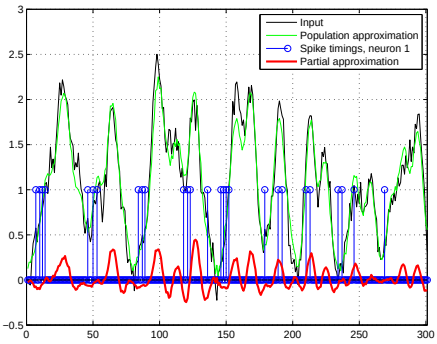
Figure 6.6: Learning of five encoding and decoding filters for input as defined in Figure 6.3, the energy punishment J_1 is weighted by $\alpha = 0.1$. See Figure 6.4 for an explanation of the panels (a) to (d).



(a) Coding of neuron 2

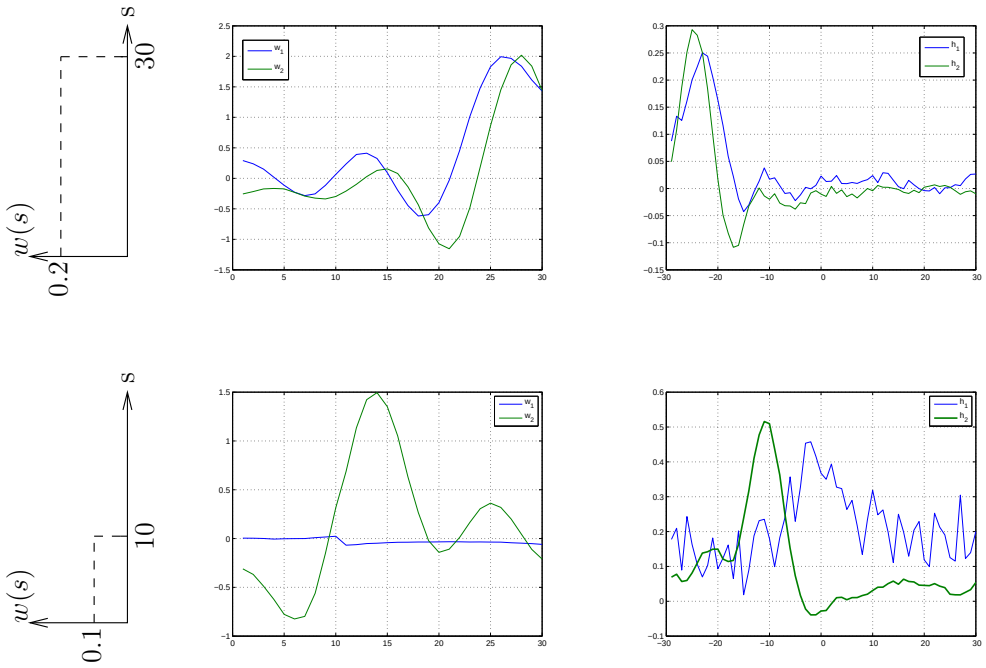


(b) Coding of neuron 5



(c) Coding of neuron 1

Figure 6.7: The representation of input of Figure 6.3 with neurons that have encoding filters and decoding filters as shown in Figure 6.4. Each panel shows the input $x(t)$ in black and the approximation $\hat{x}$ (see Equation 6.2) by means of the population of neurons in green. For the neurons which have nonzero feedforward encoding filters w_m, each panel shows the spike timings t_m^f in blue and the partial reconstruction $\sum_f h_m(t - t_m^f)$ in red. The addition of the red curves (partial approximation) of each panel gives the curve in green (population approximation). We see that the approximation of the input by means of the population activity is accurate. The partial approximations show that each neuron has a different role in the population code. Neuron 2 codes for the local mean of the input (the mean of the 300 sample points is 1). Neuron 5 codes for a coarse resolution and neuron 1 for finer details. Note that the input $x(t)$ has also two different scales. Further, neuron 5 and 1 fire bursts while neuron 2 features more a firing rate code.



Initial condition Encoding filter Decoding filter

Figure 6.8: Learning encoding and decoding filters for input of Figure 6.3 for two neurons without punishment of energy consumption. The results for two different initial conditions for w are shown. For initialization of the decoding filters h, their sample points were drawn independently from a normal distribution with standard deviation 0.1. The top row depicts the case of the same initial conditions as in Figure 6.4. Filters with large delay result. The bottom row shows an initial conditions with zero values at late time points of w. A filter with a much shorter delay results.

6.3.3 With lateral connections: Divergent connections

We consider here the representation of the input shown in Figure 6.3 by means of the neural activity of the network of neurons in Figure 6.9. The difference to the previous simulation in Section 6.3.2 is that the neurons are synaptically coupled with each other.

The learning rules of Section 6.2.1 and Section 6.2.2 are used for the learning of the feedforward encoding filters w_m and the weights of the synaptic coupling c^{mj}, respectively. No energy constraints are imposed ($\alpha = 0$). For the decoding filters h_m, the learning rule of Section 6.2.3 is used.

Figure 6.10 shows the encoding filters and decoding filters both towards the beginning and the end of the learning. Figure 6.11 shows the evolution of the lateral weights with learning.

The learned decoding filters show that the three neurons code for different aspects of the input. This learned differentiation is further illustrated in Figure 6.12 which shows the neural representation of an excerpt of input $x(t)$. Neuron 2 represents the input at a coarse resolution. Neuron 1 codes for an intermediate resolution and represents significant features of the input. Neuron 3 codes for fine details and leads to a sharpening of the peaks in the reconstruction.

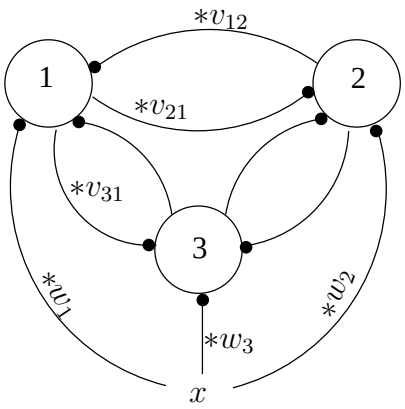
Simulation parameters are the same as in the previous simulation without lateral connections in the encoding network, see Section 6.3.2. In total, 1750 learning rounds were performed where input x had in each round length 5000.

6.4 Discussion

We have have dealt here with the second research question of Section 4.3.2. The input data is represented by multiple neurons which may have lateral connections. We presented an online learning rule for the unknown feedforward encoding filters, the lateral encoding filters, and the decoding filters: We first formulated the data representation problem as an optimization problem. Then, we found an iterative optimization scheme from which the online rule was derived.

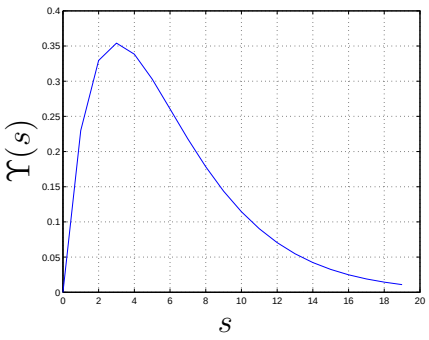
Learning rule for feedforward encoding filters w_m

We comment here on the term $y_i^m(s, f)$ which occurs in Equation (6.11). It describes the influence of $w_m(s)$ on the f-th spike timing t_i^f of neuron i. If there is no (direct or indirect) connection between neuron m and neuron i, the recursion shows that $y_i^m(s, f) = 0$ for all f. The term $y_i^m(s, f)$ becomes nonzero if



Encoding network

$v_{mj}(s) = c^{mj}\Upsilon(s)$



Form of the lateral encoding filter

Figure 6.9: Learning the representation of input x by means of a network with recurrent synaptic connections. The update rules of Section 6.2 are used to learn the feedforward filters w_1 to w_3, the weights c^{mj} of the lateral encoding filters v_{mj}, and the decoding filters h_1 to h_3. The decoding filters are not shown in the schematic.

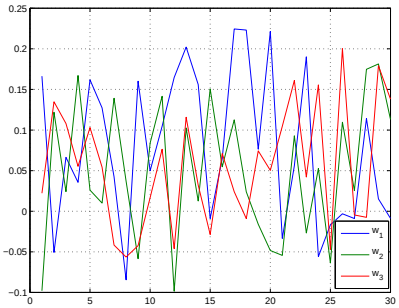
$\dot{v}_{im}(t_i^f - t_m^\nu)$, for some spike timing t_m^ν of neuron m prior to t_i^f, is nonzero. Without such a pair of spike events of neuron m and i, y_i^m decays to zero due to the Γ factor.

The remaining elements of the learning rule for data representation with multiple neurons remain the same as in case of a single neuron, which we have treated in Chapter 5.

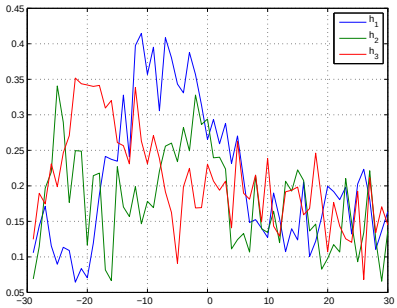
Learning rule for lateral filters v_{mj}

We distinguish three cases which can happen in the learning rule for a lateral filter $v_{mj}(s)$. The first case is the change in v_{mj} due to a spike of neuron m. The second case is the change in v_{mj} due to a spike of a neuron $i \neq m, \neq j$. The third case happens when neuron j is spiking.

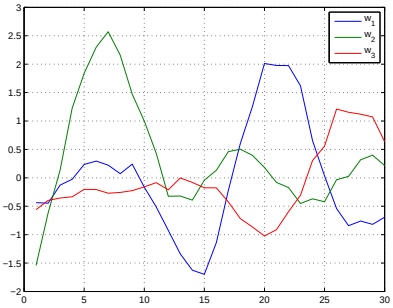
If neuron m is emitting its f-th spike at t_m^f, the term $\mathbf{z_m^{mj}}(f)$ of Equation (6.18) enters together with $\bar{e}(t_m^f)$ into the update. The role of $\bar{e}$ has been discussed previously in Section 5.2.3. Equation (6.18) shows that $\mathbf{z_m^{mj}}(f)$ depends on its past and the other quantities $\mathbf{z_n^{mj}}(f)$, $n \neq m$. But it depends also on the "external input" $\boldsymbol{\Upsilon}(t_m^f - t_j^\nu)$. This external input depends also on the spike timing t_j^ν of neuron j prior to the spike event of neuron m.



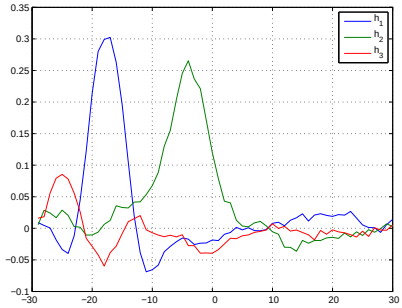
(a) Encoding filter, beginning



(b) Decoding filter, beginning



(c) Encoding filter, end



(d) Decoding filter, end

Figure 6.10: Learning the recurrent network of Figure 6.9 for input as defined in Figure 6.3. Top row: Encoding and decoding filters towards the beginning of the learning. Bottom row: After learning. Energy consumption is not punished. For initialization of the learning rule, after discretization, each sample point of the feedforward encoding filters w_m was drawn independently from an uniform distribution on [0 0.2]. The sample points of the decoding filters h_m were drawn independently from a normal distribution with standard deviation 0.1.

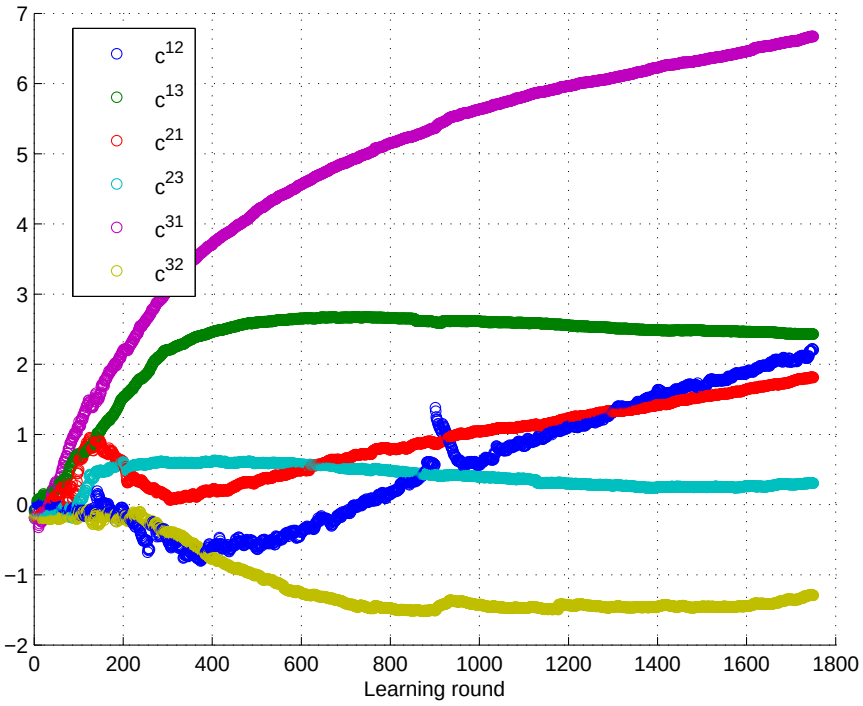
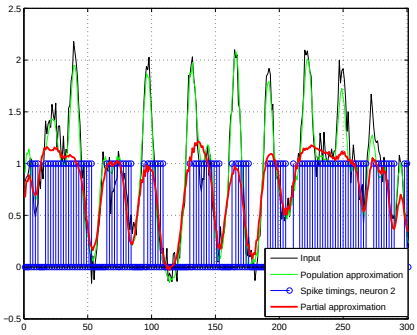
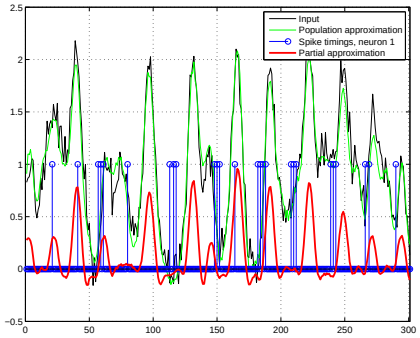


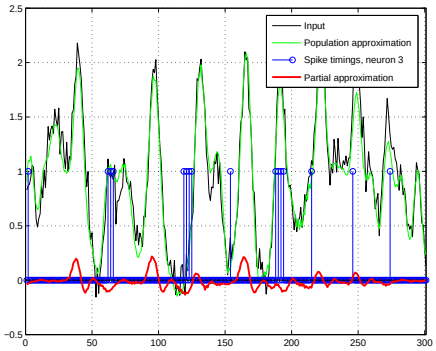
Figure 6.11: Lateral weights evolution. All the weights c^{mj} of the lateral encoding filters v_{mj} were initialized with zero. The curves show several regions of interest: There is rapid growth in the beginning. Then, the growth slows down for most curves, and in some cases, the absolute value of the weight decreases. This leads for c^{12} (shown in blue) to a period where neuron 2 inhibits neuron 1 before switching to an excitatory behavior.



(a) Coding of neuron 2



(b) Coding of neuron 1



(c) Coding of neuron 3

Figure 6.12: The representation of input of Figure 6.3 with neurons that have encoding filters and decoding filters as shown in Figure 6.10, and lateral weights as shown in Figure 6.11 at the end of the learning process. Each panel shows the input $x(t)$ in black and the approximation $\hat{x}$ (see Equation 6.2) by means of the population of neurons in green. Each panel shows also the spike timings t_m^f in blue and the partial reconstruction $\sum_f h_m(t - t_m^f)$ in red. The addition of the red curves (partial approximation) of each panel gives the curve in green (population approximation). We see that the approximation of the input by means of the population activity is accurate. The partial approximations show that each neuron has a different role in the population code. Neuron 2 represents the input at a coarse resolution. Neuron 1 codes for an intermediate resolution and represents significant features of the input. Neuron 3 codes for fine details and leads to a sharpening of the peaks in the reconstruction. Note that the input $x(t)$ has also two different scales. Further, neuron 1 and 3 fire bursts while neuron 2 features more a firing rate code.

If neuron i is emitting its f-th spike at t_i^f, the term $\mathbf{z_i^{mj}}(f)$ of Equation (6.19) enters together with $\bar{e}(t_i^f)$ into the update. The quantity $\mathbf{z_i^{mj}}(f)$ decays to zero because of the Γ factor. It can obtain a non-zero value if $\dot{v}_{im}(t_i^f - t_m^\nu)$, for a pair of spike events of neuron i and m, is large enough.

The case of neuron j is similar as for the neurons $i \neq m$. However, an additional term due to the energy constraint enters into the update. This term pushes $v_{mj}(s)$ towards zero.

Simulations without lateral connections in the encoding network

We discuss the origin of the observed suppression of some of the feedforward encoding filters w in the simulations of Section 6.3.2.

We have seen in Section 5.2.3 that the plasticity of the encoding filter w depends on the decoding filter h over $\bar{e}$. The encoder exerts also influence on the learning of the decoder since it provides the learning data by triggering spikes. This can be seen by inspecting Equation 5.26 where vector $\mathbf{y}$ is zero if no spikes happen during $[n - N_p \; n + N_d]$. If the learning data for h is consistent with the input which triggered the spike both encoder and decoder develop. Else, the decoder remains poorly developed which hinders the development of the encoding filter over $\bar{e}$. Poorly developed encoding filters provide bad learning data so that a destructive feedback circle emerges which may be the reason for the observed suppression.

Key to the development of the encoding filters are the "spikes at the right time". From the above discussion, we see that the presence of the (hypothetical) decoder, which provides a running commentary on the spike trains, leads to a competition of the encoding filters for the spike timing.

Simulations with lateral connections in the encoding network

First, we compare the learned decoding filters where the network had lateral connections with the learned decoding filters for the network without lateral connections. Then, we analyze the effect of the lateral connections on the encoding.

Figure 6.13 shows the comparison of the decoding filters. We are interested in the shape of the filters. We have seen in Figure 6.8 that different initial conditions may lead to filters with different delays. Hence, we have shifted the filters which were obtained for the network without lateral connections so that they maximally overlap with the filters which were learned for the network with lateral connections. The figure shows that for the intermediate resolution, which codes the significant

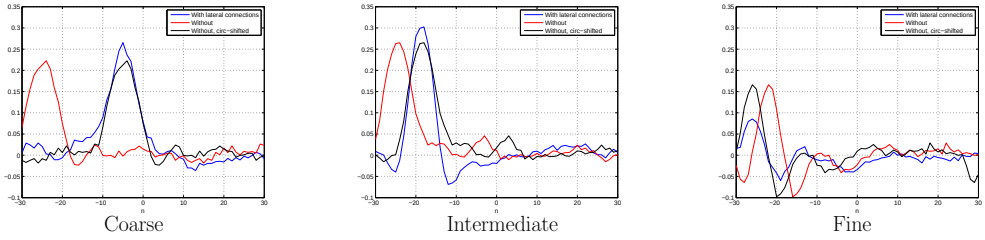


Figure 6.13: Decoding filters learned for a network with lateral connections (blue) versus decoding filters learned for a network without lateral connections (red and shifted version in black).

features, the decoding filter learned for the network with lateral connections is more sharply tuned than the decoding filter that is obtained for the network without lateral connections. The same holds for the finest scale. For the coarse scale, however, we observe a broader tuning of the decoding filter.

Figure 6.14 shows the effect of the lateral connections on the encoding. Holding both the encoding and decoding filters fixed, we encoded the same input once with the original values of c^{mj}, and once with all the lateral weights c^{mj} set to zero. The figure shows that without lateral connections, the neurons lack input current in order to trigger enough spikes for the accurate representation of larger excursions in the input. The lateral connections provide thus an internal amplification of the input (network gain). The figures show as well that the excitatory connection between neuron 2 (coding for the coarse resolution) and neuron 1 (coding for the intermediate resolution) is at the origin for the network gain.

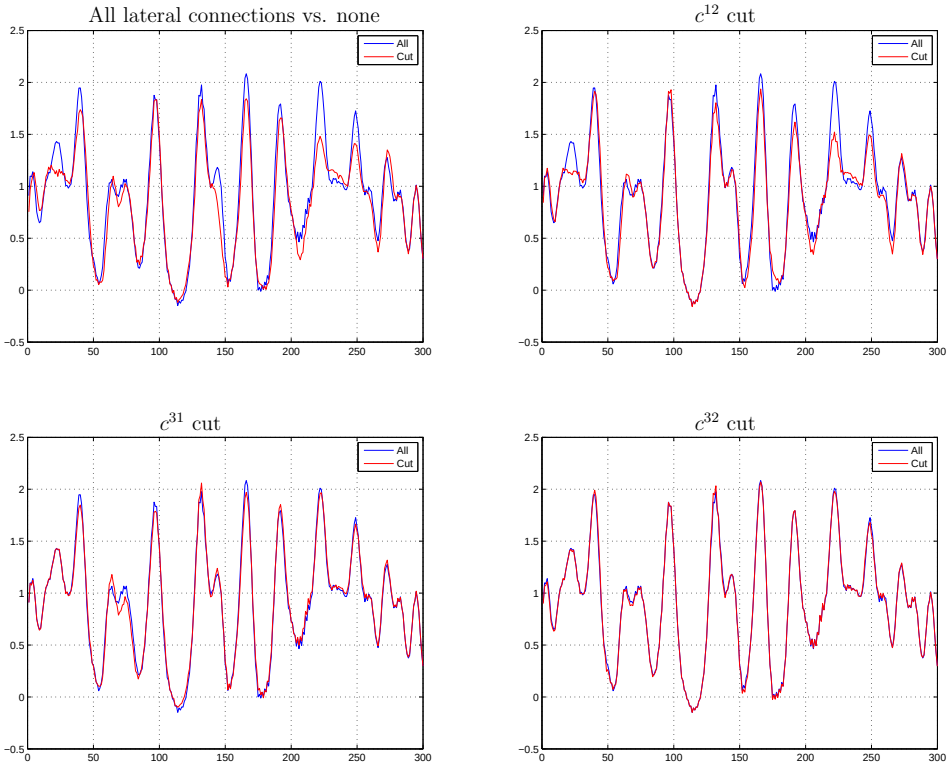


Figure 6.14: Illustrating the effect of lateral connections on the encoding. Cutting the connection from neuron 2 to neuron 1 accounts for most of the differences which occur in the encoding when all the connections are cut. Cutting the connections onto neuron 3 does not effect the encoding to a large extent.

6.5 Appendix: Calculation of the functional derivatives

Functional derivative $\delta J_e/\delta w_m(s)$

We calculate here the functional derivative for $m = 1$. Compared to the case of a single neuron which we have treated in the Appendix Section 5.5, a perturbation of $w_1(s)$ to $w_1(s) + \epsilon\varphi_1(s)$ may result in a change of the spike timings of all other neurons. This may happen when they receive lateral input from neuron 1. As in the Appendix Section 5.5, we make for the perturbation δt_i^f of the f-th spike of neuron i the ansatz

$$\begin{aligned} \delta t_i^f &= \epsilon a_i^1(f) + O(\epsilon^2) && (6.26)\\ &= \epsilon \int_0^{T_w} y_i^1(s,f)\varphi_1(s)\mathrm{d}s + O(\epsilon^2) && (6.27) \end{aligned}$$

The calculations of the Appendix Section 5.5 show that

$$\frac{\delta J_e}{\delta w_1(s)} = -\frac{1}{T}\sum_i \sum_f \bar{e}(t_i^f) y_i^1(s,f). \tag{6.28}$$

In the following paragraphs we calculate $y_i^1(f,s)$.

The perturbation δt_1^f The spiking timing $t_1^f > T_w$ is defined over

$$\begin{aligned} \theta &= \eta_0 \exp\left[-\frac{t_1^f - t_1^{f-1}}{\tau}\right] + \int_0^{T_w} x(t_1^f - s) w_1(s)\mathrm{d}s + \\ &\quad \sum_{j\neq 1}\sum_\nu v_{1j}(t_1^f - t_j^\nu) + In_1(t_1^f). \end{aligned} \tag{6.29}$$

After perturbation, the spike timings change by an amount $\delta t_i^f = \epsilon a_i^1(f) + O(\epsilon^2)$ so that

$$\begin{aligned} \theta &= \eta_0 \exp\left[-\frac{t_1^f - t_1^{f-1}}{\tau}\right] \exp\left[-\frac{\epsilon(a_1^1(f) - a_1^1(f-1)) + O(\epsilon^2)}{\tau}\right] \\ &\quad + \int_0^{T_w} x(t_1^f + \epsilon a_1^1(f) + O(\epsilon^2) - s)(w(s) + \epsilon\varphi_1(s))\mathrm{d}s \\ &\quad + \sum_{j\neq 1}\sum_\nu v_{1j}(t_1^f + \epsilon a_1^1(f) - t_j^\nu - \epsilon a_j^1(\nu) + O(\epsilon^2)) \\ &\quad + In_1(t_1^f + \epsilon a_1^1(f) + O(\epsilon^2)) \end{aligned} \tag{6.30}$$

The constant $\epsilon > 0$ can be made arbitrarily small, so that via Taylor series[1] and Equation (6.29) we obtain

$$\begin{aligned} 0 &= \epsilon\Bigg\{\dot{u}_1(t_1^f)a_1^1(f) + a_1^1(f-1)\frac{\eta_0}{\tau}\exp\left[-\frac{t_1^f - t_1^{f-1}}{\tau}\right] - \sum_{j\neq 1}\sum_{\nu}\dot{v}_{1j}(t_1^f - t_j^\nu)a_j^1(\nu) + \\ &\int_0^{T_w} x(t_1^f - 1)\varphi_1(s)\mathrm{d}s\Bigg\} + O(\epsilon^2) \end{aligned} \tag{6.31}$$

We see that $a_1^1(f)$ must satisfy

$$\begin{aligned} a_1^1(f) &= \frac{1}{\dot{u}_1(t_1^f)}\Bigg[-\frac{\eta_0}{\tau}\exp\left[\frac{t_1^f - t_1^{f-1}}{\tau}\right]a_1^1(f-1) - \int_0^{T_w} x(t_1^f - 1)\varphi_1(s)\mathrm{d}s + \\ &\sum_{j\neq 1}\sum_{\nu}\dot{v}_{1j}(t_1^f - t_j^\nu)a_j^1(\nu)\Bigg] \end{aligned} \tag{6.32}$$

Using Equation (6.27), we obtain

$$\begin{aligned} y_1^1(s,f) &= \frac{1}{\dot{u}_1(t_1^f)}\Bigg[-x(t_1^f - 1) + \Gamma(t_1^f, t_1^{f-1})y_1^1(s,f-1) + \\ &\sum_{j\neq 1}\sum_{\nu}\dot{v}_{1j}(t_1^f - t_j^\nu)y_j^1(s,\nu)\Bigg] \end{aligned} \tag{6.33}$$

where

$$\Gamma(t_1^f, t_1^{f-1}) = -\frac{\eta_0}{\tau}\exp\left[\frac{t_1^f - t_1^{f-1}}{\tau}\right]. \tag{6.34}$$

The perturbation $\delta t_i^f,\ i \neq 1$ The spike timing t_i^f is as in Equation (6.29) defined over

$$\begin{aligned} \theta &= \eta_0\exp\left[-\frac{t_i^f - t_i^{f-1}}{\tau}\right] + \int_0^{T_w} x(t_i^f - s)w_i(s)\mathrm{d}s + \\ &\sum_{j\neq i}\sum_{\nu}v_{ij}(t_i^f - t_j^\nu) + In_i(t_i^f). \end{aligned} \tag{6.35}$$

Perturbation of $w_1(s)$ does change t_i^f and t_j^ν but w_i stays the same. The same calculations as for δt_1^f lead to

$$\begin{aligned} y_i^1(s,f) &= \frac{1}{\dot{u}_i(t_i^f)}\Bigg[\Gamma(t_i^f, t_i^{f-1})y_i^1(s,f-1) + \\ &\sum_{j\neq i}\sum_{\nu}\dot{v}_{ij}(t_i^f - t_j^\nu)y_j^1(s,\nu)\Bigg]. \end{aligned} \tag{6.36}$$

[1] We assume that the optional noise current is smooth enough.

Functional derivative $\delta J_E/\delta w_m(s)$

We assume that small changes in the feedforward filter $w_m(s)$ may change the timings of the spikes, but that they do not change the mean firing rate $\bar{f}$ in Equation (6.6). We calculate under that assumption the functional derivative of J_E for $m = 1$. From Equation (6.6), we see that the only term in J_E which depends on w_1 is $||w_1||_1$. Hence

$$\frac{\delta J_E}{\delta w_1(s)} = \text{sign}(w_1(s)). \tag{6.37}$$

Derivative $\delta J_e/\delta \mathbf{c^{mj}}$

We calculate here the functional derivative for $(m, j) = (1, 2)$. As for the case of the feedforward filters w_m, we perturbate $\mathbf{c^{12}}$ to $\mathbf{c^{12}} + \epsilon \mathbf{a^{12}}$, for an $\epsilon > 0$ that can be made arbitrarily small and an arbitrary vector $\mathbf{a^{12}}$, and calculate the corresponding change in J_e. The change in the cost function J_e results from the perturbation in the firing times t_i^f. For the perturbation δt_i^f we make the ansatz

$$\delta t_i^f = \epsilon \mathbf{z_i^{12}}(f)^T \mathbf{a^{12}} + O(\epsilon^2). \tag{6.38}$$

The same reasoning as in the Appendix Section 5.5 leads to

$$\frac{\delta J_e}{\delta \mathbf{c^{12}}} = -\frac{1}{T} \sum_i \sum_f \bar{e}(t_i^f) \mathbf{z_i^{12}}(f). \tag{6.39}$$

In the following paragraphs we calculate $\mathbf{z_i^{12}}(f)$

The perturbation δt_1^f We calculate the perturbation $\delta t_1^f = \epsilon \mathbf{z_1^{12}}(f)^T \mathbf{a^{12}} + O(\epsilon^2)$ which results from the perturbation in $\mathbf{c^{12}}$. By definition of the spike timings, we obtain

$$\begin{aligned}
\theta \;=\; & \eta_0 \exp\left[-\frac{t_1^f - t_1^{f-1}}{\tau}\right] \exp\left[-\frac{\epsilon(\mathbf{z_1^{12}}(f) - \mathbf{z_1^{12}}(f-1))^T \mathbf{a^{12}}}{\tau} + O(\epsilon^2)\right] + \\
& \int_0^{T_w} x(t_1^f - s + \epsilon \mathbf{z_1^{12}}(f)^T \mathbf{a^{12}} + O(\epsilon^2)) w_1(s) \mathrm{d}s + \\
& \sum_{i \neq 1} \sum_\nu v_{1i}(t_1^f + \epsilon \mathbf{z_1^{12}}(f)^T \mathbf{a^{12}} - t_i^\nu - \epsilon \mathbf{z_i^{12}}(\nu)^T \mathbf{a^{12}} + O(\epsilon^2)) + \\
& \epsilon \sum_\nu \boldsymbol{\Upsilon}(t_1^f + \epsilon \mathbf{z_1^{12}}(f)^T \mathbf{a^{12}} - t_2^\nu - \epsilon \mathbf{z_2^{12}}(\nu)^T \mathbf{a^{12}} + O(\epsilon^2))^T \mathbf{a^{12}} \\
& + In_1(t_1^f + \epsilon \mathbf{z_1^{12}}(f)^T \mathbf{a^{12}} + O(\epsilon^2))
\end{aligned} \tag{6.40}$$

The constant $\epsilon > 0$ can be made arbitrarily small, so that via Taylor series[2] and Equation (6.29) we obtain

$$\begin{aligned} 0 &= \epsilon\Bigg\{ \dot{u}_1 \mathbf{z_1^{12}}(f)^T \mathbf{a^{12}} + \frac{\eta_0}{\tau} \exp\left[-\frac{t_1^f - t_1^{f-1}}{\tau} \right] \mathbf{z_1^{12}}(f-1)^T \mathbf{a^{12}} + \\ &\quad \sum_\nu \boldsymbol{\Upsilon}(t_1^f - t_2^\nu)^T \mathbf{a^{12}} - \sum_{i \neq 1} \sum_\nu \dot{v}_{1i}(t_1^f - t_i^\nu) \mathbf{z_i^{12}}(\nu)^T \mathbf{a^{12}} \Bigg\} + O(\epsilon^2) \end{aligned} \quad (6.41)$$

We see that $\mathbf{z_1^{12}}(f)^T \mathbf{a^{12}}$ must satisfy

$$\begin{aligned} \mathbf{z_1^{12}}(f)^T \mathbf{a^{12}} &= \frac{1}{\dot{u}_1(t_1^f)} \Bigg[-\sum_\nu \boldsymbol{\Upsilon}(t_1^f - t_2^\nu) + \Gamma(t_1^f, t_1^{f-1}) \mathbf{z_1^{12}}(f-1) + \\ &\quad \sum_{i \neq 1} \sum_\nu \dot{v}_{1i}(t_1^f - t_i^\nu) \mathbf{z_i^{12}}(\nu) \Bigg]^T \mathbf{a^{12}}, \end{aligned} \quad (6.42)$$

where $\Gamma(t_1^f, t_1^{f-1})$ is defined in Equation (6.34). Hence

$$\begin{aligned} \mathbf{z_1^{12}}(f) &= \frac{1}{\dot{u}_1(t_1^f)} \Bigg[-\sum_\nu \boldsymbol{\Upsilon}(t_1^f - t_2^\nu) + \Gamma(t_1^f, t_1^{f-1}) \mathbf{z_1^{12}}(f-1) + \\ &\quad \sum_{i \neq 1} \sum_\nu \dot{v}_{1i}(t_1^f - t_i^\nu) \mathbf{z_i^{12}}(\nu) \Bigg]. \end{aligned} \quad (6.43)$$

The perturbation $\delta t_i^f,\ i \neq 1$ The spike timing t_i^f was defined in Equation (6.35) as

$$\begin{aligned} \theta &= \eta_0 \exp\left[-\frac{t_i^f - t_i^{f-1}}{\tau} \right] + \int_0^{T_w} x(t_i^f - s) w_i(s)\ uds + \\ &\quad \sum_{j \neq i} \sum_\nu v_{ij}(t_i^f - t_j^\nu) + In_i(t_i^f). \end{aligned} \quad (6.44)$$

Perturbation of v_{12} does change the spike timings t_i^f and t_j^ν but the functions v_{ij} remain the same since $i \neq 1$. Considering this, the same calculations as for $\mathbf{z_1^{12}}(f)$ lead to

$$\mathbf{z_i^{12}}(f) = \frac{1}{\dot{u}_i(t_i^f)} \left[\Gamma(t_i^f, t_i^{f-1}) \mathbf{z_i^{12}}(f-1) + \sum_{j \neq i} \sum_\nu \dot{v}_{ij}(t_i^f - t_j^\nu) \mathbf{z_j^{12}}(\nu) \right] \quad (6.45)$$

[2] We assume that the optional noise current is smooth enough.

Derivative $\delta J_E/\delta \mathbf{c^{mj}}$

We assume that small changes in the lateral filter $v_{mj}(s)$ may change the timings of the spikes, but that they do not change the mean firing rate $\bar{f}$ in Equation (6.6). We calculate under that assumption the functional derivative of J_E for $(m, j) = (1, 2)$. From Equation (6.6), we see that the only term in J_E which depends on v_{12} is $\bar{f}_2||v_{12}||_1$. Hence

$$\frac{\partial J_E}{\partial c_n^{12}} = \bar{f}_2 \int_0^{T_v} \text{sign}(v_{12}(t))\Upsilon_n(t)\text{d}t, \tag{6.46}$$

and in vectorized form

$$\frac{\delta J_E}{\delta \mathbf{c^{12}}} = \bar{f}_2 \int_0^{T_v} \text{sign}(v_{12}(t))\boldsymbol{\Upsilon}(t)\text{d}t. \tag{6.47}$$

Chapter 7

The Case of a Canonical Integrator

7.1 Recapitulation of the third research question

We deal in this chapter with the third research question of Section 4.3.2. The goal is to learn a representation of simple sensory input where the encoding happens via a single canonical integrator neuron (also called quadratic integrate and fire neuron, see the review in [Izhikevich, 2006]). The behavior of this kind of neuron model is described by the differential equation given in Equation (3.17). The canonical integrator neuron can operate either in monostable or bistable mode, see Figure 3.8. Figure 7.1 visualizes the research task.

Encoding of the input $x(t)$ is done by firing single spikes. The spike timings constitute the output signal of the encoder. For a proper representation of the input by means of the spike timings, the input should be reconstructible from the neural activity. Learning the representation is based on the minimization of the reconstruction error, with an added penalty to minimize energy consumption during the encoding process. The research task is to formulate the optimization problem and find an iterative optimization rule.

7.2 Cost functional for data representation

7.2.1 Part one of the cost functional: reconstruction error

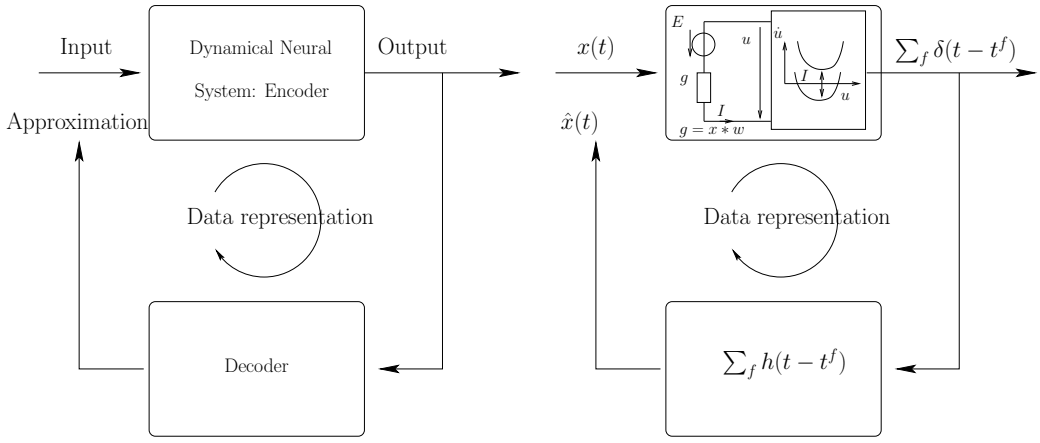
The assumed neuron model follows the differential equation (see the review in [Izhikevich, 2006])

$$\dot{u} = a_1(u - u_R)(u - u_T) - a_2 g(t)(u - E) \tag{7.1}$$

for $u < u_p$. Refer to Equation (3.17) for an explanation of the constants $a_1 > 0$, $a_2 > 0$, u_R, and $u_T > u_R$. In the equation is $g(t)$ given by

$$g(t) = \int_0^{T_w} x(t - s)w(s)\mathrm{d}t. \tag{7.2}$$

If the voltage u passes the peak value u_p at time t^f, we do the reset $u(t^f + \triangle) \leftarrow c$. The small delay $\triangle$ is the time during which the voltage returns to c. This part of the trajectory of u is not simulated in the canonical integrator neuron. The time t^f is the spike timing. The input current $I = -g(u - E)$ arises from an ion channel with reversal potential E (introduced in Section 3.1.1). The conductance g of that channel depends on the input $x(t)$ via the convolution with the unknown encoding filter w.



(a) General scheme (b) Scheme in this chapter

Figure 7.1: (a) Input data x is transformed by a dynamical neural system. This encoding process should be such that the input can be decoded from the output. The energy consumption during the encoding process should also be minimized. The encoder and decoder together provide data representation. (b) Here, we encode by means of the canonical integrator model, see Equation (3.17). The input $x(t)$ affects over an unknown filter w the conductance g of an ion channel over which current $I = g(V - E)$, see Equation (3.2), enters into the neuron. The input current I may give rise to a self-amplification of voltage u leading to an action potential. The time t^f of an action potential is recorded and used for the decoding by means of the unknown decoding filter h. Learning rules for the unknown encoding filter w and decoding filter h are derived from the minimization of the mean squared reconstruction error, with an added penalty to minimize energy consumption during the encoding process.

The reconstruction $\hat{x}$ is sought under the form

$$\hat{x}(t) = \sum_{f:t-T_p<t^f<t+T_d} h(t-t^f), \tag{7.3}$$

which is the same as in Equation (5.2) of Chapter 5.

The first part of the cost functional which is due to the reconstruction error is then given by

$$J_e(h,w) = \frac{1}{2T}\int_0^T (\hat{x}(t)-x(t))^2 \mathrm{d}t, \tag{7.4}$$

where T is a fixed time horizon.

7.2.2 Part two of the cost functional: energy consumption

We measure here the energy consumption by means of the total ion load per time T of the neuron (see Section 3.1.2),

$$J_E(w) = \frac{1}{T}\int_0^T |I(t)|\mathrm{d}t. \tag{7.5}$$

7.3 Learning rule

7.3.1 Encoding filter w

Key to the update rule for w is the calculation of the functional derivative $\delta J/\delta w(s)$, where the total J is the sum of the reconstruction error J_e, defined in Equation (7.4), and of the energy cost J_E, defined in Equation (7.5). In the Appendix Section 7.5, we calculate the functional derivative for J_e and the energy costs J_E. The decoding filter h is held fixed. The functional derivative of J_e is

$$\frac{\delta J_e}{\delta w(s)} = -\frac{1}{T}\sum_f \bar{e}(t^f) y_f(s), \tag{7.6}$$

where

$$\bar{e}(t^f) = \int_{t^f-T_d}^{t^f+Tp} (\hat{x}(t)-x(t))\dot{h}(t-t^f)\mathrm{d}t, \tag{7.7}$$

$$\begin{aligned} y_f(s) = \frac{1}{\dot{u}(t^f)}\Big[a_2 \int_{t^{f-1}+\triangle}^{t^f} x(\tau-s)(u(\tau)-E)U(\tau,t^f)\mathrm{d}\tau \\ +\dot{u}(t^{f-1}+\triangle)U(t^{f-1}+\triangle,t^f)y_{f-1}(s)\Big], \end{aligned} \tag{7.8}$$

$$U(t^1,t^2) = \exp\left[2a_1\int_{t^1}^{t^2} u(s) - \frac{u_R+u_T}{2} - \frac{a_2}{2a_1}g(s)\mathrm{d}s\right], \tag{7.9}$$

The functional derivative of J_E is

$$\frac{\delta J_E}{\delta w(s)} = \frac{1}{T}\int_0^T \mathrm{sign}(I(t))x(t-s)(u(t)-E)\mathrm{d}t. \tag{7.10}$$

We propose the following online rule: After spike k at t^k, update the encoder by

$$w_k(s) = w_{k-1}(s) + \mu\left(\bar{e}(t^k)y_k(s) - \alpha\frac{\delta\check{J}_E}{\delta w(s)}\right) \tag{7.11}$$

where $\mu > 0$ is the step size and α weights the influence of the energy constraint. The algorithm is initialized with $y_0 = 0$ and $w = w_0$. The term $\delta\check{J}_E/\delta w(s)$ is obtained from $\delta J_E/\delta w(s)$ by integrating not from 0 till T, but from t^{k-1} till t^k, i.e. by

$$\frac{\delta\check{J}_E}{\delta w(s)} = \int_{t^{k-1}}^{t^k} \mathrm{sign}(I(t))x(t-s)(u(t)-E)\mathrm{d}t. \tag{7.12}$$

7.3.2 Decoding filter h

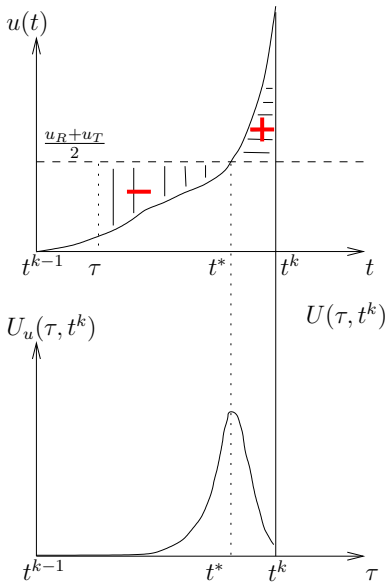
Exactly the same algorithm as in Section 5.2.2 can be used for the learning of the decoding filter h.

7.4 Discussion

We discuss here the learning rule for the encoding filter w and compare it to the learning rule of Section 5.2.1, where we have presented the learning rule for a single spike response neuron model.

We point out two differences. The first difference lies in the fact that here, the input current is given by $I = g(u - E)$, where the encoding filter w determines g by a convolution with the input x. In Section 5.2.1, the input current I was independent from the membrane voltage u, and given directly by the convolution of the encoding filter w with the input x. The dependency of the input current on u enters into the update rule via the $(u(\tau) - E)$ term of Equation (7.8).

The second difference has its roots in the fact that, in the canonical integrator neuron, the onset of the spike is also simulated. In the spike response model, which we have used for the learning rule of Section 5.2.1, the simulation of the action potential is completely omitted. This leads to the presence of the function $U(\tau, t^k)$ in Equation (7.8). Figure 7.2 shows that $U(\tau, t^k)$ can be factorized into the product of $U_u(\tau, t^k)$ with $U_g(\tau, t^k)$, and that $U_u(\tau, t^k)$ attains a maximum at $\tau = t^* < t^k$. This means that the input $x(t^* - s)$ enters most strongly weighted



$$U(\tau, t^k) = \exp\left[2a_1 \int_\tau^{t^2} u(s) - \frac{u_R+u_T}{2} \mathrm{d}s\right] \exp\left[-a_2 \int_\tau^{t^2} g(s)\mathrm{d}s\right] = U_u(\tau, t^k) \cdot U_g(\tau, t^k)$$

Figure 7.2: Explanation of the U function in Equation (7.8). The U function has two parts. The first part U_u is determined by the voltage trajectory u. The second part U_g is determined by the trajectory of g. We discuss here only U_u. The area where u is below $(u_R + u_T)/2$, which is the bifurcation point for u in the canonical integrator neuron, is subtracted from the area where u is above $(u_R + u_T)/2$. The areas which are compared are determined by τ. The closer τ comes to the spike timing t^k, the more the negative area shrinks. The plot shows that the U function attains a maximum value at $t = t^*$. The time t^* is the time where u crosses $(u_R + u_T)/2$. With increasing values of τ, the positive area shrinks as well, and U decreases.

into the update of w, when we neglect the role of $U_g(\tau, t^k)$. It is furthermore not only $x(t^* - s)$, which is used for the update, but a local average around $\tau = t^*$. For the spike response model, it was only $x(t^k - s)$, the input prior to the spike timing t^k, which entered into the update.

7.5 Appendix: Calculation of the functional derivatives

Functional derivative $\delta J_e/\delta w(s)$

The cost functional J_e has the same form as in Chapter 5, but the involved spike timings are calculated differently. We perturbate $w(s)$ as in the Appendix Section 5.5 to $w(s) + \delta w(s)$ where

$$\delta w(s) = \epsilon\varphi(s), \tag{7.13}$$

for a small constant $\epsilon > 0$ and a sufficiently smooth, but else arbitrary function $\varphi(s)$. Repeating the same steps as in the Appendix Section 5.5 leads to Equation(5.44), i.e.

$$\frac{\delta J_e}{\delta w(s)} = -\frac{1}{T}\sum_f \bar{e}(t^f) y_f(s), \tag{7.14}$$

where $y_f(s)$ is here however different from Chapter 5. We calculate in the next paragraphs $y_f(s)$.

The perturbation $g_w(t)$ The conductance $g(t)$ was defined as

$$g(t) = \int_0^{T_w} x(t-s)w(s)\mathrm{d}t. \tag{7.15}$$

The perturbation of $w(s)$ to $w(s) + \epsilon\varphi(s)$ leads thus to a perturbation of $g(t)$ to $g(t) + \epsilon g_w(t)$ where

$$g_w(t) = \int_0^{T_w} x(t-s)\varphi(s)\mathrm{d}t. \tag{7.16}$$

Small signal behavior of u The voltage u of the canonical integrator neuron satisfies the differential equation

$$\dot{u} = a_1(u-u_R)(u-u_T) - a_2 g(t)(u-E), \tag{7.17}$$

where the ion current I is $I = g(u-E)$. Let us assume g is perturbated to $g_p(t) = g(t) + \epsilon g_w(t)$, where $\epsilon > 0$. We are interested in the corresponding perturbation in u. If we make the ansatz $u_p(t) = u(t) + \epsilon u_w(t) + O(\epsilon^2)$, we obtain the following differential equation for u_w

$$\dot{u}_w = 2a_1\left(u - \frac{u_R+U_T}{2} - \frac{a_2}{2a_1}g\right)u_w - a_2 g_w(u-E). \tag{7.18}$$

Inserting into the differential equation shows that

$$u_w(t) = AU(t_0, t) + \int_{t_0}^{t} -a_2 g_w(u - E)U(\tau, t)\mathrm{d}\tau, \tag{7.19}$$

where $A = u_w(t_0)$ and

$$U(\tau, t) = \exp\left[2a_1 \int_{\tau}^{t} u(s) - \frac{u_R + u_T}{2} - \frac{a_2}{2a_1} g(s)\mathrm{d}s\right], \tag{7.20}$$

is a solution.

The perturbation δt^f and the calculation of $y_f(s)$ We calculate here the perturbation in t^f which results from a perturbation of g into $g_p(t) = g(t) + \epsilon g_w(t)$. The expression for g_w of Equation (7.16) leads then to $y_f(s)$.

We make for the perturbated spike timing t_p^f the ansatz

$$t_p^f = t^f + \epsilon b_f + O(\epsilon). \tag{7.21}$$

We want to calculate b_f and assume b_{f-1} to be known. For simplicity of notation, we set here $f = 2$. Figure 7.3 visualizes the setting.

By definition of the spike timings, the perturbated voltage trajectory satisfies

$$u_{\text{peak}} - c = \int_{t_p^1 + \triangle}^{t_p^2} \dot{u}_p(t)\mathrm{d}t. \tag{7.22}$$

Using $u_p(t) = u(t) + \epsilon u_w(t) + O(\epsilon^2)$, we have

$$\begin{aligned} u_{\text{peak}} - c &= \int_{t_p^1 + \triangle}^{t_p^2} \dot{u}(t) + \epsilon \dot{u}_w + O(\epsilon^2)\mathrm{d}t & (7.23) \\ &= \int_{t^1 + \triangle}^{t^2} \dot{u}(t)\mathrm{d}t - \int_{t^1 + \triangle}^{t_p^1 + \triangle} \dot{u}(t)\mathrm{d}t + \int_{t^2}^{t_p^2} \dot{u}(t)\mathrm{d}t + \\ &\quad \epsilon \int_{t_p^1 + \triangle}^{t_p^2} \dot{u}_w(t)\mathrm{d}t + O(\epsilon^2), & (7.24) \end{aligned}$$

where we continue integrating $u(t)$ till time t_p^2 (shown as dashed curve in Figure 7.3). Since the unperturbated trajectory $u(t)$ satisfies

$$u_{\text{peak}} - c = \int_{t^1 + \triangle}^{t^2} \dot{u}(t)\mathrm{d}t, \tag{7.25}$$

we have

$$\begin{aligned} 0 &= -\int_{t^1+\triangle}^{t_p^1+\triangle} \dot{u}(t)\mathrm{d}t + \int_{t^2}^{t_p^2} \dot{u}(t)\mathrm{d}t + \\ &\quad \epsilon \int_{t_p^1+\triangle}^{t_p^2} \dot{u}_w(t)\mathrm{d}t + O(\epsilon^2) \qquad (7.26) \\ &= -\int_{t^1+\triangle}^{t_p^1+\triangle} \dot{u}(t)\mathrm{d}t + \int_{t^2}^{t_p^2} \dot{u}(t)\mathrm{d}t + \\ &\quad \epsilon \int_{t_p^1+\triangle}^{t^2} \dot{u}_w(t)\mathrm{d}t + \epsilon \int_{t^2}^{t_p^2} \dot{u}_w(t)\mathrm{d}t + O(\epsilon^2) \qquad (7.27) \end{aligned}$$

We make now the approximations

$$\begin{aligned} \int_{t^1+\triangle}^{t_p^1+\triangle} \dot{u}(t)\mathrm{d}t &= \dot{u}(t^1+\triangle)(\epsilon b_1 + O(\epsilon^2)), \qquad (7.28) \\ \int_{t^2}^{t_p^2} \dot{u}(t)\mathrm{d}t &= \dot{u}(t^2)(\epsilon b_2 + O(\epsilon^2)), \qquad (7.29) \\ \int_{t^2}^{t_p^2} \dot{u}_w(t)\mathrm{d}t &= \dot{u}_w(t^2)(\epsilon b_2 + O(\epsilon^2)), \qquad (7.30) \end{aligned}$$

so that

$$0 = \epsilon \left\{ -\dot{u}(t^1+\triangle)b_1 + \dot{u}(t^2)b_2 + u_w(t^2) - u_w(t_p^1+\triangle) \right\} + O(\epsilon^2). \qquad (7.31)$$

Note that $\dot{u}(t^2)$ is the derivative of the extended u at t^2 (shown as dashed curve in Figure 7.3). It corresponds to the derivative from the left of the original $u(t)$.

For the further evaluation of this expression, we need to know $u_w(t_p^1+\triangle)$. For that, we note that on the one hand

$$\begin{aligned} u(t_p^1+\triangle) - c &= u(t^1+\triangle) + \dot{u}(t^1+\triangle)\epsilon b_1 - c + O(\epsilon^2) \qquad (7.32) \\ &= \dot{u}(t^1+\triangle)\epsilon b_1 + O(\epsilon^2). \qquad (7.33) \end{aligned}$$

And on the other hand

$$\begin{aligned} c &= u_p(t_p^1+\triangle) \qquad (7.34) \\ &= u(t_p^1+\triangle) + \epsilon u_w(t_p^1+\triangle) + O(\epsilon^2), \qquad (7.35) \end{aligned}$$

so that

$$\epsilon u_w(t_p^1+\triangle) + O(\epsilon^2) = c - u(t_p^1+\triangle), \qquad (7.36)$$

and hence

$$u_w(t_p^1+\triangle) = -\dot{u}(t^1+\triangle)b_1. \qquad (7.37)$$

When we insert that expression into Equation (7.31), we obtain

$$0 = \epsilon \left\{ \dot{u}(t^2) b_2 + u_w(t^2) \right\} + O(\epsilon^2), \tag{7.38}$$

from where we obtain a necessary condition for b_2 to hold, namely

$$b_2 = -\frac{u_w(t^2)}{\dot{u}(t^2)}. \tag{7.39}$$

With Equation (7.19), using $A = u_w(t_p^1 + \triangle) = -\dot{u}(t^1 + \triangle) b_1$, we obtain

$$b_2 = \frac{1}{\dot{u}(t^2)} \left[b_1 \dot{u}(t^1 + \triangle) U(t_1 + \triangle, t^2) + \int_{t^1+\triangle}^{t^2} a_2 g_w(u - E) U(\tau, t^2) \mathrm{d}\tau \right]. \tag{7.40}$$

If we insert the expression for g_w from Equation(7.16), we obtain after switching the order of integration

$$\begin{aligned} b_2 &= \frac{1}{\dot{u}(t^2)} \Big[b_1 \dot{u}(t^1 + \triangle) U(t_1 + \triangle, t^2) + & (7.41) \\ &\int_0^{T_w} \mathrm{d}s \varphi(s) a_2 \int_{t^1+\triangle}^{t^2} \mathrm{d}\tau x(\tau - s)(u(\tau) - E) U(\tau, t^2) \Big]. & (7.42) \end{aligned}$$

We can introduce the function $y_f(s)$ as in the Appendix Section 5.5 via

$$b_f = \int_0^{T_w} y_f(s) \varphi(s) \mathrm{d}s, \tag{7.43}$$

so that we obtain the recursion

$$\begin{aligned} y_2(s) &= \frac{1}{\dot{u}(t^2)} \Big[a_2 \int_{t^1+\triangle}^{t^2} x(\tau - s)(u(\tau) - E) U(\tau, t^2) \mathrm{d}\tau + & (7.44) \\ &y_1(s) \dot{u}(t^1 + \triangle) U(t_1 + \triangle, t^2) \Big]. & (7.45) \end{aligned}$$

This equation defines y_f for a given y_{f-1}, where $y_0 = 0$.

Functional derivative $\delta J_E / \delta w(s)$

We write J_E as

$$\frac{1}{T} \int_0^T \mathrm{sign}(I(t)) I(t) \mathrm{d}t. \tag{7.46}$$

The current $I(t)$ is given by

$$I(t) = \int_0^{T_w} x(t - s) w(s) (u(t) - E) \mathrm{d}s, \tag{7.47}$$

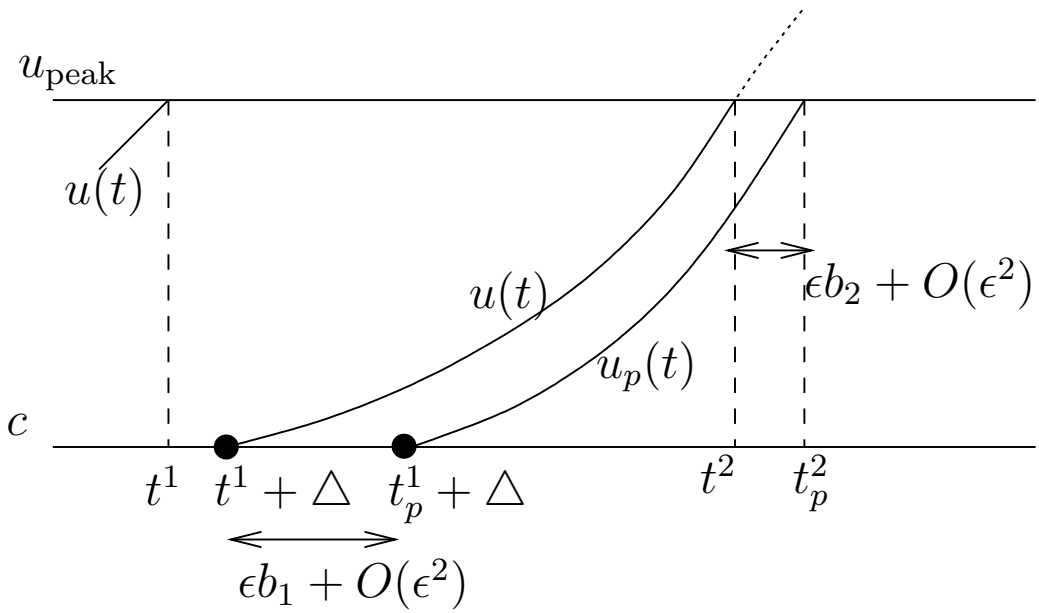


Figure 7.3: We assume that a perturbation in g shifts the spike timing t^1 to the perturbated spike timing t_p^1 so that we restart the integration at the reset voltage c at $t_p^1 + \triangle$ where $\triangle > 0$ is a small delay during which the voltage returns to c (this part of the trajectory of u is not simulated in the canonical integrator neuron). Given b_1, we want to calculate b_2. Note that the values b_k can be > 0 or < 0.

so that

$$\frac{\delta I(t)}{\delta w(s)} = x(t-s)(u(t)-E). \tag{7.48}$$

Hence,

$$\frac{\delta J_E}{\delta w(s)} = \frac{1}{T}\int_0^T \text{sign}(I(t))x(t-s)(u(t)-E)\mathrm{d}t. \tag{7.49}$$

Chapter 8

Conclusions

We give here a summary of thesis, and discuss the main points. Discussion of details can be found at the end of the corresponding chapters.

Summary

We derived learning rules for data representation with dynamical neural systems. The input is encoded by means of a neural system into spike timings, from which the decoder reconstructs the input.

The decoder and the encoding dynamical neural system has free parameters which should be chosen such that the representation of the input by means of the spike timings (neural representation) is as accurate as possible, and, possibly, such that the energy consumption during the encoding process is minimized. The optimal choice of the parameters depends on the properties of the input data.

The derived learning rules (algorithms) determine the free parameters from the input data based on these optimality criteria.

1. The first considered dynamical neural system consisted of a single monostable integrator, which emits a spike whenever the convolution of the input with an encoding filter, given some refractory current from the previous spike, passes a threshold. The resulting spike train is convolved with a decoding filter to obtain an approximation of the input. We allowed a delay for the reconstruction. The encoding and decoding filter are unknown and learned with the derived algorithms from the input (Chapter 5).

2. The second considered dynamical neural system consisted of multiple monostable integrators, possibly interconnected by lateral filters. The input into each neuron had thus two parts: The feedforward part which is obtained through the convolution of the input with a feedforward encoding filter. And the lateral part which is obtained through the summation of the convolutive products of the presynaptic spike trains with the corresponding lateral filters. The spike train of each neuron is decoded as in the single neuron case, and the sum of the decoded spike trains is used as approximation of the input. The feedforward, the lateral and the decoding filters are unknown and learned with the derived algorithms from the input (Chapter 6).

3. The third considered dynamical neural system consisted of a single, bi- or monostable, canonical integrator (with another name: quadratic integrate and fire neuron). The convolution of the input with an encoding filter determines the conductance of an ion channel through which the input current

enters into the neuron. The onset of action potentials are also simulated with this kind of neuron model. The neuron emits a spike when the input current gives rise to a self-amplification of the membrane voltage which pushes the voltage to its peak value, after which the voltage is reset. The resulting spike train is convolved with a decoding filter to obtain an approximation of the input. The encoding and decoding filter are unknown and learned with the derived algorithms from the input (Chapter 7).

We related the learning rules for the canonical integrator to the learning rules for the (formal) monostable integrator in Section 7.4. Furthermore, we explored in Section 5.3 the influence of different punishment schemes for energy consumption on the learning of the encoding and decoding filter. In Section 6.3, we illustrated the use of the derived learning rules for the study of neural representation of sensory input by a population of neurons. After learning, the input was accurately represented by the population of neurons, and the learning rules led to self-organized neural differentiation.

Discussion

We have reviewed in Chapter 2 a number of mathematical data representation methods. They represent the input in a new basis which is adapted to the structure of the data. We describe them as mathematical because the encoding, which calculates the coordinates for the new basis, does not correspond to a dynamical neural system (see Section 3.1). Most of the transforms are linear. In neuroscientific terms, they can be thought of as a linear firing-rate neuron model. In many cases, time dependency of the input is however not modeled. And when it is, the encoding is acausal.

Neural integration denotes the combination of receptor signals to form a neural representation of the sensory stimulus which was sensed by the receptors (Section 3.2). In Chapter 4, we pointed out that data representation is a useful model for neural integration. The application of mathematical data representation methods to natural stimuli was shown in the same chapter to account for the change of neural selectivity which happens in the process of neural integration.

The neuroscientific interpretation of the data representation results is limited by the missing link of the encoding to neuroscience. The development of data representation methods with dynamical neural systems alleviates this which opens the door for more studies into neural representation of sensory input.

Bibliography

J.J. Atick and A.N. Redlich. What does the retina know about natural scenes? *Neural Computation*, 4:449–572, 1992.

David Attwell and Simon B. Laughlin. An energy budget for signaling in the grey matter of the brain. *J Cereb Blood Flow Metab*, 21(10):1133–1145, October 2001.

Stephen A. Baccus and Markus Meister. Retina versus cortex: Contrast adaptation in parallel visual pathways. *Neuron*, 42(1):5–7, April 2004.

Rosario M. Balboa and Norberto M. Grzywacz. Power spectra and distribution of contrasts of natural images from different habitats. *Vision Research*, 43(24): 2527–2537, November 2003.

P.F. Baldi and K. Hornik. Learning in linear neural networks: a survey. *IEEE Transactions on Neural Networks*, 6(4):837–858, 1995.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

H.B. Barlow. Possible principles underlying the transformations of sensory messages. In W. Rosenblith, editor, *Sensory Communication*, pages 217–234. MIT Press, 1961.

Guo-qiang Bi and Mu-ming Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24(1):139–166, 2001.

P. Brehm and R. Eckert. Calcium entry leads to inactivation of calcium channel in paramecium. *Science*, 202::1203–6, 1978.

Stephen M. Carcieri, Adam L. Jacobs, and Sheila Nirenberg. Classification of retinal ganglion cells: A statistical approach. *J Neurophysiol*, 90(3):1704–1713, September 2003.

J.-F. Cardoso and B.H. Laheld. Equivariant adaptive source separation. *IEEE Trans. on Signal Processing*, 44(12):3017–3030, 1996.

Soumya Chatterjee and Edward M. Callaway. Parallel colour-opponent pathways to primary visual cortex. *Nature*, 426(6967):668–671, December 2003.

Y.M. Chino, E.L. Smith, J.H. Kaas, Y. Sasaki, and H. Cheng. Receptive-field properties of deafferentated visual cortical neurons after topographic map reorganization in adult cats. *J. Neurosci.*, 15(3):2417–2433, March 1995.

Susana Cohen-Cory. The developing synapse: Construction and modulation of synaptic structures and circuits. *Science*, 298(5594):770–776, October 2002.

Steven J. Cooper. Donald O. Hebb's synapse and learning rule: a history and commentary. *Neuroscience & Biobehavioral Reviews*, 28(8):851–874, January 2005.

Justin C. Crowley and Lawrence C. Katz. Development of ocular dominance columns in the absence of retinal input. *Nat Neurosci*, 2(12):1125–1130, December 1999.

Justin C. Crowley and Lawrence C. Katz. Early development of ocular dominance columns. *Science*, 290(5495):1321–1324, November 2000.

Dennis M. Dacey and Orin S. Packer. Colour coding in the primate retina: diverse cell types and cone-specific circuitry. *Current Opinion in Neurobiology*, 13(4): 421–427, August 2003.

Dennis M. Dacey, Beth B. Peterson, Farrel R. Robinson, and Paul D. Gamlin. Fireworks in the primate retina: In vitro photodynamics reveals diverse LGN-projecting ganglion cell types. *Neuron*, 37(1):15–27, January 2003.

Yang Dan, Joseph J. Atick, and R. Clay Reid. Efficient coding of natural scenes in the lateral geniculate nucleus: Experimental test of a computational theory. *J. Neurosci.*, 16(10):3351–3362, May 1996.

Aniruddha Das. Cortical maps: Where theory meets experiments. *Neuron*, 47(2): 168–171, July 2005.

P. Dayan and L.F. Abbott. *Theoretical Neuroscience*. The MIT Press, 2001.

Steven H. Devries and Denis A. Baylor. Mosaic arrangement of ganglion cell receptive fields in rabbit retina. *J Neurophysiol*, 78(4):2048–2060, October 1997.

K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks: Theory and Applications*. John Wiley & Sons, 1996.

Barry J. Dickson. Molecular mechanisms of axon guidance. *Science*, 298(5600): 1959–1964, December 2002.

Dawei Dong and Joseph Atick. Statistics of natural time-varying images. *Network: Computation in Neural Systems*, 6(3):345–358, August 1995.

John P. Donoghue. Plasticity of adult sensorimotor representations. *Current Opinion in Neurobiology*, 5(6):749–754, December 1995.

Gidon Felsen and Yang Dan. A natural approach to studying vision. *Nat Neurosci*, 8(12):1643–1646, December 2005.

David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J Opt Soc Am, A*, 4::2379–2394, 1987.

M. Frazier. *An introduction to wavelets through linear algebra*. Springer, 2001.

Robert M. Friedman, Li Min Chen, and Anna Wang Roe. Modality maps within primate somatosensory cortex. *Proceedings of the National Academy of Sciences*, 101(34):12724–12729, August 2004.

Robert C. Froemke and Yang Dan. Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature*, 416(6879):433–438, March 2002.

M.N. Geffen, S.E. Devries, and M. Meister. Retinal ganglion cells can rapidly change polarity from off to on. *PLoS Biol*, 5:640–650, 2007.

I.M. Gelfand and S.V. Fomin. *Calculus of Variations*. Dover, 2000.

U. Gerster, D.G. Stavenga, and W. Backhaus. Na+/k+-pump activity in photoreceptors of the blowfly calliphora: a model analysis based on membrane potential measurements. *Journal of Comparative Physiology A:*, 180(2):113–122, 1997.

W. Gerstner and W. K Kistler. *Spiking Neuron Models*. Cambridge University Press, 2002.

R.M. Gray and L.D. Davisson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004.

R.J. Greenspan. *An Introduction to Nervous Systems*. Cold Spring Harbor, 2007.

M. Gutmann and K. Aihara. Toward data representation with spiking neurons. *Artificial Life and Robotics*, In press, 2008.

M. Gutmann, A. Hyvärinen, and K. Aihara. Learning a nonlinear multi-scale edge detection system from natural stimuli. In *Shanghai International Symposium on Nonlinear Science and Applications*, 2005.

Peter Hancock, Roland Baddeley, and Leslie Smith. The principal components of natural images. *Network: Computation in Neural Systems*, 3(1):61–70, February 1992.

S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 2001.

G. Heidemann. The principal components of natural images revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):822–826, 2006.

Stewart H. C. Hendry and R. Clay Reid. The koniocellular pathway in primate vision. *Annual Review of Neuroscience*, 23(1):127–153, 2000.

J.C. Horton and D.R. Hocking. An adult-like pattern of ocular dominance columns in striate cortex of newborn monkeys prior to visual experience. *J. Neurosci.*, 16(5):1791–1807, March 1996.

Toshihiko Hosoya, Stephen A. Baccus, and Markus Meister. Dynamic predictive coding by the retina. *Nature*, 436(7047):71–77, July 2005.

Mark Hubener, Doron Shoham, Amiram Grinvald, and Tobias Bonhoeffer. Spatial relationships among three columnar systems in cat area 17. *J. Neurosci.*, 17(23): 9270–9284, December 1997.

Andrew D. Huberman. Mechanisms of eye-specific visual circuit development. *Current Opinion in Neurobiology*, 17(1):73–80, February 2007.

J. Hurri and A. Hyvärinen. Temporal and spatiotemporal coherence in simple-cell responses: A generative model of natural image sequences. *Network: Computation in Neural Systems*, 14(3):527–551, 2003a.

Jarmo Hurri and Aapo Hyvärinen. Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, 15(3):663–691, 2003b.

A. Hyvärinen. One-unit contrast functions for independent component analysis: A statistical analysis. In *Neural Networks for Signal Processing VII (Proc. IEEE NNSP Workshop '97, Amelia Island, Florida)*, pages 388–397, 1997.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen and P.A. Hoyer. A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41(18):2413–2423, 2001.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, 1st edition, 2001a.

A. Hyvärinen, M. Gutmann, and P.O. Hoyer. Statistical model of natural stimuli predicts edge-like pooling of spatial frequency channels in V2. *BMC Neuroscience*, 6:12, 2005.

A. Hyvärinen, P.O. Hoyer, J. Hurri, and M. Gutmann. Statistical models of images and early vision. In *Proceedings of the Int. Symposium on Adaptive Knowledge Representation and Reasoning*, 2005.

Aapo Hyvärinen and Patrik Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Comp.*, 12(7):1705–1720, July 2000.

Aapo Hyvärinen and Erkki Oja. Independent component analysis by general nonlinear hebbian-like learning rules. *Signal Processing*, 64(3):301–313, February 1998.

Aapo Hyvärinen, Patrik O. Hoyer, and Mika Inki. Topographic independent component analysis. *Neural Comp.*, 13(7):1527–1558, July 2001b.

Aapo Hyvärinen, Jarmo Hurri, and Jaakko Väyrynen. Bubbles: a unifying framework for low-level statistical properties of natural image sequences. *J. Opt. Soc. Am. A*, 20(7):1237–1252, July 2003.

E. Izhikevich. Resonant-and fire-neurons. *Neural Networks*, 14:883–894, 2001.

E. Izhikevich. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press, 2006.

E. Kandel, J. Schwartz, and T. Jessel, editors. *Principles of Neural Science*. Mc Graw Hill, 4th edition, 2000.

L. C. Katz and C. J. Shatz. Synaptic activity and the construction of cortical circuits. *Science*, 274(5290):1133–1138, November 1996.

Lawrence C. Katz and Justin C. Crowley. Development of cortical circuits: Lessons from ocular dominance columns. *Nat Rev Neurosci*, 3(1):34–42, January 2002.

P. Lennie and J.A. Movshon. Coding of color and form in the geniculostriate visual pathway. *Journal of the Optical Society of America A*, 22:2013–2033, 2005.

Peter Lennie. The cost of cortical computation. *Current Biology*, 13(6):493–497, March 2003.

M.A. MacNeil, J.K. Heussy, R.F. Dacheux, E. Raviola, and R.H. Masland. The population of bipolar cells in the rabbit retina. *The Journal of Comparative Neurology*, 472(1):73–86, 2004.

S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

Richard H. Masland. Neuronal diversity in the retina. *Current Opinion in Neurobiology*, 11(4):431–436, August 2001a.

Richard H. Masland. The fundamental plan of the retina. *Nat Neurosci*, 4(9): 877–886, September 2001b.

K. Miller. *Models of Neural Networks III*, chapter Receptive fields and maps in the visual cortex: models of ocular dominance and orientation columns, pages 55–78. Springer-Verlag, 1996.

K.D. Miller and D.J.C. MacKay. The role of constraints in hebbian learning. *Neural Computation*, 6(1):100–126, 1994.

K.D. Miller, J.B. Keller, and M.P. Stryker. Ocular dominance column development: analysis and simulation. *Science*, 245(4918):605–615, August 1989.

U. Nuding and C. Zetzsche. Learning the selectivity of V2 and V4 neurons using non-linear multi-layer wavelet networks. *Biosystems*, 89(1-3):273–279, 2007.

K. Obermayer, G. Blasdel, and K. Schulten. Statistical-mechanical analysis of self-organization and pattern formation during the development of visual maps. *Phys. Rev. A*, 45(10):7568–7589, May 1992.

E. Oja, H. Ogawa, and J. Wangviwattana. Principal component analysis by homogeneous neural networks, part 1: The weighted subspace criterion. *IEICE Trans. on Information and Systems*, E75-D(3):366–375, 1992a.

E. Oja, H. Ogawa, and J. Wangviwattana. Principal component analysis by homogeneous neural networks, part 2: Analysis and extensions of the learning algorithms. *IEICE Trans. on Information and Systems*, E75-D(3):376–382, 1992b.

Anna A. Penn, Patricio A. Riquelme, Marla B. Feller, and Carla J. Shatz. Competition in retinogeniculate patterning driven by spontaneous activity. *Science*, 279(5359):2108–2112, March 1998.

P. Perona. Deformable kernels for early vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(5):488–499, 1995.

Laurent Perrinet. Finding independent components using spikes: A natural result of hebbian learning in a sparse spike coding scheme. *Natural Computing*, 3(2): 159–175, June 2004.

Franck Polleux. Genetic mechanisms specifying cortical connectivity: Let's make some projections together. *Neuron*, 46(3):395–400, May 2005.

Jason L. Puchalla, Elad Schneidman, Robert A. Harris, and Michael J. Berry. Redundancy in the population code of the retina. *Neuron*, 46(3):493–504, May 2005.

Purves, editor. *Neuroscience*. Sinauer, 3rd edition, 2004.

J.P. Rauschecker and W. Singer. The effects of early visual experience on the cat's visual cortex and their possible explanation by hebb synapses. *J Physiol*, 310: 215?–39, 1981.

H. Reinhard. *Elements de mathematiques du signal*. Dunod, 1997.

F. Rieke, D. Warland, R. de R. van Steveninck, and W. Bialek. *Spikes: Exploring the neural code*. MIT Press, 1997.

Rebecca L. Rockhill, Frank J. Daly, Margaret A. MacNeil, Solange P. Brown, and Richard H. Masland. The diversity of ganglion cells in a mammalian retina. *J. Neurosci.*, 22(9):3831–3843, May 2002.

R.W. Rodieck. *The First Steps in Seeing.* Sinauer, 1998.

T.E. Rolls and G. Deco. *Computational neuroscience of vision.* Oxford University Press, 2002.

D. L. Ruderman and W. Bialek. Statistics of natural images: Scaling in the woods. *Phys Rev. Letters*, 73, 1994.

Daniel Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517–548, November 1994.

Daniel L. Ruderman. Origins of scaling in natural images. *Vision Research*, 37 (23):3385–3398, December 1997.

T.D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

Ronen Segev, Jason Puchalla, and Michael J. Berry. Functional organization of ganglion cells in the salamander retina. *J Neurophysiol*, 95(4):2277–2292, April 2006.

Evan Smith and Michael S. Lewicki. Efficient coding of time-relative structure using spikes. *Neural Computation*, 17(1):19–45, 2005.

Evan C. Smith and Michael S. Lewicki. Efficient auditory coding. *Nature*, 439 (7079):978–982, February 2006.

Samuel G. Solomon and Peter Lennie. The machinery of colour vision. *Nat Rev Neurosci*, 8(4):276–286, April 2007.

Samuel G. Solomon, Jonathan W. Peirce, Neel T. Dhruv, and Peter Lennie. Profound contrast adaptation early in the visual pathway. *Neuron*, 42(1):155–162, April 2004.

Sen Song and L. F. Abbott. Cortical development and remapping through spike timing-dependent plasticity. *Neuron*, 32(2):339–350, October 2001.

Sen Song, Kenneth D. Miller, and L. F. Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nat Neurosci*, 3(9):919–926, September 2000.

N.V. Swindale. The development of topography in the visual cortex: a review of models. *Network: Computation in Neural Systems*, 7:161–247, 1996.

A. van der Schaaf and J.H. van Hateren. Modelling the power spectra of natural images : statistics and information. *Vision Research*, 36:2759–2770, 1996.

J. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. Royal Soc. Lond. B*, 265:359–366, 1998.

J.H. van Hateren. Real and optimal neural images in early vision. *Nature*, 360 (6399):68–70, November 1992a.

J.H. van Hateren. Theoretical predictions of spatiotemporal receptive fields of fly lmcs, and experimental validation. *J.Comp.Physiol. A*, 171:157–170, 1992b.

J.H. van Hateren. A theory of maximizing sensory information. *Biol.Cybernetics*, 68:23–29, 1992c.

J.H. van Hateren and D.L. Ruderman. Independent component analysis of image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proc.R.Soc.Lond. B*, 265:2315–2320, 1998.

Benjamin Vincent, Roland Baddeley, Tom Troscianko, and Iain Gilchrist. Is the early visual system optimised to be energy efficient? *Network: Computation in Neural Systems*, 16(2 - 3):175–190, June 2005.

Benjamin T. Vincent and Roland J. Baddeley. Synaptic energy efficiency in retinal processing. *Vision Research*, 43(11):1285–1292, May 2003.

Heinz Wassle. Parallel processing in the mammalian retina. *Nat Rev Neurosci*, 5 (10):747–757, October 2004.

N.M. Weinberger. Dynamic Regulation of Receptive Fields and Maps in the Adult Sensory Cortex. *Annual Review of Neuroscience*, 18(1):129–158, 1995.

Michael Weliky, William H. Bosking, and David Fitzpatrick. A systematic map of direction preference in primary visual cortex. *Nature*, 379(6567):725–728, February 1996.

David G. Wilkinson. Multiple roles of eph receptors and ephrins in neural development. *Nat Rev Neurosci*, 2(3):155–164, March 2001.

Lei Xu. Least mean square error reconstruction principle for self-organizing neural-nets. *Neural Networks*, 6(5):627–648, 1993.

B. Yang. Projection approximation subspace tracking. *IEEE Trans. on Signal Processing*, 43(1):95–107, 1995.

Haishan Yao, Lei Shi, Feng Han, Hongfeng Gao, and Yang Dan. Rapid learning in cortical coding of visual scenes. *Nat Neurosci*, 10(6):772–778, June 2007.

Joshua M. Young, Wioletta J. Waleszczyk, Chun Wang, Michael B. Calford, Bogdan Dreher, and Klaus Obermayer. Cortical reorganization consistent with spike timing-but not correlation-dependent plasticity. *Nat Neurosci*, 10(7):887–895, July 2007.

Hongbo Yu, Brandon J. Farley, Dezhe Z. Jin, and Mriganka Sur. The coordinated mapping of visual space and response features in visual cortex. *Neuron*, 47(2): 267–280, July 2005.