Information Processing and Learning in Networks of Cortical Columns

DISSERTATION
zur
Erlangung des Grades 
„Doktor der Naturwissenschaften“
an der Fakultät für Physik und Astronomie
der Ruhr-Universität Bochum

von
Jörg Lücke
aus
Paderborn

Bochum 2004
Acknowledgment

I thank J. D. Bouecke for his help in acquiring simulation results and I thank I. J. Wundrich, R. P. Würtz, and C. von der Malsburg for many discussions and useful suggestions. Further thanks go to B. Bolder, D. Garrett, S. Wu, and J. Zhu who improved this work by their feedback on different chapters. Special thanks go to A. C. Möller for her at least as important non-scientific support.
## Contents

List of Symbols 5

1 Introduction 6

2 Neural Dynamics of the Macrocolumn 9
   2.1 Model of the Minicolumn 9
   2.2 Model of the Macrocolumn 11
   2.3 Stability Analysis 12
   2.4 Afferent Fibers 14
   2.5 Conclusion and Discussion 18

3 Self-Organizing Receptive Fields 20
   3.1 Hebbian Plasticity 20
   3.2 Experiments of Receptive Field Self-Organization 24
      3.2.1 Pattern Classification 26
      3.2.2 Extraction of Features 28
   3.3 Hierarchical Self-Organization 35
   3.4 Experiments of Hierarchical Receptive Field Self-Organization 38
      3.4.1 Input with Hierarchical Similarity Structure 39
      3.4.2 The Continuous Bars Test 43
   3.5 Conclusion and Discussion 48

4 Clustering with Receptive Field Self-Organization 52
   4.1 Place Coding of Continuous Variables 52
   4.2 Receptive Field Self-Organization 53
      4.2.1 Visualization 54
      4.2.2 Clustering of More Complex Data 57
      4.2.3 Data Point Superpositions 57
      4.2.4 Higher Dimensional Data 58
   4.3 Conclusion and Discussion 59

5 Abstraction of Macrocolumn Dynamics 61
   5.1 Basic Assumptions 61
      5.1.1 Stationary Points and Stability 62
      5.1.2 An Explicit Activation Function 65
      5.1.3 An Explicit Macrocolumn Dynamics 66
   5.2 Afferent Connections 70
   5.3 Hierarchical Self-Organization 75
   5.4 Clustering 79
   5.5 Conclusion 81
Contents

6 Rapid Dynamic Link Matching 83
   6.1 Principles of Dynamic Link Matching . . . . . . . . . . . . . . 84
   6.2 Representation of a Feature Vector . . . . . . . . . . . . . . . . 88
   6.3 A Network of Two Macrocolumns . . . . . . . . . . . . . . . . . 90
   6.4 A Macrocolumn Network for Similarity Matching . . . . . . . . . . 92
   6.5 Matching in a Macrocolumn Network . . . . . . . . . . . . . . . . 94
   6.6 Topological Matching . . . . . . . . . . . . . . . . . . . . . . . . 98
       6.6.1 Emergence of Topological Interconnectivity . . . . . . . . . . 101
       6.6.2 Topological Matching . . . . . . . . . . . . . . . . . . . . 103
   6.7 Conclusion and Future Work . . . . . . . . . . . . . . . . . . . . 108

7 Summary 111

Appendix 112

List of Figures 117

References 118
List of Symbols

- $n_i^\alpha$: activity of neuron $i$ in minicolumn $\alpha$
- $n_i^E$: activity of input neuron $i$
- $T_{ij}^\alpha$: internal connection from neuron $j$ to $i$ in minicolumn $\alpha$
- $R_{ij}^\alpha$: afferent fiber from input neuron $j$ to neuron $i$ of minicolumn $\alpha$
- $R_i^\alpha$: receptive field of minicolumn $\alpha$
- $\vec{I}$: input vector
- $s$: number of internal connections received by a neuron
- $r$: number of afferent fibers received by a neuron
- $c$: synaptic weight of an internal connection
- $\tilde{c}$: synaptic weight of an afferent fiber
- $\kappa$: ratio of internal to external input, coupling constant
- $k$: number of minicolumns
- $m$: number of neurons per minicolumns
- $N$: number of input neurons, number of layer units
- $p_\alpha$: mean neural activity of minicolumn $\alpha$
- $P$: mean neural activity of the macrocolumn
- $\mathcal{I}$: inhibition function between minicolumns
- $\Theta_o$: the neurons’ constant threshold
- $\Theta_{ho}$: neuronal threshold noise
- $\nu$: gain factor of intercolumnar inhibition
- $\nu_c$: critical value of $\nu$
- $T$: period length of a $\nu$-cycle
- $\mathcal{P}_\nu$: activity of a minicolumn in a stationary point
- $\chi$: threshold for synaptic plasticity
- $\alpha, b_\chi$: scalar parameters of $\chi$ dynamics
- $\langle \gamma \rangle_T$: time average over the last $\nu$-cycles
- $\mathcal{E}$: synaptic growth factor

- $\vec{F}$: feature vector
- $\mathcal{L}_i$: neural unit $i$ of layer $\mathcal{L}$, $\mathcal{L} = \mathcal{I}$ is input layer, $\mathcal{L} = \mathcal{M}$ is model layer
- $E_{\alpha}^{\mathcal{L}_i}$: input to minicolumn $\alpha$ of the feature encoding macrocolumn in $\mathcal{L}_i$
- $I_{\mathcal{L}_i,\mathcal{L}_j}^{\mathcal{L}_i}$: input to minicolumn $j$ of the position encoding macrocolumn in $\mathcal{L}_i$
- $p_{\alpha}^{\mathcal{L}_i}$: activity of minicolumn $\alpha$ of the feature encoding macrocolumn in $\mathcal{L}_i$
- $W_{\mathcal{L}_i,\mathcal{L}_j}^{\mathcal{L}_i}$: activity of minicolumn $j$ of the position encoding macrocolumn in $\mathcal{L}_i$
- $C_E$: parameter determines ratio of jet input to input of the other layer
- $C_I$: parameter determines ratio of jet similarity input to topological input
- $R_{\alpha}^{\mathcal{L}_i,\mathcal{L}_j}$: interconnection between the layers
- $T_{ab}^{\mathcal{L}_i,\mathcal{L}_j}$: interconnection within the layers
1 Introduction

Simulations of artificial neural networks (ANNs) are a standard way to study neural information processing. Although a large amount of data about biological neural networks is available there remain uncertainties regarding the way in which neurons process incoming action potentials, the way the neurons are interconnected, and the way in which interconnections change dynamically over time. These uncertainties have generated a broad variety of different models of neural networks. They are based on different assumptions for connectivity (e.g., feed-forward or symmetrically interconnected), neuron models (e.g., McCulloch-Pitts, integrate-and-fire, Hodgkin-Huxley), and different modification rules for synaptic weight changes (e.g., Hebbian learning, back-propagation). ANNs similar to the Hopfield network (Hopfield, 1982; Hopfield and Tank, 1986) or to perceptrons (e.g. Fukushima et al., 1983) afford deep functional insight on the basis of mathematical analysis that, first, allowed the networks to be successful in various technical applications and, second, influenced our views on learning and information processing in biological neural networks significantly. However, it has become obvious that the classical ANNs fall short in modeling the generalization abilities or computation times of biological networks. Many important reactions in the brain take place in times so short that individual neurons only have time to transmit very few or just a single action potential (see Potter, 1976; Subramaniam et al., 1995; Thorpe et al., 1996b; Nowak and Bullier, 1997, for reaction times in the visual system). If graded signals are to be processed, models based on a single neuron rate code fail to model the measured reaction times. Further, most ANNs do not reflect biologically plausible connectivity because they were motivated by the view that biological information processing is continuously distributed over the cortical surface or that information is processed strictly feed-forward through layers of equal neurons. However, in the last decades a large amount of anatomical and physiological data was accumulated suggesting that the cortex is hierarchically organized in cellular columns as principal building blocks (see Mountcastle, 1997, and Buxhoeveden and Casanova, 2002, for overviews and Jones, 2000, for a critical discussion). Columnar organization is advantageous, first, with respect to the implementation of a neural population rate code able to overcome the computational speed limitations of single neuron rate codes and, second, with respect to connectivity and robustness. With evolutionary growth of the brain, individual building blocks had to connect to more and more other elements. Groups of neurons can support many more connections than individual neurons and a network based on neural columns as principal units can be expected to be much more robust against the loss of connections or drop-out of neurons. In the cerebral cortex of mammals neural columns can be identified on different scales. The minicolumn is believed to be the smallest neural entity consisting of several tens up to a few hundred neurons, which are stacked perpendicular to the cortical surface (see, e.g., Peters and Yilmaze, 1993; Peters and Payne, 1993 for what they called pyramidal cell module, see Fig. 1 for an illustration). The minicolumns themselves combine to what is called macrocolumn or segregate (Favorov and Diamond, 1990, see Mountcastle, 1997, for an overview). Like minicolumns, macrocolumns can be identified both anatomically and physiologically (Favorov and Diamond, 1990; Elston and Rosa, 2000; Lubke et al., 2000)
Figure 1: Pyramidal cell modules in cat and monkey primary visual cortex. The pyramidal cells contributing apical dendrites to the clusters are shown in black. The connectivity of pyramidal cell modules motivates the connectivity of model minicolumns. (The figure is taken from (Peters and Yilmaze, 1993) with friendly allowance of A. Peters).
Introduction

and are shown to process stimuli from the same source, such as an area of the visual field or a patch of the body surface (Favorov and Whitsel, 1988; Favorov and Diamond, 1990). In the primary somatosensory cortex of the cat macrocolumns have been found to contain approximately 80 minicolumns (Mountcastle, 1997). Although mini- and macrocolumns are best studied in primary sensory areas they are found in higher cortical areas as well (Peters et al., 1997; Constantinidis et al., 2001) and are believed to represent the basic building blocks of all areas of cortices of higher vertebrates (Mountcastle, 1997; Buxhoeveden and Casanova, 2002). The main part of a minicolumn is a collection of excitatory cells grouped around bundles of dendrites (Peters and Yilmaze, 1993) and axons (Peters and Sethares, 1996). Together with physiological findings (Thomson and Deuchars, 1994) this suggest that the excitatory cells of a minicolumn are mutually strongly interconnected (Mountcastle, 1997; Buxhoeveden and Casanova, 2002, see also Douglas et al., 1995). For inhibitory feedback double-bouquet cells and basket (clutch) cells play a central role (DeFelipe et al., 1989; DeFelipe et al., 1990; Peters and Sethares, 1997; Budd and Kisvarday, 2001). Dendritic branch and axonal field analysis suggests that the inhibitory cells are stimulated by the activities within the excitatory cells of their minicolumn and project back to a number of minicolumns in their neighborhood.

According to the studies above, the cortex is a network of interconnected neural modules of same architecture. Especially in the primary visual cortex the columnar structure itself and its lateral interconnectivity could be investigated by detailed anatomical analysis and advanced imaging methods. Inter-columnar interconnections were found to be very specific (Schmidt et al., 1997), to gradually form (Callaway and Katz, 1990; Chapman et al., 1996), and to reflect the diversity of the environment in which the subject grew up, e.g. (Schmidt et al., 1997). Based on this architecture the cortex is able to process information and to generate abstractions on the basis of examples. However, even in the brain area which is best investigated, in the visual cortex, the mechanisms underlying cortical information processing are far from being understood.

In this work we study a neural network model based on the columnar organization of the cortex. We define a model of minicolumns and macrocolumns motivated by the above anatomical and physiological investigations and analytically study the resulting neural dynamics. Hebbian plasticity of afferent fibers to minicolumns is subsequently shown to result in neural receptive field (RF) self-organization, which can account for structuring inter-columnar connectivity. We close by studying a model network of macrocolumns with specific architecture which can explain key features of neural processing in the visual cortex.
2 Neural Dynamics of the Macrocolumn

In this section we first define and analyze the dynamics of a model of a single mini-
column and then proceed by studying the dynamical properties of the macrocolumn
as a set of inhibitorily coupled minicolumns. The results of this section are partly
published in (Lücke et al., 2002) and (Lücke and von der Malsburg, 2004).

2.1 Model of the Minicolumn

We take a minicolumn to consist of excitatory neurons which are randomly intercon-
ected as motivated by the above mentioned findings (see Mountcastle, 1997, and
Buxhoeveden and Casanova, 2002, for review). The excitatory neurons are modeled
as McCulloch-Pitts neurons with a refractory period of one time step. The dynamics of a minicolumn consisting of \( m \) neurons is described by the following set of
difference equations:

\[
\begin{align*}
\mathcal{H}(x) & := \begin{cases} 
0 & \text{if } x \leq 0 \\
1 & \text{if } x > 0 
\end{cases} \\
n_i(t + 1) & = \mathcal{H} \left( \sum_{j=1}^{m} T_{ij} n_j(t) - \Theta \right) \cdot \mathcal{H}(1 - n_i(t)) 
\end{align*}
\]

with \( t \in \{0, 1, \ldots \} \). For the interconnection \( T_{ij} \) we assume that each neuron receives \( s \) synapses from other neurons of the minicolumn. We further assume that the
dendrites and axons interconnect randomly such that a natural choice for the proba-
bility to receive a synapse from a given pre-synaptic neuron is \( \frac{1}{m} \) (compare Anninos
et al., 1970). The strength of one connection we take to be equal to the constant \( c \).
Note that for any choice of \( c \) the threshold \( \Theta \) can be chosen such that the resulting
dynamics is the same. Without loss of generality we therefore choose \( c = \frac{1}{s} \) such
that \( \frac{1}{m} \sum_{i=1}^{m} T_{ij} = 1 \). To further analyze the dynamics we describe equations (2)
in terms of the activation probability of a neuron, \( p(t) \), at time \( t \). The probability
depends first on the number of received inputs and second on the probability that
the neuron was active in the preceding time step. Due to the interconnection \( T_{ij} \)
the probability \( f_{bn}(x) \) of a neuron to receive \( x \) non-zero inputs from its pre-synaptic
neurons is given by the binomial distribution

\[
f_{bn}(x) = \binom{s}{x} p^x (1 - p)^{s-x}.
\]

For \( s \gg 1 \) the distribution can be approximated by a Gaussian distribution (theorem
of Moivre-Laplace) given by the probability density:

\[
f_g(x) = \frac{1}{\sqrt{2 \pi \sigma}} e^{-\frac{1}{2} \left( \frac{x-a}{\sigma} \right)^2}, \quad a = sp, \quad \sigma = \sqrt{sp(1-p)}.
\]
the probability $p_B(t+1)$ that it is not refractory at $(t+1)$. The probability $p_B(t+1)$ is directly given by the complement of the probability that the neuron was active the time step before, $p_B(t+1) = (1 - p(t))$.

To further proceed with the analysis of the dynamics so-called coherence effects have to be considered, i.e., effects which are caused by repeating (cycling) neural activity states. Such effects are a direct consequence of the random but fixed interconnection matrix $(T_{ij})$ and thus interdependent neural activation and refraction probabilities. As we will discuss at the end of the section, the coherence effects can be suppressed, e.g., by neural threshold noise. If the effects are sufficiently suppressed, we can assume the processes with probabilities $p_A(t+1)$ and $p_B(t+1)$ to be approximately independent, $p(t+1) \approx p_A(t+1) p_B(t+1)$. The assumption permits a compact description of dynamics (2) in terms of the activation probability $p(t)$:

\[ p(t+1) \approx p_A(t+1) p_B(t+1) \]

\[ \approx \int_{s=0}^{\infty} f_s(x) \, dx \, (1 - p(t)) \]

\[ = \int_{s=0}^{\infty} \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2} \left( \frac{x-s}{\sigma} \right)^2} \, dx \, (1 - p(t)), \quad a = sp(t), \quad \sigma = \sqrt{sp(t)(1-p(t))} \]

\[ = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{x^2}{2}} \, dx \, (1 - p(t)). \]

Using the Gaussian error integral parameterized by $s$,

\[ \Phi_s(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} \, dy, \]

we get:

\[ p(t+1) \approx \Phi_s \left( \frac{p(t) - \Theta}{\sqrt{p(t)(1-p(t))}} \right) (1 - p(t)). \]

The inhibitory feedback to the minicolumnar activity is modeled indirectly as a rise of the threshold $\Theta$. It is taken to be present already in the next time step and to be equally sensed by all neurons, which can be motivated by the axonal distribution of inhibitory double-bouquet neurons (DeFelipe et al., 1990; Peters and Sethares, 1997). The inhibitory neurons receive input from the excitatory neurons of the minicolumn. The inhibitory feedback we choose to depend linearly on the over-all activity, $\frac{1}{m} \sum_{i=1}^{m} n_i(t)$, of the minicolumn,

\[ \Theta = \nu \frac{1}{m} \sum_{i=1}^{m} n_i(t) = \nu p(t) + \Theta_o, \]

where $\nu$ is the proportionality factor of inhibition and $\Theta_o$ the constant threshold of the neurons. The choice represents a natural approximation of the feedback and allows for a further analytical treatment. Note that the parameter $\nu$ will play a crucial role in the following. If we use (6) to define a difference equation which approximately describes the dynamics of $p$, we get by inserting (7):

\[ p(t+1) = \Phi_s \left( \frac{(1-\nu) p(t) - \Theta_o}{\sqrt{p(t)(1-p(t))}} \right) (1 - p(t)). \]
The difference equation (8) can be shown to possess a point of non-zero stable stationary activity for a wide range of parameters $s$, $\Theta_o$, and $\nu$, given by:

$$P_\nu := \max \{ p \mid p = \Phi_s \left( \frac{(1-\nu)p - \Theta_o}{\sqrt{p(1-p)}} \right)(1 - p) \}.$$  \hspace{1cm} (9)

$P_\nu$ can be numerically determined and its value is in good agreement with activity probabilities obtained by directly simulating (2) with inhibition (7). The dynamics (2) with (7) has to be simulated with additional neural threshold noise in order to suppress the coherence effects mentioned above. Fluctuations of the inhibitory feedback caused by a finite number of simulated neurons, $m$, also contribute to the noise but are on their own for a wide range of parameters insufficient for an appropriate suppression of the effects. Note that the coherence effects are most efficiently suppressed if the interconnection matrix $(T_{ij})$ is re-randomized at each time step. In this case the assumption $p(t+1) = p_A(t+1) p_B(t+1)$ and also the computation of $p_A(t+1)$ via the binomial distribution can be adapted from (Anninos et al., 1970). There the matter is thoroughly discussed for a dynamics with another type of inhibition, but the essential arguments carry over to our dynamics. Neural threshold noise as an alternative to re-randomization was first described in (Lücke et al., 2002). In specific simulations with fixed interconnections and threshold noise we have validated that their behavior closely matches that of simulations with successively re-randomized interconnections, which shows that the analytical results are also applicable for the biologically realistic case of fixed interconnections, as used in this work.

### 2.2 Model of the Macrocolumn

As motivated by the distribution of synapses of inhibitory neurons (DeFelipe et al., 1989; DeFelipe et al., 1990; Peters and Sethares, 1997; Budd and Kisvarday, 2001) the macrocolumn is modeled as a collection of inhibitorily coupled minicolumns. With the same assumption as above the dynamics is given by the following set of $k \cdot m$ difference equations,

$$n_i^\alpha(t+1) = \mathcal{H} \left( \sum_{j=1}^{m} T_{ij}^\alpha n_j^\alpha(t) - \mathcal{I}(t) - \Theta_o \right) \cdot \mathcal{H} (1 - n_i^\alpha(t)),$$  \hspace{1cm} (10)

where $\alpha = 1, \ldots, k$ labels the minicolumns, $i = 1, \ldots, m$ the neurons of the minicolumn, and where $\mathcal{I}(t)$ denotes a time dependent inhibition. Note that equation system (10) assumes that there are no direct connections between excitatory neurons of the different minicolumns. For a fixed $\alpha$ the interconnection $(T_{ij}^\alpha)$ is of the same type as in the previous section. By the statistical considerations of Sec. 2.1 we can replace the $k \cdot m$ difference equations by a set of $k$ difference equations in terms of activation probabilities $p_\alpha$ of neurons of different minicolumns:

$$p_\alpha(t+1) = \Phi_s \left( \frac{p_\alpha(t) - \mathcal{I}(t) - \Theta_o}{\sqrt{p_\alpha(t)(1 - p_\alpha(t))}} \right)(1 - p_\alpha(t)).$$  \hspace{1cm} (11)
The inhibitory feedback $I(t)$ is modeled as the maximum of the over-all activities in the minicolumns,

$$I(t) = \nu \max_{\beta=1,\ldots,k} \{p_\beta(t)\}, \quad p_\beta = \frac{1}{m} \sum_{i=1}^{m} n_\beta^i(t),$$

where the maximum operation is assumed to be implemented by the system of inhibitory neurons of the macrocolumn. The maximum operation can be biologically implemented in various ways (Yu et al., 2002). Some possibilities are based on shunting inhibition (compare also Reichardt et al., 1983) whereas others use subtractive inhibition. On the functional side, inhibition proportional to the maximal minicolumnar activity (12) results in a qualitatively different and favorable behavior compared to a dynamics with inhibition proportional to the average activity as was studied in (Lücke et al., 2002). The dynamical difference and its functional implications will be further discussed later in this section.

### 2.3 Stability Analysis

The dynamical properties of the macrocolumn model can now be studied with a stability analysis of a system of coupled non-linear difference equations (13):

$$p_\alpha(t+1) = \Phi_s \left( \frac{p_\alpha(t) - \nu \max_{\beta} \{p_\beta(t)\} - \Theta_\alpha}{\sqrt{p_\alpha(t)(1-p_\alpha(t))}} \right) (1-p_\alpha(t)) =: G_\alpha(\bar{p}(t))$$

First note that the system possesses the following set of potentially stable stationary points,

$$\mathcal{Q} = \{q | \forall i = 1, \ldots, k \quad (q_i = 0 \lor q_i = \mathcal{P}_\nu) \} = \{0, \mathcal{P}_\nu\}^k,$$

e.g., for $k = 3$, $(0,0,0)$, $(\mathcal{P}_\nu,0,0)$, $(0,\mathcal{P}_\nu,0)$, $(\mathcal{P}_\nu,\mathcal{P}_\nu,0)$, $(\mathcal{P}_\nu,0,\mathcal{P}_\nu)$, where $\mathcal{P}_\nu$ is given in (9). The magnitude of $\mathcal{Q}$ is $|\mathcal{Q}| = 2^k$. The vector with smallest norm, e.g. for $k = 3$, $(0,0,0)$, will be called $\vec{q}_{\min}$, and the vector with largest norm, $(\mathcal{P}_\nu,\mathcal{P}_\nu,\mathcal{P}_\nu)$, will be called $\vec{q}_{\max}$. The set of $\mathcal{Q}$ without the trivial stationary point $\vec{q}_{\min}$ will be denoted by $\mathcal{Q}^+$, $\mathcal{Q}^+ = \mathcal{Q} - \{\vec{q}_{\min}\}$. To analyze the stability of the stationary points in $\mathcal{Q}$ we first approximate $I(\bar{p}) = \nu \max_{\beta} \{p_\beta\}$ with a differentiable function:

$$\tilde{I}_\rho(\bar{p}) := \nu \left( \sum_{\beta=1}^{k} (p_\beta)^\rho \right)^{\frac{1}{\rho}} \Rightarrow \lim_{\rho \to \infty} \tilde{I}_\rho(\bar{p}) = I(\bar{p}).$$

If $l(\vec{q})$ denotes the number of non-zero entries of $\vec{q} \in \mathcal{Q}$, we get:

$$\left( \frac{\partial}{\partial p_\alpha} \tilde{I}_\rho \right)(\vec{q}) = \left( \frac{\partial}{\partial p_\alpha} \nu \left( \sum_{\beta=1}^{k} (p_\beta)^\rho \right)^{\frac{1}{\rho}} \right)_{|p=\vec{q}}$$

$$= \nu \left( \sum_{\beta=1}^{k} (p_\beta)^\rho \right)^{\frac{1}{\rho}-1} p_\alpha^{\rho-1} |_{p=\vec{q}}$$
symmetric stable stationary points tract only a small volume of neighboring phase space. For increasing values can then be obtained by long but straightforward calculations.

In the point of structural instability, for small values of the Jacobian of

\[ \lambda_0 = 0, \quad \lambda_1 = \frac{1}{2\sqrt{\mathcal{P}_\nu (1-\mathcal{P}_\nu)}} \left( 1 - \nu + \frac{1 - 2\mathcal{P}_\nu}{\mathcal{P}_\nu} \Theta_0 \right) \Phi'_s(h(\nu)) - \Phi_s(h(\nu)) \]

\[ \lambda_2 = \frac{1}{2\sqrt{\mathcal{P}_\nu (1-\mathcal{P}_\nu)}} \left( 1 + \left( 1 - 2\mathcal{P}_\nu \right) \nu + \frac{1 - 2\mathcal{P}_\nu}{\mathcal{P}_\nu} \Theta_0 \right) \Phi'_s(h(\nu)) - \Phi_s(h(\nu)) \]

where \( h(\nu) = \frac{(1-\nu) \mathcal{P}_\nu - \Theta_0}{\sqrt{\mathcal{P}_\nu (1-\mathcal{P}_\nu)}} \).

For a given vector \( \vec{q} \in \mathcal{Q} \), \( \lambda_0 \) is of multiplicity \( (k - l(\vec{q})) \), \( \lambda_1 \) is of multiplicity 1, and \( \lambda_2 \) is of multiplicity \( (l(\vec{q}) - 1) \). For fixed parameters \( s \) and \( \Theta_0 \), the magnitudes of all eigenvalues are smaller than one for \( \nu \) smaller than a critical value \( \nu_c \). For \( \nu > \nu_c \), \( \lambda_2 \) gets larger than one, which implies that all \( \vec{q} \in \mathcal{Q} \) with \( l(\vec{q}) \geq 2 \) become unstable. Hence, a set of \( (2^k - k - 1) \) stationary points of \( \mathcal{Q}^+ \) lose their stability at the same value \( \nu_c \). In Fig. 2 and 3 the properties of the dynamics are visualized using bifurcation diagrams. The critical value \( \nu_c \) can be computed numerically and is, for \( s = 15 \) and \( \Theta_0 = \frac{1}{s} \), given by \( \nu_c \approx 0.69 \).

For \( k = 2 \) the set \( \mathcal{Q}^+ \) consists of three non-trivial stationary points, which are all stable for \( \nu < \nu_c \). Apart from the points in \( \mathcal{Q}^+ \) there exist two unstable stationary points, which have been numerically computed and are given by the dotted lines in Fig. 2. For small values of \( \nu \) the unstable points lie in the vicinity of the anti-symmetric stable stationary points \( (\mathcal{P}_\nu, 0) \) and \( (0, \mathcal{P}_\nu) \), which indicates that they attract only a small volume of neighboring phase space. For increasing \( \nu \) the unstable points approach the symmetric stable stationary point \( \vec{q}_{\text{max}} = (\mathcal{P}_\nu, \mathcal{P}_\nu) \), which indicates that the phase space volume of points attracted by \( \vec{q}_{\text{max}} \) gets gradually smaller. In the point of structural instability, \( \nu = \nu_c \), \( \vec{q}_{\text{max}} \) finally loses its stability when it meets the unstable stationary points in a subcritical bifurcation.

To visualize the crucial and analytically derived property that \( (2^k - k - 1) \) stationary points of \( \mathcal{Q}^+ \) lose their stability for the same value of \( \nu \), the bifurcation diagram of a network for \( k = 3 \) is given in Fig. 3. In the diagram all stationary points of \( \mathcal{Q}^+ \) are plotted and we find for \( \nu < \nu_c \) the set’s \( (2^k - 1) \) stable stationary points. Apart from the points in \( \mathcal{Q}^+ \) we get a number of unstable stationary points, which all lie, for small \( \nu \), at the same distance from \( \vec{q}_{\text{max}} \) and in the vicinity of the other points of

\[ \Rightarrow \left( \frac{\partial}{\partial p_\alpha} \mathcal{I} \right) (\vec{q}) = \left\{ \begin{array}{ll} \frac{\nu}{\rho} & \text{for } \alpha = 1, \ldots, l(\vec{q}) \\ 0 & \text{for } \alpha = (l(\vec{q}) + 1), \ldots, k \end{array} \right. \]
Figure 2: Bifurcation diagram of equation system (13) for parameters $s = 15$, $\Theta_0 = \frac{1}{2}$ and $k = 2$. The points of $Q^+ = \{ (P_\nu, 0), (0, P_\nu), (P_\nu, P_\nu) \}$ are plotted together with the two unstable points of the dynamics. To obtain a 2-dimensional bifurcation diagram the stationary points for given $\nu$ are projected onto the 1-dimensional space with normal vector $\frac{1}{\sqrt{2}} (1, 1)$. The only stationary point not plotted is $\bar{q}_{\text{min}}$ because it projects onto the same point as $\bar{q}_{\text{max}}$. The solid lines mark stable stationary points, the dotted lines mark unstable points.

$Q^+$. As $\nu$ increases, the unstable points are getting closer to the stable points with $l(q) \geq 2$ non-zero entries. At $\nu = \nu_c$ these stable points of $Q^+$ lose their stability when they are met by the unstable points in the point of structural instability.

Note that an inhibition proportional to the mean minicolumnar activity instead of the maximum as in (12) results in a dynamics whose stationary points lose their stability for different values of $\nu$ (compare Lücke et al., 2002). This dynamic property is reflected by eigenvalues of the dynamics’ Jacobian which depend on $l(q)$ for $\mathcal{I}_\rho(\bar{p})$ with $0 < \rho < \infty$ (see Appendix A).

### 2.4 Afferent Fibers

For the neuron dynamics (10) we have gained, by our stability analysis, far reaching insight into the dynamical properties of the macrocolumnar model. The knowledge can now be exploited to investigate the dynamical behavior of the system if it is subject to perturbations in the form of externally induced input (see Fig. 4 for an illustration of afferent input to a collection of pyramidal cell modules). As it is cus-
2 Neural Dynamics

Figure 3: Bifurcation diagram of equation system (13) for parameters $s = 15$, $\Theta_0 = \frac{1}{2}$, and $k = 3$. For each $\nu$ the stationary points of the 3-dimensional phase space are projected to the plane given by the normal vector $\frac{1}{\sqrt{3}}(1, 1, 1)$. The vectors $\overline{p}_1, \overline{p}_2,$ and $\overline{p}_3$ are projections of the trihedron of the phase space onto this space. The only stationary point of the system which is not plotted is $\overline{q}_{\text{max}}$ because it projects onto the same point as $\overline{q}_{\text{min}}$. The unstable stationary points are plotted as dotted lines, the stable stationary points, which are always elements of $Q^+$, are plotted as solid lines. For $\nu < \nu_c$ all elements of $Q^+$ are stable but for $\nu > \nu_c$ only $k = 3$ stable points remain. All other stable points lose their stability in subcritical bifurcations for the same value of $\nu$.

In biology, we will denote the positive contribution of a presynaptic neuron to the input of the postsynaptic neuron excitatory postsynaptic potential (EPSP) and we will say that a neuron emits a spike at time $t$ if it is active at time $t$. For the dynamics as investigated in the previous section there are essentially three different modes of operation possible:

- For $\nu > \nu_c$ the macrocolumn can serve as a memory unit being able to stabilize $k$ different macroscopic states, i.e., stable stationary points of equation system (13). The switching between the states is possible by sending a sufficiently large quantity of EPSPs to the respective minicolumn.

- For $\nu < \nu_c$ the macrocolumn is able to stabilize $2^k - 1$ different macroscopic states. The transition between the states would be possible by inducing a sufficiently large quantity of EPSPs to an appropriate subset of minicolumns. If $\nu$ is chosen to be only slightly smaller than $\nu_c$ and if one starts with the stable
Figure 4: Sketch of an X-cell axon arbor terminating in layer IV, where it spreads over about 250 pyramidal cell modules. On the left is a pyramidal cell module of the cat, which contains about 200 neurons. (The figure is taken from (Peters and Payne, 1993) with friendly allowance of A. Peters.)
stationary point $\bar{q}_{\text{max}}$, already small differences in the input to the minicolumns are sufficient for the macrocolumn to change to a corresponding macroscopic state.

- If, for an initial state $\bar{q}_{\text{max}}$, $\nu$ is continuously increased from $\nu < \nu_c$ up to a value equal or larger than $\nu_c$, the system will change to another stable stationary point for some value of $\nu < \nu_c$ depending on the input. For larger values of $\nu$ the system can again change between stable points until $\nu$ is finally larger than (or close to) $\nu_c$ and the dynamics is forced to one of the remaining stable stationary points where just one minicolumn is active. Consider, for instance, external input to a macrocolumn with $k = 3$ minicolumns. If the numbers of EPSPs induced per time step, $M^\alpha_{\text{EPSP}}$, are different for the three minicolumns, e.g., $M^1_{\text{EPSP}} = M^2_{\text{EPSP}} = M^3_{\text{EPSP}} = 0 : 1 : (1 + \epsilon)$, for $0 < \epsilon \ll 1$, the dynamics will stabilize the initial state $\bar{q}_{\text{max}}$ for small $\nu$. If $\nu$ gets larger, the system will change to the stable point $(0, P_\nu, P_\nu)$ for some $\nu_1 < \nu_c$ because this point is less deflected by the input than $\bar{q}_{\text{max}}$. The deflection of $(0, P_\nu, P_\nu)$ caused by the input is sufficiently large, however, if $\nu$ is further increased. The system will therefore finally stabilize the point $(0, 0, P_\nu)$.

The third possibility is the one with the most useful features. For given inputs the dynamics first successively switches off the minicolumns with smallest inputs. These macro-state transitions occur the earlier the larger the differences between the inputs are and can therefore encode neural population rate differences. If a new stable stationary point is reached, the process of switching off a minicolumn continues, each time without the perturbing influence of the input of the already quiescent columns.

For a dynamics whose stationary points lose their stability for different values of $\nu$ (see Lücke et al., 2002) the number of active minicolumns is determined by $\nu$ and not by the input. Note in this context that (12) is not the only type of inhibition that results in a single critical value of $\nu$. Other inhibition functions, e.g., the average of active columns,

$$I_{ac}(t) := \nu \frac{1}{|A|} \sum_{\beta \in A} p_\beta(t), \quad A = \{\beta | p_\beta > \bar{p}\},$$

with $0 < \bar{p} \ll 1$, can also be shown to possess this property. In general, the contribution of quiescent minicolumns to the inhibition has to be negligible as a prerequisite for a dynamics with qualitative behavior comparable to the one with inhibition (12). The simplicity of (12) and its good functional performance were the reasons to choose an inhibition proportional to the maximum in this work.

The macro state transitions which depend on input differences but which are induced by an increased parameter $\nu$ are all performed near to symmetry breaking points. The transitions are theoretically infinitely sensitive to input differences such that a macrocolumn can serve as an ideal decision unit (see also Lücke et al., 2002). For appropriate parameters $s$ and $\Theta_\nu$ the stabilization of stationary activity is performed in few iteration steps such that the time to increase $\nu$ from its minimal to its maximal value can be as short as a few tens of time steps, which makes decisions very fast in addition. If the inhibition parameter oscillates, the macrocolumn can
repeatedly select the strongest input or inputs. In the next chapter this mode of operation is exploited and further discussed for a macrocolumn with explicitly modeled afferent fibers. In simulations with parameters $s = 15$, $\Theta_a = \frac{1}{2}$, $m = 100$, the system reactions were still very sensitive for an oscillation with period length $T = 25$ time steps and were still reliable for $T = 10$ (see Sec. 3 for further details).

2.5 Conclusion and Discussion

From an elementary neuron model and a random but column-based interconnection we derived a neural dynamics which allows a fast and reliable decision making with respect to minicolumnar inputs. The dynamics is best exploited with an oscillating gain factor of the inhibition.

As discussed in the introduction we designed our model of the cortical macrocolumn in accordance with relevant neuroanatomical and neurophysiological facts. We have seen that on discrimination tasks the resulting system can overcome two serious problems raised by the concept of single neurons as the brains’ decision units, reaction time and limited fault tolerance. The essential components of our model are column-based interconnections, discrete neural spike signals, and oscillatory activity. These neural characteristics which are usually seen as independent of each other, are shown here to form a natural alliance, with important functional consequences. The model requires little genetic information, being based on sparse, asymmetric and random interconnections within the minicolumn. Our model makes several simplifying assumptions, using an abstract neuron model, discrete time and direct inhibition. Experimental predictions of the model should therefore be treated with caution. A fundamental property of our system is the ability to sustain neural activity without input. The property is based on a random interconnection matrix within a minicolumn. A relatively high number of EPSPs per time step results in a relatively high number of EPSPs in the next. The amount of EPSPs is controlled by inhibitory feedback and refractoriness of the neurons. As studies of continuous time systems suggest (e.g. Wilson and Cowan, 1973), this mechanism can be implemented in a continuous time version of the presented minicolumn model (see Sec. 5), such that, with a continuous inhibition between the minicolumns as given in (12), the qualitative dynamical behavior of the discrete time model carries over to a continuous one.

Our system realizes neural populations with well-defined global behavior, while realistically using local update rules for individual neurons. The resulting population code is based on a collective firing rate, evaluated by the macrocolumnar dynamics as average over each minicolumn’s population at a particular phase relative to oscillating inhibition. We tentatively identify our inhibitory cycle with cortical oscillations in the gamma frequency range, approx. 30-60 Hz, or theta range, approx. 7-12Hz. Recent neurophysiological experiments (Perez-Orive et al., 2002, see Singer, 2003, for review) support this view of a phase-coupled population rate code. The oscillation in our model is taken to be generated by a not further specified subsystem of the brain (see, e.g., Körner et al., 1999, for possible sources of the oscillation). The inhibition oscillation induces decision making in our system, which is different from the role of oscillations in other column based models (e.g., Fukai,
A central issue for understanding the brain is the neural code. The currently dominant view is the single neuron hypothesis (Barlow, 1972), according to which essential decisions of the brain can be linked directly to firing decisions of individual neurons. A fundamental difficulty for this view are reaction times of the brain. These can be so short that single neurons can fire only once. This makes it impossible to express graded signals (see, however, the time-of-arrival hypothesis of Thorpe, 1988, which also advocates a firing phase). On the other hand, a population code can be the basis for very fast information processing. In our model with a standard set of parameters, individual neurons typically fire only 2-10 times (for $T = 25$) before the macrocolumn makes a decision, and yet the decision is based in a precise graded fashion on the input (if $T$ is reduced to $T = 10$ the system shows qualitatively the same behavior but neurons spike only 1-4 times before the first macro-state transition).

The other fundamental weakness of the single neuron hypothesis is the lack of robustness against damage and accidents of wiring. The usual proposal to repair this weakness is a population code, and our model may be seen as an essential step at establishing one. The minicolumn has a collective receptive field. This makes it fault tolerant with respect to accidents in the afferent connections; the same can be said about intra-cortical connections. Moreover, the recurrent minicolumnar interconnections makes them robust to lesion or imperfections in ontogenesis.
We now explicitly consider afferent fibers to the model macrocolumn and study their modifications if they are subject to synaptic plasticity. Results of this section were published in (Lücke and von der Malsburg, 2004) and (Lücke, 2004b).

3.1 Hebbian Plasticity

We consider the situation that the excitatory neurons of the macrocolumn, \( n_i^E \), receive input from an input layer of \( N \) external neurons \( n_j^E \) which are of the same type as the excitatory neurons of the minicolumns. In the following we think of the neurons of the input layer as extra-cortical neurons in order to analyze the dynamical properties of the macrocolumn in a more convenient way. However, the input neurons can also be considered to be excitatory neurons of other macrocolumns, which would account for lateral excitation within the cortex. In Sec. 6 we will explicitly consider macrocolumn-macrocolumn connections.

The external or input neurons will be denoted by \( n_j^E \) (\( j = 1, \ldots, N \)) and an afferent fiber from input neuron \( n_j^E \) to neuron \( i \) of minicolumn \( \alpha \) will be denoted by \( R_{ij}^\alpha \). The dynamics (10) together with afferent fibers and input neurons is now given by:

\[
\begin{align*}
n_i^\alpha(t+1) & = \mathcal{H} \left( \sum_{j=1}^m T_{ij}^\alpha n_j^\alpha(t) + \sum_{j=1}^N R_{ij}^\alpha n_j^E(t) - \Theta(t) \right) \mathcal{H}(1 - n_i^\alpha(t)), \quad (18) \\
\Theta(t) & = \nu \max_{\alpha=1,\ldots,k} \{ p_\alpha(t) \} + \Theta_o + \Theta_{no}. \quad (19)
\end{align*}
\]

The afferents from external neurons \( (R_{ij}^\alpha) \) will dynamically change in time but we require that a neuron always receives exactly \( r \) synapses of weight \( \tilde{c} = \frac{w}{s} \) from external neurons. The ratio between the number of external and the number of internal connections\(^3\),

\[
\kappa = \frac{\sum_{j=1}^N R_{ij}^\alpha}{\sum_{j=1}^m T_{ij}^\alpha}, \quad (20)
\]

is taken to be significantly smaller than one which means that the internal input of an active minicolumn is on average significantly larger than the external input.

The external neurons \( n_j^E \) are essentially of the same type as the neurons of the macrocolumn, i.e., they are either inactive or active, \( n_j^E(t) \in \{0, 1\} \), and they are refractory for one time step after they have spiked:

\[
n_j^E(t+1) = B_j(w_i) \mathcal{H}(1 - n_j^E(t)), \quad w_i = \frac{1}{s} I_i, \quad (21)
\]

where \( B_j(w_i) \in \{0, 1\} \) is a (for each input neuron different) Bernoulli process\(^4\) with probability \( W(0) = 1 - w_i \) and \( W(1) = w_i \). The neuron \( n_i^E(t+1) \) is active only if,

\(^2\)in the same way as their activities

\(^3\)Note that due to the required boundary conditions \( \kappa \) is independent of \( i \) and \( \alpha \)

\(^4\)a sequence of independent and identically distributed Bernoulli trials
first, \( B_i(w_i) \) is equal to one and, second, if the neuron was not active the time-step before. The probability \( w_i \) determines the activity probability of the neuron. Note that for time constant activity probability \( p \) of a neuron \( n^E_j \) applies \( p = w_i(1 - p) \) because \( B_i(w_i) \) has to be equal to one and the neuron has to be non-refractory. The external neurons translate an input vector \( \vec{I} \in [0, 1]^N \) into spike rates such that a neuron \( n^E_j \) has a probability of \( \frac{1}{3}I_j \) to be active. The maximal activity of input neurons \((21)\) is \( \frac{1}{2} \) but we choose a maximal activity of \( \frac{1}{3} \) to maintain a stochastic behavior of a neuron also in the case of \( I_i = 1 \).

Figure 5: A Sketch of a macrocolumn with \( k = 3 \) minicolumns, parameters \( m = 8, s = 3, \) and \( r = 2, \) connected to \( N = 25 \) external neurons. The inhibition is symbolically sketched. The randomly initialized RF, \( \vec{R}^1, \) of minicolumn \( \alpha = 1 \) is fully displayed whereas RFs \( \vec{R}^2 \) and \( \vec{R}^3 \) are not. Lines within the input layer are only displayed for visualization purposes. B Two dimensional visualization of the RF of one neuron in minicolumn \( \alpha = 1 \) as grey-level image. C Two dimensional visualization of \( \vec{R}^1 \) as grey-level image.

The \( N \) dimensional vector of all afferent fibers to neuron \( n^E_i, \) \( \vec{R}^\alpha_i = (R^\alpha_{i1}, \ldots, R^\alpha_{iN}), \) will be called the receptive field (RF) of the neuron and the sum of the RF vectors of all neurons of a minicolumn \( \alpha \) will be called the RF of the minicolumn, \( \vec{R}^\alpha = \sum_{i=1}^{m} \vec{R}^\alpha_i \) (see Fig.5B,C). In Fig.5A a sketch of a macrocolumn with \( k = 3 \) minicolumns is displayed. The input neurons are arranged in a two-
dimensional array because we will use two-dimensional grey-level images as input vectors to conveniently display simulation results. Note, however, that the set of input neurons is not equipped with any neighborhood relationship and can therefore be arranged in an arbitrary manner.

Instead of reanalyzing the dynamics statistically it is sufficient for small $\kappa$ to treat the external input to the macrocolumn as perturbation of the internal dynamics. The macrocolumn will be operated by repeatedly increasing the inhibition factor $\nu$ from a minimal value $\nu_{\text{min}}$ to a maximal value $\nu_{\text{max}}$. The system is hereby forced to select the column(s) with strongest input at the end of each period or $\nu$-cycle (as we will call it from now on). In the beginning of a $\nu$-cycle the system has to be in the state $q_{\text{max}}$, which can be achieved under the influence of noise by setting $\nu$ to a sufficiently small value before starting to increase $\nu$ from $\nu_{\text{min}}$ (see Fig. 6B). If for a macrocolumn of, e.g., $k = 3$ minicolumns, the RFs, $\tilde{R}^\alpha$, are already given, the system is able to distinguish even strongly overlapping input patterns. The system first switches off the minicolumns with RFs very different to the presented stimulus and then decides between the remaining minicolumns with RFs similar to the stimulus (see Fig. 6). In this way the system can also gracefully handle simultaneously presented patterns. If a superposition of two patterns corresponding to the RFs of two minicolumns is presented, the dynamics switches off all irrelevant minicolumns except the two corresponding ones, whose activities are symmetrized. It then depends on the choice of $\nu_{\text{max}}$ whether this is the final state or whether the symmetry is broken to favor one of the patterns.

We now proceed by introducing Hebbian plasticity of the afferent fibers to match neurophysiological experiments which show input-dependent changes of neuron RFs. As the RFs of neurons, $\tilde{R}^\alpha_i$, change, the RFs of the minicolumns, $\tilde{R}^\alpha$, consequently change in time as well. As update rule for the synaptic change, $\Delta R^\alpha_{ij}(t) = R^\alpha_{ij}(t) - R^\alpha_{ij}(t-1)$, we use elementary Hebbian plasticity, i.e., an afferent connection $\tilde{R}^\alpha_{ij}$ is increased if the pre-synaptic neuron was active at the time-step directly preceding the firing of the post-synaptic neuron. In the beginning of a $\nu$-cycle all minicolumns have about the same activity. Only after the selection process, i.e., at lower levels of activity due to minicolumn deactivation, the activity state notably reflects input differences. Therefore, synaptic plasticity has to be predominant at low levels of macrocolumnar activity, $A(t) = \sum_{\alpha=1}^{k} \sum_{i=1}^{m} n^\alpha_i(t)$, in order to generate discriminating RFs. A simple and, as it turned out, functionally advantageous way to do this is enabling synaptic modification only if $A(t)$ is smaller than a threshold $\xi$. As activity oscillations are ubiquitous in the cortex, it seems plausible that synaptic plasticity is phase coupled (see Wespatat et al., 2004, for recent evidence of phase coupled synaptic modification). As mentioned above, we demand as boundary condition for the dynamics that the number of synapses received by a minicolumnar neuron is limited to $r$ in order to avoid unlimited synaptic growth.

We get as dynamic equations for the synaptic weights ($\alpha = 1, \ldots, k$; $i = 1, \ldots, m$; $j = 1, \ldots, N$):

$$\Delta R^\alpha_{ij}(t) = B_i(\epsilon) n^E_i(t) n^E_j(t-1) \quad \text{iff } A(t) < \xi,$$

$$\forall i, \alpha : \frac{1}{c} \sum_{j=1}^{N} R^\alpha_{ij}(t) = r. \quad (23)$$
Figure 6: Operation and dynamic behavior of a system with parameters, \(m = 100\), \(s = 15\), \(r = 7\), \(\Theta_o = \frac{1}{s}\), with \(k = 3\) minicolumns, and an input layer of \(N = 16 \times 16\) neurons. In A the RFs, \(\tilde{R}^\alpha\), of the minicolumns \(\alpha = 1, 2, 3\) are given as two-dimensional plots of the \(16^2\) vector entries. The entries are visualized as gray levels (black = 0). To make the RFs more conceivable we have chosen them to be of the form of simple two dimensional patterns. The input pattern is chosen to correspond to the RF of minicolumn \(\alpha = 3\). During the operation of the system all neurons of the input layer which correspond to white pixels spike with probability \(\frac{1}{3}\), neurons which correspond to black pixels are not spiking. In B the periodical change of the parameter of inhibition, \(\nu\), is visualized. After a short period with \(\nu = 0.1\) which serves to reset the dynamics to \(\tilde{q}_{\text{max}}\), \(\nu\) is linearly increased from \(\nu_{\text{min}}\) to \(\nu_{\text{max}} = 1.12\). Three \(\nu\)-cycles with period length \(T_\nu = 25\) are displayed. The dynamic behavior of the system is visualized in C where the activities \(p_\alpha(t)\) for the minicolumns \(\alpha = 1, 2, 3\) are plotted against time. In the beginning of a \(\nu\)-cycle the dynamics tend to symmetrize the activities as predicted by the theoretical results and the bifurcation diagrams. The symmetry is first broken when minicolumn \(\alpha = 1\) is switched off because its RF receives the smallest number of EPSPs from the presented input. Afterwards the stationary point \((0, \bar{P}_\nu, \bar{P}_\nu)\) is stabilized, i.e., the remaining two minicolumn activities are symmetrized, until minicolumn \(\alpha = 2\) becomes quiescent because it receives less input than minicolumn \(\alpha = 3\). The qualitative behavior for each \(\nu\)-cycle is the same but quantitative differences exist due to threshold noise of the neurons and due to finitely many neurons per minicolumn.
As our synaptic weights are discrete values, \( B_t(\mathcal{E}) \) is not a real valued growth factor but a probability\(^5\) that the synaptic weight is increased by \( \frac{\nu}{\nu} \). If \( R_{ij}^n \) is increased, the neuron \( \eta_i^j \) removes randomly one of its afferent from the input layer in order to fulfill boundary condition (23).

We operate the system by periodically changing \( \nu \) as in Fig. 6B:

\[
\nu(t) = \begin{cases} 
\nu_{\text{init}} & \text{if } \tilde{t} < T_{\text{init}} \\
(\nu_{\text{max}} - \nu_{\text{min}}) \frac{t - T_{\text{init}}}{T - T_{\text{init}}} + \nu_{\text{min}} & \text{if } \tilde{t} \geq T_{\text{init}},
\end{cases}
\]  

(24)

where \( T_{\text{init}} \) is the length of an initialization phase and where \( \tilde{t} = \text{mod}(t, T) \) denotes the common residue, i.e., \( \tilde{t} = t - t_o \), where \( t_o \) is the largest value smaller or equal \( t \) which is dividable by the oscillation’s period length \( T \). Throughout the duration of a \( \nu \)-cycle we present a pattern \( \tilde{I} \in [0,1]^N \) randomly chosen from a set of input patterns. An input neuron is then spiking according to (21).

The RFs of the neurons are randomly initialized and are modified according to (22) and (23). If the set of training patterns is structured, e.g., in the sense that it contains a small number of patterns as in Fig. 7B, we can observe a specialization of the RFs of the minicolumns to the different input patterns. In Fig. 7C the modification of the RFs, \( \tilde{R}_\alpha^\alpha \), of a macrocolumn with \( \alpha = 1, 2, 3 \) minicolumns is displayed and it can be seen that the system organizes its RFs such that the macrocolumn becomes a decision unit for the input patterns.

In the beginning an input pattern effects all minicolumns equally such that the system selects a subset of minicolumns by symmetry breakings. As soon as, initiated by random selection, a RF specializes for one class of input patterns, the corresponding minicolumn is more likely to be activated by patterns of this class, which further increases the specialization of the RF. This is the positive feedback loop of the self-organizing process, which amplifies small fluctuation and finally leads to an ordered state of the RFs.

Additionally to self-organizing aspects, we have a competition due to the minicolumn selection process and competition between afferent fibers induced by (23). In order to avoid mutual weakening of different patterns stored in the same minicolumn RF, the system specializes its RFs to adequately different input patterns.

### 3.2 Experiments of Receptive Field Self-Organization

We have seen that the system is able to specialize its RFs to different input patterns. So far we presented three different patterns to a network of three minicolumns (Fig. 7). We will now investigate two more general situations. In the first experimental setting we will present to the network different patterns which can be grouped into different classes. In the second setting the network’s task is to extract basic constituents of a class of patterns generated by combining different bars, a task known as the bars test (Földiák, 1990).

For both tasks one and the same network is used with the same set of parameters. All experiments use an input layer of \( N = 16 \times 16 \) input neurons. If an input pattern

---

\(^5\)To be more precise, for each \((i,j,\alpha)\) \( B_t(\mathcal{E}) \in \{0,\bar{c}\} \) is a Bernoulli process with probability \( W(0) = 1 - \mathcal{E} \) and \( W(\bar{c}) = \mathcal{E} \).
Figure 7: A Sketch of a macrocolumn with $k = 3$ minicolumns connected to an input layer of $N = 25$ neurons. The $m = 8$ neurons per minicolumn are randomly interconnected, each minicolumnar neuron receives $s = 3$ synapses from within its minicolumn. The inhibition is symbolically sketched as one inhibitory neuron receiving input from all minicolumns and projecting back to all of them. Each minicolumnar neuron receives $r = 2$ synapses from neurons of the input layer. The randomly initialized RF, $\tilde{R}^1$, of minicolumn $\alpha = 1$ is fully displayed whereas RFs $\tilde{R}^2$ and $\tilde{R}^3$ are not. Lines within the input layer are only displayed for visualization purposes, there are no connections of neurons within the input layer. B Set of three different input patterns of $16 \times 16$ pixels. C Modifications of RFs of a macrocolumn with $k = 3$ minicolumns and parameters $m = 100$, $s = 15$, $r = 7$, $\Theta_o = \frac{1}{s}$, $\mathcal{E} = 0.03$, $\xi = 55$, and $N = 256$. For 0 $\nu$-cycles the random initialization of the RFs is displayed. After five $\nu$-cycles (and five presentations of patterns randomly chosen from the set of input patterns) $\tilde{R}^1$ is slowly specializing to pattern 2 and after 10 and 15 $\nu$-cycles $\tilde{R}^2$ and $\tilde{R}^3$ specialize to the patterns 3 and 1, respectively. After 15 $\nu$-cycles the RF specialization further increases until the maximal specialization is reached after about 100 $\nu$-cycles. From 100 $\nu$-cycles on the degree of specialization remains unchanged.
\( \vec{I} \in [0,1]^N \) is presented the input neurons spike according to (21). Note that for the next two experiments we use \( \vec{I} \in \{0,1\}^N \) but we will later also consider input patterns with intermediate values.

We use a network with \( m = 100 \) neurons per minicolumn. Each neuron receives \( s = 15 \) synapses from pre-synaptic neurons of the same minicolumn and \( r = 7 \) synapses from neurons of the input layer. We choose the synaptic weights \( c = \frac{1}{s} \) and \( \tilde{c} = \frac{r}{s} \) to be equal for the following two experiments, \( w = 1 \), which results in a coupling (20) of \( \kappa \approx 0.47 \). Note that in later experiments we will work with a fixed \( \kappa \) and we will adjust the value of \( w \) accordingly.

The neurons’ constant threshold is set to \( \Theta_0 = \frac{1}{8} \approx 0.067 \), it is chosen such that a single EPSP is not sufficient to activate a neuron. The constant threshold is subject to Gaussian threshold noise with zero mean and a variance of \( \sigma_0^2 = 0.01 \).

The oscillation of the inhibition (24) is determined by the parameters \( \nu_{\text{init}} = 0.1 \), \( \nu_{\text{min}} = 0.5 \), \( \nu_{\text{max}} = 1.12 \), and the length of a \( \nu \)-cycle is \( T = 25 \) time steps with \( T_{\text{init}} = 3 \). As mentioned above the initialization phase serves to reset the dynamics to the stationary point \( \mu_{\text{max}} \). Self-organization of RFs is also possible with other, e.g., sinusoidal, types of \( \nu \)-oscillations. For the system to work appropriately it is only necessary that \( \nu \) gets small enough such that all minicolumns are activated again and that \( \nu \) is gets gradually larger during an oscillation such that sufficiently many minicolumns are deactivated.

Hebbian plasticity (22) and (23) is determined by the synaptic modification rate \( \mathcal{E} = 0.03 \) and the parameter \( \tilde{\xi} = 55 \) which determines the network activity for which synaptic modification is possible. The latter is chosen such that synaptic modification is only enabled close to the end of a \( \nu \)-cycle (note that a 2-3 times larger value for \( \xi \) with simultaneously reduced \( \mathcal{E} \) results in a system with comparable qualitative behavior).

All parameters are independent of the number of minicolumns \( k \) which we allow to change for different experiments. The parameters are partly chosen to reflect anatomical data as in the case of the number of neurons per minicolumns \( m = 100 \) and partly to optimize performance in the experiments. In the following we will refer to these parameters as the standard set of parameters.

### 3.2.1 Pattern Classification

We have seen that the system is able to specialize its RFs to be sensitive to a number of input patterns. More realistic input would not consist of a repeated presentation of exactly the same patterns as in Fig. 7B,C but rather of different patterns which can be grouped into different classes. In Fig. 8 a pattern classification experiment for such a kind of input is illustrated. For input patterns \( V^a \in \{0,1\}^{256} \) as displayed in Fig. 8A we can define the distance measure,

\[
    d_S(V^a, V^b) := \frac{|A \Delta B|}{|A \cup B|},
\]

where \( A = \{i | V^a_i = 1\} \), \( B = \{i | V^b_i = 1\} \), and where \( (A \Delta B) = (A \cup B) - (A \cap B) \) is the symmetric difference of sets. For input as in Fig. 8A it has turned out that dis-
Figure 8: In A the set of input patterns is displayed. During each $\nu$-cycle one randomly chosen pattern of this set is presented. B Distance matrix generated using the distance measure $d_S$. The line and column index enumerates the 42 input patterns in the same order as they appear in A. In C the modification of the RFs of a macrocolumn with $k = 6$ minicolumns and the standard set of parameters is displayed. After 250 and 1000 $\nu$-cycles four and six different pattern classes are represented, respectively. The RFs’ degree of specialization further increases thereafter and it can be seen that RFs $\tilde{R}^1$ and $\tilde{R}^2$ further subdivide the pattern class formerly represented by $\tilde{R}^1$ only. D Final RF specialization (after 250 $\nu$-cycles) if a macrocolumn with $k = 3$ minicolumns is used with the same input. E Final RF specialization (after 10000 $\nu$-cycles) if a macrocolumn with $k = 9$ is used.
3 Self-Organizing Receptive Fields

A typical modification of RFs of a macrocolumn with six minicolumns is displayed and it can be observed that the system builds up representations of all classes identifiable in Fig. 8B. If fewer minicolumns than pattern classes are available, the system builds up larger classes of mutually similar patterns (see Fig. 8D). If more minicolumns than major classes are available, the system further subdivides the pattern classes (see Fig. 8E). The subdivision may in this respect slightly differ from simulation to simulation. E.g., for \( k = 9 \) the ‘square class’ is in many cases only represented by one and the ‘plus class’ by three instead of two minicolumns as in Fig. 8E. In the experiment it can further be observed that the final representation rather depends on the substructure of the pattern classes than on their size, e.g., the ‘plus’ pattern appears more frequently than the ‘St. Andrew’s cross’ pattern but the ‘St. Andrew’s cross’ tends to be represented by more RFs (see Fig. 8C,E). Furthermore, the classification is independent of the number of white pixels per pattern because the impact of the patterns on the minicolumns is normalized by boundary condition (23). The independence is only affected by patterns approximately filling the whole input layer or by patterns having a number of white pixels close to zero.

3.2.2 Extraction of Features

So far we have seen that if the training patterns contain \( v \) classes of patterns, the system is able to identify these classes if \( k \geq v \). There are situations, however, where the training patterns cannot easily be grouped into pattern classes. This is the case, for instance, if we present from a number of \( v \) patterns not only the patterns themselves but also all possible superpositions. If \( k < 2^v \), the system is not able to store all patterns in different RFs. We have already mentioned in Sec. 3.1, however, that the internal dynamics of a macrocolumn is especially suitable to take into account pattern superpositions such that we can, nevertheless, expect the system to generate appropriate RFs. A method to evaluate the ability of a network to handle input which can only be represented by a combination of different patterns is the bars test. It was first introduced in (Földiák, 1990) and soon became a benchmark test for generalization and combinatorial abilities of learning systems. The training patterns of the bars test consist of horizontal and vertical bars. On a quadratic input layer, \( \frac{b}{2} \) (non-overlapping) horizontal and \( \frac{b}{2} \) (non-overlapping) vertical bars can be displayed (with \( b \) an even integer) each with probability \( p_o \) and all of equal size (see Fig. 9A for some examples with \( b = 8 \)). We will use bars tests with different numbers of bars \( b \) and will use a bar appearance probability of \( p_o = \frac{2}{b} \). Note that overlapping horizontal and vertical bars do not add up linearly because two overlapping white pixels do not add but result in a white pixel, as well. The bars test is passed if, after a training phase, the system has built up representations of all bars and is able to correctly classify input consisting of superpositions of the learned bars.

We can operate our system without modification, with the same set of parameters as for the experiment in Fig. 8, and it turns out that it passes the bars test without difficulties. The only prerequisite for a correct representation is that the number of minicolumns \( k \) is greater or equal to the number of different bars, \( k \geq b \). If \( k > b \)

---

\(^6\)See (Yianilos, 2002) for a proof that the distance function is a metric.
Figure 9: A A selection of 33 typical input patterns of the bars test of 8 different bars. B Typical example of the self-organization of the RFs of a macrocolumn with 10 minicolumns and the standard set of parameters. During each $\nu$-cycle a randomly generated input pattern of the upper type is presented. After about 250 $\nu$-cycles the network has already found representations of seven bars. After 1000 $\nu$-cycles representations of all bars are found and are further stabilized.
the RFs of some minicolumns remain uncommitted or specialize for a bar already
represented by another minicolumn. For a bars test with \( b = 8 \) different bars Fig. 9B
shows the modification of the RFs of a macrocolumn with \( k = 10 \) minicolumns.
Starting from random initialization the RFs specialize to different single bars even
though the input patterns consist mainly of bar superpositions. In Fig. 9B a repre-
sentation of all bars is clearly visible after 1000 \( \nu \)-cycles and the representation can
be seen to further stabilize thereafter. During the learning phase a RF sometimes
specializes to a combination two or more bars as can be seen by looking at RF \( \tilde{R}^8 \)
in Fig. 9 (after 500 \( \nu \)-cycles). Such a RF is not stable, however, because the parts of
the RF which correspond to different bars compete via equation (23). The RF there-
fore rapidly specializes for one bar if another RF becomes sensitive for the other.
An example is given by RFs \( \tilde{R}^7 \) and \( \tilde{R}^8 \) in Fig. 9 from 500 to 1000 \( \nu \)-cycles. In
the experiment of Fig. 9 two RFs remain unspecialized. In other experiments or for
a longer learning phase the two super-numerary RFs often specialize to an already
represented bar and increase redundancy in this way.

The bars test was used in different versions with different numbers of bars and
different systems. In (Hinton et al., 1995), for instance, 8 bars where used, (Hochre-
iter and Schmidhuber, 1999) and others used 10 bars, (Hinton and Ghahramani,
1997) 12 bars, and (Földiák, 1990) and others used 16 bars. To allow for compari-
son with these systems we measured the performance of our system for bars tests
of 8, 10, 12, 14, and 16 bars. For all tests we used the same system with the stan-
dard set of parameters and an input layer of \( 16 \times 16 \) neurons. The different bars
tests required different bar widths in order to cover the input layer appropriately.
For the bars test with \( b = 8 \) bars a bar width of four pixels was used (see Fig. 9),
for \( b = 10 \) three pixels, and for \( b = 12, 14, 16 \) bars were of a width of two pixels.
Consequently, the input layer is not uniformly covered for 10, 12, and 14 bars. In
Fig. 10A,B,C the results of different test series are presented. In Fig. 10A the num-
er of minicolumns is equal to the number of different bars, in Fig. 10B the number
of minicolumns exceeds the number of bars by 2, and in Fig. 10C a surplus of 4
minicolumns is available. A measurement point in the diagrams corresponds to the
number of \( \nu \)-cycles after which there is a 50% probability that all bars are repre-
sented, e.g., in 200 runs with 8 bars and \( k = 10 \) minicolumns there were 100 runs in
which a representation was found after 1050 \( \nu \)-cycles (see first measurement point
in Fig. 10B). A bar is taken to be represented by a minicolumn if the minicolumn
remains active in 9 of 10 \( \nu \)-cycles if the bar is presented. A macrocolumn is said
to have found a representation of all bars if all bars are represented by at least one
minicolumn and no minicolumn represents two different bars. In all runs the system
finally found a correct representation. Once a representation was found it remained
stable in the sense that the minicolumns remained specialized for the same bars. For
a given experimental setting there can be relatively large differences between indi-
vidual runs, however. For 8 bars and \( k = 10 \), e.g., a correct representation of the
bars was not found after about 2100 \( \nu \)-cycles in 20% of the 200 runs (indicated by
the upper bound of the error bar) whereas another 20% of the experiments found
representations already after 400 \( \nu \)-cycles (lower bound). The reason for this is that
all bars but one find representations very early whereas the remaining bar might con-
sume a long time to be represented – an effect which is, for instance, also observable.
Figure 10: In A, B, and C results of bars tests with \( b = 8, 10, 12, 14, 16 \) bars and a macrocolumn with standard set of parameters are displayed. In A the number of minicolumns of the used macrocolumns is always equal to the number of different bars. In B the number of minicolumns exceeds the number of bars by two and in C there is a surplus of four minicolumns. In D, E, and F results of a bars tests with \( b = 8 \) bars and a macrocolumn with \( k = 10 \) minicolumns and standard parameters are given. In D the input patterns are perturbed with bit flip noise of 0 to 12%. In E the bar widths are varied and in F the generation of the input patterns is altered in the way that, parameterized by \( \gamma \), half of the bars appear with a different probability than the other half (\( \gamma = 0 \) for equal probabilities). The measurement points of A, B, and C were obtained by taking 200 runs into account, the measurement points of D, E, and F with 100 runs, each. As result the number of \( \nu \)-cycles is given after which a representation of all bars is found with a probability of 0.5. The lower and upper bounds of the error bars correspond to a probability of 0.2 and 0.8, respectively. For each run a newly generated macrocolumn with newly initialized RFs was used.
in (Spratling and Johnson, 2002) for the noisy bars test.

In Fig. 10B,C a large reduction of learning time can be observed if the number of minicolumns is larger than that of presented bars. A surplus of 2 minicolumns results in a reduction to less than half of the learning time for no surplus and a surplus of 4 minicolumns results in a learning time of coarsely a fourth.

For the results in Fig. 10A,B,C we used a newly generated bars image for every $\nu$-cycle. The same experiments can be carried out, however, by choosing randomly from a fixed set of a number of $u$ generated bars images. If $u$ is several times larger than $b$ the results are qualitatively and quantitatively comparable. For a bars test with 8 bars and $k = 10$ minicolumns, for instance, $u = 50$ input patterns are fully sufficient to build up a correct representation of single bars.

The results of Fig. 10A,B,C further show that learning time in terms of $\nu$-cycles decreases if the number of bars does. This can be expected because the system has to learn a decreasing number of independent input constituents. On the other hand, there is an increasing overlap of bars, which makes it harder for the system to differentiate between two bars (compare Dayan and Zemel, 1995, and Hochreiter and Schmidhuber, 1999). The positive effect of fewer constituents is predominant in our system. The negative effect of more overlap can be made responsible, however, for an increase of learning time if the bar widths are varied as in an experiment discussed below.

Once a system has learned a correct representation of the bars it can be used to analyze bars images. To test the accuracy of the recognition we trained a macrocolumn with images generated according to the bars test until it found a representation. After some additional learning to further stabilize the representation it was tested with newly generated bars images of the same type as the training images. If an image is presented, the minicolumns corresponding to the bars appearing in the image remain active longer than minicolumns associated to bars not appearing in the test image. At the end of a $\nu$-cycle a minicolumn is either active or not. A test image is considered to be correctly recognized if for all minicolumns which correspond to bars appearing in the image, the probability to remain active is above average and if the probabilities of minicolumns corresponding to all other bars lie below average. Six macrocolumns of 16 minicolumns trained with bars images of 16 bars were each tested 100000 times with images generated according to the bars test except that, for convenience, we required each image to contain at least one bar. The networks could classify the input correctly in all but two of the 600000 cases. In the first case one of seven bars was not recognized and in the second one of eight bars.

In the usual bars test an individual bar is always displayed identical, the bars are of the same size, and all bars occur with exactly the same probability. Systems solving the bars test can therefore be suspected to use these artificial assumptions. The system (Földiáik, 1990), for instance, not only exploits the fact that the bars are occurring with the same probability but also needs to know the exact value of the bars’ probability of occurrence. How much a system relies on the assumptions of the bars test can be tested by relaxing them and we present three test series showing the corresponding behavior of our system. For all three series we use a bars test with $b = 8$ bars and a macrocolumn with $k = 10$ minicolumns and standard set of parameters.
Figure 11: A A selection of 33 typical input patterns of a bars test with $b = 8$ different bars and 8% bit flip noise. B Typical RF specialization corresponding to this input. RFs of a macrocolumn with 10 minicolumns and the standard set of parameters are displayed. After about 500 $\nu$-cycles representations of all bars are recognizable and after about 2000 $\nu$-cycles the maximal degree of specialization is reached.
For the robustness against perturbed bar images we presented input images with bit flip noise during the learning phase (see Fig. 11). In Fig. 10D the learning time is plotted for different degrees of noise. As can be observed, low levels of noise even have positive effects. However, with an increasing noise level the final degree of specialization of the minicolumns’ RFs is reduced. In Fig. 11, the final specialization degree corresponds to the displayed RFs after about 2000 or 5000 $\nu$-cycles. If compared to the final degree of specialization in Fig. 9B, it can be seen that in the noisy case the RFs have more overlap. The overlap increases with increasing noise which leads to an increasing instability of a representation of all bars until the system cannot find a representation of the bars anymore. For the standard set of parameters and for a bars test with 8 bars the bar representations become unstable above about 12% bit flip noise and no representations can be found for noise levels above about 15%. By decreasing the learning rate $\mathcal{E}$ the robustness against noise can be increased such that representations can be found for noise levels above 15%.

In the second test series the bar size is varied. For $b = 8$ the bars are usually $w = 4$ pixels wide. If $\bar{w} = (w_1, w_2, w_3, w_4)$ denotes the bar widths for the four vertical as well as for the four horizontal bars, we can define $\delta w = \sum_{i=1}^{4} |w_i - 4|$ as a measure for the bar width variation. In Fig. 10E the results for the test series $\bar{w} = (4, 4, 4, 4), (3, 4, 4, 5), (3, 3, 5, 5), (2, 3, 5, 6), (1, 3, 5, 7)$ are given. The learning time increases with increasing $\delta w$ presumably because the maximal bar overlap increases, e.g., for $\bar{w} = (1, 3, 5, 7)$ the horizontal 7 pixel wide bar covers nearly half of the 1 pixel wide vertical bar.

The robustness of the system against relaxation of the assumption that all bars occur with equal probability is investigated in the third test series. We reduce the appearance probability of four randomly chosen bars to the value $p = p_0 (1 - \gamma)$, and increase the appearance probability of the four other bars by the same value $p = p_0 (1 + \gamma)$. Here $\gamma$ is a parameter in the interval $[0, 1]$ and $p_0 = 0.25$ is the bar appearance probability. In Fig. 10F the results for $\gamma = 0.0, \ldots, 0.8$ are given, for $\gamma = 0.9$ the corresponding measurements are 3750, 8050, and 32850 $\nu$-cycles for probabilities to get a correct representation of 0.2, 0.5, and 0.8, respectively. The measurements show that the system learns reliable a correct representation even if half of the bars appear nearly 20 times more frequently than the others and it only needs a longer learning phase if half of the bars occur more than four times more frequently.

The bar appearance probability can also be varied globally. If all bars appear with the same probability $p_0$ and if $p_0$ is increased to larger values, the probability to find a few or a single bar in an input image gets gradually smaller. For $k = 10, b = 8$, and the standard set of parameters, learning time increases if $p_0$ gets larger than 0.25. For $p_0 = 0.375$ the system found a stable representation in all of 100 runs and needed less than 1600 $\nu$-cycles to find representations in 50% of them. In addition to a longer learning phase, the RFs representing the different bars become less disjunct until the final representation gets unstable for values of $p_0$ larger than about 0.45. Representations for input with higher values of $p_0$ can be found, however, if the synaptic modification rate $\mathcal{E}$ is reduced, which in general stabilizes the representation. For $\mathcal{E} = 0.005$ instead of $\mathcal{E} = 0.03$ the system always finds stable representations for $p_0 = 0.5$ (after less than 18600 $\nu$-cycles in 50% of 100 runs).
However, even for very low values of $E$ there is limit at $p_o$ somewhat larger than 0.5 from which on no stable representation can be found anymore.

We have seen that one and the same network solves problems such as pattern classification and basic feature extraction. As demonstrated in the bars test, the network can build up a representation of the input, which allows to classify patterns by using distributed neural coding. The network found correct representations of all bars in all 5700 simulations which were carried out to acquire the data given in Fig. 10. After the learning phase the classification for the usual bars test shows a reliability of virtually 100%. All experimental data given in Fig.8 to Fig.11 was obtained with the same parameters. Different sets of parameters lead to different results and for an individual task the parameters can be optimized to obtain shorter learning times or a higher robustness. We have chosen, however, to use for all experiments the standard set of parameters in order to demonstrate universality and robustness of the system’s dynamics.

3.3 Hierarchical Self-Organization

In the dynamics for Hebbian plasticity the threshold $\xi$ plays a crucial role. As mentioned earlier it has the meaning of a reciprocal differentiation pressure. On the one hand, if $\xi$ is large the RFs on average start to modify already for relatively low values of $\nu$, i.e., already in the middle of the minicolumn selection process. RFs thus specialize relatively fast but are only coarsely discriminating. On the other hand, low values of $\xi$ result in strongly discriminating RFs but RFs need a long time to actually become specialized to a certain pattern. The value of $\xi$ chosen in the beginning of Sec. 3.2 is an intermediate one.

In this section we will extend the dynamics of Hebbian plasticity (22) and (23) by a dynamic threshold which we will call $\chi(t)$ in order to distinguish the dynamic threshold from the constant threshold $\xi$ in (22). Apart from the now time dependent threshold the synaptic plasticity dynamics remains the same and now read:

$$\Delta R_{ij}^\alpha(t) = B_i(E) n_i^\alpha(t) n_j^E(t-1)$$

$$\forall i, \alpha : \frac{1}{C} \sum_{j=1}^N R_{ij}^\alpha(t) = r.$$  

Here $P(t)$ is the over-all activity probability in the macrocolumn, $P(t) = \frac{1}{km} \sum_{i,j} n_i^\alpha(t)$ which is the normalized form of $A(t)$, $P(t) = \frac{1}{km} A(t)$.

The dynamics for $\chi$ we choose such that $\chi$ is large in the beginning of the self-organization process and that it decreases if RF specialization increases, i.e., specialization pressure is increased with increasing specialization of the RFs.

$\chi$ is only updated at time steps $\tau \in \{T-1, 2T-1, \ldots\}$ at the end of each $\nu$-cycle. For a time-depending function $f$ let us first define the average over the last $C \nu$-cycles:

$$\langle f \rangle_{\tau} := \frac{1}{C} \sum_{c=0}^{C-1} f(\tau - cT)$$
At the beginning of learning, the system starts with large $\chi$, $\chi(0) = \chi_0 = \frac{1}{3}$, which is subsequently decreased by the dynamics:\footnote{We demand $\chi(t) \leq \chi_0$ as boundary condition in order to prevent $\chi$ from increasing during the first updates.}

$$\Delta \chi(\tau) := \chi(\tau + 1) - \chi(\tau)$$  
\hspace{1cm} (29)

$$\Delta \chi(\tau) = -\lambda(\tau) \left( \chi(\tau) - a_\chi P(\tau) \right),$$  
\hspace{1cm} (30)

$$\lambda(\tau) = \lambda_0 \left( \frac{\min_\beta \{ \langle p_\beta \rangle_\tau \}}{\langle P \rangle_\tau} - b_\chi \right).$$  
\hspace{1cm} (31)

where $a_\chi$ and $b_\chi$ are scalar parameters and $\tau \in \{ T - 1, 2T - 1, \ldots \}$ are time-steps which always mark the end of a $\nu$-cycle (with period $T$). Only at these time steps $\chi$ and $\lambda$ are updated. The function $\min_\beta \{ \ldots \}$ is an average over the least active minicolumns (see below). For small and positive $\lambda$, $\chi$ is monotonously decreased until it stabilizes around the time average of $(a_\chi P(\tau))$. The velocity with which the threshold $\chi$ decreases and hence the velocity the differentiation of RFs increases is controlled by $\lambda$ which itself is dependent on the minicolumn activity averages at the end of the past $\nu$-cycles. The time dependent $\lambda$ slows down the differentiation process if some minicolumns are repeatedly quiescent in the minicolumn selection process and start to cease their RF specialization. For a small number of minicolumns $\min_\beta \{ \ldots \}$ in (31) can be the usual minimum, for arbitrary $k$ it has turned out, however, that the average over all minicolumn activities smaller than a quarter of the macrocolumn activity $P(\tau)$ gives better results:

$$L_\tau = \{ \beta \mid \langle p_\beta \rangle_\tau < \frac{1}{4} P(\tau) \text{ or } \langle p_\beta \rangle_\tau = \min_\tau \{ \langle p_\gamma \rangle_\tau \} \}$$  
\hspace{1cm} (32)

$$\min_\beta \{ \langle p_\beta \rangle_\tau \} = \frac{1}{|L_\tau|} \sum_{\beta \in L_\tau} \langle p_\beta \rangle_\tau$$  
\hspace{1cm} (33)

Hebbian plasticity (26) and (27) together with time-depending and self-controlled differentiation pressure (29) to (31) results in hierarchical self-organization of minicolumnar RFs. An example is shown in Fig.12. When synaptic modification starts the RFs are modified and fall into two groups after 10 to 50 $\nu$-cycles. Group one, consisting of RFs $\tilde{R}^1$ and $\tilde{R}^2$, specialize to patterns of the upper left corner and group two, consisting of RFs $\tilde{R}^3$, $\tilde{R}^4$, and $\tilde{R}^5$ specialize to the patterns of the lower right corner (in Fig.12C the RFs are appropriately ordered). In the interval between 50 and 200 $\nu$-cycles RFs $\tilde{R}^1$ and $\tilde{R}^2$ are sensitive to different mixtures of patterns $\tilde{F}^1$ and $\tilde{F}^2$ and RFs $\tilde{R}^3$, $\tilde{R}^4$, and $\tilde{R}^5$ are sensitive to different mixtures of patterns $\tilde{F}^3$, $\tilde{F}^4$, and $\tilde{F}^5$. During this learning period the RFs are repeatedly changing between sensitivities to different mixtures of patterns of their group but never to mixtures of patterns of different groups. From about 200 $\nu$-cycles on the RFs start to specialize to one of the patterns their group is sensitive to. $\tilde{R}^1$ and $\tilde{R}^2$ specialize to patterns $\tilde{F}^1$ and $\tilde{F}^2$, respectively, and $\tilde{R}^3$ and $\tilde{R}^5$ specialize to $\tilde{F}^3$ and $\tilde{F}^5$. RF $\tilde{R}^4$ specializes somewhat slower to pattern $\tilde{F}^4$ because of the overlap with patterns $\tilde{F}^3$ and $\tilde{F}^5$ (see Fig. 12B). From about 400 $\nu$-cycles on, all RFs have specialized for an input pattern and their degree of specializations further increases to a final value thereafter.
Figure 12: A The five different input patterns of the database. B Outlines of the input patterns. C Modification of the RFs of a macrocolumn with $k = 5$ minicolumns. D RFs of a macrocolumn with $k = 3$ minicolumns after 1000 $\nu$-cycles. E RFs of a macrocolumn with $k = 7$ minicolumns after 1000 $\nu$-cycles.

If there are fewer minicolumns than input patterns, the RFs still fall into two groups and specialize to one or a superposition of two or more input patterns afterwards (see Fig.12D). If there are more minicolumns than input patterns available, two or more RFs specialize to the same input pattern or occasionally to superpositions of two patterns (see Fig.12E). Note that a macrocolumn with more than five minicolumns usually needs longer than 1000 $\nu$-cycles to fully specialize its RFs. In the experiment of Fig.12B with $k = 5$ minicolumns, the case may occur that three RFs specialize to the upper left corner patterns such that there are only two RFs available for the lower right. A surplus of minicolumns can therefore be of advantage.

Hierarchical self-organization is also possible with small and constant $\lambda$. The RFs would take a longer time to fully differentiate in this case. However, a suitable constant $\lambda$ would still result in hierarchical self-organization which is in general superior to non-hierarchical. Further, a constant $\lambda$ would not require a time average over minicolumnar and macrocolumnar activities (see (31)). Note that we will come back to this discussion in Sec. 5.3.

Possible biological mechanisms for a change of $\lambda$ and a change of the threshold for synaptic plasticity $\chi$ might be modulations in the efficiency of AMPA- and NMDA-mediated forms of glutamatergic synaptic transmission (Watt et al., 2000). As discussed in (Abbott and Nelson, 2000), neurons scale down NMDA currents in response to enhanced activity, which in turn can make it more difficult to evoke LTP and easier to induce LTD. In our model the hierarchical formation of RFs essentially
requires a threshold $\chi$ which slowly decreases in time. Flexibility and learning time are significantly enhanced if the change of $\chi$ is appropriately coupled to mini- and macrocolumnar activity. Equations (29) to (33) represent one possibility\(^8\) to realize a well-working such coupling but it can, of course, only be regarded as an abstract and maybe more or less exact approximation of the dynamics which is really at work in cortical neurons.

### 3.4 Experiments of Hierarchical Receptive Field Self-Organization

Patterns $\vec{I}^1$ to $\vec{I}^5$ of Fig. 12A represent very simple input. In this section we investigate the behavior of the hierarchical system for different kinds of more complex input and we will also compare the results with results of non-hierarchical input. All input vectors are again visualized as two-dimensional patterns in order to conveniently display the vectors as well as the RFs of minicolumns as two-dimensional grey-level images. For all experiments we use a set of parameters which has shown to be flexible and has shown to result in a good performance in different tests. As we will work with inputs of different size $N$ we will, however, use different weights $\tilde{c}$ for afferent fibers than for internal connections. This is necessary because we need to keep the coupling $\kappa$ (20) of the input neurons to the macrocolumn dynamics independent of the input size $N$. Another value which we require to be independent of the input size $N$ (and also independent of the number of minicolumns $k$) is the average number of afferent fibers to a minicolumn per input neuron,

$$S_d = \frac{r m}{N}.$$  

(34)

$S_d$ is a kind of resolution of the minicolumnar RFs. The larger $S_d$ the more gradual are the values of the minicolumnar RFs $\vec{R}_a$ (compare Fig. 5). Very large $S_d$ are well approximating continuous valued minicolumnar RF weights but they require much longer computation times.

With fixed values for $\kappa$ and $S_d$ we can determine the values for $r$ (number of afferent connections per minicolumnar neuron) and their weights $\tilde{c}$:

$$r = \frac{N}{m} S_d, \quad \tilde{c} = \frac{m \kappa}{N S_d}.$$ 

(35)

For the following experiments we will again use $m = 100$ neurons per minicolumn. Again, each neuron receives $s = 15$ synapses from pre-synaptic neurons of the same minicolumn with weight $c = \frac{1}{s}$. We further choose $\kappa = 0.47$ and $S_d = 2.73$ because these values correspond to the values used for the experiments in Sec. 3.2, i.e., for $N = 16 \times 16$ we get $r = 7$ and $\tilde{c} \approx \frac{1}{s} = c$.

Most of the other parameters are identical or similar to the ones given in Sec. 3.2: the neurons’ constant threshold is $\Theta_o = \frac{1}{2}$ and the Gaussian threshold noise $\Theta_{no.}$ has a variance of $(\sigma_{no.})^2 = 0.01$. The oscillation of the inhibitory gain factor $\nu$ is determined by $\nu_{\text{init}} = 0.1$, $\nu_{\text{min}} = 0.5$, $\nu_{\text{max}} = 1.1$, $T_{\text{init}} = 3$, and $T = 25$. For the

\(^8\)see Sec. 5.3 for another possibility
Hebbian plasticity we use the parameters $E = 0.02$, $\chi_0 = \frac{1}{3}$, $\lambda_0 = 0.01$, $a_\chi = 0.85$, and $b_\chi = 0.15$. The time average in (28) is computed over the last $C = 200 \nu$-cycles. Note that all parameters are again independent of the number of minicolumns $k$.

### 3.4.1 Input with Hierarchical Similarity Structure

In this section hierarchically structured input is presented, i.e., the input contains a hierarchy of groups of input patterns which members are more similar to each other than to patterns of other groups. Using artificial data and hand-written digits, we show that hierarchical self-organization leads to an appropriate representation of the input in the case that much less RFs than input patterns exist. Furthermore, we compare hierarchical and non-hierarchical learning in this section.

**Artificial Data.** Let us consider a generalized experiment of the one in Fig. 12. As input on a layer of $16 \times 16$ input neurons we use a $7 \times 7$ square at 24 different positions. The squares are located at all possible positions within the grey region in Fig. 13B. In Fig. 13C RF self-organization is shown for afferent fibers which are subject to Hebbian plasticity defined by equations (26) to (33). As is shown in Fig. 12C the RFs first specialize coarsely and subsequently refine to a final specialization degree. In the experiment of Fig. 13 there are more input patterns than RFs, however. The RFs, therefore, specialize such that they nevertheless can, up to a high-degree, distinguish between the different inputs.

In contrast to Fig. 13C non-hierarchical RF self-organization is shown in Fig. 13D. Dynamics and parameters of the simulation in Fig. 13D are identical to the ones of the simulation in Fig. 13C except that the dynamical $\chi(t)$ in (26) is replaced by a constant one, $\chi = 0.55 \frac{1}{k}$. It can be seen that some RFs very quickly specialize to specific input patterns but that the majority of RFs remains unspecialized. Only after some time a formerly unspecialized RF specializes to some pattern and competes with similar RFs thereafter. Even after $2 \times 10^5$ or $2 \times 10^6 \nu$-cycles the representation is far from being as accurate as after 2000 or 20000 $\nu$-cycles in the hierarchical case.

The principle advantage of the hierarchical dynamics is that it, first, induces competition between large groups of RFs and only afterwards increases competition within specialized groups. This ensures that RFs compete with other similar RFs whereas in the non-hierarchical dynamics unspecialized RFs compete with specialized RFs (see Fig. 13D), which makes differentiation qualitatively worse and causes a much longer learning time.

For the simulation of Fig. 13D we have chosen the value of $\chi$ which was used in (Lücke and von der Malsburg, 2004). For smaller values of constant $\chi$ RFs specialize in the above experiment even slower than in Fig. 13D. For a larger value of $\chi$ the RFs specialize faster but the system’s ability to discriminate between correlated input patterns decreases. Even if we adjust the value of the constant $\chi$ to the above experiment in order to receive the best possible compromise between velocity of RF specialization and discrimination strength, we can neither qualitatively nor quantitatively reproduce the results of hierarchical self-organization.

The differences between hierarchical and non-hierarchical learning become less pregnant if the input patterns cover a larger fraction of the input layer or if the num-
Figure 13: **A** Five of the 24 possible input patterns. **B** Illustration of possible inputs. The $7 \times 7$ square can be at any position within the grey region. **C** Hierarchical self-organization of $k = 12$ minicolumnar RFs. **D** Non-hierarchical self-organization of $k = 12$ minicolumnar RFs.

Hand-Written Digits. We now investigate RF self-organization for a database of input patterns $\tilde{I}$ consisting of the hand-written digits zero, one, and seven. The database used is the freely available subset MNIST$^9$ of the NIST database. It consists of 60000 $28 \times 28$ large grey-level images. For the experiment we only used the digits zero, one, and seven (see Fig. 14A for some examples). We have chosen this data because it, first, consists of two-dimensional patterns and is therefore easily displayable, second, it is commonly accessible, and, third, it can be expected to contain a hierarchy of subclasses of mutually similar patterns. In Fig. 14B RF self-organization is displayed for a set of $N = 28 \times 28$ input neurons and a macrocolumn.

---

$^9$MNIST database of hand-written digits, NEC, yann.lecun.com/exdb/mnist/
Figure 14: A Subset of input patterns of the hand-written digits zero, one, and seven of the MNIST database. B RF modification of a macrocolumn with $k = 8$ minicolumns if input of the form as displayed in A is presented. C RFs of a macrocolumn with $k = 4$ minicolumns after 5000 $\nu$-cycles. D RFs of a macrocolumn with $k = 12$ minicolumns after 5000 $\nu$-cycles. E RFs of a macrocolumn with $k = 12$ minicolumns and non-hierarchical dynamics with $\chi = 0.55 \frac{1}{k}$ after 5000 $\nu$-cycles.
with standard set of parameters and $k = 8$ minicolumns. From 0 to 100 $\nu$-cycles the randomly initialized RFs specialize to the region where the input neurons are most active. Some RFs coarsely show preferences to stimuli in the form of vertical bars, e.g., $\bar{R}^3$ and $\bar{R}^4$, and others are rather sensitive to circular stimuli, e.g., $\bar{R}^1$. During the first 100 $\nu$-cycles the RFs are rather unstable and can frequently change between various kinds of input pattern mixtures. In the period from 100 to 500 $\nu$-cycles the RF change is still high but steadily decreases until, after about 500 $\nu$-cycles, two groups of RFs have formed. Group 1, consisting of $\bar{R}^1$, $\bar{R}^6$, $\bar{R}^7$, and $\bar{R}^8$, is sensitive to zero-type patterns whereas group 2 is sensitive to one and seven-type inputs or mixtures of them. The groups remain stable in the sense that no RF of one group changes such that it becomes sensitive to a pattern of the other group. From 500 to 1000 $\nu$-cycles the RFs of each group further specialize to subpatterns of their input type. This is especially obvious for group 2 where $\bar{R}^2$ and $\bar{R}^3$ form a subgroup specialized to one-type patterns and $\bar{R}^5$ and $\bar{R}^6$ form a subgroup specialized to seven-type patterns. From 1000 to 5000 $\nu$-cycles the specialization continues and the formed subgroups decay into subgroups themselves. After 5000 $\nu$-cycles each RF is sensitive to a different characteristic type of input patterns, e.g., RFs $\bar{R}^2$ and $\bar{R}^3$ are both sensitive to one-type input but $\bar{R}^5$ is rather sensitive to the subtype of vertical lines whereas $\bar{R}^2$ is sensitive to diagonal patterns of hand-written ones.

Note that the course of self-organization is only sensitive to the overlap of input patterns and that there is no neighborhood relationship between the input pixels. The experiment would produce the same results if all input pixels were permuted. Induced by the overlap, a neighborhood relationship only exists between the patterns themselves and as they are two-dimensional in nature, hierarchical self-organization is reflecting this relationship.

If we apply a macrocolumn with less minicolumns to the data, the final subdivision is not as fine, see Fig. 14C for $k = 4$, and if there are more minicolumns available, the subdivision is finer, see Fig. 14D for $k = 12$. For $k = 8$ the system most often specialized four, two, and two RFs to patterns of type zero, one, and seven, respectively. Sometimes it specialized three RFs to seven-type input but usually only two for one-type input.

For comparison, the RFs of non-hierarchical self-organization with constant $\chi = 0.55 \frac{1}{k}$ are shown after 5000 $\nu$-cycles in Fig. 14E. After $5 \times 10^4$ or $5 \times 10^5$ $\nu$-cycles the number of specialized RFs increases but the representation does not reach the accuracy as with hierarchical self-organization after 5000 $\nu$-cycles.

We have seen that hierarchical RF self-organization works well even for strongly varying patterns. The RFs form classes of patterns by hierarchically decaying into increasingly smaller groups. Although the groups solely form on the basis of pattern overlap they represent abstractions of typical hand-written instances of the different digits. The overlap similarity relationship is independent of any n-dimensional space structure such that the system can during learning be expected to become sensitive to significant classes of more general three or n-dimensional input or of input with much more complex topological structure.

---

10Digit recognition systems are generally using more invariant input representations of hand-written digits, of course.
3.4.2 The Continuous Bars Test

In Sec. 3.4.1 hierarchical RF self-organization found an appropriate representation for the input patterns although many more input patterns existed than RFs were available. The RFs specialized to patterns which are, in the sense of pattern overlap, located, e.g., at about equidistant positions of the subspace of the input space in which the patterns occur (compare Fig. 13C). In this section we will present a class of input patterns which can only be appropriately represented if the system extracts basic features from the input and uses distributed neural coding for pattern representation.

Overlapping Vertical Bars. Let us first more systematically study the ability of the system to accurately represent the subspace of training patterns. On a set of \( N = 16 \times 16 \) input neurons we take an input vector to consist of exactly one vertical four pixel wide bar (see Fig. 15A). We use cyclic boundary conditions such that there are 16 different such inputs. Note that a similar experiment without cyclic boundary conditions was discussed in (Spratling, 1999). In Fig. 16A RF self-organization for

![Figure 15](image)

Figure 15: A 10 randomly chosen input patterns of vertical four pixel wide bars. B RF modification of a macrocolumn with \( k = 8 \) minicolumns if input of the form as displayed in A is presented.

the same input database is visualized in another way. The sensitivity of a RF is displayed as an arrow which orientation corresponds to the mean x-axis pixel position of bars the RF is sensitive to, \( \phi = \frac{x}{10} \times 360^\circ \), and the length of the arrow corresponds to the degree of specialization\(^{11} \). Both visualizations show that the RFs rapidly cover the input pattern space within the first 50 \( \nu \)-cycles. After 50 \( \nu \)-cycles the input space is rather unevenly covered, however, and there are RFs specialized for the same region. From 50 to 500 \( \nu \)-cycles the specialization degree increases until the input pattern space is evenly covered by the RFs, a property which is especially well observable in Fig. 16A. For a number of minicolumns smaller or larger than \( k = 8 \)

\(^{11}\)0 means no bar and 1 means exactly one bar is preferred.
Figure 16: A Visualization of RF specialization for a macrocolumn with $k = 8$ minicolumns if input of the type as in Fig. 15A is presented. Each arrow corresponds to a RF. An arrow’s angle corresponds to the mean x-axis position of the vertical RF, $\phi = \frac{\pi}{16} \times 360^\circ$. An arrow’s length is zero if no bar is preferred and one (the circles radii) if exactly one bar is preferred. The RFs of one macrocolumn during one simulation are displayed after 10, 50, and 500 $\nu$-cycles. B Visualization of macrocolumns with $k = 4, 8, \text{and} 12$ minicolumns after 5000 $\nu$-cycles if input of the type as in Fig. 15A is presented.

RF self-organization is similar and the space of input patterns is, respectively, more sparsely or more densely covered (see Fig. 16B).

The experiment shows that even if the input does not contain any prominent classes of mutually similar patterns, the system organizes its RFs such that the input patterns can be processed appropriately. The RFs organize in a one-dimensional fashion, i.e., each RF has exactly two nearest neighbors with respect to RF similarity. The RFs can therefore be displayed like in Fig. 16. Note that the one-dimensional nature of the RF self-organization is solely induced by the one-dimensional overlap relation between the bars. The overlap interrelations of the input patterns themselves induce the dimensionality or, more general, the topology of the RF self-organization much like the distance interrelation of a metric space gives rise to its topology. If the input of a $16 \times 16$ array of input neurons does not consist of all possible vertical bars but of all possible squares of a certain size, e.g. $4 \times 4$, the RFs evenly cover a two-dimensional space\textsuperscript{12} after learning.

\textsuperscript{12}a two-dimensional torus for cyclic boundary conditions
The Continuous Bars Test. We now combine the bars test of Sec. 3.2 and the above demonstrated ability to represent continuous bars to what we will call the continuous bars test. The input patterns we use consist of superpositions of horizontal and vertical bars of a width of four pixels. On an input array of $16 \times 16$ neurons with cyclic boundary conditions a total of $b = 32$ different bars can occur. Each bar we take to appear in an input image with probability $p_o$ (see Fig. 17A for some examples with $p_o = \frac{1}{b}$). In Fig. 17B RF self-organization is shown for $b = 32$

![Input patterns](image)

Figure 17: A 20 randomly chosen input patterns of a bars test with 8 bars and continuously varying bar positions. B RF modifications of a macrocolumn with $k = 10$ minicolumns if input of the form as displayed in A is presented. C RFs of a macrocolumn with $k = 8$ minicolumns after 5000 $\nu$-cycles. D RFs of a macrocolumn with $k = 12$ minicolumns after 5000 $\nu$-cycles. E RFs of a macrocolumn with $k = 12$ minicolumns and non-hierarchical learning after 5000 $\nu$-cycles.

bars with $p_o = \frac{1}{b}$ and $k = 10$ minicolumns. In the time from 0 to 500 $\nu$-cycles the RFs are not very stable and their degree of specialization is very low. During learning they become more and more stable, however, and from about 500 $\nu$-cycles on their specialization degree increases until a final RF configuration is established somewhen between 2000 and 5000 $\nu$-cycles. In all simulations with this input and $k = 10$ the x-axis positions of the vertical RFs and the y-axis positions of the hor-
Horizontal RFs were evenly spaced after 5000 $\nu$-cycles (see Fig. 18 for an example). Note that the system is stable against various perturbations of the bars test’s parameters such as noise, bar width variations or variations of $p_o$. A macrocolumn with $k = 10$ minicolumns with an array of $N = 16 \times 16$ input neurons and standard set of parameters usually converges for a continuous bars test with $b = 32$ bars and $p_o = \frac{1}{b}$ to a final RF specialization within the first 2000 $\nu$-cycles. If one RF specializes to a superposition of horizontal and vertical bars like $\vec{R}_{\theta}$ in Fig. 17B, the system needs up to about 3000 $\nu$-cycles. Note that it is not possible to determine a time after which the system has found a representation of all bars. For other numbers of minicolumns, see Fig. 17C,D for $k = 8$ and $k = 12$, the outcome of the experiment is the same, i.e., half of the RFs evenly cover the space of vertical bars and the other half the space of horizontal bars. Occasionally, however, the numbers of vertical and horizontal RFs differ by a small number.

In Fig. 17E the RFs of non-hierarchical self-organization are shown for $\chi = 0.5 \frac{b}{k}$ after 5000 $\nu$-cycles. As can be observed, the system does not use all its RFs to represent the space of input patterns, which leads to a less accurate representation than in the hierarchical case.

**The Classical Bars Test.** The hierarchical system can, of course, also be applied to the classical bars test and we will shortly put forward some results in order to be able to compare hierarchical RF self-organization with non-hierarchical and with results of other suggested systems. The input data of the bars test consist of patterns

![Figure 18: Visualization of RF specialization for a macrocolumn with $k = 10$ minicolumns after 5000 $\nu$-cycles if input of the type as in Fig. 17A is presented. Each arrow corresponds to a RF. An arrow in the left-hand-side (lhs) diagram corresponds to a vertical RF and an arrows in the right-hand-side (rhs) diagram corresponds to a horizontal RF. An angle of an arrow in the lhs diagram corresponds to the x-axis position of a vertical RF, $\phi_{\text{left}} = \frac{2\pi}{16} \times 360^\circ$, and the angle of an arrow in the rhs diagram corresponds to the y-axis position of a horizontal RF, $\phi_{\text{right}} = \frac{\pi}{16} \times 360^\circ$. An arrow’s length is zero if no bar is preferred and one (the circles radii) if exactly one bar is preferred.

![Diagram showing vertical and horizontal RFs with angles and lengths indicating the specialization process.](image-url)
Figure 19: A 20 randomly chosen input patterns of a bars test with 8 bars. B Hierarchical and non-hierarchical learning of bars. The plots show self-organization in the course of a bars test with $k = 10$ minicolumns and $b = 8$ bars. A bar is taken to be represented if a minicolumn remains active in at least nine of ten $\nu$-cycles if the bar is shown. The number of represented bars is measured at different time steps (every 5 $\nu$-cycles before 500 $\nu$-cycles, every 10 between 500 and 1000, and every 20 after 1000 $\nu$-cycles) and the average number of represented bars is computed using 200 simulations. The black graph shows hierarchical learning and the grey graph shows non-hierarchical learning with $\chi = 0.55\frac{1}{k}$. Left limits of the error bars show the learning time after which there is a 20% probability that 3, 4, 5, 6, and 7 bars are represented, respectively. Right limits show the time after which there is an 80% probability. Black error bars belong to hierarchical learning and grey bars to non-hierarchical.
of superpositions of disjunct horizontal and disjunct vertical bars. In an input pattern (16\times 16\text{ pixel in this case}) each bar occurs with the same probability $p_o$ (here $p_o = \frac{1}{k}$) and all bars are of equal size (four pixel width in our case).

On first sight, hierarchical self-organization does not promise to be superior to non-hierarchical self-organization in this experiment because the input does not contain a hierarchy of similarity classes. However, in all experimental settings the hierarchical approach shows a shorter learning time and smaller variations between simulations. For $b = 8$ bars and $k = 10$ minicolumns there is a 50\% probability that the system has found all bars after about 650 $\nu$-cycles. After about 500 $\nu$-cycles there is a 20\% probability and after 800 we have a probability of about 80\% that all bars are represented (we used 100 simulations for the measurements). A non-hierarchical system with constant $\chi = 0.55 \frac{1}{k}$ needs for the same bars test longer for learning: it requires about 1000 $\nu$-cycles for a 50\% probability that all bars are represented (about 2400 $\nu$-cycles to find all bars with a probability of 80\%).

In Fig. 19B RF specialization for $b = 8$ bars and $k = 10$ minicolumns for hierarchical and non-hierarchical ($\chi = 0.55 \frac{1}{k}$) learning is plotted. As can be seen, the learning time average of both systems is comparable until about six bars are represented. The non-hierarchical system subsequently needs on average a very long time to represent the remaining bars (see Spratling and Johnson, 2002, for a similar effect). For other numbers of bars or minicolumns (including perturbations such as noise of bar width variations) the qualitative differences between hierarchical and non-hierarchical learning are the same and in all test settings hierarchical learning has shown to be superior to non-hierarchical learning, although hierarchical self-organization is more suitable for data with hierarchical similarity structure (see, e.g., Fig. 14B).

### 3.5 Conclusion and Discussion

Activity dependent Hebbian plasticity (26) to (31) coupled to a dynamics of spiking neurons based on columnar organization and background oscillation was shown to result in self-organization of minicolumnar RFs. Self-organization appropriately adapted the minicolumn sensitivities to the patterns of different databases. Hereby, the database can contain nested similarity classes as in Sec. 3.4.1, continuously overlapping patterns, superpositions of patterns, or superpositions of continuously overlapping patterns (all Sec. 3.4.2). The experiments further show that hierarchical learning is functionally advantageous to non-hierarchical learning as presented. Hierarchical self-organization develops stronger discriminating RFs, uses all RFs to represent the input, and is faster than non-hierarchical self-organization. The superiority of the hierarchical approach is especially significant for larger numbers (about $k > 10$) of minicolumns, which further supports the necessity of an hierarchical approach if we consider an estimation of about 80 minicolumns per macrocolumn, e.g., in primary cortical areas (Favorov and Diamond, 1990; Mountcastle, 1997).

The ability of the hierarchical system to adapt its neuron sensitivities such that the input pattern space is uniformly covered is similar to the ability of self-organizing maps (SOMs) (Kohonen, 1990; Kohonen, 1995) to appropriately cover a space of input data. However, SOMs require a predefined distance measure for the input...
space as well as a predefined dimensionality of the SOM output space. The system presented here forms an appropriate topological representation of the input space solely on the basis of input pattern overlap. It can, therefore, also process input data whose topology is not equivalent to the one of an n-dimensional Euclidean space. Thus, the system is rather comparable to a Growing Neural Gas\textsuperscript{13} (GNG) (Fritzke, 1995) with respect to its topological flexibility. Note, however, that also for the GNG a predefined distance measure between points of the input space is used whereas our system uses overlap of extended patterns (see, however, Sec. 4).

A second and more significant difference between the system presented here and SOMs or GNGs is the ability of our system to learn from pattern superpositions and to extract the basic constituents of the patterns as was demonstrated in the bars test. Furthermore, the neural architecture of our column based system with spiking neurons is quite different from the one of SOMs or GNGs and we refer to Sec. 2.5 for a discussion of functional and neuroscientific aspects of the macrocolumnar neural dynamics. A system which was applied to input like in the first experiment in Sec. 3.4.2 was described in (Spratling, 1999). In contrast to our system, the computational units of the network presented there had to be explicitly one-dimensionally interconnected in order to solve the task.

There are various unsupervised systems able to find clusters of mutually similar or neighboring input. If the input data is continuously distributed over a manifold with or without well-defined dimensionality, SOMs and GNGs are, respectively, a popular choice to appropriately represent the data. Again other systems have successfully been applied to the bars test, e.g. (Földiák, 1990; Dayan and Zemel, 1995; Hinton et al., 1995; Hinton and Ghahramani, 1997; Charles and Fyfe, 1998; Hochreiter and Schmidhuber, 1999; O’Reilly, 2001; Spratling and Johnson, 2002), and thus demonstrated their ability for distributed neural coding. The type of self-organization presented in Sec. 3.3 results, for input as in Sec. 3.4.1, in a natural nested series of RF subdivisions and in a high discrimination ability due to competition between already similar RFs. In the bars test RF self-organization as presented here is highly competitive to other suggested systems and for continuously overlapping input patterns hierarchical self-organization results in RFs which evenly cover the corresponding input pattern space.

RF self-organization functionally distinguishes the system from the majority of other especially neuronal systems which can be applied to the one or other input type. The system presented in this section, with one set of parameters, can not only successfully be applied to all different input types just discussed but, furthermore, is able to process input consisting of mixtures of the different types. In the continuous bars test the system’s RFs formed two groups, one specialized to horizontal and one group specialized to vertical bars. Simultaneously, each group’s RFs organized such that they evenly covered the space represented by their group (see Fig. 18). In being able to pass the continuous form of the bars test the system offers unequaled opportunities especially if compared to other artificial neural networks.

In this section we considered self-organization of afferents to a single model macrocolumn. In the neocortex, however, a number on the order of roughly $10^6$\textsuperscript{13}a further development of SOMs

\textsuperscript{13}a further development of SOMs
mutually interconnected macrocolumns is estimated (Mountcastle, 1997). In a large number of experiments, the interconnection structure between the macrocolumns or rather between the minicolumns were investigated, (see e.g. Rathjen et al., 2002, for a recent study in the visual cortex). The interconnections were found to change in time (Callaway and Katz, 1990; Chapman et al., 1996) and to reflect the diversity of the environment in which the subject grew up, e.g. (Schmidt et al., 1997). The statistical prevalence of collinear contours in real world images (see, e.g., Krüger, 1998; Kaschube et al., 2001), for instance, is reflected by the prevalence of interconnections between corresponding orientation sensitive columns in the visual cortex (see, e.g., Schmidt et al., 1997). In the primary visual cortex the interconnections can be shown to be unspecific at first, i.e., no specialization to specific columns, which is measured via the clustering of horizontal interconnections, occurs during the first postnatal days or weeks (Ruthazer and Stryker, 1996; Chapman et al., 1996). The interconnections only gradually specialize to form interconnections between orientation tuned columns. They first form coarsely clustered interconnections which gradually refine in time. The refinement can hereby be shown to be the result of synaptic plasticity rather than cell death and to be input dependent (Ruthazer and Stryker, 1996; Chapman et al., 1996). Although the interconnection structure and its development is best studied in primary sensory areas the same mechanisms are believed to underly horizontal interconnection organization in all cortical areas.

For the inter-columnar connections to be able to specialize to appropriate cells, an input-driven organization principle can be assumed which has properties similar to the type of self-organization presented here. In networks of macrocolumns hierarchical self-organization of minicolumnar RFs has the advantage to gradually structure the network from coarse to fine. In non-hierarchical self-organization a newly (and formerly poorly) specialized RF would cause other inter-macrocolumnar connections to reorganize in order to consider the new RF. Reorganization is much less efficient than refinement induced by hierarchical learning, however. Furthermore, a coarse to fine process is not only observed neuroanatomically but is also consistent with our view of high level learning.

The RFs of biological minicolumns of a macrocolumn have to differentiate to appropriately represent input which can be expected to consist of superpositions of spike patterns with continuous overlaps. E.g., orientation columns in the primary visual cortex are sensitive to basic constituents of natural images which are often mathematical abstracted using Gabor wavelet-filters (Jones and Palmer, 1987). Another example are columns in MT\footnote{Note that more examples can be found in somatosensory areas, auditory areas etc.} which are sensitive to different directions of motion (Albright et al., 1984). The biological columns equidistantly cover the space of all possible orientations and all possible directions of motion, respectively. Natural input consist, for instance in the case of orientation columns, of superpositions of stimuli for different orientations which almost certainly lie in between of the represented orientations. This is, however, the situation which was studied in the continuous bars test in which hierarchical self-organization has shown to result in appropriate RFs. A more detailed analysis, e.g., of natural image processing, also has to take into account preprocessing which takes place already before the opti-
Self-Organizing Receptive Fields

3

cal nerve fibers connect to cortical neurons. A future study of RF self-organization using such preprocessed input can be expected to produce interesting results.

Our system models self-organization of RFs of a single cortical macrocolumn (Favorov and Diamond, 1990; Mountcastle, 1997) and is therefore not directly comparable to large scale models (Baxter and Dow, 1989; Obermayer et al., 1990; Tanaka, 1990; Niebur and Wörgötter, 1993; Miller, 1994; Bednar et al., 2002; Prodöhl et al., 2003, and others) which intend to study map creation mainly in the visual cortex (see Erwin et al., 1995; Swindale, 1996, for overviews). We rather suggest hierarchical self-organization as an underlying mechanism which can lead to structured afferent connections as well as to structured inner-cortical connections. In this context, the presented synaptic plasticity has proven to adapt minicolumnar RFs to various kinds of inputs in a for pattern representation especially suitable and, compared to other systems, extraordinarily flexible manner. Hierarchical self-organization matches properties required from the plasticity of biological RFs and is modeling their hierarchical formation. Our model, motivated by macrocolumn connectivity, has neurodynamical properties that solve important conceptual problems of neurophysiology. The spiking character of neurons, column based interconnection structure, oscillatory inhibition, and Hebbian plasticity are shown to combine to form an advanced information processing system which allows to solve problems such as pattern classification and feature extraction, where it was shown to be highly competitive to other recent systems in different versions of the bars benchmark test.
4 Clustering with Receptive Field Self-Organization

Spike patterns of input neurons so far encoded two-dimensional black-and-white or grey-level level images. These types of input made it possible to conveniently visualize the input as well as the RFs of minicolumns. However, the input neurons of the system are not subject to any kind of neighborhood relationship and can, therefore, be used to encode many different kinds of input.

In this section we will use the input neurons to encode continuous valued input vectors and apply the macrocolumn model dynamics to the problem of data clustering, i.e., unsupervised data classification. Note that the results of this section were published in (Lücke, 2004a). First, we will need to introduce the encoding of \( D \)-dimensional continuous valued input vectors by spike patterns of the set of input neurons.

4.1 Place Coding of Continuous Variables

There are various approaches to model the representation of continuous variables in neural networks. The different coding schemes range from spike rates of single neurons, time to first spike, fractions of spiking neurons in a neuron population to temperature scale and place coding (see, e.g., Kandel et al., 1991; Maass and Bishop, 2001, for overviews). In our model we use a neural place code. In a linearly ordered set of neurons the position of the most active neurons is representing the value of the input. As neuron model we use the discrete time model with refraction time of Sec. 3.1:

\[
I_{\text{badc}} = P_{L_9;}\, I = < L_\text{V} > \, P_{n_9;} \quad (36)
\]

where \( \text{V}_9 \) is given in (1) and where \( B_i(w_i) \in \{0,1\} \) is a Bernoulli process with probability \( W(0) = 1 - w_i \) and \( W(1) = w_i \). As demanded from the input neurons in (21), \( n^{E}_i(t + 1) \) is equal to one only if, first, \( B_i(w_i) \) is equal to one and, second, if the neuron has not been active the time-step before.

To place code a continuous variable \( x \in [0,1] \) we use a linearly ordered set of \( M \) neurons of the above type and define the probabilities \( w_i \) as follows (compare (21)):

\[
w_i = \frac{\pi_i}{1-p_i}, \quad p_i = \frac{1}{3} \exp \left( -\frac{(x_i-x)^2}{2\sigma^2} \right), \quad (37)
\]

\[
x_i = \frac{1+2b}{M-1} (i-1) - b. \quad (38)
\]

The neural encoding is illustrated in Fig. 20. Note that the weights \( w_i \) are not directly set by the input as in (21) but are indirectly translating a continuous value. Definitions (37) and (38) result in a set of active neurons which are symmetrically grouped around the center \( x \) and whose activity probabilities decrease in a Gaussian manner. The equations translate the continuous input \( x \) to activities of neurons at discrete positions \( x_i \). To equally treat all input values in \([0, 1]\) the neurons cover the extended interval \([-b, 1+b]\). The first equation in (37) sets the probabilities \( w_i \) such that the actual activity probabilities of the neurons are given by \( p_i \). The maximal possible activity is \( \frac{1}{3} \). Note that equations (37) and (38) model the encoding of a stimulus by a population of broadly tuned neurons, which is a common and actively
discussed encoding in experiment and theory (see, e.g., Eurich and Wilke, 2000, for further reading). In Fig. 20 the activity of a set of \( M = 100 \) input neurons is dis-

![Figure 20: Encoding of a continuous number \( x \in [0, 1] \) using the parameters \( b = 0.2, \sigma = 0.1, \) and \( M = 100 \) input neurons. A Gaussian function sets the activity probabilities of the neurons at positions \( x_1 \) to \( x_M \). A typical resulting spike pattern is shown in the lower part of the image. Every second line of pixels visualizes the activity of the \( M \) neurons (black lines were inserted for visualization purposes). A black pixel marks an inactive neuron and a white pixel an active one.](image)

played for an input value \( x = 0.34 \). To estimate the activity probability of a single neuron several time steps are necessary. Note, however, that in a single time step the neural activity of the whole set of neurons is appropriately representing the input. The more input neurons there are the more accurate the representation.

To represent a vector \( \vec{I} \) with \( D \) different vector entries in the interval \([0, 1]\), \( \vec{I} \in [0, 1]^D \), we encode the entries using \( D \) different sets of \( M \) linearly ordered neurons. Each set we take to represent the entry as described above. The \( N = DM \) neurons (36), which encode the vectors \( \vec{I} \in [0, 1]^D \), now serve as input neurons of the neural network model of the cortical macrocolumn. See Fig. 21 for a sketch of a system with \( k = 3 \) minicolumns encoding \( D = 2 \) dimensional input vectors. We use hierarchical synaptic plasticity as defined in Sec. 3.3.

### 4.2 Receptive Field Self-Organization

We can now apply the system described above to the problem of unsupervised data classification. Given a database of \( D \)-dimensional inputs \( \vec{I} \in [0, 1]^D \) the task is to group the input vectors \( \vec{I} \) into classes of mutually similar vectors. We operate the
Figure 21: Sketch of a macrocolumn with its continuous number encoding input neurons. A $D = 2$ dimensional input vector $\vec{I}$ is translated into spike patterns of $N = DM = 24$ neurons (the grey levels of the neurons indicate activity probabilities). The neurons serve as input neurons for a macrocolumn of $k = 3$ minicolumns. A minicolumnar neuron receives $s = 3$ synapses from internal neurons and $r = 2$ afferents from the input neurons. Only the RF of minicolumn $\alpha = 1$, which is specialized to input vectors similar to $\vec{I}$, is displayed. Inhibition is visualized symbolically.

system as described in Sec. 3.3 with the same parameters as used for the experiments in Sec. 3.4.

During each $\nu$-cycle we present a vector $\vec{I}$ which is randomly chosen from the database, i.e., we compute the values $w_i$ in (37) with a randomly chosen $\vec{I}$ once at the beginning of each $\nu$-cycle. After starting with a random initialization of afferent fibers ($R_{ij}^{\alpha}$) the minicolumnar RFs specialize in the course of self-organization. The system first tries to subdivide the data vectors into coarse subclasses and tries to find an increasingly finer subdivision as time proceeds.

4.2.1 Visualization

If a vector $\vec{I}$ is presented to the system, it reacts by deactivating the minicolumns with least suitable RFs during a $\nu$-cycle. For a given vector $\vec{I}$ this process is not deterministic due to the Bernoulli process in (36) and neural threshold noise in (19).
To quantify the response of the system to a given input vector $\vec{I}$ we, therefore, use the probabilities $q_\alpha(\vec{I})$ that minicolumn $\alpha$ is active at the end of a $\nu$-cycle if input $\vec{I}$ is presented. If $\vec{I}$ is presented during each of $n$ $\nu$-cycles, $q_\alpha(\vec{I})$ can be estimated by counting the times $\tilde{n}$ that minicolumn $\alpha$ remains active, $q_\alpha(\vec{I}) \approx \frac{\tilde{n}}{n}$ (we use $n = 50$ and will also denote the estimation by $q_\alpha(\vec{I})$). To visualize cluster separation by RF self-organization we define the scalar functions:

$$Q(\vec{I}) := \frac{\max_\beta \{q_\beta(\vec{I})\}}{\sum_\beta q_\beta(\vec{I})}, \quad Q_\alpha(\vec{I}) := \frac{q_\alpha(\vec{I})}{\sum_\beta q_\beta(\vec{I})}.$$  

(39)

$Q(\vec{I})$ is equal to one if always one and the same minicolumn remains active. $Q(\vec{I})$ is smaller than one if more than one minicolumn remains active or if the minicolumns which remain active are different from $\nu$-cycle to $\nu$-cycle.

In the following we will visualize the RF self-organization process for $D = 2$ by measuring the values of $Q(\vec{I})$ for all inputs $\vec{I}$ on a grid of $100 \times 100$ points in the interval $[0, 1] \times [0, 1]$. The value of $Q(\vec{I})$ defines the grey level of a corresponding pixel and the system reaction can thus be displayed as a $100 \times 100$ pixels grey-level image. Note that to display the complete system behavior grey-level images of all $Q_\alpha(\vec{I})$ would have to be displayed. Sufficient insight is already gained, however, by displaying the corresponding image of the scalar $Q(\vec{I})$.

For a $D = 2$ dimensional database of 300 data points cluster separation is shown in Fig. 22. After random initialization of minicolumnar RFs the system’s behavior is unspecific, which is reflected by relatively uniformly distributed values of $Q(\vec{I})$. After 25 $\nu$-cycles the system’s reactions become more specific, which is indicated by higher values of $Q(\vec{I})$ in the vicinity of the input data points. After 50 $\nu$-cycles a first subdivision of the input data is clearly visible. If a data point of the cluster in the lower part of the image is presented, there is one specific minicolumn activated with a probability of virtually 100%. A presentation of a data point in the upper half of the image results in an activation of one of the other minicolumns or in both of them. From 200 to 500 $\nu$-cycles the system subdivides the data points in the upper half into two classes. After 500 $\nu$-cycles RF specialization further increases up to a final degree after 1000 to 5000 $\nu$-cycles.

After the system has reached its final degree of RF specialization, it can be used as a classifier for data points of the database or for other data points of the same type. A data point is classified by the minicolumn or minicolumns which remain active in the end of a $\nu$-cycle if $\vec{I}$ is presented. If a data point of the upper right cluster is presented, the system reacts, in the course of a $\nu$-cycle, by first switching off the minicolumn which corresponds to the lower cluster and subsequently switches off the minicolumn corresponding to the upper left cluster. If a data point is presented which lies in between the upper two clusters, the system switches off the minicolumn corresponding to the lower cluster and leaves the other two minicolumns active\(^{15}\). This shows that classification like learning is hierarchical in nature and that the values $Q_\alpha(\vec{I})$ for a data point $\vec{I}$ contain information about decision reliability as well as information about the location of $\vec{I}$ relative to the clusters.

\(^{15}\)Note that this behavior can only be observed by also monitoring all functions $Q_\alpha(\vec{I})$.  

Figure 22: Clustering with a macrocolumn of $k = 3$ minicolumns. Minicolumnar RF self-organization is visualized using the scalar function $Q(\vec{I})$. $\vec{I}$ is a point of the two-dimensional interval $[-0.2, 1.2] \times [-0.2, 1.2]$ and $Q(\vec{I})$ is an indicator of the reliability of the system’s reaction. The crosses mark the data points used as input for the simulation. To generate an image after a given learning time the values of $Q$ are estimated on a raster of $100 \times 100$ input vectors $\vec{I} \in [-0.2, 1.2]^2$. For each raster point the system’s reaction was measured for 50 $\nu$-cycles.
4.2.2 Clustering of More Complex Data

In the experiment of Fig. 22 the system separated three clusters with about the same number of data points which were distributed in an approximately Gaussian manner. In Fig. 23A the classification of a database with very unequal numbers of data points per cluster with non-Gaussian distribution is shown. Class 5, e.g., contains about 200 data points (see Fig. 23A) whereas class 1 contains merely about 20.

![Figure 23: Clustering with a macrocolumn of $k = 5$ minicolumns. The separation is visualized in the same way as in Fig. 22 using the function $Q(\tilde{I})$. A Cluster separation after 10000 $\nu$-cycles if during each $\nu$-cycle one single data point was shown. B Cluster separation after 10000 $\nu$-cycles if during a $\nu$-cycle superpositions ($p_o = 2.0$) of data points were shown.](image)

If for the database of Fig. 23A a macrocolumn with $k = 3$ minicolumns is used, the system tries to find three appropriate classes (a number of classes much less suitable). If more than $k = 5$ minicolumns are used, the system tries to find more classes. The number of different classes that are finally found depends on the data itself. If more minicolumns are available than clusters exist, the system specializes more than one minicolumnar RF to the same cluster. Thus, the minicolumns which correspond to the same cluster remain simultaneously active if a data point of the cluster is presented. Subsequent stages of neural information processing based on population activity correlation would, therefore, automatically merge all minicolumns corresponding to the same cluster.

4.2.3 Data Point Superpositions

In the experiments above one input vector per $\nu$-cycle was presented during the learning phase. If a database is given, there is no reason to present different data points simultaneously. For a biological neural network the situation is quite different, however, because there is no absolute control over the input data. If the input
neurons of the biological system encode sensory data, it usually consists of superpositions of stimulations from different sources. A biological network must therefore be able to cope with data point and, hence, spike pattern superpositions. To account for such a situation we train our system with data point superpositions. Given a database of \( L \) data points we first randomly choose a set of data points \( \{ \vec{I}^j | j \in \mathcal{J}_r \} \) and subsequently compute the activity probability, \( p^d_i \), of the \( i \)-th neuron of the neuron set of vector entry \( d \in [1, \ldots, D] \):

\[
\mathcal{J}_r = \{ j \in [1, L] | B^j_\tau \left( \frac{p_0}{L} \right) = 1 \}, \quad p^d_i = \max_{j \in \mathcal{J}_r} \{ p_i(I^j_d) \},
\]

where \( \vec{I}^j \) with \( j \in \{ 1, \ldots, L \} \) is a data point and \( p_i(x) \) is given in (37). \( B^j_\tau \) is a (for each \( j \) and \( \tau \) different) Bernoulli process with probability \( W(0) = 1 - \frac{p_0}{L} \) and \( W(1) = \frac{p_0}{L} \). For convenience we do not use empty input vectors for learning.

In the experiment of Fig. 23B we used the same network and the same database as for the experiment of Fig. 23A but with spike patterns generated according to (40) with \( p_0 = 2.0 \). Although the majority of inputs consists of data point superpositions the final classification of the database is qualitatively the same as for non-superimposed input. If the network is trained with input vector superpositions, the class boundaries become less sharp and the system needs longer to build up appropriate classes (it needs about 2800 \( \nu \)-cycles compared to about 1200 \( \nu \)-cycles for non-superimposed data). Learning time increases if \( p_0 \) gets larger and for the database of Fig. 23 clustering within 10000 \( \nu \)-cycles starts to fail for \( p_0 > 3.0 \).

### 4.2.4 Higher Dimensional Data

So far we studied clustering of databases with two-dimensional input vectors \( \vec{I} \). The system can, of course, be applied to data consisting of any vectors of finite dimension. The most popular higher dimensional database for clustering is probably Fisher’s iris database (Fisher, 1936). It consists of 150 input vectors of dimensionality \( D = 4 \). The input vectors \( \vec{I}^j \) (where \( j \in \{ 1, \ldots, L \} \) and \( L = 150 \)) are not restricted to the interval \([0, 1]^D\) such that they have to be rescaled:

\[
J^\text{max}_d = \max_{j=1, \ldots, L} \{ J^j_d \}, \quad J^\text{min}_d = \min_{j=1, \ldots, L} \{ J^j_d \}, \quad J^j_d = \frac{J^j_d - J^\text{min}_d}{J^\text{max}_d - J^\text{min}_d},
\]

The vectors \( \vec{I}^j \) can now be used as input vectors for a system with \( DM = 400 \) input neurons. Learning of a system with \( k = 3 \) minicolumns with parameters as given in Sec. 3.4 results in self-organization of minicolumnar RFs. The system was trained for 10000 \( \nu \)-cycles. After 5000 \( \nu \)-cycles the learning rate \( \mathcal{E} \) was linearly decreased to zero for other 5000 \( \nu \)-cycles\(^{16}\). If an input vector \( \vec{I} \) is presented after learning, it is assigned to the class with maximal \( q_\alpha(\vec{I}) \) (or likewise \( Q_\alpha(\vec{I}) \)). The resulting classification of the database can subsequently be compared with the actual

\(^{16}\)Note that an automatic stopping criterion based on the variance of \( \chi(\tau), \langle (\chi(\tau) - \langle \chi(\tau) \rangle)^2 \rangle \), produced approximately the same results.
Table 1: Comparison of iris database classifications

<table>
<thead>
<tr>
<th>Method</th>
<th>Correct Classifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOM</td>
<td>85.33% ± 0.1%</td>
</tr>
<tr>
<td>k-Means</td>
<td>88.6% ± 0.1%</td>
</tr>
<tr>
<td>Macrocolumn</td>
<td>90.6% ± 1.7%</td>
</tr>
<tr>
<td>Spiking RBF</td>
<td>92.6% ± 0.9%</td>
</tr>
</tbody>
</table>

labels of the data points and the percentage of correct classifications can be computed. In 100 simulations the system’s correct-classification rate had a mean value of 90.6%±1.7% and was always larger than 87.3%. In Tab. 1 percentages of correct classifications are also given for other unsupervised methods\(^{17}\). Our system can also be trained with data point superpositions (40) of the iris database. Even for relatively high superposition factors \(p_o\) it reaches good classification results (90.6%±1.8% correct classifications for \(p_o = 2.0\) and 89.6%±2.2% for \(p_o = 3.0\)). For values \(p_o > 3.0\) the system in some cases fails to find appropriate classifications within 10000 \(\nu\)-cycles. For low superposition rates the system, surprisingly, reaches even better results than in the case of no superposition (e.g., 90.9%±1.6% for \(p_o = 1.5\))\(^{18}\).

The results of this paragraph show that classification on the basis of minicolumn RF self-organization works well also for higher-dimensional natural data and is, furthermore, competitive to other unsupervised classification systems.

### 4.3 Conclusion and Discussion

We have seen how minicolumnar RF self-organization of the model macrocolumn can successfully be applied to clustering. The clustering procedure is fuzzy\(^{19}\) and hierarchical – a combination which, in itself, distinguishes the system from the vast majority of clustering techniques (compare Jain et al., 1999). On a linear processor simulation time for one \(\nu\)-cycle scales only linearly with the number of minicolumns \(k\) and the input space dimensionality, \(O(Dk)\), i.e., the system remains applicable to very high-dimensional data, which represents an interesting feature if considering that hierarchical clustering methods usually scale at least quadratically. Note, however, that learning time in terms of \(\nu\)-cycles also increases with \(k\).

For the application to clustering, capabilities of the model have their limitations, e.g., in a fixed number of minicolumns \(k\) which sets a limit for the maximal number of classes which can be found. Additionally, the probability that a cluster is rep-

\(^{17}\)The data in Tab. 1 is taken from (Bohte et al., 2002) where SOM was run with three output neurons and k-Means was set to \(k = 3\). SOM and k-Means results can probably be improved with parameter tuning. Spiking RBF reaches the best classification rate but fails in clustering in about one of ten simulations.

\(^{18}\)Note that for each result we considered 100 simulations and that the system for \(p_o \leq 3.0\) and in the case of no superposition always found a clustering within 10000 \(\nu\)-cycles.

\(^{19}\)The functions \(Q_{\alpha}(\hat{T})\) can be used to assign class membership probabilities to a data point.
resented by two or more RFs increases for large \( k \). By measuring the correlations between minicolumn activities while iterating through the data points such multiple representations can be identified, however. This approach also offers a way to classify more elongated clusters.

The size of a cluster can vary over some orders of magnitude but a limitation exists with respect to the volume of the input space which is assigned to one class. Due to a fixed standard deviation, \( \sigma \), of the Gaussian representation (37) there exists a finite minimal size of the input space volume assigned to one class. Cluster size limitations could be accounted for with other coding schemes (compare, e.g., Bohte et al., 2002) but with respect, e.g., to sensory data natural limitations of cluster size and cluster spacing usually exist.

To represent a \( D \)-dimensional input vector the use of a \( D \)-dimensional array of neurons and the usage of an Euclidean like metric could be considered, as well. For general databases, e.g., the iris database, it is questionable, however, whether an Euclidean distance measure represents a good choice (see Jain et al., 1999, for a discussion of different distances). For the iris database and for the two-dimensional databases of Fig. 22 and Fig. 23 the input vector encoding as used in our system produces appropriate results. The separate encoding of each vector entry (compare also Bohte et al., 2002) has the advantages that, first, the required number of input neurons \( N \) only scales linearly with dimension \( D \) and, second, a separate encoding can directly handle incomplete input vectors by just leaving the corresponding set of neurons inactive.

We have seen that minicolumnar RF self-organization can successfully be applied to the problem of clustering. It forms suitable classes for different artificial databases and is competitive to other systems, e.g., in the benchmark task of the iris database classification. The macrocolumn model, hereby, distinguishes itself from other clustering methods by combining important and essential abilities required from biological information processing systems: the input processing is robust against perturbations and lesions, the reaction times are fast, the system learns hierarchical, and it is able to learn from data point superpositions.
5 Abstraction of Macrocolumn Dynamics

In Sec. 2 and Sec. 3 we defined and analyzed a model of the cortical macrocolumn. We have explicitly modeled the neurons of a macrocolumn along with their interconnections. The dynamics’ key features were derived and it was coupled to a dynamics of Hebbian plasticity of afferent fibers to the macrocolumn. In a large number of experiments we could show that the resulting self-organization of RFs can be applied to different feature extraction and classification problems.

The question remains, however, what properties of the macrocolumn are essential for the dynamics to be applicable to the different kinds of experiments. What properties of the macrocolumn are crucial and what other types of dynamic equations might lead to a similar behavior? And as real time is (presumably) continuous – what is the continuous analogon of the discrete dynamics defined in Sec. 2 and Sec. 3?

To answer these question we will, in the following, begin with a very general approach for the dynamics and only get more explicit if necessary.

5.1 Basic Assumptions

Consider a biological cortical macrocolumn with \( k \) minicolumns which is not coupled to any external input. Let us denote the activity of each minicolumn of the macrocolumn by \( p_\alpha \) in analogy to the discussion of average activity probabilities in the preceding sections. Note, however, that \( p_\alpha \) is from now on not a probability but a real valued dynamic variable.

As discussed in Sec. 2 neurophysiological and neuroanatomical experiments suggest that at least a subset of the excitatory neurons of a minicolumn act as monolithic units to encode, e.g., visual frequency and orientation content in primary visual cortex. We therefore assume that the neurons of the macrocolumn are on average only influenced by the activity of their own minicolumn \( p_\alpha \) and by activities \( p_\beta \ (\beta \in \{1, \ldots, k\} \setminus \{\alpha\}) \) of all other minicolumns of the macrocolumn:

\[
\frac{d}{dt} p_\alpha = F_\alpha(p), \quad \forall \alpha \in \{1, \ldots, k\}. \tag{43}
\]

Motivated by neuroanatomical findings and by the model studied in the earlier sections we further assume a macrocolumn to consist of equal minicolumns and we take each minicolumn to be equally and inhibitorily coupled to all minicolumn activities, i.e., we assume that there is no neighborhood relationship between the minicolumns. An equation system which models without loss of generality such a macrocolumn is given by:

\[
\frac{d}{dt} p_\alpha = f(p_\alpha, h(p)), \tag{44}
\]

with function \( f : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \) and \( h : \mathbb{R}^k \to \mathbb{R} \) where \( h \) is invariant under all permutations of its arguments (\( I = \{1, \ldots, k\} \)):

\[
\forall \vec{p} \in \mathbb{R}^k \forall \sigma : I \to I \text{ permutations} : h(x_1, \ldots, x_k) = h(x_{\sigma(1)}, \ldots, x_{\sigma(k)}). \tag{45}
\]
The time course of $p_\alpha$ is thus for all minicolumns equally dependent on itself and on a coupling function $^20h : \mathbb{R}^k \rightarrow \mathbb{R}$.

The function $h$ models the inhibitory input to a minicolumn, i.e., it models the effect of inhibitory postsynaptic potentials (IPSPs) on active neurons of a minicolumn. If a minicolumn $\alpha$ is inactive, $p_\alpha = 0$, there are (in the absence of spontaneous activity) no EPSPs which might result in an activation of the minicolumn. We therefore assume:

$$\forall y \in \mathbb{R} : f(0, y) = 0.$$  \hfill (46)

As explicit inhibitory coupling between the minicolumns we will now choose the same function as in Sec. 2 because it resulted in the desired type of structural instability:

$$h(p) = \nu \max_{\beta=1,\ldots,k} \{p_\beta\}.$$  \hfill (47)

Again $\nu \in \mathbb{R}$ is an inhibitory gain factor which will play a crucial role for the dynamics. Note that (47) satisfies the assumption in (45).

### 5.1.1 Stationary Points and Stability

To analyze the dynamic behavior of (44) with (47) we first look for stationary points of the system. For $k = 1$ and a function $g(q) = f(q, \nu q)$ with finitely many zero points, the stationary points $q_i$ of (44) are given by:

$$q_1 = \max \{q \in \mathbb{R} \mid 0 = g(q)\},$$  \hfill (48)

$$q_i = \max \{q \in \mathbb{R} \mid 0 = g(q) \land \forall j < i : q \neq q_j\}. \hfill (49)$$

For $k \geq 1$ let us, therefore, consider the following sets of phase space points for fixed $\nu \in \mathbb{R}$:

$$\mathcal{P}_i^0 := \max \{q \in \mathbb{R} \mid 0 = f(q, \nu q)\},$$

$$\mathcal{P}_i^0 := \max \{q \in \mathbb{R} \mid 0 = f(q, \nu q) \land \forall j < i : q \neq \mathcal{P}_j^0\},$$

$$\mathcal{P}_i^1 := \max \{q \in \mathbb{R} \mid 0 = f(q, \nu \mathcal{P}_i^0) \land q < \mathcal{P}_i^0 \land (\forall r < j : q \neq \mathcal{P}_r^1)\},$$

$$Q_i := \{q \in \mathbb{R}^k \mid \max_{r \in I} \{q_r\} = \mathcal{P}_i^0 \land (\forall r \in I \exists j \in \mathbb{N}_0 : q_r = \mathcal{P}_r^1)\}$$

$$Q := \bigcup_i Q_i$$

where $I = \{1, \ldots, k\}, \mathbb{N}_0 = \mathbb{N} \cup \{0\}, i \in \mathbb{N}, j \in \mathbb{N}_0,$ and $r \in I$. Note that $\mathcal{P}_i^1$ does not necessarily exist for all $j$. The following example will be of relevance later.

#### Example 1

Suppose $\hat{f}(q) := f(q, \nu q)$ has three zeros, namely $q_1 > q_2 > q_3 = 0$. Suppose further that $g_1(q) = f(q, \nu q_1)$ has with $\bar{q}_1$ one non-trivial zero smaller than $q_1$ and that $g_2(q) = f(q, \nu q_2)$ and $g_3(q) = f(q, 0)$ do not have any non-trivial zeros smaller

$^20$More generally we could take $f : \mathcal{U} \times V \rightarrow \mathbb{R}$ and $h : \mathbb{W}^k \rightarrow V$ with $\mathcal{U}, V, \mathbb{W} \subset \mathbb{R}$ but as we will only use functions $f$ and $g$ defined on $\mathbb{R}$ we will set $\mathcal{U}, V, \mathbb{W} = \mathbb{R}$ for simplicity.
than $q_2$ and 0, respectively. Note that because of (46) $\hat{f}, g_1, g_2, g_3$ all have the trivial zero point $q = 0$. For dynamics (44) with $h$ given in (47) we get the following phase space points:

$$ \mathcal{P}_1^0 = q_1, \mathcal{P}_1^1 = \bar{q}_1, \mathcal{P}_2^2 = 0,$$
$$ \mathcal{P}_2^0 = q_2, \mathcal{P}_2^1 = 0,$$
$$ \mathcal{P}_3^0 = 0,$$

$$ Q_1 = \{(P_1^0, \ldots, P_1^0, P_1^1, \ldots, P_k^1, 0, \ldots, 0) \text{ and permutations } |l \geq 1; m_1, m_2 \geq 0 \}$$

$$ Q_2 = \{(P_2^0, \ldots, P_2^0, 0, \ldots, 0) \text{ and permutations } |l \geq 1, m \geq 0 \}$$

$$ Q_3 = \{(0, \ldots, 0) \}$$

As can be seen, the $\mathcal{P}_i^0$'s represent the zeros of the function $\hat{f}(q) = f(q, \nu q)$ and the $\mathcal{P}_i^1$ with $(j > 0)$ represent zeros of the functions $g_i(q)$ and such that all $g_i(q) = f(q, \nu P_i^0)$ also have finitely many zeros.

If $Q$ is defined as in (50) then $\bar{q} \in \mathbb{R}^k$ is a stationary point of

$$ \frac{d}{dt} p_\alpha = f(p_\alpha, \nu \max_{\beta=1, \ldots, k} \{ q_\beta \}) , \alpha = 1, \ldots, k, \quad (51) $$

if and only if $\bar{q} \in Q$.

**Proof**

If $\bar{q} \in Q_i$ then there exists a vector entry $q_r = P_i^0$ such that all other vector entries $q_s$ are smaller or equal $q_r$. Thus we get by inserting $\bar{q}$ into (51):

$$ \frac{d}{dt} p_s |_{\beta=q} = f(q_\beta, \nu q_r) = f(P_i^0, \nu P_i^0) = 0 \quad \text{by definition of } \mathcal{P}_i^0,$$

$$ \frac{d}{dt} p_s |_{\beta=q} = f(q_\beta, \nu q_r) = f(P_i^0, \nu P_i^0) = 0 \quad \text{by definition of } \mathcal{P}_i^0$$

which proves that $\bar{q}$ is a stationary point.

For the inverse implication suppose that $\bar{q} = (q_1, \ldots, q_k)$ is a stationary point of (51) then we get for $q_r = \max_{r'} \{ q_{r'} \}$ by definition of a stationary point:

$$ \frac{d}{dt} p_s |_{\beta=q} = f(q_\beta, \nu q_r) = 0$$

Hence there exists a $\mathcal{P}_i^0$ which is equal to $q_r$. For any other entry $q_s$ of $\bar{q}$ we get if $q_s = q_r$ that $q_s = \mathcal{P}_i^0$ or if $q_s < q_r$ we get by definition of a stationary point:

$$ \frac{d}{dt} p_s |_{\beta=q} = f(q_\beta, \nu P_i^0) = 0$$

Thus, there exists a $j$ such that $q_s = \mathcal{P}_i^j$ because the $\mathcal{P}_i^j$ are the zero points of the function $f(q, \nu P_i^0)$, $\bar{q}$ is therefore a vector which entries are equal to $\mathcal{P}_i^j$ for different values of $j$ with maximal entry $\mathcal{P}_i^l$. Thus there exists a $Q_i$ such that $\bar{q} \in Q_i$.

$\square$
Hence, given a dynamics (51) with a function \( f : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \) with suitable properties, we can iteratively drive all stationary points by using (50).

We now proceed by computing the stabilities of stationary points contained in (50). The stability of stationary points is of major importance in the theory of (especially non-linear) differential equations because if all stationary points of a system and their stabilities are known, at least its qualitative behavior can be understood in many cases. There are different approaches to compute the stabilities. Given a system of ordinary differential equations we can linearize the equations in the stationary point and compute the approximate behavior in the point’s vicinity. To do so, we can explicitly compute the eigenvalues of the system’s Jacobian or we use a more indirect method like the Hurwitz criterion. Another approach to analyze the qualitative dynamic behavior is to deduce a Lyapunov-function or, in the special case of a gradient system, a potential (see, e.g., Guckenheimer and Holmes, 1983; Jetschke, 1989, for a discussions of the different approaches).

Due to the symmetries of dynamics (51) we choose the most direct method in this case. We compute the Jacobian in all stationary points of the system and directly compute its eigenvalues. Using this approach we will obtain simple expressions of the eigenvalues in terms of derivatives of function \( f : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \) in (51).

First note that the entries of the Jacobian of (51) are given by:

\[
\left( \partial_\beta F_\alpha \right)(\vec{p}) = \left( \partial_1 f \right)(p_\alpha, h(\vec{p})) \delta_{\alpha\beta} + \left( \partial_2 f \right)(p_\alpha, h(\vec{p}))(\partial_\beta h)(\vec{p}),
\]

where we used the notation \( \partial_\beta = \frac{\partial}{\partial x_\beta} \) and \( \left( \partial_i f \right)(x_1, x_2) = \frac{\partial}{\partial x_i} f(x_1, x_2) \). For a stationary point \( \vec{q} \in Q_i \),

\[
\vec{q} = \left( p_{i0}^0, \ldots, p_{im_i}^0, p_{i1}^1, \ldots, p_{im_i}^1, \ldots, p_{i0}^j, \ldots, p_{im_i}^j \right),
\]

we can with

\[ h_\rho(\vec{p}) := \nu \left( \sum_{\beta=1}^{k} (p_\beta)^\rho \right)^{\frac{1}{\rho}} \Rightarrow \lim_{\rho \to \infty} h_\rho(\vec{p}) = h(\vec{p}), \]

where \( \rho = \{2, 4, \ldots \} \), compute (see Appendix B for details):

\[
\left( \partial_\alpha h \right)(\vec{q}) = \begin{cases} \nu & \text{for } \alpha = 1, \ldots, l \\ 0 & \text{for } \alpha = (l + 1), \ldots, k \end{cases}
\]

We therefore get a symmetric Jacobian, \( F'(\vec{q}) \), and the characteristic polynomial for \( \vec{q} \in Q_i \) can be computed to be (see Appendix B):

\[
\det \left( F'(\vec{q}) - \lambda \mathbf{I} \right) = (c - d + ld - \lambda) \left( c - d - \lambda \right)^{l-1} \prod_j (e_j - \lambda)^{m_j}
\]

\[
c = \left( \partial_1 f \right)(p_{i0}^0, \nu p_{i0}^0) + \nu \frac{\nu}{l} \left( \partial_2 f \right)(p_{i0}^0, \nu p_{i0}^0)
\]

\[
d = \nu \frac{\nu}{l} \left( \partial_2 f \right)(p_{i0}^0, \nu p_{i0}^0)
\]

\[
e_j = \left( \partial_1 f \right)(p_{ij}^j, \nu p_{ij}^j)
\]
Thus we finally get the eigenvalues:

\[
\begin{align*}
\lambda_1 &= (\partial_1 f)_{(p^0, \nu, p^0)} + \nu (\partial_2 f)_{(p^0, \nu, p^0)} \quad \text{multiplicity 1} \\
\lambda_2 &= (\partial_1 f)_{(p^0, \nu, p^0)} \quad \text{multiplicity } (l-1) \\
\lambda_{2+j} &= (\partial_1 f)_{(p^0, \nu, p^0)} \quad \text{multiplicity } m_j
\end{align*}
\]

Because of assumption (46) there is always a $P_i^j$ identical zero. The associated $\lambda_{2+j}$ will for convenience therefore also be denoted by $\lambda_0$ and its multiplicity by $m_0$:

\[
\lambda_0 := \lambda_{2+j} = (\partial_1 f)_{(0, \nu, p^0)} \quad \text{multiplicity } m_0
\]

The eigenvalues determine the stability of a stationary point $\vec{q} \in Q_i$ (53). Note that stabilities do not directly depend on the signature $l(\vec{q}), m_1(\vec{q}), ..., m_J(\vec{q})$. They are only dependent via the their multiplicities given in (60).

Consequently, we can define subsets of $Q_i$ which contain stationary points with equal stability. E.g. the points of the subsets

\[
Q_i^{l+} = \{ \vec{q} \in Q_i | l(\vec{q}) \geq 2, m_j \neq 0, \forall r \neq j : m_r(\vec{q}) = 0 \}
\]

of $Q_i$ contain points with equal stability if we further assume that they are hyperbolic, i.e. that none of their eigenvalues is identical to zero. Note that the existence of such subsets is independent of the choice of a specific function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.

The eigenvalues (60) are of a particularly simple form – they are given by the first and second derivative of the function $f$ in (51). This means that as soon as a function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ with the required properties is given, we can in a straightforward way not only compute the stationary points of dynamics (51) given by (50) but we, furthermore, can easily compute their stabilities using (60).

### 5.1.2 An Explicit Activation Function

We now choose a specific function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ for dynamics (51). To ease the choice we exploit that in the case of $k = 1$ (51) describes the activity dynamics of a single isolated minicolumn:

\[
\frac{d}{dt} p = f(p, \nu) = \hat{f}(p).
\]

We expect a minicolumn to be able to maintain non-zero neural activity as in Sec. 2. Furthermore, it can be expected that due to finite thresholds of neurons low levels of activity are not able to activate a minicolumn if it is in its stable inactive state. A function reflecting these expectations is sketched in Fig. 24. The simplest explicit choice of a function like the one displayed is obviously a polynomial, $G(p)$, of order three:

\[
G(p) = a p (p - c_1 - c_2 p^2) \text{ where } a, c_2 > 0, c_1 \geq 0.
\]

To choose an appropriate function $f(p, h(\vec{p}))$ consistent with (63) we have to appropriately generalize $G(p)$ using $h(\vec{p})$. The function $h(\vec{p})$ is assumed to act inhibitorily.
but we can not just subtract it from the rhs in (64) because this would violate assumption (46). The simplest choice in accordance with (63) and (46) is to insert $h(\bar{p})$ as factor of the polynomial’s linear term. We get:

$$f(p_{\alpha}, h(\bar{p})) = a p_{\alpha} (p_{\alpha} - h(\bar{p}) - \Theta - b p_{\alpha}^2) \text{ where } a, b > 0, \Theta \geq 0.$$

Note that $\hat{f}(p) = f(p, \nu p)$ is for $\nu < 1$ indeed a polynomial of the form (64) and that a polynomial $\hat{f}(p)$ with constant term would violate assumption (46). The function $\hat{f}$ has finitely many zeros and is smaller than zero for $p_{\alpha} \to \infty$ because $a, b > 0$.

Function (65) defines an explicit macrocolumn dynamics which satisfies all made assumptions:

$$\frac{d}{dt} p_{\alpha} = a p_{\alpha} (p_{\alpha} - \nu \max_{\beta=1,\ldots,k} \{p_{\beta}\} - \Theta - b p_{\alpha}^2)$$

where $a, b > 0, \Theta \geq 0$, and $\nu \in (0, 1)$. $p_{\alpha}$ increases for low inhibition and decreases for large inhibition. For the activity to increase, $p_{\alpha}$ has to be larger than a non-negative threshold, however. The term $(-b p_{\alpha}^2)$ prevents unlimited growth and can be interpreted as the effect of neuron refraction times.

### 5.1.3 An Explicit Macrocolumn Dynamics

A special case of (65) is to choose $b = 1$ and $\Theta = 0$.

$$f(p_{\alpha}, h(\bar{p})) = a p_{\alpha} (p_{\alpha} - h(\bar{p}) - p_{\alpha}^2) \text{ with } a > 0$$

A dynamics with phase velocity\(^{21}\) (67) approximates the case of a large number of neurons per minicolumns such that with excitatory interconnections the threshold $\Theta$

\(^{21}\)In our case the function $f(p_{\alpha}, h(\bar{p}))$ is equal to the phase velocity (see, e.g. Arnol’d, 1992, for an exact definition).
can be neglected. The resulting dynamics is given by:

\[
\frac{d}{dt} p_\alpha = a p_\alpha (p_\alpha - \nu \max_{\beta=1,\ldots,k} \{p_\beta\} - p_\alpha^2), \quad \text{where } a > 0, \text{ and } \nu \in (0, 1). \tag{68}
\]

The dynamics will prove to possess all desired properties and the computation of stationary points and their stabilities will turn out to be especially easy. The more extensive computations for the general case (66) are given in Appendix C and we will refer to them later.

Function (67) satisfies the requirements of Proposition 1. Computation of stationary points and their stabilities is therefore straightforward. Using definitions (50) we get

\[
P_1^0 = 1 - \nu, \quad P_2^0 = 0, \quad \text{and if } \nu < \frac{1}{2}: \quad P_1^1 = \nu, \quad P_2^2 = 0. \tag{69}
\]

Thus the stationary points of the system for \(\nu < \frac{1}{2}\) are given by

\[
Q_1 = \{ (P_1^0, \ldots, P_1^0, P_1^1, \ldots, 0, \ldots, 0) \text{ and permutations } |l| \geq 1, m_1, m_2 \geq 0 \},
\]

\[
Q_2 = \{ (0, \ldots, 0) \},
\]

and for \(\nu > \frac{1}{2}\) by

\[
Q_1 = \{ (P_1^0, \ldots, P_0^0, 0, \ldots, 0) \text{ and permutations } |l| \geq 1, m \geq 0, l + m = k \},
\]

\[
Q_2 = \{ (0, \ldots, 0) \}.
\]

Using (60) the stabilities of the points in \(Q_1\) and \(Q_2\) are for \(\nu < \frac{1}{2}\) given by the eigenvalues:

\[
\begin{align*}
\lambda_0 &= -a \nu P_1^0 = -a \nu (1 - \nu) \quad \text{mult. } m_0 \\
\lambda_1 &= a (2(1 - \nu) P_1^0 - 3(P_1^0)^2) = -a (1 - \nu)^2 \quad \text{mult. } 1 \\
\lambda_2 &= a ((2 - \nu) P_1^0 - 3(P_1^0)^2) = a (1 - \nu) (2\nu - 1) \quad \text{mult. } (l - 1) \\
\lambda_3 &= a (2P_1^1 - \nu P_1^0 - 3(P_1^1)^2) = a \nu (1 - 2\nu) \quad \text{mult. } m_1
\end{align*}
\tag{72}
\]

Note that we used convention (61), i.e., \(m_0 = m_2\). For \(\nu > \frac{1}{2}\) we get the same eigenvalues except of \(\lambda_3\) which does not exist. The stationary point \((0, \ldots, 0)\) of \(Q_2\) has as only eigenvalue \(\lambda = 0\) and it turns out to be unstable with polynomial behavior in the vicinity of \((0, \ldots, 0)\).

The eigenvalues for \(\nu \in (0, 1)\) and \(a = 1\) are plotted in Fig. 25. As can be observed by looking at the plot or directly at (72), there is for all \(\nu \in (0, 1) \setminus \{ \frac{1}{2} \}\) an eigenvalue \(\lambda_2\) or \(\lambda_3\) larger than zero. There is nevertheless a large set of stationary point which can be shown to be stable. Motivated by the existence of subsets like (62) we define:

\[
Q^+: = \{ \bar{q} \in Q_1 \mid \ell(\bar{q}) \geq 2, m_1(\bar{q}) = 0 \}
\tag{73}
\]

\[
Q^{++}: = \{ \bar{q} \in Q_1 \mid \ell(\bar{q}) = 1, m_1(\bar{q}) = 0 \}
\tag{74}
\]
Figure 25: Eigenvalues of $F'(\vec{q})$ for $a = 1$ at stationary points $\vec{q}$. Values of $\lambda_0$, $\lambda_1$, $\lambda_2$, and $\lambda_3$ depend on the value of the bifurcation parameter $\nu$. Eigenvalue $\lambda_3$ does not exist for $\nu > \frac{1}{2}$.

which are subsets of $Q$ for all $\nu \in (0, 1)$. Because of (72) all stationary in $Q^+$ are stable for $\nu < \frac{1}{2}$ but become unstable for $\nu > \frac{1}{2}$ because of $\lambda_2$. The stationary points in $Q^{++}$ remain stable for all $\nu$. The points in the set

$$Q^- := \left\{ (P_{1,\ldots,l}, P_{1,\ldots,l}, 0, \ldots, 0) \text{ and permutations} \mid l \geq 1, m_1 > 0 \right\}, \quad (75)$$

which only exist for $\nu < \frac{1}{2}$, are for all $\nu \in (0, \frac{1}{2})$ unstable because of $\lambda_3 > 0$.

In Fig. 26 we plotted the phase velocity of (68) for two different values of $\nu$ and $k = 2$. If $\nu$ is smaller than the critical value $\nu_c = \frac{1}{2}$, we get the three non-trivial stable stationary points of $Q^{++}$ and $Q^+$ as well as the two unstable stationary points of $Q^-$. If $\nu$ is increased to a value larger than $\nu_c$, the unstable points in $Q^-$ merge with the stable point in $Q^+$ in the point of structural instability $\nu_c = \frac{1}{2}$ and we get an unstable symmetric stationary point $(P_1^0, P_1^0)$. This dynamic behavior exactly matches the behavior of the macrocolumn model with $k = 2$ minicolumns as described in Sec. 2 (compare Fig. 2).

For higher dimensions we know because of the multiplicities in (72) that all stationary points in $Q^+$ loose stability for the same value $\nu_c$ ($\nu_c = \frac{1}{2}$ in this case). The dynamics, therefore, generalizes to higher dimensions exactly as the macrocolumn dynamics of Sec. 2 does (see Fig. 3 for $k = 3$). Thus the equation system (68) represents a derivation of a macrocolumn dynamics from basic principles which is reproducing the crucial features of the model of Sec. 2 which used explicit assumptions about neuron models and connectivity.
Figure 26: Phase velocities $\vec{F}(\vec{p}), F_\alpha(\vec{p}) = f(p_\alpha, h(\vec{p}))$, of dynamics (68). A Phase velocity for $\nu = 0.4 < \nu_c$. Black points mark stationary points as given in (70). B Phase velocity for $\nu = 0.6 > \nu_c$. Black points mark stationary points as given in (71).
5.1.4 More Complex Dynamics

Let us now shortly discuss the more complex case of a dynamics (66) with \( \nu \neq b > 0 \) and \( \Theta > 0 \). In this case we get for \( \nu \) smaller than a critical value \( \nu_c \) three instead of two sets \( Q_i \):

\[
Q_1 = \{ (P_0^0, \ldots, P_l^0, P_1^1, \ldots, P_l^1, 0, \ldots, 0) \text{ and permutations } | l \geq 1; m_1, m_2 \geq 0 \}
\]

\[
Q_2 = \{ (P_0^0, \ldots, P_l^0, 0, \ldots, 0) \text{ and permutations } | l \geq 1, m \geq 0, l + m = k \}
\]

\[
Q_3 = \{ (0, \ldots, 0) \}
\]

Note that this is the situation as earlier in Example 1. An analysis along the lines for dynamics (68) (see Appendix C) then shows that \( \bar{q} = 0 \) is now a stable point and that the points in \( Q_1 \) can be split in qualitatively the same stable and unstable subsets \( Q^+, Q^{++}, \) and \( Q^- \). The formerly non-existent stationary points in \( Q_2 \) are all unstable for a wide range of parameters. The nature of the additional stationary points can easily be understood by looking at the phase space diagram in Fig. 27. As in Fig. 26 we get the stable stationary points of \( Q^+, Q^{++}, \) and \( Q^- \) of \( Q_1 \) but we additionally get unstable points closer to the origin as can be seen in Fig. 27B.

The stable manifolds of the saddle points in \( Q_2 \), \( (P_0^0, 0) \) and \( (0, P_0^0) \) for \( k = 2 \), separate the phase space volume which is attracted by the origin, \( (0, \ldots, 0) \), from the phase space volume which is attracted by other stable stationary points. If the system is initialized with a phase space point not attracted by \( (0, \ldots, 0) \), it can not enter the attraction region of the origin.

The more general dynamics (66) also reproduces the dynamic behavior of the system in Sec. 2 but we will use dynamics (68) in the following.

5.2 Afferent Connections

After the derivation of an abstract and continuous macrocolumnar dynamics in the first part of this section we will now introduce an abstraction of afferent connections to the macrocolumn. First, we need an abstraction of the input neurons (21). Instead of explicit input neurons we now use a number of \( \ell \) input units with activities \( p_j^E \).

Each input unit can be considered as a set of input neurons in which each neuron has a probability proportional to \( \bar{q} \) to be active. Note, however, that \( \bar{q} \) is a continuous dynamic variable from now on and not a probability as in Sec. 3. An input unit with activity \( p_j^E \) is connected to minicolumn \( \alpha \) with activity \( p_\alpha \) by the afferent fiber \( R_{\alpha j} \). The strength of the coupling of the macrocolumnar dynamics to the input is in analogy to (20) denoted by \( \kappa \). Using (44) we get the dynamics:

\[
\frac{d}{dt} p_\alpha (t) = f(p_\alpha (t), h(\bar{\mu}(t))) + \kappa \sum_{j=1}^{N} R_{\alpha j} (t) p_j^E (t) \quad (76)
\]

Note that we will often omit the time index \( t \) in the following. For the functions \( h(\bar{\mu}) \) and \( f(p_\alpha, h(\bar{\mu})) \) we can choose any functions which reproduce the desired bifurcation behavior. As was proven in the previous chapter, one possibility is to choose...
5 Abstraction

Figure 27: Phase velocity $\tilde{F}(\bar{p})$, $F_\alpha(\bar{p}) = f(p_\alpha, h(\bar{p}))$, of dynamics (66) for $k = 2$, $\nu = 0.4$, $b = 0.5$, and $\Theta = 0.04$. Black points mark stationary points as computed in Appendix C. A Phase velocity in $[-0.05, 1.15] \times [-0.05, 1.15]$. B Phase velocity in $[-0.01, 0.085] \times [-0.01, 0.085]$. 
as inhibition function \( h(\bar{p}) \) the maximum operation (47) and as function \( f(p_\alpha, h(\bar{p})) \) the function given in (67). As mentioned above, it is also possible to use the more complex function (65) but we choose to use the less complex case.

There is one difference between the abstract dynamics on the one side and the explicit model of Sec. 2 and Sec. 3 on the other side which has not yet been considered. The explicit model contains noise as integral part. Its input neurons are stochastically activated (21) and the minicolumnar neurons are modeled using neural threshold noise (19). Together with random minicolumnar interconnection the dynamics of the minicolumn activities \( p_\alpha \) is noisy. A noisy behavior can because of various stochastic influences also be expected from activities of biological minicolumns. We, therefore, extend equation system (76) by adding Gaussian white noise because this is the most elementary way to account for random perturbations. Together with (47) and (67) we get as dynamics for minicolumnar activity:

\[
\frac{d}{dt} p_\alpha = \alpha p_\alpha (p_\alpha - \nu \max_{\beta = 1, \ldots, k} \{ p_\beta \} - p_\alpha^2) + \kappa \sum_{j=1}^{N} R_{\alpha j} p_j^E + \sigma \eta_t, \tag{77}
\]

where \( \sigma \) parameterizes the influence of the Gaussian white noise \( \eta_t \) and \( \kappa \) parameterizes the coupling to external input.

As abstraction of Hebbian plasticity (22) and normalization (23) consider the difference equations:

\[
\begin{align*}
\tilde{R}_{\alpha j}(t + \Delta t) &= R_{\alpha j}(t) + \mathcal{E} p_\alpha(t) p_j^E(t) \Delta t \tag{78} \\
R_{\alpha j}(t + \Delta t) &= \frac{\tilde{R}_{\alpha j}(t + \Delta t)}{\sum_{l=1}^{N} R_{\alpha l}(t + \Delta t)} \tag{79}
\end{align*}
\]

Equations (78) and (79) represent average versions of equations (22) and (23), respectively. Equation (79) ensures that

\[
\forall t : \sum_{j=1}^{N} R_{\alpha j}(t) = 1. \tag{80}
\]

In the limit \( \Delta t \to 0 \) we get:

\[
\begin{align*}
&\lim_{\Delta t \to 0} \frac{1}{\Delta t} (R_{\alpha j}(t + \Delta t) - R_{\alpha j}(t)) \\
&= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left( \frac{R_{\alpha j}(t) + \mathcal{E} p_\alpha(t) p_j^E(t) \Delta t}{\sum_{l=1}^{N} R_{\alpha l}(t) + \mathcal{E} p_\alpha(t) p_l^E(t) \Delta t} - R_{\alpha j}(t) \right) \\
&= \frac{1}{\sum_{l=1}^{N} R_{\alpha l}} \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left( R_{\alpha j} (1 - \sum_{l=1}^{N} R_{\alpha l}) + \mathcal{E} p_\alpha p_j^E \Delta t - (\sum_{l=1}^{N} \mathcal{E} p_\alpha p_l^E \Delta t) R_{\alpha j} \right)
\end{align*}
\]

Using (80) we get with

\[
\frac{d}{dt} R_{\alpha j} = \mathcal{E} p_\alpha p_j^E - (\sum_{l=1}^{N} \mathcal{E} p_\alpha p_l^E) R_{\alpha j} \tag{81}
\]

\[\text{22Gaussian white noise is an independent and identically distributed stochastic process whose finite dimensional distributions have a Gaussian probability density } \mathcal{N}(x) = (\sqrt{2\pi}\sigma)^{-1} \exp(-\frac{x^2}{2\sigma^2}) \text{ (see, e.g. Honerkamp, 1994, for details). } \sigma \eta_t \text{ is Gaussian white noise with variance } \sigma^2.\]
5 Abstraction

a continuous dynamics for synaptic plasticity. Note that $\sum_j R_{\alpha j}(t)$ converges to one if it is not initially equal to one:

$$\sum_{j=1}^{N} R_{\alpha j}(t) = 1 - \exp \left( - \int_0^{t} f(t') dt' \right), \quad f(t) = \sum_{l=1}^{N} E_p \alpha(t) p_l^E(t) \gtrsim 0. \quad (82)$$

As it was the case in (22) we want synaptic plasticity to be enabled only if the macrocolumnar activity $A(t) = \sum_{\alpha=1}^{k} p_\alpha(t)$ falls below a threshold $\chi$.

Using dynamics (77) for the neural dynamics and dynamics (81) with threshold condition for the dynamics of synaptic plasticity we get the dynamics of the complete system. We, again, operate the system with oscillating inhibitory gain factor $\nu$ such that we arrive at a complete dynamic description of the system given by:

$$\nu(t) = \begin{cases} 
0 & \text{if } \bar{t} < T_{\text{init}} \\
\left( \nu_{\text{max}} - \nu_{\text{min}} \right) \frac{\bar{t} - T_{\text{init}}}{T_{\text{ref}}} + \nu_{\text{min}} & \text{if } \bar{t} \geq T_{\text{init}}, 
\end{cases} \quad (83)$$

$$\frac{d}{dt} p_\alpha = \alpha p_\alpha \left( p_\alpha - \nu(t) \max_{\beta=1,\ldots,k} \{ p_\beta \} - p_\alpha^g \right) + \kappa \sum_{j=1}^{N} R_{\alpha j} p_j^E + \sigma \eta_t, \quad (84)$$

$$\frac{d}{dt} R_{\alpha j} = \left( E_p \alpha p_j^E - \left( \sum_{l=1}^{N} E_p \alpha p_l^E \right) R_{\alpha j} \right) \mathcal{H}(\chi - A(t)), \quad \mathcal{E} = \frac{\epsilon}{N}, \quad (85)$$

where $\bar{t} = \text{mod}(t, T)$ (see Eqn. (24)). In (85) $\epsilon$ is a synaptic growth factor which is independent of the number of input units $N$. Equations (84) and (85) are a system of non-linear differential equations coupled to an oscillation given by (83). In simulations the oscillation is chosen to be slow compared to the dynamics of $p_\alpha$. For $\nu = 0$ the system quickly converges to the stable stationary point $(\mathcal{P}_1^0, \ldots, \mathcal{P}_1^0)$ in the case of no input and to a phase point in the vicinity of this stable point in the case of input (note that the input is always small due to small $\kappa$ and $p_j^E \in [0, 1]$). For $0 < \nu \ll \nu_c$, the system follows the trajectory of the stable stationary point of the system with $\nu(t)$. Near to the bifurcation point $\nu_c$ the system can rapidly change its activities $p_\alpha$ because of symmetry breaking due to input and because of stable stationary points in $Q$ which attract increasingly small regions of the phase space (compare $(\mathcal{P}_1^0, \mathcal{P}_1^0)$ in Fig. 26). The system’s reaction near and at the point of structural instability is the desired behavior which amplifies input differences to the minicolumns.

As in Sec. 3 we study the system behavior by exposing it to different kinds of input. From a given database with different two-dimensional input patterns we present a randomly chosen pattern during each $\nu$-cycle. Note that it is not essential that the presentation of a pattern is synchronized with the $\nu$-cycles. If it is not synchronized, the presentation time of a pattern has to be at least several times longer than the period length of a $\nu$-cycle.

Because of the abstraction of the input neurons we can directly use an input image given by the vector $\tilde{I}$ to set the activities of the macrocolumn, $p_j^E = I_j$. Before we can start simulating the dynamics we have to choose a suitable set of parameters. Let us first estimate the magnitude of the parameter $a$ (the inverse time constant) in (84). Note that in (10) we took one time step to be about 1 ms. For dynamics (84) the comparison to real time is not directly possible. We will, however, indirectly use the refraction time of neurons to coarsely estimate the parameter $a$. Suppose we have no
5 Abstraction

self-excitation in an isolated minicolumn. In this case only the term \(-ap^3\) of (84), which models negative feedback because of neuron refraction times, survives. Now let the minicolumn be in its state of maximal activity at \(t = 0\). If we assume that this state is given by \(p = 1\), we get as time course for the minicolumn activity:

\[
p(t) = \frac{1}{\sqrt{2at + 1}}
\]  

(86)

For an isolated biological minicolumn with removed intra-columnar excitation we expect that the activity \(p(t)\) decays to a value close to zero in about \(1\) ms. We now estimate a value for \(a\) by supposing that the activity \(p(t)\) falls to a value of \(5\%\) after \(1\) ms. This results with (86) in a value for \(a\) of \(a \approx 200\text{ms}^{-1}\). It is well understood that this is only a very coarse estimate due to the arbitrariness to choose the activity after \(1\) ms but one obtains the order of magnitude of \(a\).

The value of the coupling \(\kappa\) is taken to be only a small fraction of the value for \(a, \kappa = 1.0\text{ms}^{-1}\), and standard deviation \(\sigma\) of the Gaussian white noise \(\sigma \eta_k\) is taken to be only a fraction of \(\kappa, \sigma = 0.12\text{ms}^{-1}\).

For the oscillation of \(\nu\) (83) we choose as in Sec. 3 a period length of \(T = 25\) ms and a time of \(T_{\text{init}} = 2\) ms with \(\nu = 0\) to reset the dynamics to the symmetric stationary point. After initialization we choose \(\nu\) to increase from \(\nu_{\text{min}} = 0.3\) to a value \(\nu_{\text{max}} = 0.55\) which is slightly larger than the critical value \(\nu_c = 0.5\). For the dynamics of Hebbian plasticity (85) we choose \(\epsilon = 0.2\) and a threshold of \(\chi = 0.55\).

Simulations. Equations (83) to (85) can now be numerically simulated. The most direct way for simulating differential equations is to use the Euler method (see, e.g., Honerkamp, 1994, for a discussion of the Euler method in the context of stochastic differential equations with additive and multiplicative noise). In our case, the time interval \(\Delta t\) for the numerical simulation has to be \(\Delta t = \frac{1}{50}\) ms or smaller in order to avoid coarse imprecisions due to rapid changes of \(p_{\alpha}(t)\) in bifurcation points which can occur in time intervals of less than \(1\) ms.

In the first experiment let us again use the input patterns of Fig. 8A. By simulating dynamics (83) to (85) with parameters as given above we indeed get RF self-organization as can be seen in Fig. 28B. The RFs specialize to different classes of input patterns. If we have fewer minicolumns than major classes exist, we get coarser RFs (see Fig. 28C) and if we have more minicolumns we get RFs with higher specialization degrees (see Fig. 28D).

In the second experiment we use once more the bars test (see Sec. 3.2.2) in order to demonstrate that the abstracted dynamics maintains the ability of learning a distributed code for input obtained by combining different constituents. We operate the system using the same parameters as in the previous experiment and we use a bars test with \(b = 8\) bars and bar appearance probability \(p_o = 0.25\) (the same as in Sec. 3.2.2). As can be seen in Fig. 29, RF self-organization results in a representation of all bars. In 200 considers simulations with \(k = 10\) minicolumns a bars test with above parameters required less than 600 \(\nu\)-cycles in 50\% of the cases to represent all bars (less than 410 in 20\% and less than 950 \(\nu\)-cycles in 80\% of the simulations). Again the system found a correct representation for all bars in all simulations and is
Figure 28: In A the set of input patterns is displayed (same set as in Fig. 8A). During each \( \nu \)-cycle one randomly chosen pattern of this set is presented. In B the modification of the RFs of an abstract macrocolumn with \( k = 6 \) minicolumns and the standard set of parameters is displayed. After 1000 \( \nu \)-cycles six different pattern classes are represented. The RFs’ degree of specialization further increases thereafter to a final degree. C Final RF specialization (after 250 \( \nu \)-cycles) if an abstract macrocolumn with \( k = 3 \) minicolumns is used with the same input. D Final RF specialization (after 10000 \( \nu \)-cycles) if an abstract macrocolumn with \( k = 9 \) is used.

robust against various perturbations to the bars test. Note that the results for the bars test show a significant improvement compared to the explicit non-hierarchical system of Sec. 3 which requires, for the same bars test, less than 400, 1050, and 2100 \( \nu \)-cycles in 20%, 50%, and 80% of 200 simulations, respectively. Thus dynamics (83) to (85) represent not only an abstraction but also an improvement of the explicit non-hierarchical dynamics of Sec. 3. A further difference to self-organization as in Sec. 3.2.2 is that supernumerous RFs of the abstract model frequently specialize to combinations of two or more bars (compare \( \bar{R}^{3} \) and \( \bar{R}^{10} \) in Fig. 29B). Supernumerous RFs of the explicit model tend to specialize to an already represented bar or remain unspecialized.

### 5.3 Hierarchical Self-Organization

Let us now extend the Hebbian plasticity dynamics to a dynamics with time dependent differentiation pressure as we have done in Sec. 3.3. For this reason, we replace \( \chi \) in (85) by a dynamic threshold \( \chi(t) \) and change the value of \( \chi(t) \) after each \( \nu \)-cycle
Figure 29: **A** A selection of 33 typical input patterns of the bars test of 8 different bars (same example pictures as in Fig. 9). **B** Typical example of the self-organization of the RFs of an abstract macrocolumn with 10 minicolumns and the standard set of parameters. During each υ-cycle a randomly generated input pattern of the upper type is presented. After about 250 υ-cycles the network has already found representations of seven bars. After 1000 υ-cycles representations of all bars are found and are further stabilized.
by a value $\Delta \chi = \chi(t) - \chi(t - T)$:

$$
\Delta \chi = -\lambda_\chi (\chi - a_\chi A(t)) \text{ iff } \text{mod}(t, T) = 0,
$$

(87)

where $\lambda_\chi$ and $a_\chi$ are scalar parameters. $\chi$ is initialized with a relatively high value, $\chi(t = 0) = 0.6 k$, and is decreased until it reaches a value of about $a_\chi A(t)$. With decreasing $\chi$ the differentiation pressure for the RFs increases. The value of $a_\chi$ we choose to be $a_\chi = 1.9$ and the value for the decay rate of $\chi$ we set to $\lambda_\chi = 0.0005$. It is chosen such that differentiation pressure is not decreased too fast and such that, at the same time, learning is not too slow.

Instead of a time dependent $\lambda_\chi$ as in (31) we introduce a dynamics for $\nu_{\text{max}}$. As for equation (87) we change the value for $\nu_{\text{max}}$ only at the end of each $\nu$-cycle by $\Delta \nu_{\text{max}} = \nu_{\text{max}}(t) - \nu_{\text{max}}(t - T)$:

$$
\Delta \nu_{\text{max}} = -\lambda_\nu (a_\nu - A(t)) \text{ iff } \text{mod}(t, T) = 0.
$$

(88)

The now time dependent $\nu_{\text{max}}$ also influences the differentiation pressure by adjusting the minicolumn selection process to the input. A time dependent $\nu_{\text{max}}$ also suppresses the tendency of the system to sometimes cease differentiation of some RFs. This allows on the other hand a larger $\lambda_\chi$ in (87) and, hence, faster learning.

In dynamics (29) to (31) we used a time-dependent $\lambda_\chi$ to control the increase of differentiation pressure. Functionally this resulted in a fast hierarchical learning but averaging over time was required. Dynamics (87) and (88) is much simpler and does not require averaging. However, it will result in a hierarchical learning which takes longer in terms of $\nu$-cycles than a hierarchical learning with dynamics (29) to (31).

The parameter $a_\nu$ is set to $a_\nu = 0.55$ which results in a value of $\nu_{\text{max}}$ close to the critical value $\nu_c$. The change rate of $\nu_{\text{max}}$ is set to $\lambda_\nu = 0.02$ and $\nu$ is at $t = 0$ initialized with the critical value $\nu = \nu_c = 0.5$. All other parameters are chosen as for the last two experiments. Only the level of white noise\(^{23}\) is set to a much smaller value of $\sigma = 0.02$.

**Simulations.** We again use the Euler method to simulate the differential equation system (83) to (85) and use equations (87) and (88) with parameters given above for self-adjusting differentiation pressure.

In the first experiment we, again, use the input of experiment of Sec. 3.4.1 (see Fig. 14A). In Fig. 30B RF self-organization is shown for this input data. We again can observe that the system first specializes groups of RFs to coarse classes and continuous to find ever smaller classes of patterns if time proceeds. After the RFs have reached their final degree of specialization they look similar to the RFs of the explicit model of Sec. 3 (compare Fig. 14B). Due to a continuous synaptic plasticity dynamics and continuous valued synaptic strengths the RFs look more homogeneous in the abstract model, however. For smaller number of minicolumns $k$ the RFs become coarser (see Fig. 30C) and for larger $k$ the specialization degree of RFs increase (see Fig. 30D).

\(^{23}\)For non-hierarchical learning a larger noise level helps to keep all RFs in the self-organization process. For hierarchical learning the slowly decreasing differentiation pressure ensures that all RFs remain in the self-organizing process.
Figure 30: A Subset of input patterns of the hand-written digits zero, one, and seven of the MNIST database. B RF modification of an abstract macrocolumn with $k = 8$ minicolumns if input of the form as displayed in A is presented. C RFs of an abstract macrocolumn with $k = 4$ minicolumns after 10000 $\nu$-cycles. D RFs of an abstract macrocolumn with $k = 12$ minicolumns after 10000 $\nu$-cycles.
By comparing the time scales of Fig. 14B and Fig. 30B it can be observed that the abstracted model needs longer for learning. In contrast to the dynamics for $\chi$ of the explicit model (29) to (31) we removed, as mentioned above, the dynamics for differentiation velocity $\lambda$, (31), in the $\chi$-dynamics of the abstracted model in order obtain the more elementary dynamic equations (87) and (88). As a consequence the functionally advantageous hierarchical self-organization slows down the learning if compared to non-hierarchical self-organization. E.g., for the classical bars test with $b = 8$ bars and bar appearance probability of $p_a = 0.25$ using $k = 10$ minicolumns the abstract non-hierarchical system requires less than 600 $\nu$-cycles in 50% of 200 simulations to represent all bars (less than 410 in 20% and less than 950 $\nu$-cycles in 80% of the simulations) whereas the abstract hierarchical system requires less than 1280 $\nu$-cycles in 50% of 200 simulations for the same test (for 20% and 80% the system requires 1050 and 1520 $\nu$-cycles, respectively). In the case of the explicit model, the more complex $\chi$-dynamics (29) to (31) learning is faster in the hierarchical case.

In the next experiments we will apply the abstract model to the continuous bars test again. We use the same experiment as in Fig. 17, i.e., bars of four pixel width at random positions on a $16 \times 16$ pixel input layer (see Fig. 31A for some examples). As can be observed in Fig. 31 an abstract macrocolumn with $k = 10$ minicolumns finds a representation within 5000 to 10000 $\nu$-cycles. Again, this is, due to the new dynamics (87) and (88) about twice as long in terms of $\nu$-cycles compared to the explicit hierarchical system of Sec. 3.3.

With the exception of a longer learning time for relatively small $k$, abstract hierarchical self-organization has all the advantages over non-hierarchical self-organization as were described and discussed for the explicit system in Sec. 3. Hierarchical self-organization shows appropriate representations of the input already in intermediate levels of learning, it has a larger discrimination ability, and is equal or faster in learning time for larger $k$. For more detail we refer to the discussion in Sec. 3.

Exemplarily for the abstract case results of hierarchical and non-hierarchical learning are shown in Fig. 32 for the input used in the experiment of Fig. 13. In general the difference between non-hierarchical and hierarchical learning seems to be less pronounced for the abstract system than for the explicit one. E.g., in Fig. 32 non-hierarchical RF self-organization finally converges to a representation of the input which is nearly as accurate as the final representation of the hierarchical self-organization. The explicit system used in Sec. 3.4.1 was even after a much longer learning time not able to find a representation of such a quality. However, by comparing RFs at earlier time steps, e.g., after 2000 or 5000 $\nu$-cycles, the differences between hierarchical and non-hierarchical learning become more pronounced, again.

### 5.4 Clustering

To apply the abstract system to the problem of data clustering we merely have to use the place coding introduced in Sec. 4. As we do not have explicit input neurons in the abstracted model, we can directly set the activities $p_j^F$ of the input units. To encode a value $x \in [0, 1]$ we use a set of $M$ input units with activities (compare (37)
5 Abstraction

Figure 31: A 20 randomly chosen input patterns of a bars test with 8 bars and continuously varying bar positions. B RF modifications of an abstract macrocolumn with \( k = 10 \) minicolumns if input of the form as displayed in A is presented. C RFs of an abstract macrocolumn with \( k = 8 \) minicolumns after 10000 \( \nu \)-cycles. D RFs of an abstract macrocolumn with \( k = 12 \) minicolumns after 10000 \( \nu \)-cycles.

and (38):

\[
\begin{align*}
  p_j^E & = \exp \left( - \frac{(x_j - x)^2}{2\sigma^2} \right), \\
  x_j & = \frac{1 + 2b}{M - 1} (j - 1) - b.
\end{align*}
\]  

(89)

(90)

For a \( D \) dimensional input vector \( \vec{I} \in [0,1]^D \) we use \( D \) sets of \( M \) input units each encoding one vector entry. If the input vector \( \vec{J} \) does not lie in the interval \( [0,1]^D \) we rescale the vector entries according to (41) and (42).

We can now use the abstracted hierarchical system with above parameters and apply it, with \( \sigma \) and \( b \) as used in Sec. 4, to the iris benchmark test. As in Sec. 4 we operate the system for 5000 \( \nu \)-cycles with \( E \) given above and then linearly decrease synaptic plasticity to zero for another 5000 \( \nu \)-cycles. For Fisher’s iris database the system reaches 91.4\% \( \pm \) 0.1 correct classifications after learning and does not fail to appropriately classify the database in all 100 simulations considered for this measurement. This value of correct classifications represents an improvement compared to the system used in Sec. 4 and makes the abstract system even more competitive.
5 Abstraction

Figure 32: RF self-organization of the abstracted system for hierarchical and non-hierarchical dynamics. **A** Five of the 24 possible input patterns. **B** Illustration of possible inputs. The $7 \times 7$ square can be at any position within the grey region. **C** Hierarchical self-organization of $k = 12$ minicolumnar RFs. **D** Non-hierarchical self-organization of $k = 12$ minicolumnar RFs.

to other suggested clustering methods, compare Tab. 1. Presumably because of less noise in the abstract system, not only the mean value of the simulation results is higher but also the standard deviation is significantly lower ($0.1$ compared to $1.7$ in the explicit system). The result shows that the abstraction is at least in this case superior to the explicit system used in Sec. 4.

5.5 Conclusion

In this section we derived a neural dynamics motivated by cortical connectivity. In contrast to explicitly defining a model macrocolumn, we derived the dynamics from a small set of basic assumptions on macrocolumn connectivity and plasticity of afferent fibers. The resulting system of differential equations (83) to (85) represents a continuous time version of the difference equation system (18) and (19) together with (26) and (27). In simulations of the dynamics using the same input data as for the dynamics of the preceding sections, the abstracted dynamics (83) to (85) could reproduce and improve RF self-organization results. The abstraction has proven to capture the essential dynamical features required for RF self-organization.
and represents a more compact and easier to handle dynamics than its predecessor. Furthermore, dynamics (83) to (85) is in a sense more independent of the concept of minicolumns and macrocolumn, i.e., any neural entities and connectivities giving rise to an equation system (83) to (85) possess the equivalent information processing and learning capabilities.
6 Rapid Dynamic Link Matching

In Sec. 3 we introduced explicit afferent connections \( R_{ij} \) to the macrocolumn model (18). The afferents were taken to originate from neurons \( n_j^{\text{ex}} \) of a set of external neurons termed \textit{input neurons} (see also Fig. 5). The neurons could be arranged arbitrarily, e.g., in Sec. 3 they were used to encode two-dimensional grey-level patterns and in Sec. 4 they encoded continuous valued input vectors. As was noted when they were first introduced, the external neurons can also model neurons of other macrocolumns instead of modeling neurons of sensory systems. A network of macrocolumns can, therefore, be modeled as a set of \( N \) macrocolumns in which each macrocolumn has as input neurons sensory neurons and all excitatory neurons of all other \((N - 1)\) macrocolumns.

For modeling macrocolumn networks we will make use of the model abstraction derived in Sec. 5 because the abstracted dynamical equations are easier to handle. In this section we will consider a macrocolumn network architecture which addresses the capability of biological networks to handle invariance. In the preceding chapters we have demonstrated that a single macrocolumn is capable of appropriately representing different kinds of input, if necessary, by using a distributed neural code. The representation of a given input is, however, not invariant under transformations such as translation of the input pattern. In fact, as the input neurons can encode any type of input, it is not clear which input transformations should leave the system output unchanged. However, for special types of inputs, e.g., for visual input, we quite explicitly know these types of transformations. A neural system processing visual information has to recognize objects more or less independent of position in the visual field, size, rotation etc. If a system can achieve this, we call it \textit{invariant} with respect to the corresponding transformations.

For artificial neural networks the invariance problem is a traditionally difficult one and classical network models such as Hopfield type associative memories (Hopfield, 1982; Hopfield and Tank, 1986) and perceptrons (e.g. Fukushima et al., 1983) or newer developments such as SOMs (Kohonen, 1995) are not invariant to transformations as mentioned above. Computational models applied to object recognition are usually based on one of two paradigms, they are \textit{correspondence-based} or \textit{feature-based}. Examples of correspondence-based systems are (Ullman, 1989; Hummel and Biederman, 1992; Olshausen et al., 1993) whereas systems like (Fukushima et al., 1983; Elliffe et al., 2002; Mel, 1997; Westphal and Würtz, 2004) are primarily feature-based (see, however, Elliffe et al., 2002, for a feature based system which uses a correspondence-based mechanism for learning). These two paradigms are often seen as alternative approaches (refer to Zhu and von der Malsburg, 2004, for a recent discussion) but they might just reflect different integral parts of our visual system.

The \textit{dynamic link matching} (DLM) network (Lades et al., 1993) is a correspondence-based system which has proven to be very successful in tasks of visual object recognition and in particular in the long standing benchmark of face recognition (Wiskott and von der Malsburg, 1996; Wiskott et al., 1997). The model is based on the short-term establishment of neural interconnections, \textit{links}, between neural units which are subject to corresponding stimuli.
DLM systems (von der Malsburg, 1981; Lades et al., 1993; Konen et al., 1994; Würtz, 1995b; Würtz, 1995a; Wiskott and von der Malsburg, 1996; Aonishi and Kurata, 1998; Aonishi and Kurata, 2000; Zhu and von der Malsburg, 2002; Zhu and von der Malsburg, 2004) invariantly recognize objects and distinguish faces. Its engineering predecessor elastic graph matching (EGM), e.g. (von der Malsburg, 1988; Lades et al., 1993; Wiskott et al., 1997), has continuously proven to be highly competitive (see, e.g., Philips et al., 2000; Messer et al., 2004) and has become the basis of commercial applications. Although it proved to be successful in many real world tasks, there remain unsolved problems if considering DLM as a model for information processing in the brain. All DLM implementations so far rely on implicitly non-neuronal computations and are far too slow to account for neurophysiological visual recognitions times which are known to lie in the 100ms range (Potter, 1976; Subramaniam et al., 1995; Thorpe et al., 1996a). Only the recently suggested system (Zhu and von der Malsburg, 2001; Zhu and von der Malsburg, 2004) based on an abstract non-neuronal analytical treatment achieved shorter recognition times when measured in terms of iteration steps required for finding correspondences (see Zhu and von der Malsburg, 2004, for details).

### 6.1 Principles of Dynamic Link Matching

Let us first present the concepts of DLM. The most basic DLM network consists of two layers the *input layer* and the *model layer*. Each layer itself consists of neural modules which are taken to represent *feature vectors* at different positions of an image. For a two-dimensional image these positions are usually points of a regular grid (see, e.g., Fig. 33) and for a one-dimensional input, as is always used as first step in development (see, e.g., Zhu, 2004; Zhu and von der Malsburg, 2004), the positions are given by a set of equally spaced points (compare Fig. 33). The basic DLM system consisting of one input and one model layer is essentially applied to the so-called *correspondence problem*, i.e., to the problem to find pairs of points in two different images of the same object, which correspond to the same points on the real object (see Fig. 34). For object recognition the basic system can then be extended in a straightforward way (see, e.g., Wiskott, 1995, for details).

For a position \( \bar{x} \) in an image a feature vector \( \mathcal{J}(\bar{x}) \) represents texture properties of the image in the vicinity of \( \bar{x} \). Usually the value of each vector entry \( \mathcal{J}_\alpha(\bar{x}) \) is given by the magnitude of the response of a Gabor filter operation with certain frequency and orientation (see Wiskott, 1995, for details). In principle a feature vector can, however, also contain other types of extracted features, e.g., color. A feature vector constructed using Gabor wavelet responses is also termed *jet* and, for simplicity, we will use this notation in the following also for a general feature vector.

The input layer represents a given input image by its extracted feature vectors and the model layer represents a model image in the same way\(^24\). Note that the set of extracted feature vectors of the input image will, for simplicity, also be called the

\(^24\)It can be shown that such a representation preserves the information crucial for object or face identity if a suitable Gabor filtering and a suitable spacing of the feature extraction positions is chosen (Wundrich, 2004), see also (Wundrich et al., 2004).
A B

Figure 33: Grey-level images with feature extraction points. A Two-dimensional image with $8 \times 10$ feature extraction points on a regular two-dimensional grid. A feature vector $\mathbf{J}(\mathbf{x})$ is usually computed in each of the grid points using two-dimensional Gabor wavelets. B One-dimensional image of 128 pixels with 9 feature extraction points. The image is horizontally expanded for visualization purposes. A feature vector $\mathbf{J}_i = \mathbf{J}(x_i)$ is usually computed using one-dimensional Gabor wavelets.

transformed input image or sometimes also just input image. The same applies for the model image.

The set of neural modules of the input layer is taken to be fully (all-to-all) and usually equally interconnected with the set of neural modules of the model layer (see Fig. 35). Given a transformed input and model image the task is to select out the links which interconnect corresponding points (see Fig. 34). To accomplish this task it is not enough to simply select out the links connecting the most similar jets because even in an input image which is an only slightly perturbed version of the model image the interconnections between the most similar jets will most likely not connect corresponding points.

To find the right correspondences it is necessary to encode the neighborhood relationship between the feature encoding units in a layer. We need a system dynamics which simultaneously considers jet-similarity and neighborhood relationship to select from all possible links the ones interconnecting the corresponding points.

In implementations of DLM (von der Malsburg, 1981; Lades et al., 1993; Würtz, 1995b; Wiskott and von der Malsburg, 1996), neighborhood relationship is coded by localized Amari-field like (Amari, 1977; Amari, 1983) so-called blobs of high neural activity within a layer. The layer dynamics is then coupled to a dynamics of short time-scale Hebbian plasticity which strengthens and weakens synapses depending on a similarity $S(\mathbf{J}, \mathbf{J}')$ between input and model jet and depending on simultaneous activity of the corresponding input and model unit. The latter dependence results via the blob dynamics (Wiskott, 1995) within the layers in a final connectivity state which is one-to-one and topological, i.e., neighboring units in the input layer are
connected to neighboring units in the model layer.

A detailed discussion of the dynamic equations of the different implementations would go far beyond the scope of this work and we, therefore, only mentioned the basic mechanisms (see Wiskott, 1995, for details). DLM implementations including a recent abstract version (Zhu and von der Malsburg, 2004) differ in detail but all have in common that they use feature vectors $\mathbf{J}(x)$ constructed using Gabor filter responses and that they are computing jet similarities using explicit formulas equal or similar to

$$S(\mathbf{J}, \mathbf{J}') = \frac{\sum_{\alpha=1}^{k} J_{\alpha}J'_{\alpha}}{\sqrt{\sum_{\alpha=1}^{k}(J_{\alpha})^2 \sum_{\alpha=1}^{k}(J'_{\alpha})^2}}, \quad (91)$$

where $J_{\alpha} (\alpha = 1, \ldots, k)$ are the amplitudes of the complex value resulting from a Gabor wavelet transformation.

The dynamic equations of the different DLM implementations are by definition translation invariant and robust with respect to various other transformations (see, e.g., Wiskott and von der Malsburg, 1996; Zhu and von der Malsburg, 2004, for discussions). The systems’ invariance to many such transformations is limited, however, by limited invariance of the extracted feature vectors.

As demonstrated, e.g., in (Wiskott and von der Malsburg, 1996) and (Konen et al., 1994) DLM implementations can be applied to various real world tasks such as object and face recognition in which they are competitive to other feature-based and correspondence-based systems. As models for visual information processing there remain several problems to solve, however.

The basic DLM architecture consisting of an input and a model layer can be regarded as a network model to study how the brain achieves short-term changes of its
Figure 35: Visualization of a basic DLM network. The system consists of two layers with neural units which encode feature vectors of an input and a model layer. The interconnection between input and model layer is all-to-all in the beginning. Via feature vector comparison and link competition the interconnection structure is dynamically reduced to a neighborhood preserving one-to-one interconnectivity, a match, which interconnects corresponding points in input and model image.

connectivity structure to combine associated neural stimuli for further processing. A more realistic architecture would of course consist of both, a hierarchy of more than two layers and a matching mechanism between the layers to interconnect corresponding activity patterns. First, however, a neural mechanism which implements the basic DLM architecture and which is consistent with neurobiological and psychophysical experiments has to be found because DLM implementations have three major drawbacks with respect to neurally plausible implementability:

- DLM implementations so far require fast changing synaptic weights and synapto-synaptic competition in order to build up a topological layer interconnection. However, synaptic weights would have to change reliably on the basis of very few spikes to correspond to neurophysiological computation times (see also below). It is not clear how this could be achieved in biological networks and DLM implementations do not give an explanation. In the system (Zhu and von der Malsburg, 2004) a fast map creation using higher-order synapto-
synaptic interactions was suggested. These interactions are also difficult to explain, however. It was argued in (Zhu and von der Malsburg, 2004) that astrocytes can serve as an implementation but in neuroscience astrocytes are usually not considered to play a functional role in biological information processing (see Antanitus, 1998, however) and especially not on the time-scale needed for DLM (see, e.g., Rose et al., 2003).

- DLM implementations lack an explanation of how the similarity between feature vectors (91) is neurally computed. In all implementations the jet similarity explicitly appears in the dynamic equations for links but there is no biological explanation. Although the problem is discussed, e.g. in (Zhu and von der Malsburg, 2004), there is no satisfying answer given in any of the DLM system descriptions.

- DLM implementations are, as mentioned earlier, far too slow to account for psychophysically measured computation times. In tasks of visual object recognition subjects are able to recognize objects with a rate of 10 per second (Potter, 1976; Subramaniam et al., 1995; Thorpe et al., 1996a), which implies that the visual system processes visual stimuli in time intervals at least as short as 100ms. If the visual system implemented a mechanism of matching on the bases of interactions as suggest, e.g., in the system (Wiskott and von der Malsburg, 1996), it would require about 3 – 10s for one recognition, which is about two orders of magnitudes to slow. Matching times for other DLM systems are comparably slow.

These severe drawbacks of DLM implementations indicate that the right biological correlates have yet to be found. Implementations so far make use of single neurons to represent feature vectors but it is not clear how activity of a neuron can actually encode a feature vector. Without a neural feature vector representation it seems impossible to neuronally compute feature vector similarities.

At the same time single neuron interactions are disadvantageous for a fast link dynamics. If we suppose that the exchange of EPSPs between two neurons determines the strengthening or weakening of their interconnection, we would have to wait till the neurons spike several times before a meaningful change of the connection strength is possible. This process has to be iterated until a suitable interconnection is established. Thus, even if all other interactions needed could be justified, and even if iteration times can be drastically reduced using higher-order link interactions as in (Zhu and von der Malsburg, 2004), the matching process would still be far too slow if neural interactions are based on single neuron spike rates.

### 6.2 Representation of a Feature Vector

Choosing single neurons as neural units for the representation of feature vectors does not only result in a very slow matching dynamics but can also be seen as reason for the use of a non-neuronal similarity computation identical or similar to (91). But what is the right neural object to represent a feature vector?
To answer the question we just have to look at neuroscientific experiments of visual perception again. As was discussed in Sec. 1, neurophysiological experiments suggest a columnar organization of the visual cortex. Neurons stacked orthogonal to the cortical surface tend to be sensitive to stimuli with similar orientation and similar frequency. RFs of such cells are often modeled using Gabor filter responses (Jones and Palmer, 1987), i.e., the sensitivities of all neurons in an orientation column in the visual cortex can be modeled by the response of a single Gabor filtering of an image patch using a Gabor wavelet with certain frequency and orientation.

A feature vector in DLM, on the other hand, contains different Gabor filter responses as vector entries. This inevitably leads to the conclusion that a collection of orientation columns has to be taken as the neural module to represent a feature vector.

Already in Sec. 3.5 we shortly discussed the interrelation between orientation columns and the macrocolumn model. Note, however, that the term hypercolumn\textsuperscript{25} is more commonly used in the visual cortex. All neurons of a minicolumn of our macrocolumn model have similar RFs (see Sec. 3). Furthermore, visual stimuli of certain orientations and frequencies are considered to be the basic constituents of visual input (see, e.g., Simoncelli and Schwartz, 1999) such that RFs specialized to such stimuli can be considered to be formed in a RF self-organization process similar to the one studied in Sec. 3. The macrocolumn model, as defined and discussed in the preceding sections is, therefore, the natural neural unit to represent a feature vector in a neuronally plausible DLM implementation.

As macrocolumn model we use the abstracted continuous dynamic model as derived in Sec. 5 (see Eqn. (77)):

\begin{align}
\frac{d}{dt} p_\alpha &= f(p_\alpha, \nu \max_{\beta=1,\ldots,k} \{p_\beta\}) + \kappa \tilde{\mathcal{J}}_\alpha(x) + \sigma \eta_t, \tag{92} \\
f(p, h) &= a p (p - h - p^2). \tag{93}
\end{align}

Instead of afferent fibers ($R_{ej}$) from external units $p_j^E$ we directly couple the minicolumn activity to the feature vector entry $\mathcal{J}_\alpha(x)$. In the case that the entry $\mathcal{J}_\alpha(x)$ is constructed using a Gabor filter response, Eqn. (92) models an orientation column which receives input from an area of the visual field with center $x$. Note that we take a general feature vector $\tilde{\mathcal{J}}(x)$ to consist of real valued vector entries between zero and one, $\tilde{\mathcal{J}}(x) \in [0, 1]^k$, and that $\tilde{\mathcal{J}}(x)$ is the zero-mean version of $\tilde{\mathcal{J}}(x)$:

\begin{align}
\tilde{\mathcal{J}}_\alpha(x) &= \mathcal{J}_\alpha(x) - \frac{1}{k} \sum_{\beta=1}^{k} \mathcal{J}_\beta(x) \tag{94}
\end{align}

As the position $x$ of feature vector $\tilde{\mathcal{J}}(x)$ is defined by fixed positions in a one or two-dimensional model and input image, we will label the feature vectors by the index of their corresponding neural units, e.g., $\tilde{\mathcal{J}}_{\mathcal{I}^i}$ will be the feature vector represented by unit $i$ of input layer $\mathcal{I}$.

\textsuperscript{25}A hypercolumn is rather neurophysiologically defined and usually also includes occular dominance columns and color blobs. For simplicity occular dominance and color is not considered here.
As activation function of the minicolumn, Eqn. (93), we use function (67) because we have shown in Sec. 5 that the function possesses the desired bifurcation behavior. In principle, however, we can use any function with qualitatively the same bifurcations. As discussed in Sec. 5, the dynamics of (92) symmetrizes the activities $p_{\alpha}$ for small $\nu$ and deactivates $p_{\alpha}$'s depending on the input $(\kappa \tilde{J}_{\alpha}(\vec{x}) + \sigma \eta_{l})$ if $\nu$ is increased. Using Eqn. (92) we represent a feature vector by its effect on minicolumn activities $p_{\alpha}$ and we can now proceed by considering interconnections between macrocolumns for computing feature vector similarities on a neural basis.

### 6.3 A Network of Two Macrocolumns

Let us first consider a simple elementary macrocolumn interconnection. Consider two macrocolumns $\mathcal{I}$ and $\mathcal{M}$ ($\mathcal{I}$ for input and $\mathcal{M}$ for model) with the same number of minicolumns $k$. Let macrocolumn $\mathcal{I}$ represent a feature vector $\vec{J}_{\mathcal{I}}$, then the minicolumn activities $p_{\alpha}$ are given by (see (92)):

$$\frac{d}{dt} p_{\alpha} = f(p_{\alpha}, \nu \max_{\beta=1,\ldots,k} \{p_{\beta}\}) + \kappa \tilde{J}_{\alpha}(\vec{x}) + \sigma \eta_{l}.$$  \hspace{1cm} (95)

If the macrocolumn additionally receives input from macrocolumn $\mathcal{M}$ we get:

$$E_{\alpha}^{\mathcal{I}} = C_{E} \tilde{J}_{\alpha}^{\mathcal{I}} + (1 - C_{E}) \sum_{\beta=1}^{k} R_{\alpha\beta}^{\mathcal{I},\mathcal{M}} p_{\beta}^{\mathcal{M}}$$  \hspace{1cm} (96)

$$\frac{d}{dt} p_{\alpha} = f(p_{\alpha}, h(\vec{p})) + \kappa E_{\alpha}^{\mathcal{I}} + \sigma \eta_{l},$$  \hspace{1cm} (97)

where $E_{\alpha}^{\mathcal{I}}$ is the total input to minicolumn $\alpha$ of macrocolumn $\mathcal{I}$ and where $(R_{\alpha\beta}^{\mathcal{I},\mathcal{M}})$ defines the input from macrocolumn $\mathcal{M}$ to minicolumn $\alpha$. $C_{E}$ is a real parameter in the interval $[0, 1]$.

The second macrocolumn $\mathcal{M}$ also receives input from two sources, from a feature vector $\vec{J}_{\mathcal{M}}$ and from macrocolumn $\mathcal{I}$:

$$E_{\alpha}^{\mathcal{M}} = C_{E} \tilde{J}_{\alpha}^{\mathcal{M}} + (1 - C_{E}) \sum_{\beta=1}^{k} R_{\alpha\beta}^{\mathcal{I},\mathcal{M}} p_{\beta}^{\mathcal{I}}$$  \hspace{1cm} (98)

$$\frac{d}{dt} p_{\alpha} = f(p_{\alpha}, h(\vec{p})) + \kappa E_{\alpha}^{\mathcal{M}} + \sigma \eta_{l}.$$  \hspace{1cm} (99)

The afferents from macrocolumn $\mathcal{I}$ to column $\mathcal{M}$ and visa versa, we take to be feature conserving, i.e., minicolumns sensitive to the same subfeatures (to the same feature vector entries) are interconnected:

$$R_{\alpha\beta}^{\mathcal{M},\mathcal{I}} = R_{\alpha\beta}^{\mathcal{I},\mathcal{M}} = \delta_{\alpha\beta} - \frac{1}{k}.$$  \hspace{1cm} (100)

The additional negative term results in contributions of the second term in (98) which are identical to zero in the case that all external minicolumns are equally active.

In Fig. 36 the system is sketched for the case of $k = 4$ minicolumns per macrocolumn. In the system both macrocolumns are subject to the same oscillation of
Figure 36: Visualization of the explicit feature preserving interconnections between two macrocolumns of $k = 4$ minicolumns.

their proportionality factor of inhibition $\nu$ (compare (83)):

$$
\nu(t) = \begin{cases} 
0 & \text{if } \tilde{t} < T_{\text{init}} \\
\left(\nu_{\text{max}} - \nu_{\text{min}}\right) \frac{\tilde{t} - T_{\text{init}}}{T - T_{\text{init}}} + \nu_{\text{min}} & \text{if } \tilde{t} \geq T_{\text{init}}
\end{cases}
$$

(101)

where $\tilde{t} = \text{mod}(t, T)$ (see Eqn. (24)). For low values of $\nu$ the macrocolumns are in the symmetric stationary state, i.e., all minicolumns in a macrocolumn are equally active. If $\nu$ is increased, the macrocolumns are forced to break the symmetry depending on their inputs. If all minicolumns are active, the effect of one macrocolumn on the other macrocolumn is neutralized because of (100) and the system reaction is just dependent on the feature vector inputs. A macrocolumn will, therefore, first switch off minicolumns which are associated with the smallest jet entries.

As soon as the first symmetry breakings occur, e.g., macrocolumn $\mathcal{M}$ receives input from feature vector entries $\mathcal{J}_a^\mathcal{M}$ and additionally from the minicolumns of the other macrocolumn. Because of the chosen one-to-one connectivity between the macrocolumns we get in the case that $\mathcal{J}_a^\mathcal{I} = \mathcal{J}_a^\mathcal{M} = (1, 1, 0, 0)^T$, high input to the first two minicolumns and a high probability that the state $\vec{p}^\mathcal{I} = \vec{p}^\mathcal{M} = (\vec{P}_1^0, \vec{P}_1^0, 0, 0)^T$ is stabilized in both minicolumns early in a $\nu$-cycle. If both macrocolumns have stabilized this state, it remains stable even for higher values of $\nu$ because all inputs cooperate. If the feature vectors are different, e.g., $\mathcal{J}_a^\mathcal{I} = (1, 1, 0, 0)^T$ and $\mathcal{J}_a^\mathcal{M} = (0, 0, 1, 1)^T$, macrocolumn $\mathcal{I}$ can be expected to stabilize the state $\vec{p}^\mathcal{I} = (\vec{P}_1^0, \vec{P}_1^0, 0, 0)^T$ and macrocolumn $\mathcal{M}$ state $\vec{p}^\mathcal{M} = (0, 0, \vec{P}_1^0, \vec{P}_1^0)^T$. In this case the macrocolumns are inhibiting each other via their interconnection and the stabilized states decay early in a $\nu$-cycle. The degree of mutual cooperation between the two macrocolumns, which is reflected by a long survival of a stabilized activity state, can, therefore, serve as a measure for the similarity of the represented features. Note that strong cooperation and stabilization of two macrocolumns means a relatively large number of exchanged EPSPs whose post-synaptic minicolumns are active.
6.4 A Macrocolumn Network for Similarity Matching

Instead of evaluating exactly how similar two feature vectors are, a correspondence based recognition system must be able to find, in a set of feature vectors, the most similar feature vector with respect to a given model feature vector $\mathbf{f}_M$. A first possibility would be to interconnect, e.g., $N = 3$ macrocolumns, $\mathcal{I}_1, \mathcal{I}_2,$ and $\mathcal{I}_3,$ to a single model macrocolumn $\mathcal{M}$ using the same interconnection as defined in Eqn. (100) (see Fig. 36). Such a network indeed best stabilizes the activity of the macrocolumn with an activity state most similar to the activity in the model macrocolumn. However, the network only works reliably for very low numbers of input macrocolumns. For larger $N$, the interference of inputs from macrocolumns $\mathcal{I}_1$ to $\mathcal{I}_N$ to macrocolumn $\mathcal{M}$ results in the deactivation of position encoding minicolumns based on input ambiguities.

For a network which reliably finds the most similar feature vector even in a large set of input feature vectors we need a mechanism to switch off the effect of input macrocolumns on the model macrocolumn and on the basis of macrocolumn activity state comparison. The implementation of such a mechanism is possible by making use of an additional macrocolumn which controls the inputs of macrocolumns $\mathcal{I}_1$ to $\mathcal{I}_N$. For this macrocolumn with $N$ minicolumns we will use the term position encoding macrocolumn. A network with $N = 3$ input macrocolumns is sketched in Fig. 37. Macrocolumn $\mathcal{M}$ receives input from $N$ input macrocolumns $\mathcal{I}_1, \ldots, \mathcal{I}_N$ and each such input is weighted by the activity $\tilde{\mathbf{w}}_{\mathcal{M},\mathcal{I}_j}$ of the corresponding minicolumn of the position encoding macrocolumn. The dynamics of the feature encoding macrocolumns now reads:

\[
\frac{d}{dt} p_{\alpha} = f(p_{\alpha}, \nu \max_{\beta=1,\ldots,k} \{p_{\beta}\}) + \kappa \tilde{\mathbf{f}}_{\alpha} + \sigma \eta_t
\]

\[
E_{\alpha}^{\mathcal{M}} = C_E \tilde{\mathbf{f}}_{\alpha} + (1 - C_E) \sum_{j=1}^{N} \sum_{\beta=1}^{k} \tilde{\mathbf{w}}_{\mathcal{M},\mathcal{I}_j} R_{\alpha\beta}^{\mathcal{M},\mathcal{I}_j} p_{\beta}^{\mathcal{I}_j}
\]

\[
R_{\alpha\beta}^{\mathcal{M},\mathcal{I}_j} = \delta_{\alpha\beta} - \frac{1}{k},
\]

The interconnections $R_{\alpha\beta}^{\mathcal{M},\mathcal{I}_j}$ are again feature preserving (see (100)),

\[
\frac{d}{dt} p_{\alpha} = f(p_{\alpha}, \nu \max_{\beta=1,\ldots,k} \{p_{\beta}\}) + \kappa E_{\alpha}^{\mathcal{M}} + \sigma \eta_t
\]

The dynamics of the position encoding macrocolumn we take to be identical to the dynamics of the feature encoding macrocolumns, i.e., we use equations (92) and (93) to model its changes of the minicolumnar activities $W_{\mathcal{M},\mathcal{I}_j}$ ($j = 1, \ldots, N$):

\[
\frac{d}{dt} W_{\mathcal{M},\mathcal{I}_j} = f(W_{\mathcal{M},\mathcal{I}_j}, \nu \max_{i=1,\ldots,N} \{W_{\mathcal{M},\mathcal{I}_i}\}) + \kappa I_{\mathcal{M},\mathcal{I}_j} + \sigma \eta_t,
\]

where $I_{\mathcal{M},\mathcal{I}_j}$ denotes the total input to minicolumn $j$ of the position encoding macrocolumn in unit $\mathcal{M}$. Note that $\tilde{\mathbf{w}}_{\mathcal{M},\mathcal{I}_j}$ in Eqn. (103) is the zero-mean transformed of
Figure 37: A network for similarity matching. On the input side the network consists of three feature encoding macrocolumns $I_1$, $I_2$, and $I_3$ which are representing three input feature vectors. On the model side the system consists of a feature encoding macrocolumn $M$ which is representing a model jet and of a position encoding macrocolumn which controls the incoming connections of the model macrocolumn.

$W^{M,I_j}$ given by

$$W^{M,I_j} = W^{M,I_j} - \frac{1}{N} \sum_{l=1}^{N} W^{M,I_l}. \quad (107)$$

The subtractive term has a similar effect as the subtractive terms in (94) and (105). As mentioned at the end of Sec. 6.3, a measure for the activity state similarity of two macrocolumns is the number of exchanged EPSPs whose post-synaptic neurons lie in active minicolumns. The interconnection of our system is, therefore, chosen such that the number of ‘successful’ EPSPs is received as minicolumn inputs of the position encoding macrocolumn:

$$I^{M,I_j} = \sum_{\alpha=1}^{k} \sum_{\beta=1}^{k} p^{M}_{\alpha} R^{M,I_j}_{\alpha\beta} p^{I_j}_{\beta} . \quad (108)$$

Note that to implement (103) and (108) multiplicative interconnections are required, i.e., the effect of $W^{M,I_j}$ on $E^{M}_{\alpha}$ in (103) is not additive but multiplicative and the
same applies for the effect of $p^\alpha_\nu$ on $I^{M,T_j}$ in (108). Multiplicative synaptic effects are commonly described, however, and we can rely on a large number of scientific contributions which describe this type of synaptic impact theoretically (see, e.g., Salinas and Abbott, 1996) as well as experimentally (e.g. Gabbiani et al., 2002).

The system of equations (102) to (108) can be simulated with simultaneous oscillation of $\nu$ as given in (101). In the initialization phase, for very small $\nu$ all minicolumns of the network are activated. If $\nu$ is increased, the macrocolumns undergo symmetry breakings, i.e., they start to deactivate their minicolumns with weakest inputs. The first minicolumns which are switched off are minicolumns of the feature encoding macrocolumns. The minicolumns of the position encoding macrocolumn all remain active because the inputs $I^{M,T_j}$ are all about equal to zero (compare Eqn. (100)). Only after some minicolumns of the feature encoding macrocolumns have been deactivated such that these macrocolumns truly represent the feature vectors, the term (108) results in different inputs to the position encoding macrocolumn. Subsequently, the position encoding minicolumns with lowest input $I^{M,T_j}$ are switched off, i.e., $W^{M,T_j}$ is zero for some $j$. Zero $W^{M,T_j}$ means, on the other hand, that the corresponding macrocolumn $T_j$ does not influence the feature encoding model macrocolumn $M$ anymore. The process continues until finally just one $W^{M,T_{j_0}}$ survives the selection process. Via Eqn. (103) the surviving, i.e. non-zero, $W^{M,T_{j_0}}$ keeps the coupling between input macrocolumn $T_{j_0}$ and macrocolumn $M$ active whereas all other links are deactivated.

As different simulations have shown, the input macrocolumn representing the feature vector $\vec{J}^{T_j}$, which is the most similar27 to the model feature vector $\vec{J}^M$, is indeed always the one whose connection $W^{M,T_{j_0}}$ survives the selection process. This mechanism will be the basis for more complex networks which will be discussed below. The network discussed in this section can then be regarded as a limiting case of the more complex systems. For detailed parameter settings and simulations results we, therefore, refer to experiments below.

6.5 Matching in a Macrocolumn Network

As discussed in Sec. 6.1 the elementary DLM system consists of an input layer and a model layer of feature encoding neural units. In the section above the model layer merely consisted of a single feature representing neural unit – the model macrocolumn $M$. In this section we consider a system which consists of $N$ input units $T_1, \ldots, T_N$ and also of $N$ model units $M_1, \ldots, M_N$. Each unit itself consists of a feature encoding macrocolumn with minicolumn activities $p^{Li}_\alpha$ ($L \in \{T,M\}$) and a position encoding macrocolumn with minicolumn activities $W^{Li,L_j}$ ($L_j \in \{T,M\}\backslash\{L\}$). Note that we will denote the feature encoding macrocolumn of the $i$th unit of layer $L$ by $p^{Li}$ and the position encoding macrocolumn by $W^{Li}$. With suitable values of $\nu_{\text{max}}$ the system can also be operated such that several $W^{M,T_{j_0}}$ survive. 

\footnote{26With suitable values of $\nu_{\text{max}}$ the system can also be operated such that several $W^{M,T_{j_0}}$ survive.}

\footnote{27Here, similarity between feature vectors has to be understood heuristically and not in the sense of (91). The similarity function (91) has turned out to be useful if the vector entries correspond to Gabor wavelet responses (see, e.g., Lades et al., 1993). However, also for general feature vectors (91) predicts well which connection $W^{M,T_j}$ finally survives but it can be expected that the implicit similarity computations of the network are different in detail to the computations given by (91).}
6 Rapid Dynamic Link Matching

A position encoding macrocolumn \( W^{\mathcal{L}i} \) controls the incoming connections to its feature encoding macrocolumn \( p^{\mathcal{L}i} \) exactly as the position encoding macrocolumn of the system in Sec. 6.4. The difference is that we now have \( N \) model units and that all \( N \) input units, \( \mathcal{I}_1, \ldots, \mathcal{I}_N \), have control over their incoming connections.

The resulting dynamics is a straightforward generalization of equations (102) to (108). For the model layer they now read:

\[
E_{\alpha i}^{\mathcal{M}i} = C_E \tilde{J}_{\alpha i}^{\mathcal{M}i} + (1 - C_E) \sum_{j=1}^{N} \sum_{k=1}^{K} \tilde{W}^{\mathcal{M}i,\mathcal{I}j} R_{\alpha j}^{\mathcal{M}i,\mathcal{I}j} p_{\beta j}^{\mathcal{M}i} \tag{109}
\]

\[
\frac{d}{dt} p_{\alpha i}^{\mathcal{M}i} = f \left( p_{\alpha i}^{\mathcal{M}i}, \nu \max_{\beta=1,\ldots,k} \{ p_{\beta j}^{\mathcal{M}i} \} \right) + \kappa E_{\alpha i}^{\mathcal{M}i} + \sigma_{\text{no.}i} \tag{110}
\]

\[
I_{\alpha i,\mathcal{I}j}^{\mathcal{M}i} = \sum_{\alpha=1}^{N} \sum_{k=1}^{K} R_{\alpha j}^{\mathcal{M}i,\mathcal{I}j} p_{\alpha i}^{\mathcal{M}i} p_{\beta j}^{\mathcal{M}i} \tag{111}
\]

\[
\frac{d}{dt} \tilde{W}^{\mathcal{M}i,\mathcal{I}j} = f \left( \tilde{W}^{\mathcal{M}i,\mathcal{I}j}, \nu \max_{l=1,\ldots,N} \{ \tilde{W}^{\mathcal{M}i,\mathcal{I}l} \} \right) + \kappa I_{\alpha i,\mathcal{I}j}^{\mathcal{M}i} + \sigma_{\text{no.}i} \tag{112}
\]

For the input layer the same equations apply if we replace \( \mathcal{M} \) by \( \mathcal{I} \) and \( \mathcal{I} \) by \( \mathcal{M} \) (simultaneously). Note that we will use the notation \( \mathcal{L} \in \{ \mathcal{I}, \mathcal{M} \} \) and \( \mathcal{L}' \in \{ \mathcal{I}, \mathcal{M} \} \setminus \{ \mathcal{L} \} \) in the following. Eqn. (107) now reads

\[
\tilde{W}^{\mathcal{L}i,\mathcal{L}'j} = W^{\mathcal{L}i,\mathcal{L}'j} - \frac{1}{N} \sum_{l=1}^{N} W^{\mathcal{L}i,\mathcal{L}'l}, \tag{113}
\]

and the connectivity between the layers is again feature preserving (see (105)):

\[
R_{\alpha \beta}^{\mathcal{L}i,\mathcal{L}'j} = \delta_{\alpha \beta} - \frac{1}{k} \tag{114}
\]

\( \tilde{J}_{\alpha} \) denote, again, the entries of zero-mean versions of feature vectors \( \tilde{J} \in [0,1]^k \):

\[
\tilde{J}_{\alpha i}^{\mathcal{L}i} = J_{\alpha i}^{\mathcal{L}i} - \frac{1}{k} \sum_{\beta=1}^{k} J_{\beta i}^{\mathcal{L}i} \tag{115}
\]

The system with the above dynamics consists of units \( \mathcal{I}i \) and \( \mathcal{M}j \), which successively switch off the incoming connections or links to their corresponding feature encoding macrocolumns. At a given time a unit \( \mathcal{I}i \) hereby switches off the link \((\mathcal{L}'j \rightarrow \mathcal{L}i)\) originating from a remaining connected unit \( \mathcal{L}'j \). The unit \( \mathcal{L}'j \) which is disconnected is with high probability the unit among the still connected units which represents the jet \( \tilde{J}^{\mathcal{L}'j} \) with least similarity.

In Fig. 39 the activity state of the system is shown during a \( \nu \)-cycle at \( \nu = 0.5 \). In Fig. 39A,B the feature vectors \( \tilde{J}^{\mathcal{L}i} \) and \( \tilde{J}^{\mathcal{M}i} \) are shown as two-dimensional grey-level images. For the simulation in Fig. 39 we use \( N = 30 \) \((j = 1, \ldots, N)\) feature vectors per layer and \( k = 20 \) \((\alpha = 1, \ldots, k)\) entries per feature vector. Fig. 39A shows, from left to right, input jets, the activities \( p_{\alpha i}^{\mathcal{L}i} \) of the feature encoding macrocolumns \( p^{\mathcal{L}i} \) of the input layer, and the activities \( W^{\mathcal{I}i,\mathcal{M}j} \) of the position encoding...
Figure 38: Sketch of a network to find a one-to-one interconnectivity on the basis of feature similarities. The network consists of an input and model layer with neural units $I_1$ to $I_3$ and $M_1$ to $M_3$, respectively. Each neural unit consists of a feature encoding macrocolumn with $k = 4$ minicolumns and of a position encoding macrocolumn with $N = 3$ minicolumns. Each neural unit in the input layer receives input of each neural unit in the model layer. The inputs to a unit are controlled by its position encoding macrocolumn according to the interconnectivity as displayed in Fig. 37.
Figure 39: Activity state of the dynamics (109) to (112) for input jets \((\mathcal{J}_\alpha^{Ti})\) and model jets \((\mathcal{J}_\alpha^{Mi})\). The set of input jets is identical to the set of model jets. The activity is shown during a \(\nu\)-cycle at \(\nu = 0.5\). In A, input jets \((\mathcal{J}_\alpha^{Ti})\) are shown together with activities in the input layer, \((p_\alpha^{Ti})\) and \((W^{Ti,Mj})\). In B, model jets \((\mathcal{J}_\alpha^{Mi})\) are shown together with activities in the model layer, \((p_\alpha^{Mi})\) and \((W^{Mi,Ij})\).
macrocolumns $W^Z_i$. Fig. 39B displays, from left to right, position encoding macrocolumns $W^{M_i}$, feature encoding macrocolumn $p^{M_i}$, and feature vectors $\tilde{J}^{M_i}$ of the model layer.

For the experiment in Fig. 39 the transformed input and model image are identical, i.e., the collection of input jets is identical to the collection of model jets. We would, therefore, expect a diagonal in the images for $(W^{Z_i,M_j})$ and $(W^{M_i,Z_j})$ at least at the end of a $\nu$-cycle because each model jet $\tilde{J}^{M_i}$ has as its most similar counterpart the input jet $J^{M_i}$. A diagonal is indeed what is obtained in simulations at the end of a $\nu$-cycle if we use the same arbitrary randomly generated image as model and input image. The $20 \times 30$ transformed image of Fig. 39 is generated, however, by, firstly, generating a $20 \times 10$ image in which pixel values are equally, independently, and identically distributed in the interval $[0, 1]$. Secondly, three copies of the $20 \times 10$ image are used to construct an image with $k = 20$ columns and $N = 30$ rows. The rows of this $20 \times 30$ image are finally randomly shuffled to obtain the image used in Fig. 39. This means that for each model jet $\tilde{J}^{M_i}$ there are three identical jets with maximal similarity in the input layer and visa versa. The network, therefore, stabilizes all these connections simultaneously and we obtain for $\nu = 0.5$ the activation (or the link-configuration) $(W^{Z_i,M_j})$ and $(W^{M_i,Z_j})$ as displayed in Fig. 39A and Fig. 39B, respectively. If $\nu$ is further increased, each position encoding macrocolumn spontaneously breaks the activity state which is symmetrically keeping the approximately three remaining links active. The result is a final activity state which leaves only about one third of the minicolumns on the diagonals of $(W^{Z_i,M_j})$ and $(W^{M_i,Z_j})$ activated.

### 6.6 Topological Matching

The system of the preceding section generated a one-to-one connectivity, a match, between the one-dimensional input and model layer. The match was generated using purely neural mechanisms, i.e., without the explicit use of a similarity function such as (91). The experiment in Fig. 39 has demonstrated, however, that the dynamics (109) to (112) are not sufficient to find corresponding points between one-dimensional feature arrays of one-dimensional images. A crucial property of the dynamics which is missing is a topological constraint of the final connectivity. We do not only require the system to converge to a sparse one-to-one connectivity but we also want this connectivity to be topology preserving, i.e., we expect the final connectivity to map neighboring units in the one layer to neighboring units in the other layer.

The dynamics (109) to (112) does not in general not result in topological connectivity structures because it lacks an encoding of neighborhood relationship in the first place. The missing neighborhood information is reflected by a notable asymmetry in the equation system (109) to (112). Note that all feature encoding macrocolumns $p^{L_i}$ ($L \in \{I, M\}$) receive input from two sources, from the jet they encode and from feature encoding macrocolumns of the other layer (see Eqn. (109) and Fig. 38). The position encoding macrocolumns $W^{L_i}$, on the other hand, only receive input from one source – the feature encoding macrocolumns. A coupling of the position encoding macrocolumns to other position encoding macrocolumns, therefore, represents a
Figure 40: Sketch of a macrocolumn network for topological matching. The network consists of an input and model layer with neural units $I_1$ to $I_3$ and $M_1$ to $M_3$, respectively. Each neural unit consists of a feature encoding macrocolumn with $k = 4$ minicolumns and of a position encoding macrocolumn with $N = 3$ minicolumns. Each neural unit in the input layer receives input of each neural unit in the model layer. The inputs to a unit are controlled by its position encoding macrocolumn according to the interconnectivity as displayed in Fig. 37. Position encoding macrocolumns are coupled to the activities of feature encoding macrocolumns of both layers and to neighboring position encoding macrocolumns (compare Fig. 41).
natural extension of the dynamics (109) to (112). Furthermore, a coupling of the position encoding macrocolumns to neighboring such macrocolumns is a natural way to encode neighborhood relationship (see Fig. 40 and compare Fig. 35). The complete dynamics of the system with the additional neighborhood encoding coupling term is given by:

\[
E_{\alpha}^{\xi_{i}} = C_{E} \mathcal{F}_{a}^{\xi_{i}} + (1 - C_{E}) \sum_{j=1}^{k} \sum_{\beta=1}^{N} W_{\mathcal{L}_{i},\mathcal{L}_{j}} R_{\alpha \beta}^{\xi_{i},\xi_{j}} p_{\beta}^{\xi_{j}}
\]

(116)

\[
\frac{\text{d} p_{\alpha}^{\xi_{i}}}{\text{d}t} = f(p_{\alpha}^{\xi_{i}}, \nu \max_{\beta=1,\ldots,k} \{p_{\beta}^{\xi_{i}}\}) + \kappa E_{\alpha}^{\xi_{i}} + \sigma_{\text{no}} \eta_{t}
\]

(117)

Note that all empty entries are zero. To understand the interaction of position encoding macrocolumns to neighboring such macrocolumns is a natural way to encode neighborhood relationship (see Fig. 40 and compare Fig. 35). The complete dynamics of the system with the additional neighborhood encoding coupling term is given by:

\[
I_{\mathcal{L}_{i},\mathcal{L}_{j}}^{\xi_{i},\xi_{j}} = C_{I} \sum_{\alpha,\beta=1}^{k} p_{\alpha}^{\xi_{i}} R_{\alpha \beta}^{\xi_{i},\xi_{j}} p_{\beta}^{\xi_{j}} + (1 - C_{I}) \sum_{a,b=1}^{N} T_{\mathcal{L}_{a},\mathcal{L}_{b}}^{\xi_{i},\xi_{j}} W_{\mathcal{L},\mathcal{L}'}^{\xi_{i},\xi_{j}}
\]

(118)

\[
\frac{\text{d} W_{\mathcal{L}_{i},\mathcal{L}_{j}}^{\xi_{i},\xi_{j}}}{\text{d}t} = f(W_{\mathcal{L}_{i},\mathcal{L}_{j}}^{\xi_{i},\xi_{j}}, \nu \max_{\ell_{i}=1,\ldots,N} \{W_{\mathcal{L}_{i},\mathcal{L}'}^{\xi_{i},\xi_{j}}\}) + \kappa I_{\mathcal{L}_{i},\mathcal{L}_{j}}^{\xi_{i},\xi_{j}} + \sigma_{\text{no}} \eta_{t}
\]

(119)

Note that we used the notation \( \mathcal{L} \in \{\mathcal{I}, \mathcal{M}\} \) and \( \mathcal{L}' \in \{\mathcal{I}, \mathcal{M}\} \setminus \{\mathcal{L}\} \). The matrix \( T_{\mathcal{L}_{i},\mathcal{L}_{j}}^{\xi_{i},\xi_{j}} \) now interconnects all position encoding macrocolumns \( W_{\mathcal{L},\mathcal{L}'}^{\xi_{i},\xi_{j}} \) of the same layer. To account for a one-dimensional neighborhood relationship within a layer we choose the following explicit form of \( T_{\mathcal{L}_{i},\mathcal{L}_{j}}^{\xi_{i},\xi_{j}} \) with \( L = 4 \):

\[
T_{\mathcal{L}_{i},\mathcal{L}_{j}}^{\xi_{i},\xi_{j}} = \sum_{c=-L}^{L} \sum_{d=-L}^{L} A_{c,d} \delta_{a,i+c} \delta_{b,i+d} - \frac{1}{N}
\]

(120)

The matrix \( (A_{c,d}) \) is the \( 9 \times 9 \) matrix \( (c = -4, \ldots, +4, d = -4, \ldots, +4) \):

\[
(A_{c,d}) = \begin{pmatrix}
0 & 0 & 0.3 & 0.6 & 0.3 & 0.1 & 0.8 & 0.1 & 0 \\
0 & 1.0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0.1 \\
0 & 0 & 0 & 0.6 & 0.3 & 0.3 & 0.6 & 0.3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

(121)

Note that all empty entries are zero. To understand the interaction of position encoding macrocolumns consider the limiting case \( A_{c,d} = \delta_{c,d} - \delta_{c,0} \delta_{d,0} \) and \( N = \infty \). If we now insert (120) into (118) the topological term becomes:

\[
(1 - C_{I}) \sum_{c=-L}^{L} \sum_{d=-L}^{L} W_{\mathcal{L}^{(i+c)},\mathcal{L}'^{(j+c)}}
\]

(122)

The intra-layer connectivity, in this case, is illustrated in Fig. 41. A minicolumn \( j \) of the \( i \)th position encoding macrocolumn of layer \( \mathcal{L} \), \( W_{\mathcal{L}^{i},\mathcal{L}^{j}} \), controls the link from
6 Rapid Dynamic Link Matching

Figure 41: Visualization of the interconnection of position encoding macrocolumns within a model layer (likewise input layer). Interconnection like given in (122) is displayed. Afferents to minicolumn \( W^{M,i,j} \) for \( L = 2 \) are displayed in black to better visualize the connection structure. Afferents to other minicolumns of the same macrocolumn are displayed in grey.

6.6.1 Emergence of Topological Interconnectivity

The influence of the topological term in (118) is best studied by setting \( C_I \) to zero, i.e., we only consider interactions within each layer. In this case we get a decoupling of Eqns. (116) and (117) from Eqns. (118) and (119) and a decoupling of the two layer dynamics. Thus Eqns. (118) and (119) can be simulated for an isolated layer.

For the simulation we use a layer of \( N = 30 \) units and cyclic boundary conditions, i.e., the last unit of the layer has the first unit as successor. We choose the parameters \( \kappa = 1.0 \text{ms}^{-1} \) and \( \sigma_{\text{no.}} = 0.01 \text{ms}^{-1} \). The parameter \( a \) of the function \( f \) in (119) is chosen as in Sec. 5.2. \( a = 200 \text{ms}^{-1} \). The system is operated with oscillating \( \nu \), (101), with period length\(^{28}\) \( T = 25 \text{ms} \), \( T_{\text{init}} = 2 \text{ms} \), \( \nu_{\text{min}} = 0.4 \) and \( \nu_{\text{max}} = 0.52 \) (a value slightly above the critical value \( \nu_c = 0.5 \), see Sec. 5). A typical time course of the minicolumn activities \( W^{L};C_i,C_j \) during a \( \nu \)-cycle is shown in Fig. 42. As in Sec. 5 we use the Euler method (see, e.g., Honerkamp, 1994) with time steps \( \Delta t \leq \frac{1}{200} \text{ms} \). For all simulations in this section we use \( \Delta t = \frac{1}{100} \text{ms} \).

In the beginning all minicolumns \( W^{L};C_i,C_j \) of all macrocolumns of the system are equally active because of the symmetrizing effects of the dynamics for low \( \nu \).

\(^{28}\)In the simulations the reset of the network is supported by a higher noise level, \( \sigma = 0.5 \), in the initialization phase \( t < T_{\text{init}} \). Alternatively, we can choose a longer initialization phase to activate all minicolumns. Note, however, that not activating all minicolumns can be advantages because matching results of a \( \nu \)-cycle can in this case carry over to the next \( \nu \)-cycle.
Figure 42: Time course of network activities \((W^{L_i,L_j})\) during a \(\nu\)-cycle for no input if only neighborhood coupling is used \(C_I = 0\). The interconnection encoding dynamic variables \((W^{L_i,L_j})\) are displayed in A at \(\nu = 0.465\), in B at \(\nu = 0.47\), in C at \(\nu = 0.475\), in D at \(\nu = 0.49\), in E at \(\nu = 0.5\), and in F at \(\nu = 0.51\).
The symmetric state remains stable for a long time because input differences to the macrocolumns are only caused by the noise term in (119). Only at about \( \nu = 0.46 \), i.e., relatively close to the bifurcation point \( \nu_c = 0.5 \), the random fluctuations are sufficient to break the symmetry of some macrocolumns in the layer. Via the topological term the asymmetric activity of these macrocolumns quickly drives the neighboring macrocolumns to switch off their corresponding minicolumns (see Fig. 42A). This process continues (see Fig. 42B-D) until only a small subset of neighboring minicolumns per macrocolumn remains active (see Fig. 42D). Because of the special shape of the topological term defined in Eqns. (120) and (121) the set of active minicolumns in two neighboring macrocolumns correspond to approximately parallel links. If \( \nu \) further increases, the diagonal of active minicolumns gradually gets smaller (see Fig. 42D-E) until just one minicolumn per macrocolumn remains active (see Fig. 42F). In the full system such a final activity state corresponds precisely to a connectivity of the units of a layer \( \mathcal{L}' \) to layer \( \mathcal{L} \) which is one-to-one and neighborhood preserving. As there is no input to the system and due to cyclic boundary conditions, the final activity state, i.e., the final layer interconnectivity, is randomly chosen via the process of symmetry breaking. The topological term forces the system to finally converge to a neighborhood preserving one-to-one connectivity. The final state is sometimes reached faster than in the example of Fig. 42 and sometimes slower. Occasionally it occurs that we get two broken diagonals. A result which corresponds to a one-to-one connectivity which is only piecewise topological.

The behavior of the system is reminiscent of the self-organization of optical nerve fibers in the ontogenesis of the visual system as described in (Häussler and von der Malsburg, 1983). In dynamics (116) to (119) the connectivity between layers is explicitly controlled by neuron populations whereas the so-called Häussler equations (Häussler and von der Malsburg, 1983) contain dynamical variables describing the interconnection strengths themselves. Furthermore, we explicitly control differentiation pressure using the parameter \( \nu \) which allows the system to reset to full connectivity. The major difference is the time scale, however. The Häussler equations model the development of topological connectivity on the slow time-scale of ontogenesis of the visual system whereas the dynamics above operates on the time scale of visual perception. Despite these differences the qualitative behavior is, nevertheless, very similar and can be regarded as emerging from the general principles of DLM which itself was once suggested as a kind of short time-scale version of the Häussler equations.

### 6.6.2 Topological Matching

Let us now turn to the simulation of the complete dynamics (116) to (119). Note that using \( C_1 = 1 \) in (119) we would drop back to dynamics (109) to (112). These dynamics failed to converge to a neighborhood preserving interconnectivity in the experiment of Fig. 39. The experiment of Fig. 42 showed that the system with \( C_1 = 0 \) converges to topological interconnectivity which does, however, not interconnect units representing similar jets. An image based correspondence seeking system we expect to integrate both properties. The system must find a topological one-to-one connectivity by simultaneous evaluation of jet similarity.
Using dynamics (116) to (119) this is simply done by setting \( C_I \) to an intermediate value between zero and one. For the experiment in Fig. 44 we choose \( C_I = 0.5 \). All other parameters are chosen as above and \( C_E \) in (116) is set to \( C_E = 0.6 \), i.e., the feature encoding macrocolumns are slightly more sensitive to feature vector input than to input of the other layer. As (transformed) input and model images we use two matrices each with \( N = 30 \) rows (corresponding to \( N = 30 \) feature vectors of a one dimensional image, see Fig. 33) and \( k = 20 \) columns (corresponding to \( k = 20 \) jet entries). The model image \((\tilde{J}^M)\) is the same as the one displayed in Fig. 39A,B, i.e., it consists of randomly ordered copies of 10 different feature vectors whose entries contain equally, identically, and independently distributed random values between zero and one. The (transformed) input image \((\tilde{J}^I)\) is generated from the model image by adding Gaussian white noise with \( \sigma = 0.6 \) to the values \((\tilde{J}^M)\). Subsequently, we rescale the set of all values \((\tilde{J}^I)\) such that all jet vector entries lie in the interval \([0, 1]\) again. The resulting image \((\tilde{J}^I)\) is a strongly perturbed copy of the model \((\tilde{J}^M)\) with much smaller mean jet entry difference due to the noise and rescaling (see Fig. 43). In this way we hope to model at least some of the properties of perturbations in natural images.

The time-course of all minicolumn activities during a \( \nu \)-cycle given the input of Fig. 43 is shown in Fig. 44. In the beginning of the \( \nu \)-cycle all minicolumns of all macrocolumns are activated and symmetrized. In this state all position encoding macrocolumns receive input without significant differences due to homogeneously active feature encoding macrocolumns. The feature encoding macrocolumns \( p^I \) and \( p^M \), however, are coupled to the feature vectors \( \tilde{J}^I \) and \( \tilde{J}^M \), which results in input differences to their minicolumns. The symmetries in the feature encoding macrocolumns are, therefore, broken first (starting at about \( \nu = 0.42 \)). The more specific the feature encoding by the macrocolumns \( p^I \) and \( p^M \), the larger

![Figure 43: Visualization of input jets \( J^I \) and model jets \( J^M \) as used for the experiments in Fig. 44 and Fig. 45](image-url)
Figure 44: Time course of all network activities during a $\nu$-cycle if the feature vectors of Fig. 43 are used. The dynamic variables of the system, $(p^L_i)$, $(W^{L,M_j})$, $(W^{M_i,L_j})$, and $(p^{M_i})$ are displayed in A at $\nu = 0.45$, in B at $\nu = 0.455$, C at $\nu = 0.46$, and D at $\nu = 0.49$. Note that if one $\nu$-cycle requires about 25ms, the time difference between the activity states in A and D corresponds to about 8ms.
are the input differences to the position encoding macrocolumns $W^{T_i}$ and $W^{M_i}$ (see Eqn. (118)). Between $\nu = 0.44$ and $\nu = 0.45$ the input differences are large enough to break the activity symmetries in one or some position encoding macrocolumns. Note that this is earlier than spontaneous symmetry breaking in Fig. 42. Driven by the similarity term and the topological term in Eqn. (118) the deactivation of minicolumns quickly results in the deactivation of minicolumns of neighboring macrocolumns which control approximately parallel links between units encoding feature vectors (see Fig. 44A) with low similarity. The process continues until only few minicolumns remain active (see Fig. 44B,C). The remaining active minicolumns compete on the basis of jet similarity between the units of the link they control and between the units of neighboring approximately parallel links. Simultaneously, the feature encoding macrocolumns represent the jets with increasing accuracy. This also influences competition within position encoding macrocolumns. Note that the symmetries in $p^{M_i}$ are, hereby, broken earlier than in $p^{T_i}$ (see Fig. 44) because of the smaller jet entry differences resulting from the noise in the input image (compare Fig. 43). For $\nu = 0.49$ the activity state of $(W^{T_i,M_j})$ and $(W^{M_i,I_j})$ already corresponds to a symmetric, one-to-one, and neighborhood preserving layer to layer mapping which topologically interconnects the corresponding features. The mapping remains stable during the rest of the $\nu$-cycle.

The result is remarkable if one considers the large differences between input jets $\bar{J}^{T_i}$ and model jets $\bar{J}^{M_i}$ (see Fig. 43). On the basis of jet similarity alone ($C_i = 1$) the final connectivity is, due to the feature vector differences, far from the desired topological interconnection between corresponding units (see Fig. 45A). If only neighborhood relationships drive the selection process ($C_i = 0$) we get (for both directions different) topological mappings which do not interconnect any corresponding points between image and model (see Fig. 45B). Only if the dynamics are driven by both neighborhood relationships and feature vector similarities ($C_i = 0.5$), will the final mapping be topologically interconnecting corresponding points in image and model (see Fig. 45C).

For the input and model image constructed as the images in Fig. 43 the system finds the right correspondences in almost all $\nu$-cycles. If the noise level in the input image is even higher than for the generation of the input image in Fig. 43, the system sometimes does not find the right match, which is reflected, e.g., by the survival of discontinuous diagonals of matrices $(W^{T_i,M_j})$ or $(W^{M_i,I_j})$. The system behavior gets worse if the number $k$ of feature vector entries is decreased but improves for $k$ larger than $k = 20$. Note that technical applications (Philips et al., 2000; Messer et al., 2004) employ feature vectors of $k \geq 40$ vector entries and that biological macrocolumns are estimated to contain about $k = 80$ (see, e.g., Mountcastle, 1997).

Of course, system dynamics (116) to (119) has to be studied in many further simulations but results as shown in Fig. 44 show a very promising dynamic behavior. In particular, dynamics (116) to (119) represent the first DLM inspired entirely neural system and it removes the drawbacks of DLM systems as discussed in Sec. 6.1:

- The system does not require fast changing synaptic weights and synapto-synaptic interactions to form a topological match between the layers $\mathcal{L}$ and $\mathcal{L}'$. Instead it uses position encoding macrocolumns, which control the incoming fibers to feature encoding macrocolumns. The synapto-synaptic compe-
Figure 45: Activity states of the position encoding macrocolumn \((W^{Z_i,M_j})\) and \((W^{M_i,Z_j})\) at the end of a \(\nu\)-cycle, \(\nu = 0.52\), if the feature vectors of Fig. 43 are used. 

A Final state if exclusively jet similarities are used to find the right connectivity, \(C_I = 1.0\). 

B Final state if exclusively neighborhood coupling is used, \(C_I = 0.0\). 

C Final state if both, jet similarity and neighborhood interconnectivity, is used, \(C_I = 0.5\).
tions are replaced by multiplicative synaptic interactions which represent a well justified concept (Salinas and Abbott, 1996; Gabbiani et al., 2002). However, the use of macrocolumns especially as controlling neural units for layer interconnectivity requires a much greater number of neurons than other DLM systems. A system like the one presented needs for one layer with \( N \) neural units \( N(N+k) \) minicolumns. For \( N = 80 \) units (e.g., as grid points in Fig. 34) with \( k = 40 \) feature vector entries (as used in technical applications, see e.g., Philips et al., 2000) we require 9600 minicolumns. For comparison, one hemisphere of the cat’s area 17 is estimated to contain 160000 minicolumns (Peters and Payne, 1993) of about 200 neurons\(^{29}\). A more detailed comparison is difficult, however, because, as is pointed out above, the basic two layer system presented in this section can only be regarded as modeling the inter-layer dynamics of a biologically more realistic system with recurrent hierarchical feature processing.

- Using a neurophysiologically and neuroanatomically inspired feature vector representation by macrocolumns the system neuro-dynamically computes feature vector similarities by the interaction between macrocolumns. In contrast to all other DLM implementations no explicit similarity function like (91) is used.

- Dynamics (116) to (119) converge from an initial activity state with activated all-to-all connectivity between the layers to a connectivity state, which topologically interconnects corresponding points, within one \( \nu \)-cycle. Simulation time of a \( \nu \)-cycle if translated to real-world time corresponds to a time as short as 25ms, which is made possible by the population rate code of macrocolumns (compare Sec. 2.5) together with a fast symmetry breaking mechanism induced by bifurcation parameter \( \nu \). Even if we include signal traveling times between the neural units a match can still be found well within 100ms which corresponds to measured processing times of the visual system (Potter, 1976; Subramaniam et al., 1995; Thorpe et al., 1996a).

### 6.7 Conclusion and Future Work

**Conclusion.** In this last section we studied a network of interconnected macrocolumns. The coupling between the macrocolumns was taken to be weak (small \( \kappa \)) but the system nevertheless behaves reliably because of the process of induced symmetry breakings studied in Sec. 2 and Sec. 5. In contrast to Sec. 3 and Sec. 5 the afferent connections to the minicolumns, i.e. their RFs, are explicitly defined in this section. The matrices \( R_{\alpha j}^{L_i C_j} \) interconnect the layers and are chosen to be feature preserving and matrices \( T_{ab}^{L_i C_j} \) realize intra-layer connectivity and are chosen to encode neighborhood relationship. The explicit macrocolumn network architecture was motivated by the special purpose of this network – the application

\(^{29}\)In Sec. 1 and Sec. 2 the minicolumn model was motivated on the neuroanatomical side by the connectivity within pyramidal cell modules as described in (Peters and Payne, 1993; Peters and Yilmaze, 1993). One hemisphere of area 17 of the cat is estimated to contain 160000 such modules.
to the correspondence problem. The dynamics was inspired by the DLM network approach to this problem. Unlike other DLM systems, however, we based our dynamics on well-investigated, very sensible, and rapidly reacting building blocks – model macrocolumns. Firstly motivated by severe drawbacks of DLM implementations, we derived a system of coupled differential equations which in its final version, Eqns. (116) to (119), is surprisingly compact and symmetric. Especially compared to other DLM systems (e.g., Wiskott, 1995), it possesses few parameters whose dynamical meanings are, furthermore, intuitively conceivable (e.g., \( C_1 \) in (118)). Furthermore, the dynamics removes the crucial drawbacks of former neural DLM systems – the dynamics is now based on common and widely accepted neural interactions and the matching process takes place in neurophysiological reaction time.

**Future Work.** Dynamics (116) to (119) have proven to possess all the properties required to find correspondences in two topologically ordered sets of feature vectors. The next natural step is to apply the system to the correspondence problem of two-dimensional real-world images, e.g., Fig. 33. For that purpose we have to choose explicit feature vectors in a first step. In the case of two-dimensional grey-value real-world images, feature vectors constructed using Gabor filter responses have proven to be highly successful (see, e.g., Philips et al., 2000; Messer et al., 2004) and first experiments using one-dimensional Gabor filter versions (compare e.g. Zhu and von der Malsburg, 2004) for dynamics (116) to (119) show promising results.

A second step would consist of encoding a two-dimensional neighborhood relationship due to two-dimensional layers \( \mathcal{I} \) and \( \mathcal{M} \) in that case. In principle all that has to be changed is the topological coupling matrix \( (T_{ab}^{c_i,c_j'}) \) in (118)\(^{30}\). Matrix \( (A_{c,d}) \) of \( (T_{ab}^{c_i,c_j'}) \) has to be a two dimensional version of the softened Kronecker delta in (121) in this case. It will depend on the explicit form of this matrix how gracefully the dynamics handle differences between input and model image. An appropriate choice of \( (T_{ab}^{c_i,c_j'}) \) should also enable the system to handle differences due to scaling and deformation as well as changes due to translation and rotation.

A softened version of the Kronecker delta for \( (R_{ab}^{c_i,c_j'}) \) in (114) can also contribute to increased performance and robustness of the system. The entries in \( (R_{ab}^{c_i,c_j'}) \) could, e.g., reflect relationships between jet entries of neighboring frequencies or orientations.

The search space to find the right entries for \( (T_{ab}^{c_i,c_j'}) \) and \( (R_{ab}^{c_i,c_j'}) \) is quite large, however, and one would like to find a way to determine the entries using example images. Note at this point that the matrices are nothing but RFs of minicolumns and that we have derived a very capable mechanism of unsupervised learning for their formation in the case of a single macrocolumn. Combining the above dynamics with the derived self-organization of minicolumnar RFs, therefore, offers a way to learn a suitable jet similarity computation and to learn invariance by adapting \( (R_{ab}^{c_i,c_j'}) \) and \( (T_{ab}^{c_i,c_j'}) \), respectively. The learning is unsupervised such that it would be sufficient to present the system with a suitable set of real-world images subject to various forms of transformations.

If we can successfully apply the system to real-world correspondence problems,

---

\(^{30}\)We would, of course, rather like to use suitable two-dimensional indices \( a \) and \( b \) as well.
we could subsequently use it for object recognition. In the DLM systems object recognition is performed by making use of several model layers instead of a single one (Lades et al., 1993; Wiskott and von der Malsburg, 1996). After having established correspondence mappings of the input layer to all the different model layers, the DLM systems compute over-all similarities between the layers using the pairs of corresponding jets and an explicit similarity function such as (91). Again, in the DLM systems there is no explanation of how the feature vectors are compared neurally.

The extension to multiple model layers is of course also possible using the dynamics derived in this section. If the correspondence mappings are established, a connectivity is activated which allows a direct neuronal comparison between input and model feature vectors – the feature preserving interconnectivity \( R^{c_i, c_j}_{\alpha \beta} \) of Eqn. (116). The second term of (116) is crucial in actually transferring information between corresponding units (see also Sec. 6.3). Note at this point that the term also allows the system to keep the previously established inter-layer connectivity active for high \( \nu \) even if the inputs of the model and the input jets are removed. Thus, the network realizes a kind of associative memory of interconnection structures.

A drawback of the recognition systems based on multiple model layers is, of course, its extensive use of neurons, poor generalization ability, and the lack of plausibility as a model of biological object recognition. A more realistic object recognition system would require a combination of hierarchical feature processing and correspondence matching. A network of macrocolumns driven by a single parameter \( \nu \) for synchronized decision pressure in its units has proven to possess the prerequisites of such a system. If we can successfully combine and structure such a network using the advanced unsupervised feature extraction mechanisms studied in Sec. 3 and Sec. 5, the system will be able to learn invariant correspondence finding and appropriate recurrent feature processing. The required building blocks seem to be at hand.
7 Summary

Motivated by the columnar connectivity of the cortex we defined a model of the cortical macrocolumn based on McCulloch-Pitts like spiking neurons (Sec. 1 and Sec. 2). An analysis of the system’s dynamics has revealed a rich diversity of stationary points whose stabilities sensitively depend on the proportionality factor of inhibition. If the inhibition parameter is oscillating, the minicolumns are deactivated during one oscillation in a cascade of spontaneous symmetry breakings. If the minicolumns receive inputs from external neurons, the activity symmetries are broken on the basis of input differences. In simulations the oscillation’s period length could be very short, in the range of 25ms. Macrocolumn sensitivity to external signals is, because of the symmetry breaking process, very high even for weak coupling to external inputs.

If afferent fibers to the model macrocolumn are subject to phase coupled Hebbian plasticity (Sec. 3), a RF self-organization process is induced. Driven by input spike patterns RFs of minicolumns self-organize to pattern classes or extract features of input consisting of pattern superpositions. In the continuous bars test it was shown that the system can equidistantly cover the subspaces of extracted features – an ability of abstract learning which offers unequaled opportunities.

In Sec. 4 we showed that RF self-organization can be applied to the problem of clustering if spike patterns encode continuous variables.

In Sec. 5 the neural dynamical behavior and RF self-organization could be derived from basic assumptions on dynamical properties of macrocolumns. The dynamics of discrete time neurons could be replaced by a set of differential equations of minicolumn activities and minicolumnar RFs.

Using these abstracted dynamics we studied an architecture of interconnected macrocolumns in Sec. 6, which was motivated by the dynamic link matching approach and is applicable to the correspondence problem. In the network all macrocolumns are coupled to the same oscillating inhibition parameter which, during an oscillation, synchronously increases the decision pressure in the neural units. The chosen interconnection of the macrocolumns allows the simultaneous evaluation of feature vector similarities and neighborhood relationship. Within one oscillation the system establishes a neighborhood preserving one-to-one connectivity which interconnects corresponding points of a model and an input image.

The system suggests a specific functional relationship between computations in the cortex and its ubiquitous neural oscillations: neural oscillations synchronize and induce decision making in cortical neural modules and lead to a short-term rewiring of cortical interconnectivity to process information by context sensitive communication between corresponding neural units.

As is the case for dynamic link matching systems, the macrocolumn network is invariant to translation and robust with respect to other transformations. However, it removes the severe drawbacks of dynamic link matching implementations. The rapid matching in macrocolumn networks does not require fast synaptic plasticity and synapto-synaptic interactions, it computes feature vector similarities neuronally, and it finds correspondence mappings in neurophysiologically plausible computation times.
Appendix A

Stability Analysis of Difference Equation System (13)

The Jacobian of difference equation system with \( I \) instead of \( L \), see (15),

\[
p_{\alpha}(t+1) = \Phi_{\alpha}\left( \frac{p_{\alpha}(t) - \mathcal{I}(\bar{p}) - \Theta_{\alpha}}{\sqrt{p_{\alpha}(t)(1 - p_{\alpha}(t))}} \right) (1 - p_{\alpha}(t)) =: G_{\alpha}(\bar{p}(t)),
\]

at \( q \in \{0,q\}^k \) with \( q = \mathcal{P}_\nu \), see (14), and \( l \) zero entries, is given by:

\[
\bar{G}(\bar{q}) = \begin{pmatrix}
H_i(q) & F_i(q) & \cdots & F_i(q) \\
F_i(q) & H_i(q) & \ddots & \vdots \\
\vdots & \ddots & \ddots & F_i(q) \\
F_i(q) & \cdots & F_i(q) & H_i(q) \\
0 & 0 & \cdots & 0
\end{pmatrix},
\]

where

\[
H_i(q) := \frac{\partial}{\partial p_{\alpha}} G_{\alpha}|_{p=q}, \quad \alpha = 1, \ldots, l
\]

(124)

\[
F_i(q) := \frac{\partial}{\partial p_{\alpha}} G_{\beta}|_{p=q}, \quad \alpha, \beta = 1, \ldots, l, \quad \alpha \neq \beta
\]

(125)

Eigenvalues of the Jacobian above are given by (compare computations in Appendix B):

\[
\lambda_0 = 0
\]

(126)

\[
\lambda_1 = H_i(q) - F_i(q) + \lambda F_i(q)
\]

(127)

\[
\lambda_2 = H_i(q) - F_i(q)
\]

(128)

\( \lambda_0 \) is of multiplicity \((k-l)\), \( \lambda_1 \) is of multiplicity 1, and \( \lambda_2 \) is of multiplicity \((l-1)\).

Long but straightforward computations of (124) and (125) finally result in

\[
\lambda_0 = 0,
\]

\[
\lambda_1 = \frac{1}{2\sqrt{q(1-q)}} (1 - \frac{1}{\tau^\nu} + \frac{1}{q} \Theta_{\alpha}) \Phi'_s(h_i(\nu)) - \Phi_s(h_i(\nu)),
\]

\[
\lambda_2 = \frac{1}{2\sqrt{q(1-q)}} (1 + (1 - 2q) \frac{1}{\tau^\nu} + \frac{1}{q} \Theta_{\alpha}) \Phi'_s(h_i(\nu)) - \Phi_s(h_i(\nu)),
\]

where \( h_i(\nu) = \frac{(1 - \frac{1}{\tau^\nu}) q - \Theta_{\alpha}}{\sqrt{q(1-q)}} \).

For \( \rho \to \infty \) we finally get with \( q = \mathcal{P}_\nu \):

\[
\lambda_0 = 0,
\]
\[
\lambda_1 = \frac{1}{2 \sqrt{P_v (1 - P_v)}} (1 - \nu + \frac{1 - 2 P_v}{P_v} \Theta_o) \Phi'_\nu(h(\nu)) - \Phi_s(h(\nu)),
\]
\[
\lambda_2 = \frac{1}{2 \sqrt{P_v (1 - P_v)}} (1 + (1 - 2 P_v) \nu + \frac{1 - 2 P_v}{P_v} \Theta_o) \Phi'_s(h(\nu)) - \Phi_s(h(\nu)),
\]
where \( h(\nu) = \frac{(1 - \nu) P_v - \Theta_o}{\sqrt{P_v (1 - P_v)}} \).

**Appendix B**

**Stability Analysis of General Differential Equation Systems (51)**

The Jacobian of (51) for a stationary point \( \vec{q} \in Q_i \) (50) is given by

\[
(\partial_\beta F_\alpha)(\vec{q}) = (\partial_1 f)(q_\alpha, h(\vec{q})) \delta_{\alpha \beta} + (\partial_2 f)(q_\alpha, h(\vec{q}))(\partial_\beta h)(\vec{q}),
\]
where

\[
\vec{q} = \left( \frac{P_i^0, \ldots, P_i^0, P_i^1, \ldots, P_i^1, \ldots, P_i^J, \ldots, P_i^J}{m_1(\vec{q}), \ldots, m_J(\vec{q})} \right).
\]

Using definition (15) for \( h_\rho \) we derive with \( \rho \in \{2, 4, \ldots\} \):

\[
\left( \frac{\partial}{\partial p_\alpha} \mathcal{I}_\rho \right)(\vec{q}) = \frac{\partial}{\partial p_\alpha} \nu \left( \sum_{\beta=1}^{k} (p_\beta)^\rho \right)^{\frac{1}{\rho}} |_{\rho = \vec{q}}
\]

\[
= \nu \left( \sum_{\beta=1}^{k} (p_\beta)^\rho \right)^{\frac{1}{\rho}-1} \left( p_\alpha^\rho \right)^{\frac{1}{\rho}-1} |_{\rho = \vec{q}}
\]

\[
= \nu \left( l(\vec{q}) (P_i^0)^\rho + \sum_{j=1}^{J} m_j(\vec{q})(P_i^j)^\rho \right)^{\frac{1}{\rho}-1} \left( p_\alpha^\rho \right)^{\frac{1}{\rho}-1} |_{\rho = \vec{q}}
\]

\[
= \nu \left( l(\vec{q}) + \sum_{j=1}^{J} m_j(\vec{q}) \left( \frac{P_i^j}{P_i^0} \right)^\rho \right)^{\frac{1}{\rho}-1} \left( p_\alpha^\rho \right)^{\frac{1}{\rho}-1} |_{\rho = \vec{q}}
\]

\[
= \left\{ \begin{array}{ll}
\nu \left( l(\vec{q}) + \sum_{j=1}^{J} m_j(\vec{q}) \left( \frac{P_i^j}{P_i^0} \right)^\rho \right)^{\frac{1}{\rho}-1} \left( p_\alpha^\rho \right)^{\frac{1}{\rho}-1} & \text{for } \alpha \leq l(\vec{q}) \\
\nu \left( l(\vec{q}) + \sum_{j=1}^{J} m_j(\vec{q}) \left( \frac{P_i^j}{P_i^0} \right)^\rho \right)^{\frac{1}{\rho}-1} \left( p_\alpha^\rho \right)^{\frac{1}{\rho}-1} & \text{for } \alpha > l(\vec{q})
\end{array} \right.
\]

where \( j' \geq 1 \). It follows in the limit:

\[
\Rightarrow \left( \frac{\partial}{\partial p_\alpha} \mathcal{I} \right)(\vec{q}) = \left\{ \begin{array}{ll}
\nu \frac{1}{l} & \text{for } \alpha \leq l(\vec{q}) \\
0 & \text{for } \alpha > l(\vec{q})
\end{array} \right.
\]

because by definition (50) \( P_i^{j'} < P_i^{j} \) for \( j' \geq 1 \).

For a stationary point \( \vec{q} \in Q_i \), (130), we therefore get with
\[ \beta \leq l: \quad (\partial_\beta F_\alpha)(\vec{q}) = (\partial_1 f)(q_\alpha, h(\vec{q}))\delta_{\alpha\beta} + (\partial_2 f)(q_\alpha, h(\vec{q})) \frac{\nu}{I} \]
\[ \beta > l: \quad (\partial_\beta F_\alpha)(\vec{q}) = (\partial_1 f)(q_\alpha, h(\vec{q}))\delta_{\alpha\beta} \]

a Jacobian of the form
\[
F'(\vec{q}) = \begin{pmatrix}
B & 0 & 0 & 0 \\
A_1 & e_1 \mathbb{1} & 0 & 0 \\
\vdots & 0 & \ddots & 0 \\
A_l & 0 & 0 & e_l \mathbb{1}
\end{pmatrix},
\]

where \( e_j \mathbb{1} \) denote \( m_j \times m_j \) matrices and \( A_j \) \( l \times m_j \) matrices of the form
\[
e_j \mathbb{1} = \begin{pmatrix}
e_{j_1} & 0 & \cdots & 0 \\
0 & e_{j_2} & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & e_{j_l}
\end{pmatrix}
\text{and} \quad A_j = \begin{pmatrix}
a_{j_1} & a_{j_1} & \cdots & a_{j_1} \\
a_{j_2} & a_{j_2} & \cdots & \vdots \\
\vdots & \ddots & \ddots & a_{j_1} \\
a_{j_l} & \cdots & a_{j_1} & a_{j_l}
\end{pmatrix},
\]

and where \( B \) is an \( l \times l \) matrix of the form
\[
B = \begin{pmatrix}
c & d & \cdots & d \\
d & c & \ddots & \vdots \\
\vdots & \ddots & \ddots & d \\
d & \cdots & d & c
\end{pmatrix}.
\]

The matrix entries are given by:
\[
c = (\partial_1 f)(\mathcal{P}_i^0, \nu \mathcal{P}_i^0) + \frac{\nu}{I} (\partial_2 f)(\mathcal{P}_i^0, \nu \mathcal{P}_i^0) \tag{131}
\]
\[
d = \frac{\nu}{I} (\partial_2 f)(\mathcal{P}_i^0, \nu \mathcal{P}_i^0) \tag{132}
\]
\[
a_{j} = \frac{\nu}{I} (\partial_2 f)(\mathcal{P}_i^j, \nu \mathcal{P}_i^0) \tag{133}
\]
\[
e_{j} = (\partial_1 f)(\mathcal{P}_i^j, \nu \mathcal{P}_i^0) \tag{134}
\]

Using the formula (see, e.g., Böhme, 1993):
\[
\text{det}(\begin{pmatrix} \vec{B} & C_1 \\ C_2 & D \end{pmatrix}) = \text{det}(\vec{B}) \text{det}(D - C_2 \vec{B}^{-1} C_1), \tag{135}
\]

for square matrices \( \vec{B} = B - \lambda \mathbb{1} \) and \( D = (\bigoplus_{j=1}^{l} e_j \mathbb{1}) - \lambda \mathbb{1} \) we get because of \( C_1 = \mathbb{0} \):
\[
\text{det}(F'(\vec{q}) - \lambda \mathbb{1}) = \text{det}(\vec{B}) \text{det}(D). \tag{136}
\]

The matrix \( \vec{B} \) is of the form \( \vec{B} = dE - \mathbb{1}x \) with \( x = d - c + \lambda \) where \( E \) is a \( l \times l \) matrix whose entries are all identical one. \( dE \) has rang 1 and eigenvector \( \vec{u} = (1, \ldots, 1)^T \) with eigenvalue \( \lambda = ld \). The eigenvectors orthogonal to \( \vec{u} \) have eigenvalue \( \tilde{\lambda} = 0 \) and there are \((l - 1)\) of them. For the matrix \( \vec{B} \) we therefore get the eigenvalues \( \mu_1 = ld - x \) with multiplicity 1 and \( \mu_2 = -x \) with multiplicity \((l - 1)\). We can compute:
\[
\text{det}(\vec{B}) = \mu_1 \mu_2 = (ld - d + c - \lambda)(c - d - \lambda)^{l-1} \tag{137}
\]
\[
\text{det}(F'(\vec{q}) - \lambda \mathbb{1}) = (c - d + ld - \lambda)(c - d - \lambda)^{l-1} \prod_j (e_j - \lambda)^{m_j} \tag{138}
\]
Inserting (131) to (134) into (138), we obtain the eigenvalues:

\[
\begin{align*}
\lambda_1 &= (\partial_1 f)_{p_{\alpha}, \nu p_{\beta}} + \nu (\partial_2 f)_{p_{\alpha}, \nu p_{\beta}} & \text{multiplicity 1} \\
\lambda_2 &= (\partial_1 f)_{p_{\alpha}, \nu p_{\beta}} & \text{multiplicity } (l - 1) \\
\lambda_{2+j} &= (\partial_1 f)_{p_{\alpha}, \nu p_{\beta}} & \text{multiplicity } m_j 
\end{align*}
\]  

(139)

**Appendix C**

**Computations for** \(b \neq 1\) **and** \(\Theta \neq 0\)

Let us study the more complex dynamic system

\[
\frac{d}{dt} p_{\alpha} = a p_{\alpha} (p_{\alpha} - \nu \max_{\beta=1, \ldots, k} \{p_{\beta}\} - \Theta - b p_{\alpha}^2),
\]

(140)

where \(a, b > 0\), and \(\Theta \geq 0\). Note first that for (140) there exists a \(\nu_0 = 1 - \sqrt{4b\Theta}\) above which there are no non-trivial stationary points. We will therefore assume that \(b\) and \(\Theta\) are chosen such \(\nu_0 > 0\) and that \(\nu > 0\) is smaller than \(\nu_0\). With this assumption we get the following sets \(Q_i\):

\[
\begin{align*}
\mathcal{P}_1^0 &= \frac{1}{2b} \left( (1 - \nu) + \sqrt{(1 - \nu)^2 - 4b\Theta} \right), \\
\mathcal{P}_2^0 &= \frac{1}{2b} \left( (1 - \nu) - \sqrt{(1 - \nu)^2 - 4b\Theta} \right), \\
\mathcal{P}_3^0 &= 0 \\
\mathcal{P}_1^1 &= \frac{1}{b} - \mathcal{P}_1^0, \\
\mathcal{P}_2^2 &= 0 \\
\mathcal{P}_2^1 &= 0
\end{align*}
\]

\[
\begin{align*}
Q_1 &= \left\{ (\mathcal{P}_1^0, \ldots, \mathcal{P}_l^0, \mathcal{P}_1^1, \ldots, \mathcal{P}_m^1, 0, \ldots, 0) \text{ and permutations } |l \geq 1; m_1, m_o \geq 0 \right\} \\
Q_2 &= \left\{ (\mathcal{P}_2^0, \ldots, \mathcal{P}_l^0, 0, \ldots, 0) \text{ and permutations } |l \geq 1, m \geq 0, l + m = k \right\} \\
Q_3 &= \{(0, \ldots, 0)\}
\end{align*}
\]

Note that the solution of \(f(q, \nu \mathcal{P}_1^0) = 0, q = \frac{1}{2b} \left( 1 + \sqrt{1 - 4b\nu^2 \mathcal{P}_1^0 - 4b\Theta} \right)\), is equal to \(\mathcal{P}_1^0\) which is not obvious on first sight. \(\mathcal{P}_1^1\) therefore presents the only non-trivial solution.

As it was the case for \(b = 1\) and \(\Theta = 0\), \(\mathcal{P}_1^1\) can become trivial for \(\nu > \nu_c\) and \(Q_1\) gets smaller. Because of the existence of \(\nu_0\) it might be the case, however, that \(\nu_c\) does not exist. For the following let us assume that \(\nu_c < \nu_0\) exists. For \(\nu < \nu_c\) we
get for the stationary points in $Q_1$ the eigenvalues:

\[
\begin{align*}
\lambda_0 &= -a(\nu P_1^0 + \Theta) & \text{multiplicity } m_0 \\
\lambda_1 &= a(2(1 - \nu)P_1^0 - \Theta - 3b(P_1^0)^2) & \text{multiplicity 1} \\
\lambda_2 &= a((2 - \nu)P_1^0 - \Theta - 3b(P_1^0)^2) & \text{multiplicity } (l - 1) \\
\lambda_3 &= a(2P_1^1 - \nu P_1^1 - \Theta - 3b(P_1^1)^2) & \text{multiplicity } m_1
\end{align*}
\]

For $\nu > \nu_c$, $\lambda_3$ does not exist and $\lambda_2$ gets larger than zero.

The stability of the stationary points in $Q_2$ is given by the eigenvalues:

\[
\begin{align*}
\lambda_0 &= -a(\nu P_2^0 + \Theta) & \text{multiplicity } m_0 \\
\lambda_1 &= a(2(1 - \nu)P_2^0 - \Theta - 3b(P_2^0)^2) & \text{multiplicity 1} \\
\lambda_2 &= a((2 - \nu)P_2^0 - \Theta - 3b(P_2^0)^2) & \text{multiplicity } (l - 1)
\end{align*}
\]

And the stability of $Q_3$ is given by the eigenvalue $\lambda = -a\Theta$.

The dynamical behavior of (140) is for a wide range of parameters similar to the behavior of the system with $b = 1$ and $\Theta = 0$. Only the newly existing set $Q_2$ presents a larger difference. The points of $Q_2$ actually reflect the establishment of a region of the phase space near to the origin, which is attracted by the state of zero activity (see Fig. 27).
List of Figures

1 Pyramidal cell modules ........................................... 7
2 Bifurcation diagram for a macrocolumn with $k = 2$ ........ 14
3 Bifurcation diagram for a macrocolumn with $k = 3$ .......... 15
4 Afferents to pyramidal cell modules ........................... 16
5 Sketch of a macrocolumn and receptive field of a minicolumn 21
6 Oscillation of $\nu$ and system behavior .......................... 23
7 Sketch of a macrocolumn and receptive field self-organization 25
8 Pattern classification ............................................. 27
9 Bars test .......................................................... 29
10 Results of different bars tests ................................... 31
11 Noisy bars test ..................................................... 33
12 Hierarchical RF self-organization – simple example .......... 37
13 Comparison of hierarchical and non-hierarchical self-organization 40
14 Classification of hand-written digits ............................ 41
15 Overlapping vertical bars ......................................... 43
16 Arrow diagrams for overlapping vertical bars .................. 44
17 The continuous bars test ........................................... 45
18 Arrow diagrams for the continuous bars test .................... 46
19 Hierarchical and non-hierarchical learning in the bars test 47
20 Encoding of a continuous variable .............................. 53
21 Sketch of a macrocolumn for clustering ......................... 54
22 Time course of clustering ......................................... 56
23 Clustering of unequally distributed data .......................... 57
24 Minicolumn activation function ................................... 66
25 Eigenvalues of the abstracted dynamics ........................ 68
26 Phase velocity of the abstract dynamics ........................ 69
27 Phase velocity of a more complex abstract dynamics ........ 71
28 Pattern classification in the abstract system ..................... 75
29 Bars test – abstract system ...................................... 76
30 Hand-written digits classification – hierarchical abstract system 78
31 Continuous bars test – hierarchical abstract system .......... 80
32 Comparison of the abstract hierarchical and non-hierarchical system 81
33 Feature extraction points in images ............................. 85
34 Corresponding points in two images ............................ 86
35 Basic DLM network ............................................... 87
36 Network of two macrocolumns .................................. 91
37 Network for computing the highest jet similarity .............. 93
38 Macrocolumn network for matching ............................. 96
39 Activity in a macrocolumn network for matching .............. 97
40 Macrocolumn network for topological matching ............... 99
41 Topological interconnectivity within a layer .................... 101
42 Layer dynamics ................................................... 102
43 Input and model jets used for experiments ....................... 104
44 Rapid matching in a macrocolumn network ..................... 105
45 Matching with different values of parameter $C_l$ ............... 107
References


References


References


References


References


References


