The final version of this article has been published in
Neural Computation, 10(3):671-716 (1998) published by The MIT Press.
This version does not differ significantly from the final version.

Constrained Optimization for Neural Map Formation: A Unifying Framework for Weight Growth and Normalization

Laurenz Wiskott[1] and Terrence Sejnowski[123]

[1]Computational Neurobiology Laboratory
[2]Howard Hughes Medical Institute
The Salk Institute for Biological Studies
San Diego, CA 92186-5800
http://www.cnl.salk.edu/CNL

[3]Department of Biology
University of California, San Diego
La Jolla, CA 92093

E-mail: wiskott@salk.edu

Abstract

Computational models of neural map formation can be considered on at least three different levels of abstraction: detailed models including neural activity dynamics, weight dynamics that abstract from the neural activity dynamics by an adiabatic approximation, and constrained optimization from which equations governing weight dynamics can be derived. Constrained optimization uses an objective function, from which a weight growth rule can be derived as a gradient flow, and some constraints, from which normalization rules are derived. In this paper we present an example of how an optimization problem can be derived from detailed non-linear neural dynamics. A systematic investigation reveals how different weight dynamics introduced previously can be derived from two types of objective function terms and two types of constraints. This includes dynamic link matching as a special case of neural map formation. We focus in particular on the role of coordinate transformations to derive different weight dynamics from the same optimization problem. Several examples illustrate how the constrained optimization framework can help in understanding, generating, and comparing different models of neural map formation. The techniques used in this analysis may also be useful in investigating other types of neural dynamics.

1 Introduction

Neural maps are an important motif in the structural organization of the brain. The best studied maps are those in the early visual system. For example, the retinotectal map connects a 2-dimensional array of ganglion cells in the retina to a corresponding map of the visual field in the optic tectum of vertebrates in a neighborhood-preserving fashion. These are called topographic maps. The map from the lateral geniculate nucleus (LGN) to the primary visual cortex (V1) is a more complex map because the inputs coming from LGN include signals from both eyes and are unoriented, but most cells in V1 are tuned for orientation, an emergent property. Neurons with preferred orientation and ocular dominance in area V1 form a columnar structure, where neurons responding to the same eye or the same orientation tend to be neighbors. Other neural maps are formed in the somatosensory, the auditory, and the motor systems. All neural maps connect an input layer, possibly divided into different parts (e.g. left and right eye) to an output layer. Each neuron in the output layer can potentially receive input from all neurons in the input layer (here we ignore the limits imposed by restricted axonal arborization and dendritic extension). However, particular receptive fields develop due to a combination of genetically-determined and activity-driven mechanisms for self-organization. Although cortical maps have many feedback projections (for example, from area V1 back to the LGN), these are disregarded in most models of map formation and will not be considered here.

The goal of neural map formation is to self-organize from an initial random all-to-all connectivity a regular pattern of connectivity, as in Figure 1, for the purpose of producing a representation of the input on the output layer that is of further use to the system. The development of the structure depends on the architecture, the lateral connectivity, the initial conditions, and on the weight dynamics, including growth rule and normalization rules.

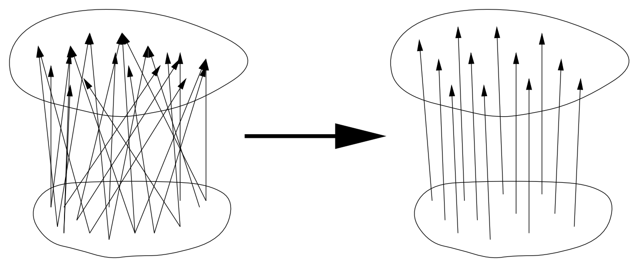

Figure 1: Goal of neural map formation: The initially random all-to-all connectivity self-organizes into an orderly connectivity that appropriately reflects the correlations within the input stimuli and the induced correlations within the output layer. The output correlations also depend on the connectivity within the output layer.

The first model of map formation, introduced by VON DER MALSBURG (1973), was for a small patch of retina stimulated with bars of different orientation. The model self-organized orientation columns, with neighboring neurons having receptive fields tuned to similar orientation. This model already included all the crucial ingredients important for map formation: (1) Characteristic correlations within the stimulus patterns, (2) lateral interactions within the output layer, inducing characteristic correlations there as well, (3) Hebbian weight modification, and (4) competition between synapses by weight normalization. Many similar models have been proposed since then for different types of map formation (see ERWIN ET AL., 1995; SWINDALE, 1996, and Table 2 for examples). We do not consider models that are based on chemical markers (e.g. VON DER MALSBURG & WILLSHAW, 1977). Although they may be conceptionally similar to those based on neural activities, they can differ significantly in the detailed mathematical formulation. Nor do we consider in detail models that treat the input layer as a low-dimensional space, say 2-dimensional for the retina, from which input vectors are drawn, (e.g. KOHONEN, 1982) (but see Sec. 6.8). The output neurons then receive only two synapses per neuron, one for each input dimension.

The dynamic link matching model (e.g. BIENENSTOCK & VON DER MALSBURG, 1987; KONEN ET AL., 1994) is a form of neural map formation that has been developed for pattern recognition. It is mathematically similar to the self-organization of retinotectal projections, but in addition each neuron has a visual feature attached, so that a neural layer can be considered as a labeled graph representing a visual pattern. Each synapse has associated with it an individual value, which affects the dynamics and expresses the similarity between the features of connected neurons. The self-organization process then not only tends to generate a neighborhood preserving map, it also tends to connect neurons having similar features. If the two layers represent similar patterns, the map formation dynamics finds the correct feature correspondences and connects the corresponding neurons.

Models of map formation have been investigated by analysis (e.g. AMARI, 1980; HÄUSSLER & VON DER MALSBURG, 1983) and computer simulations. An important tool for both methods is the objective function (or energy function) from which the dynamics can be generated as a gradient flow. The objective value (or energy) can be used to estimate which weight configurations would be more likely to arise from the dynamics (e.g. MacKay & Miller, 1990). In computer simulations the objective function is maximized (or the energy function is minimized) numerically in order to find stable solutions of the dynamics (e.g. Linsker, 1986; Bienenstock & von der Malsburg, 1987).

Objective functions, which can also serve as a Lyapunov function, have many advantages: First, the existence of an objective function guarantees that the dynamics does not have limit cycles or chaotic attractors as solutions. Second, an objective function often provides more direct and intuitive insight into the behavior of a dynamics and the effects of each term can be understood more easily. Third, an objective function allows additional mathematical tools to be used to analyze the system, such as methods from statistical physics. Finally, an objective function provides connections to more abstract models, such as spin systems, which have been studied in depth.

Although objective functions have been used before in the context of neural map formation, they have not yet been investigated systematically. The goal of this paper is to derive objective functions for a wide variety of models. While growth rules can be derived from objective functions as gradient flows, normalization rules are derived from constraints by various methods. Thus, objective functions and constraints have to be considered in conjunction and form a constrained optimization problem. We show that although two models may differ in the formulation of their dynamics, they may be derived from the same constrained optimization problem, thus providing a unifying framework for the two models. The equivalence between different dynamics is revealed by coordinate transformations. A major focus of this paper is therefore on the effects of coordinate transformations on weight growth rules and normalization rules.

1.1 Model Architecture

The general architecture considered here consists of two layers of neurons, an input and an output layer, as in Figure 2. (We use the term layer for a population of neurons without assuming a particular geometry.) Input neurons are indicated by ρ (retina), and output neurons by τ (tectum); the index ν can indicate a neuron in either layer. Neural activities are indicated by a. Input neurons are connected all-to-all to output neurons, but there are no connections back to the input layer. Thus the dynamics in the input layer is completely independent of the output layer and can be described by mean activities $\langle a_\rho \rangle$ and correlations $\langle a_\rho, a_{\rho'} \rangle$. Effective lateral connections within a layer are denoted by $D_{\rho\rho'}$ and $D_{\tau\tau'}$; connections projecting from the input to the output layer are denoted by $w_{\tau\rho}$. The second index always indicates the presynaptic neuron and the first index the postsynaptic neuron. The lateral connections defined here are called *effective*, because they need not correspond to physical connections. For example, in the input layer the effective lateral connections represent the correlations between input neurons regardless of what induced the correlations, $D_{\rho\rho'} = \langle a_\rho, a_{\rho'} \rangle$. In the example below, the output layer has short-term excitatory and long-term inhibitory connections; the effective lateral connections, however, are only excitatory. The effective lateral connections thus represent functional properties of the lateral interactions and not the anatomical connectivity itself.

To make the notation simpler we use the definitions $i = \{\rho, \tau\}$, $j = \{\rho', \tau'\}$, $A_{ij} = D_{\tau\tau'} A_{\rho'} = D_{\tau\tau'} \langle a_{\rho'} \rangle$, and $D_{ij} = D_{\tau\tau'} D_{\rho\rho'} = D_{\tau\tau'} \langle a_\rho, a_{\rho'} \rangle$ in Section 3 and later. We assume symmetric matrices $A_{ij} = A_{ji}$ and $D_{ij} = D_{ji}$, which requires some homogeneity of the architecture, i.e. $\langle a_\rho \rangle = \langle a_{\rho'} \rangle$, $\langle a_\rho, a_{\rho'} \rangle = \langle a_{\rho'}, a_\rho \rangle$, and $D_{\tau\tau'} = D_{\tau'\tau}$.

In the next section, a simple model is used to demonstrate the basic procedure of deriving a constrained optimization problem from detailed neural dynamics. This procedure has three steps. First, the neural dynamics is transformed into a weight dynamics, where the induced correlations are expressed directly in terms of the synaptic weights, thus eliminating neural activities from the dynamics by an adiabatic approximation. Second, an objective function is constructed, which can generate the dynamics of the growth rule as a gradient flow. Third, the normalization rules need to be considered and, if possible, derived from constraint functions. The last two steps depend on each other insofar as growth rule as well as normalization rules must be inferred under the same coordinate transformation. The three important aspects of this example — deriving correlations, constructing objective functions, and considering the constraints — are then discussed in greater detail in the following three sections, respectively. The reader may skip Section 2 and continue directly with these more abstract considerations beginning in Section 3. In Section 6 several examples are given for how the constrained optimization framework can be used to understand, generate, and compare models of neural map formation.

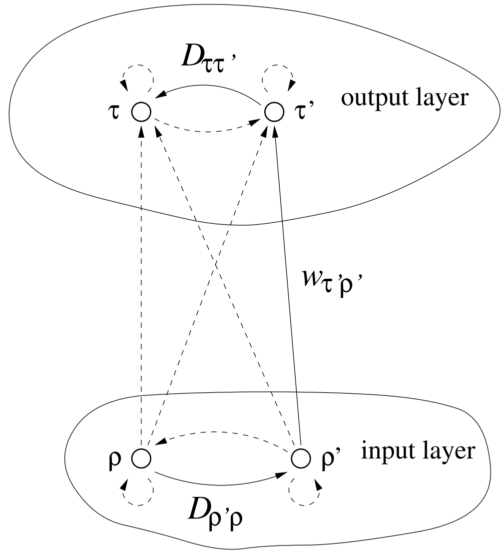

Figure 2: General architecture: Neurons in the input layer are connected all-to-all to neurons in the output layer. Each layer has *effective* lateral connections D representing functional aspects of the lateral connectivity, e.g. characteristic correlations. As an example, a path through which activity can propagate from neuron ρ to neuron τ is shown by solid arrows. Other connections are shown as dashed arrows.

2 Prototypical System

As a concrete example, consider a slightly modified version of the dynamics proposed by Willshaw & von der Malsburg (1976) for the self-organization of a retinotectal map, where the input and output layer correspond to retina and tectum, respectively. The dynamics is qualitatively described by the following set of differential equations:

Neural activity dynamics:

$$\dot{m}_\rho = -m_\rho + (k * a_{\rho'})_\rho \ , \tag{1}$$

$$\dot{m}_\tau = -m_\tau + (k * a_{\tau'})_\tau + \sum_{\rho'} w_{\tau\rho'} a_{\rho'} \ , \tag{2}$$

Weight growth rule:

$$\dot{w}_{\tau\rho} = a_\tau a_\rho \ , \tag{3}$$

Weight normalization rules:

$$\text{if } w_{\tau\rho} < 0: \qquad w_{\tau\rho} = 0 \ , \tag{4}$$

$$\text{if } \sum_{\rho'} w_{\tau\rho'} > 1: \qquad w_{\tau\rho} = \tilde{w}_{\tau\rho} + \frac{1}{M_\tau}\left(1 - \sum_{\rho'} \tilde{w}_{\tau\rho'}\right) \qquad \text{for all } \rho \ , \tag{5}$$

$$\text{if } \sum_{\tau'} w_{\tau'\rho} > 1: \qquad w_{\tau\rho} = \tilde{w}_{\tau\rho} + \frac{1}{M_\rho}\left(1 - \sum_{\tau'} \tilde{w}_{\tau'\rho}\right) \qquad \text{for all } \tau \ , \tag{6}$$

where m denotes the membrane potential, $a_\nu = \sigma(m_\nu)$ is the mean firing rate determined by a non-linear input-output function σ, $(k * a_{\nu'})$ indicates a convolution of the neural activities with the kernel k representing lateral connections with local excitation and global inhibition, $\tilde{w}_{\tau\rho}$ indicates weights as obtained by integrating the differential equations for one time step, i.e. $\tilde{w}_{\tau\rho}(t+\Delta t) = w_{\tau\rho}(t) + \Delta t\, \dot{w}_{\tau\rho}(t)$, M_τ is the number of links terminating on output neuron τ, and M_ρ is the number of links originating from input neuron ρ. Equations (1, 2) govern the neural activity dynamics on the two layers, Equation (3) is the growth rule for the synaptic weights, and Equations (4–6) are the normalization rules that keep the sums over synaptic weights originating from an input neuron or terminating on an output neuron equal to 1 and prevent the weights from becoming negative. Notice that since our discussion is qualitative, we included only the basic terms and discarded some parameters required to make the system work properly. One difference from the original model is that subtractive instead of multiplicative normalization rules are used.

2.1 Correlations

The dynamics within the neural layers is well understood (Amari, 1977; Konen et al., 1994). Local excitation and global inhibition lead to the development of a local patch of activity, called a *blob*. The shape and size of the blob depend on the kernel k and other parameters of the system and can be described by $B_{\rho'\rho_0}$ if centered on input neuron ρ_0 and $B_{\tau'\tau_0}$ if centered on output neuron τ_0. The location of the blob depends on the input, which is assumed to be weak enough that it does not change the shape of the blob. Assume the input layer receives noise such that the blob arises with equal probability $p(\rho_0) = 1/R$ centered on any of the input neurons, where R is the number of input neurons. For simplicity we assume cyclic boundary conditions to avoid boundary effects. The location of the blob in the output layer on the other hand is affected by the input

$$i_{\tau'}(\rho_0) = \sum_{\rho'} w_{\tau'\rho'} B_{\rho'\rho_0} \ , \tag{7}$$

received from the input layer and therefore depends on the position ρ_0 of the blob in the input layer. Only one blob can occur in each layer, and the two layers need to be reset before new blobs can arise. A sequence of blobs is required to induce the appropriate correlations.

Konen et al. (1994) have shown that without noise, blobs in the output layer will arise at location τ_0 with the largest overlap between input $i_{\tau'}(\rho_0)$ and the final blob profile $B_{\tau'\tau_0}$, i.e. the location for which $\sum_{\tau'} B_{\tau'\tau_0} i_{\tau'}(\rho_0)$ is maximal. This winner-take-all behavior makes it difficult to analyze the system. We therefore make the assumption that in contrast to this deterministic dynamics, the blob arises at location τ_0 with a probability equal to the overlap between the input and blob activity,

$$p(\tau_0|\rho_0) = \sum_{\tau'} B_{\tau'\tau_0} i_{\tau'}(\rho_0) = \sum_{\tau'\rho'} B_{\tau'\tau_0} w_{\tau'\rho'} B_{\rho'\rho_0} \ . \tag{8}$$

Assume the blobs are normalized such that $\sum_{\rho'} B_{\rho'\rho_0} = 1$ and $\sum_{\tau_0} B_{\tau'\tau_0} = 1$ and that the connectivity is normalized such that $\sum_{\tau'} w_{\tau'\rho'} = 1$, which is the case for the system above if the input layer does not have more neurons than the output layer. This implies $\sum_{\tau'} i_{\tau'}(\rho_0) = 1$ and $\sum_{\tau_0} p(\tau_0|\rho_0) = 1$ and justifies the interpretation of $p(\tau_0|\rho_0)$ as a probability.

Although it is plausible that such a probabilistic blob location could be approximated by noise in the output layer, it is difficult to develop a concrete model. For a similar but more algorithmic activity model (Obermayer et al., 1990) an exact noise model for the probabilistic blob location can be formulated (see Appendix A). With Equation (8) the probability for a particular combination of blob locations is

$$p(\tau_0, \rho_0) = p(\tau_0|\rho_0) p(\rho_0) = \sum_{\tau'\rho'} B_{\tau'\tau_0} w_{\tau'\rho'} B_{\rho'\rho_0} \frac{1}{R} \ , \tag{9}$$

and the correlation between two neurons defined as the average product of their activities is

$$\begin{aligned}
\langle a_\tau a_\rho \rangle &= \sum_{\tau_0\rho_0} p(\tau_0, \rho_0) B_{\tau\tau_0} B_{\rho\rho_0} && (10)\\
&= \sum_{\tau_0\rho_0} \sum_{\tau'\rho'} B_{\tau'\tau_0} w_{\tau'\rho'} B_{\rho'\rho_0} \frac{1}{R} B_{\tau\tau_0} B_{\rho\rho_0} && (11)\\
&= \frac{1}{R} \sum_{\tau'\rho'} \left(\sum_{\tau_0} B_{\tau'\tau_0} B_{\tau\tau_0} \right) w_{\tau'\rho'} \left(\sum_{\rho_0} B_{\rho'\rho_0} B_{\rho\rho_0} \right) && (12)\\
&= \frac{1}{R} \sum_{\tau'\rho'} \bar{B}_{\tau\tau'} w_{\tau'\rho'} \bar{B}_{\rho'\rho} \ , \qquad \text{with } \bar{B}_{\nu'\nu} = \sum_{\nu_0} B_{\nu'\nu_0} B_{\nu\nu_0}, && (13)
\end{aligned}$$

where the brackets $\langle\cdot\rangle$ indicate the ensemble average over a large number of blob presentations. $\frac{1}{R}\bar{B}_{\rho'\rho}$ and $\bar{B}_{\tau\tau'}$ are the effective lateral connectivities of the input and the output layer, respectively, and are symmetrical even if the individual blobs $B_{\rho\rho_0}$ and $B_{\tau\tau_0}$ are not, i.e. $D_{\rho'\rho} = \frac{1}{R}\bar{B}_{\rho'\rho}$, $D_{\tau\tau'} = \bar{B}_{\tau\tau'}$, and $D_{ij} = D_{ji} = D_{\tau\tau'}D_{\rho'\rho} = \frac{1}{R}\bar{B}_{\tau\tau'}\bar{B}_{\rho'\rho}$. Notice the linear relation between the weights $w_{\tau'\rho'}$ and the correlations $\langle a_\tau a_\rho\rangle$ in the probabilistic blob model (Eq. 13).

Substituting the correlation into Equation (3) for the weight dynamics leads to:

$$\langle \dot{w}_{\tau\rho}\rangle = \langle a_\tau a_\rho\rangle = \frac{1}{R}\sum_{\tau'\rho'} \bar{B}_{\tau\tau'} w_{\tau'\rho'} \bar{B}_{\rho'\rho} \ . \tag{14}$$

The same normalization rules (Eqs. 4–6) given above apply to this dynamics. Since there is little danger of confusion, we neglect the averaging brackets for $\langle \dot{w}_{\tau\rho}\rangle$ in subsequent equations and simply write $\dot{w}_{\tau\rho} = \langle a_\tau a_\rho\rangle$.

Although we did not give a mathematical model of the mechanism by which the probabilistic blob location as given in Equation (8) could be implemented, it may be interesting to note that the probabilistic approach can be generalized to other activity patterns, such as stripe patterns or hexagons, which can be generated by Mexican hat interaction functions (local excitation, finite-range inhibition) (von der Malsburg, 1973; Ermentrout & Cowan, 1979). If the probability for a stripe pattern arising in the output layer is linear in its overlap with the input, the same derivation follows, though the indices ρ_0 and τ_0 will then refer to phase and orientation of the patterns rather than location of the blobs.

Using the probabilistic blob location in the output layer instead of the deterministic one is analogous to the soft competitive learning proposed by Nowlan (1990) as an alternative to hard (or winner-take-all) competitive learning. Nowlan demonstrated superior performance of soft competition over hard competition for a radial basis function network tested on recognition of handwritten characters and spoken vowels, and suggested there might be a similar advantage for neural map formation. The probabilistic blob location induced by noise might help improve neural map formation by avoiding local optima.

2.2 Objective Function

The next step is to find an objective function that generates the dynamics as a gradient flow. For the above example, a suitable objective function is

$$H(\mathbf{w}) = \frac{1}{2R}\sum_{\tau\rho\tau'\rho'} w_{\tau\rho}\bar{B}_{\rho\rho'}\bar{B}_{\tau\tau'}w_{\tau'\rho'} \ , \tag{15}$$

since it yields Equation (14) from $\dot{w}_{\tau\rho} = \frac{\partial H(\mathbf{w})}{\partial w_{\tau\rho}}$ taking into account that $\bar{B}_{\nu\nu'} = \bar{B}_{\nu'\nu}$.

2.3 Constraints

The normalization rules given above ensure that synaptic weights do not become negative and that the sums over synaptic weights originating from an input neuron or terminating on an output neuron do not become larger than 1. This can be written in form of inequalities for constraint functions g:

$$g_{\tau\rho}(\mathbf{w}) = w_{\tau\rho} \geq 0 \ , \tag{16}$$

$$g_{\tau}(\mathbf{w}) = 1 - \sum_{\rho'} w_{\tau\rho'} \geq 0 \ , \tag{17}$$

$$g_{\rho}(\mathbf{w}) = 1 - \sum_{\tau'} w_{\tau'\rho} \geq 0 \ . \tag{18}$$

These constraints define a region within which the objective function is to be maximized by steepest ascent. While the constraints follow uniquely from the normalization rules, the converse is not true. In general there are various normalization rules that would enforce or at least approximate the constraints but only some of them are compatible with the constrained optimization framework. As shown in Section 5.2.1 compatible normalization rules can be obtained by the method of Lagrangian multipliers. If a constraint $g_x, x \in \{\tau\rho, \tau, \rho\}$ is violated, a normalization rule of the form

$$\text{if } g_x(\tilde{\mathbf{w}}) < 0: \qquad w_{\tau\rho} = \tilde{w}_{\tau\rho} + \lambda_x \frac{\partial g_x}{\partial \tilde{w}_{\tau\rho}} \qquad \text{for all } \tau\rho \ , \tag{19}$$

has to be applied, where λ_x is a Lagrangian multiplier and determined such that $g_x(\mathbf{w}) = 0$. This method actually leads to Equations (4–6), which are therefore a compatible set of normalization rules for the constraints above. This is necessary to make the formulation as a constrained optimization problem (Eqs. 15–18) an appropriate description of the original dynamics (Eqs. 3–6).

This example illustrates the general scheme by which a detailed model dynamics for neural map formation can be transformed into a constrained optimization problem. The correlations, objective functions, and constraints are discussed in greater detail and for a wide variety of models below.

3 Correlations

In the above example, correlations in a highly non-linear dynamics led to a linear relationship between synaptic weights and the induced correlations. We derived effective lateral connections in the input as well as the output layer mediating these correlations. Corresponding equations for the correlations have been derived for other, mostly linear activity models (e.g. Linsker, 1986; Miller, 1990; von der Malsburg, 1995), as summarized here.

Assume the dynamics in the input layer is described by neural activities $a_\rho(t) \in \mathbb{R}$, which yield mean activities $\langle a_\rho \rangle$ and correlations $\langle a_\rho, a_{\rho'} \rangle$. The input received by the output layer is assumed to be a linear superposition of the activities of the input neurons:

$$i_{\tau'} = \sum_{\rho'} w_{\tau'\rho'} a_{\rho'} \ . \tag{20}$$

This input then produces activity in the output layer through effective lateral connections in a linear fashion:

$$a_\tau = \sum_{\tau'} D_{\tau\tau'} i_{\tau'} = \sum_{\tau'\rho'} D_{\tau\tau'} w_{\tau'\rho'} a_{\rho'} \ . \tag{21}$$

As seen in the above example, this linear behavior could be generated by a non-linear model. Thus the neurons need not be linear, only the effective behavior of the correlations (cf. Sejnowski, 1976; Ginzburg & Sompolinsky, 1994). The mean activity of output neurons is:

$$\langle a_\tau \rangle = \sum_{\tau'\rho'} D_{\tau\tau'} w_{\tau'\rho'} \langle a_{\rho'} \rangle = \sum_j A_{ij} w_j \ . \tag{22}$$

Assuming a linear correlation function ($\langle a_\rho, \alpha(a_{\rho'} + a_{\rho''})\rangle = \alpha\langle a_\rho, a_{\rho'}\rangle + \alpha\langle a_\rho, a_{\rho''}\rangle$ with a real constant α) such as the average product or the covariance (Sejnowski, 1977), the correlation between input and output neurons is

$$\langle a_\tau, a_\rho \rangle = \sum_{\tau'\rho'} D_{\tau\tau'} w_{\tau'\rho'} \langle a_{\rho'}, a_\rho \rangle = \sum_j D_{ij} w_j \ , \tag{23}$$

Note that $i = \{\rho, \tau\}$, $j = \{\rho', \tau'\}$, $A_{ij} = A_{ji} = D_{\tau\tau'} A_{\rho'} = D_{\tau\tau'}\langle a_{\rho'} \rangle$, and $D_{ij} = D_{ji} = D_{\tau\tau'} D_{\rho'\rho} = D_{\tau\tau'}\langle a_{\rho'}, a_\rho \rangle$. Since the right hand sides of Equations (22) and (23) are formally equivalent, we will consider only the latter one in the further analysis, bearing in mind that Equations (22) is included as a special case.

In this linear correlation model all variables may assume negative values. This may not be plausible for the neural activities a_ρ and a_τ. However, Equation (23) can be derived also for non-negative activities and a similar equation as Equation (22) can be derived if the mean activities $\langle a_\rho \rangle$ are positive. The difference for the latter would be an additional constant, which can always be compensated for in the growth rule.

The correlation model in (Linsker, 1986) differs from the linear one introduced here in two respects. The input (Eq. 20) has an additional constant term and correlations are defined by subtracting positive constants from the activities. However, it can be shown that correlations in the model in (Linsker, 1986) are a linear combination of a constant and the terms of Equations (22, 23).

4 Objective Functions

In general, there is no systematic way of finding an objective function for a particular dynamical system, but it is possible to determine whether there exists an objective function. The necessary and sufficient condition is that the flow field of the dynamics be curl free. If there exists an objective function $H(\mathbf{w})$ with continuous partial derivatives of order two that generates the dynamics $\dot{w}_i = \partial H(\mathbf{w})/\partial w_i$, then

$$\frac{\partial \dot{w}_i}{\partial w_j} = \frac{\partial^2 H(\mathbf{w})}{\partial w_j \partial w_i} = \frac{\partial^2 H(\mathbf{w})}{\partial w_i \partial w_j} = \frac{\partial \dot{w}_j}{\partial w_i} \ . \tag{24}$$

The existence of an objective function is thus equivalent to $\partial \dot{w}_i/\partial w_j = \partial \dot{w}_j/\partial w_i$, which can be checked easily. For the dynamics given by

$$\dot{w}_i = \sum_j D_{ij} w_j \tag{25}$$

(cf. Eq. 14), for example, $\partial \dot{w}_i/\partial w_j = D_{ij} = \partial \dot{w}_j/\partial w_i$, which shows that it can be generated as a gradient flow. A suitable objective function is

$$H(\mathbf{w}) = \frac{1}{2} \sum_{ij} w_i D_{ij} w_j \tag{26}$$

(cf. Eq. 15), since it yields $\dot{w}_i = \partial H(\mathbf{w})/\partial w_i$.

A dynamics that cannot be generated by an objective function directly is

$$\dot{w}_i = w_i \sum_j D_{ij} w_j \ , \tag{27}$$

as used in (Häussler & von der Malsburg, 1983), since for $i \neq j$ we obtain $\partial \dot{w}_i/\partial w_j = w_i D_{ij} \neq w_j D_{ji} = \partial \dot{w}_j/\partial w_i$, and $\dot{w}_i$ is not curl-free. However, it is sometimes possible to convert a dynamics with curl into a curl-free dynamics by a coordinate transformation. Applying the transformation $w_i = \frac{1}{4} v_i^2$ ($\mathcal{C}^w$) to Equation (27) yields

$$\dot{v}_i = \frac{\mathrm{d}v_i}{\mathrm{d}w_i} \dot{w}_i = \sqrt{w_i} \sum_j D_{ij} w_j = \frac{1}{2} v_i \sum_j D_{ij} \frac{1}{4} v_j^2 \ , \tag{28}$$

which is curl free, since $\partial \dot{v}_i/\partial v_j = \frac{1}{2} v_i D_{ij} \frac{1}{2} v_j = \partial \dot{v}_j/\partial v_i$. Thus, the dynamics of $\dot{v}_i$ in the new coordinate system $\mathcal{V}^w$ can be generated as a gradient flow. A suitable objective function is

$$H(\mathbf{v}) = \frac{1}{2} \sum_{ij} \frac{1}{4} v_i^2 D_{ij} \frac{1}{4} v_j^2 \ , \tag{29}$$

since it yields $\dot{v}_i = \partial H(\mathbf{v})/\partial v_i$. Transforming the dynamics of $\mathbf{v}$ back into the original coordinate system $\mathcal{W}$, of course, yields the original dynamics in Equation (27):

$$\dot{w}_i = \frac{\mathrm{d}w_i}{\mathrm{d}v_i} \dot{v}_i = \frac{1}{4} v_i^2 \sum_j D_{ij} \frac{1}{4} v_j^2 = w_i \sum_j D_{ij} w_j \ . \tag{30}$$

Coordinate transformations thus can provide objective functions for dynamics that are not curl-free. Notice that $H(\mathbf{v})$ is the same objective function as $H(\mathbf{w})$ (Eq. 26) evaluated in $\mathcal{V}^w$ instead of $\mathcal{W}$. Thus $H(\mathbf{v}) = H(\mathbf{w}(\mathbf{v}))$ and H is a Lyapunov function for both dynamics.

More generally, for an objective function H and a coordinate transformation $w_i = w_i(v_i)$

$$\dot{w}_i = \frac{\mathrm{d}}{\mathrm{d}t} [w_i(v_i)] = \frac{\mathrm{d}w_i}{\mathrm{d}v_i} \dot{v}_i = \frac{\mathrm{d}w_i}{\mathrm{d}v_i} \frac{\partial H}{\partial v_i} = \left(\frac{\mathrm{d}w_i}{\mathrm{d}v_i} \right)^2 \frac{\partial H}{\partial w_i} \ , \tag{31}$$

which implies that the coordinate transformation simply adds a factor $(\mathrm{d}w_i/\mathrm{d}v_i)^2$ to the original growth term obtained in the original coordinate system $\mathcal{W}$. For the dynamics in Equation (27) derived under the coordinate transformation $w_i = \frac{1}{4} v_i^2$ ($\mathcal{C}^w$) relative to the dynamics of Equation (25) we verify that $(\mathrm{d}w_i/\mathrm{d}v_i)^2 = w_i$.

Equation (31) also shows that fixed points are preserved under the coordinate transformation in the region where $\mathrm{d}w_i/\mathrm{d}v_i$ is defined and finite but that additional fixed points may be introduced if $\mathrm{d}w_i/\mathrm{d}v_i = 0$.

This effect of coordinate transformations is known from the general theory of relativity and tensor analysis (e.g. Dirac, 1996). The gradient of a potential (or objective function) is a covariant vector, which adds the factor $\mathrm{d}w_i/\mathrm{d}v_i$ through the transformation from $\mathcal{W}$ to $\mathcal{V}$. Since $\dot{\mathbf{v}}$ as a kinematic description of the trajectory is a contravariant vector, this adds another factor $\mathrm{d}w_i/\mathrm{d}v_i$ through the transformation back from $\mathcal{V}$ to $\mathcal{W}$. If both vectors were either covariant or contravariant the back and forth transformation between the different coordinate systems would have no effect. The same argument holds for the constraints in Section 5.2. In some cases it may also be useful to consider more general coordinate transformations $w_i = w_i(\mathbf{v})$ where each weight w_i may depend on all variables v_j, as is common in the general theory of relativity and tensor analysis. Equation (31) would have to be modified correspondingly. In Figure 3, the effect of coordinate transformations is illustrated by a simple example.

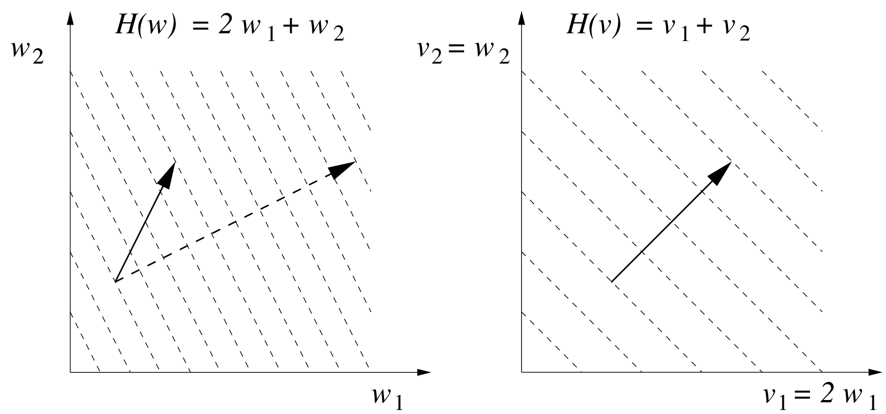

Figure 3: The effect of coordinate transformations on the induced dynamics: The figure shows a simple objective function H in the original coordinate system $\mathcal{W}$ (left) and in the new coordinate system $\mathcal{V}$ (right) with $w_1 = v_1/2$ and $w_2 = v_2$. The gradient induced in $\mathcal{W}$ (dashed arrow) and the gradient induced in $\mathcal{V}$ and then back-transformed into $\mathcal{W}$ (solid arrows) have the same component in the w_2-direction but differ by a factor of four in the w_1-direction (cf. Eq. 31). Notice that the two dynamics differ in amplitude and direction, but that H is a Lyapunov function for both.

Table 1 shows two objective functions and the corresponding induced dynamics terms they induce under different coordinate transformations. The first objective function, L, is linear in the weights and induces constant weight growth (or decay) under coordinate transformation $\mathcal{C}^1$. The growth of one weight does not depend on other weights. This term can be useful for dynamic link matching to introduce a bias for each weight depending on the similarity of the connected neurons. The second objective function, Q, is a quadratic form. The induced growth rule for one weight includes other weights and is usually based on correlations between input and output neurons, $\langle a_\tau a_\rho \rangle = \sum_j D_{ij} w_j$, and possibly also the mean activities of output neurons, $\langle a_\tau \rangle = \sum_j A_{ij} w_j$. This term is, for instance, important to form topographic maps. Functional aspects of term Q are discussed in Section 6.3.

5 Constraints

A constraint is either an inequality describing a surface (of dimensionality $RT - 1$ if RT is the number of weights) between valid and invalid region or an equality describing the valid region as a surface. A normalization rule is a particular prescription for how the constraint has to be enforced. Thus constraints can be uniquely derived from normalization rules but not vice versa.

5.1 Orthogonal Versus Non-orthogonal Normalization Rules

Normalization rules can be divided into two classes, those which enforce the constraints orthogonal to the constraint surface, i.e. along the gradient of the constraint function, and those which also have a component tangential to the constraint surface (see Fig. 4). We refer to the former ones as *orthogonal* and to the latter ones as *non-orthogonal*.

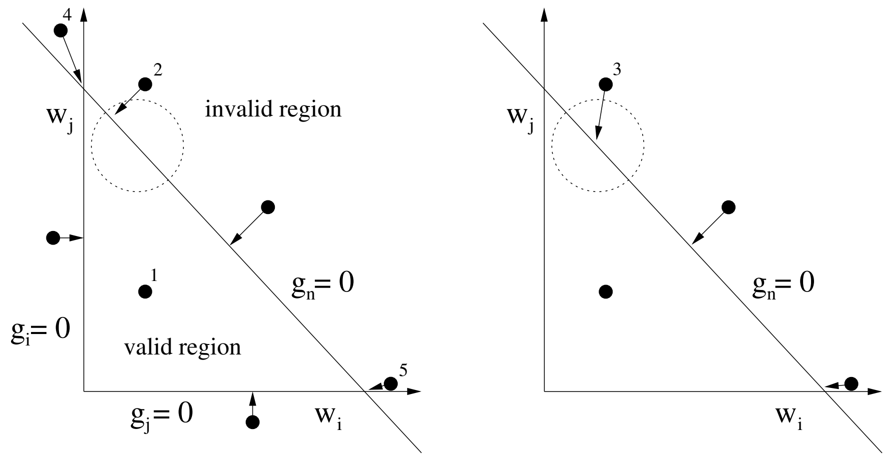

Figure 4: Different constraints and different ways in which constraints can be violated and enforced: The constraints along the axes are given by $g_i = w_i \geq 0$ and $g_j = w_j \geq 0$, which keep the weights w_i and w_j non-negative. The constraint $g_n = 1 - (w_i + w_j) \geq 0$ keeps the sum of the two weights smaller or equal to 1. Black dots indicate points in state-space that may have been reached by the growth rule. Dot 1: None of the constraints is violated and no normalization rule is applied. Dot 2: $g_n \geq 0$ is violated and an orthogonal subtractive normalization rule is applied. Dot 3: $g_n \geq 0$ is violated and a non-orthogonal multiplicative normalization rule is applied. Notice that the normalization does not follow the gradient of g_n, i.e. it is not perpendicular to the line $g_n = 0$. Dot 4: Two constraints are violated and the respective normalization rules must be applied simultaneously. Dot 5: $g_n \geq 0$ is violated, but the respective normalization rule violates $g_j \geq 0$. Again both rules must be applied simultaneously. The dotted circles indicate regions considered in greater detail in Figure 5.

Only the orthogonal normalization rules are compatible with an objective function, as is illustrated in Figure 5. For a dynamics induced as an ascending gradient flow of an objective function, the value of the objective function constantly increases as long as the weights change. If the weights cross a constraint surface, a normalization rule has to be applied iteratively to the growth rule. Starting from the constraint surface at point $\mathbf{w}'$, the gradient ascent causes a step to point $\tilde{\mathbf{w}}$ in the invalid region, where $\tilde{\mathbf{w}} - \mathbf{w}'$ is in general non-orthogonal to the constraint surface. A normalization rule causes a step back to $\mathbf{w}$ on the constraint surface. If the normalization rule is orthogonal, i.e. $\mathbf{w} - \tilde{\mathbf{w}}$ is orthogonal to the constraint surface, $\mathbf{w} - \tilde{\mathbf{w}}$ is shorter or equal $\tilde{\mathbf{w}} - \mathbf{w}'$ and the cosine of the angle between the combined step $\mathbf{w} - \mathbf{w}'$ and the gradient $\tilde{\mathbf{w}} - \mathbf{w}'$ is non-negative, i.e. the value of the objective function does not decrease. This cannot be guaranteed for non-orthogonal normalization rules, in which case the objective function of the unconstrained dynamics may not even be a Lyapunov function for the combined system, including weight dynamics and normalization rules. Thus, only orthogonal normalization rules can be used in the constrained optimization framework.

The term orthogonal is not well defined away from the constraint surface. However, the constraints used in this paper are rather simple and a natural orthogonal direction is usually available for all weight vectors. Thus the term orthogonal will also be used for normalization rules that do not project back exactly onto the constraint surface but which keep the weights close to the surface and affect the weights orthogonal to it. For more complicated constraint surfaces, more careful considerations may be required.

Whether a normalization rule is orthogonal or not depends on the coordinate system in which it is applied. This is illustrated in Figure 6 and discussed in greater detail below. The same rule can be non-orthogonal in one coordinate system but orthogonal in another. It is important to find the coordinate

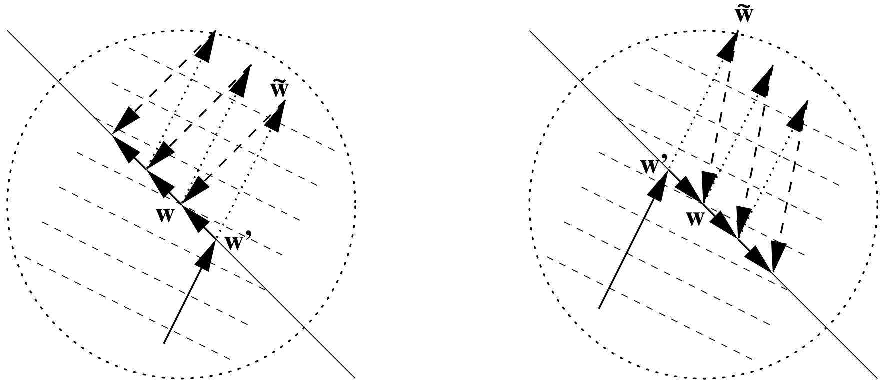

Figure 5: The effect of orthogonal versus non-orthogonal normalization rules: The two circled regions are taken from Figure 4. The effect of the orthogonal subtractive rule is shown on the left and the non-orthogonal multiplicative rule on the right. The growth dynamics is assumed to be induced by an objective function, the equipotential curves of which are shown as dashed lines. The objective function increases to the upper right. The growth rule (dotted arrows) and normalization rule (dashed arrows) are applied iteratively. The net effect is different in the two cases. For the orthogonal normalization rule the dynamics increases the value of the objective function, while for the non-orthogonal normalization the value decreases and the objective function that generates the growth rule is not even a Lyapunov function for the combined system.

system in which an objective function can be derived and the normalization rules are orthogonal. This then is the coordinate system in which the model can be most conveniently analyzed. Not all non-orthogonal normalization rules can be transformed into orthogonal ones. In (Wiskott & von der Malsburg, 1996), for example, a normalization rule is used that affects a group of weights if single weights grow beyond their limits. Since the constraint surface depends only on one weight, only that weight can be effected by an orthogonal normalization rule. Thus this normalization rule cannot be made orthogonal.

5.2 Constraints Can be Enforced in Different Ways

For a given constraint, orthogonal normalization rules can be derived using various methods. These include the method of Lagrangian multipliers, the inclusion of penalty terms, and normalization rules that are integrated into the weight dynamics without necessarily having any objective function. The former two methods are common in optimization theory. The latter is more specific to a model of neural map formation. It is also possible to substitute a constraint by a coordinate transformation.

5.2.1 Method of Lagrangian Multipliers

Lagrangian multipliers can be used to derive explicit normalization rules, such as Equations (4–6). If the constraint $g_n(\mathbf{w}) \geq 0$ is violated for $\tilde{\mathbf{w}}$ as obtained after one integration step of the learning rule, $\tilde{w}_i(t+\Delta t) = w_i(t) + \Delta t\, \dot{w}_i(t)$, the weight vector has to be corrected along the gradient of the constraint function g_n, which is orthogonal to the constraint surface $g_n(\mathbf{w}) = 0$,

$$\text{if } g_n(\tilde{\mathbf{w}}) < 0: \qquad w_i = \tilde{w}_i + \lambda_n \frac{\partial g_n}{\partial \tilde{w}_i} \qquad \text{for all } i\ , \tag{32}$$

where $(\partial g_n / \partial \tilde{w}_i) = (\partial g_n / \partial w_i)$ at $\mathbf{w} = \tilde{\mathbf{w}}$ and $\lambda_n = \lambda_n(\tilde{\mathbf{w}})$ is a Lagrangian multiplier and determined such that $g_n(\mathbf{w}) = 0$ is obtained. If no constraint is violated, the weights are simply taken to be $w_i = \tilde{w}_i$. The constraints that must be taken into account, either because they are violated or because they become violated if a violated one is enforced, are called *operative*. All others are called *inoperative* and do not need to be considered for that integration step. If there are more than one operative constraint, the normalization rule becomes

$$\text{if } g_n(\tilde{\mathbf{w}}) < 0: \qquad w_i = \tilde{w}_i + \sum_{n \in N_O} \lambda_n \frac{\partial g_n}{\partial \tilde{w}_i} \qquad \text{for all } i\ , \tag{33}$$

where N_O denotes the set of operative constraints. The Lagrangian multipliers λ_n are determined such that $g_{n'}(\mathbf{w}) = 0$ for all $n' \in N_O$ (cf. Fig. 4). Computational models of neural map formation usually take another strategy and simply iterate the normalization rules (Eq. 32) for the operative constraints individually, which is in general not accurate but may be sufficient for most practical purposes. It should also be mentioned that in the standard method of Lagrangian multipliers as usually applied in physics or optimization theory the two steps, i.e. weight growth and normalization, are combined in one dynamical equation such that $\mathbf{w}$ remains on the constraint surface. The steps were split here to obtain explicit normalization rules independent of growth rules.

Consider now the effect of coordinate transformations on the normalization rules derived by the method of Lagrangian multipliers. The constraint in Equation (17) can be written as $g_n(\mathbf{w}) = \theta_n - \sum_{i \in I_n} w_i \geq 0$ and leads to a subtractive normalization rule as in the example above (Eq. 5). Under the coordinate transformation $\mathcal{C}^w$ ($w_i = \frac{1}{4}v_i^2$), the constraint becomes $g_n(\mathbf{v}) = \theta_n - \sum_{i \in I_n} \frac{1}{4}v_i^2 \geq 0$ and in the coordinate system $\mathcal{V}^w$ the normalization rule is:

$$\text{if } g_n(\tilde{\mathbf{v}}) < 0: \qquad v_i = \tilde{v}_i - 2\left(\frac{\sqrt{\theta_n}}{\sqrt{\sum_{j \in I_n} \frac{1}{4}\tilde{v}_j^2}} - 1\right)\left(-\frac{1}{2}\tilde{v}_i\right) \tag{34}$$

$$= \frac{\sqrt{\theta_n}\ \tilde{v}_i}{\sqrt{\sum_{j \in I_n} \frac{1}{4}\tilde{v}_j^2}} \qquad \text{for all } i \in I_n\ . \tag{35}$$

Taking the square on both sides and applying the back-transformation from $\mathcal{V}^w$ to $\mathcal{W}$ leads to

$$\text{if } g_n(\tilde{\mathbf{w}}) < 0: \qquad w_i = \frac{\theta_n \tilde{w}_i}{\sum_{j \in I_n} \tilde{w}_j} \qquad \text{for all } i \in I_n\ . \tag{36}$$

This is a multiplicative normalization rule in contrast to the subtractive one obtained in the coordinate system $\mathcal{W}$ (see also Fig. 6). It is listed as Normalization Rule $\mathrm{N}^w_{\geq}$ in Table 1 (or $\mathrm{N}^w_{=}$ for constraint $g(\mathbf{w}) = 0$). This multiplicative rule is commonly found in the literature (cf. Table 2), but it is not orthogonal in $\mathcal{W}$, though it is in $\mathcal{V}^w$.

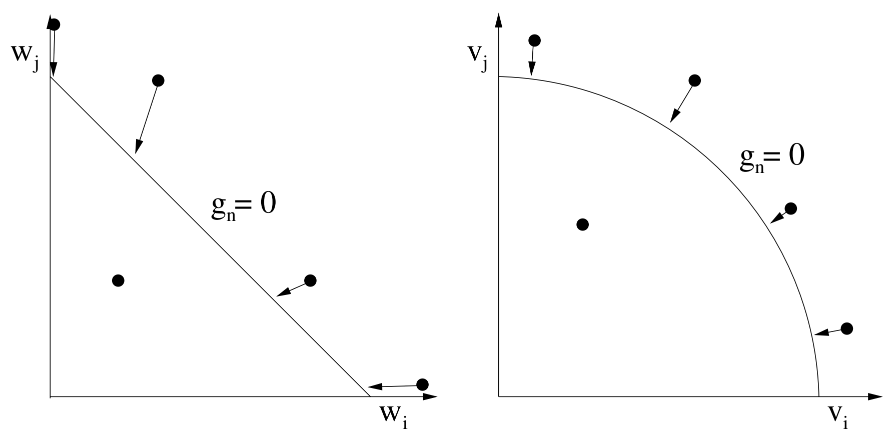

Figure 6: The effect of a coordinate transformation on a normalization rule: The constraint function is $g_n = 1 - (w_i + w_j) \geq 0$ and the coordinate transformation is $w_i = \frac{1}{4}v_i^2, w_j = \frac{1}{4}v_j^2$. In the new coordinate system $\mathcal{V}^w$ (right) the constraint becomes $g_n = 1 - \frac{1}{4}(v_i^2 + v_j^2) \geq 0$ and leads there to an orthogonal multiplicative normalization rule. Transforming back into $\mathcal{W}$ (left) then yields a non-orthogonal multiplicative normalization rule.

For a more general coordinate transformation $w_i = w_i(v_i)$ and a constraint function $g(\mathbf{w})$ an orthogonal normalization rule can be derived in $\mathcal{V}$ with the method of Lagrangian multipliers and transformed back into $\mathcal{W}$, which results in general in a non-orthogonal normalization rule:

$$\text{if constraint is violated}: \qquad w_i = \tilde{w}_i + \lambda\left(\frac{\mathrm{d}w_i}{\mathrm{d}\tilde{v}_i}\right)^2 \frac{\partial g}{\partial \tilde{w}_i} + O(\lambda^2)\ . \tag{37}$$

The λ actually would have to be calculated in $\mathcal{V}$, but since $\lambda \propto \Delta t$, second and higher order terms can be neglected for small Δt and λ calculated such that $g(\mathbf{w}) = 0$. Notice the similar effect of the coordinate transformation on the growth rules (Eq. 31) as well as on the normalization rules (Eq. 37). In both cases a factor $(\mathrm{d}w_i/\mathrm{d}v_i)^2$ is added to the modification rate. As for gradient flows derived from objective functions, for a more general coordinate transformation $w_i = w_i(\mathbf{v})$, Equation (37) would have to be modified accordingly.

We indicate these normalization rules by a subscript "$=$" (for an equality) or "$\geq$" (for an inequality), because the constraints are enforced immediately and exactly.

5.2.2 Integrated Normalization Without Objective Function

Growth rule and explicit normalization rule as derived by the method of Lagrangian multipliers can be combined in one dynamical equation. As an example consider the growth rule $\dot{w}_i = f_i$, i.e. $\tilde{w}_i(t + \Delta t) = w_i(t) + \Delta t f_i(t)$, where f_i is an arbitrary function in $\mathbf{w}$ and can be interpreted as a *fitness* of a synapse. Together with the normalization rule $\mathrm{N}^w_=$ (Eq. 36) and assuming $\sum_{j\in I} w_j(t) = \theta$ it follows that (von der Malsburg & Willshaw, 1981):

$$
\begin{aligned}
w_i(t+\Delta t) &= \frac{\theta\left[w_i(t) + \Delta t f_i(t)\right]}{\sum_{j\in I}\left[w_j(t) + \Delta t f_j(t)\right]} && (38)\\
&= w_i(t) + \Delta t f_i(t) - \Delta t \frac{w_i(t)}{\theta} \sum_{j\in I} f_j(t) + O(\Delta t^2) && (39)\\
\Longrightarrow \qquad \dot{w}_i(t) &= f_i(t) - \frac{w_i(t)}{\theta} \sum_{j\in I} f_j(t)\ , && (40)
\end{aligned}
$$

and with $W(t) = \sum_{i\in I} w_i(t)$

$$
\dot{W}(t) = \left(1 - \frac{W(t)}{\theta}\right) \sum_{j\in I} f_j(t)\ , \tag{41}
$$

which shows that $W = \theta$ is indeed a stable fixed point under the dynamics of Equation (40). However, this is not always the case. The same growth rule combined with the subtractive normalization rule $\mathrm{N}^1_=$ (Eq. 5) would yield a dynamics that only provides a neutrally stable fixed point for $W = \theta$. An additional term $(\theta - \sum_{j\in I} w_j(t))$ would have to be added to make the fixed point stable. This is the reason why this type of normalization rule is listed in Table 1 only for $\mathcal{C}^w$. We indicate these kinds of normalization rules by the subscript "$\simeq$" because the dynamics smoothly approaches the constraint surface and will stay there exactly.

Notice that this method differs from the standard method of Lagrangian multipliers, which also yields a dynamics such that $\mathbf{w}$ remains on the constraint surface. The latter only applies to the dynamics at $g(\mathbf{w}) = 0$ and always produces neutrally stable fixed points because $\sum_i \dot{w}_i(t) \frac{\partial g}{\partial w_i} = 0$ is required by definition. If applied to a weight vector outside the constraint surface, the standard method of Lagrangian multipliers yields $g(\mathbf{w}) = \mathrm{const} \neq 0$.

An advantage of this method is that it provides one dynamics for the growth rule as well as the normalization rule and that the constraint is enforced exactly. However, difficulties arise when interfering constraints are combined, i.e. different constraints that affect the same weights. This type of formulation is required for certain types of analyses (e.g. Häussler & von der Malsburg, 1983). A disadvantage is that in general there no longer exists an objective function for the dynamics, though the growth term itself without the normalization term still has an objective function that is a Lyapunov function for the combined dynamics.

5.2.3 Penalty Terms

Another method of enforcing the constraints is to add penalty terms to the objective function (e.g. Bienenstock & von der Malsburg, 1987). For instance, if the constraint is formulated as an equality $g(\mathbf{w}) = 0$, then add $-\frac{1}{2}g^2(\mathbf{w})$; if the constraint is formulated as an inequality $g(\mathbf{w}) \leq 0$ or $g(\mathbf{w}) \geq 0$, then add $\ln|g(\mathbf{w})|$. Other penalty functions, such as g^4 and $1/g$, are possible as well but those used here induce the required terms as used in the literature.

The effect of coordinate transformations is the same as in the case of objective functions. Consider, for example, the simple constraint $g_i(\mathbf{w}) = w_i \geq 0$ ($\mathrm{I}_\geq$ in Table 1), which keeps weights w_i non-negative. The respective penalty term is $\ln|w_i|$ ($\mathrm{I}_>$) and the induced dynamics under the four different transformations considered in Table 1 are $\frac{1}{w_i}$, $\frac{\alpha_i}{w_i}$, 1, and α_i.

An advantage of this approach is that a coherent objective function as well as a weight dynamics is available including growth rules and normalization rules. A disadvantage may be that the constraints are only approximate and not enforced strictly, so that $g(\mathbf{w}) \approx 0$ and $g(\mathbf{w}) < 0$ or $g(\mathbf{w}) > 0$. We therefore indicate these kinds of normalization rules by subscripts "$\approx$" and "$>$". However, the approximation can be made arbitrarily precise by weighting the penalty terms accordingly.

5.2.4 Constraints Introduced by Coordinate Transformations

An entirely different way by which constraints can be enforced is by means of a coordinate transformation. Consider, for example, the coordinate transformation $\mathcal{C}^w$ ($w_i = \frac{1}{4}v_i^2$). Negative weights are not reachable under this coordinate transformation because the factor $(\mathrm{d}w_i/\mathrm{d}v_i)^2 = w_i$ added to the growth rules (Eq. 31) as well as to the normalization rules (Eq. 37) allows the weight dynamics of weight w_i to slow down as it approaches zero, so that positive weights always stay positive (This can be generalized to positive and negative weights by the coordinate transformation $w_i = \frac{1}{4}v_i|v_i|$). Thus the coordinate transformation $\mathcal{C}^w$ (and also $\mathcal{C}^{\alpha w}$) implicitly introduces limitation constraint $\mathrm{I}_>$. This is interesting because it shows that a coordinate transformation can substitute for a constraint, which is well known in optimization theory.

The choice of whether to enforce the constraints by explicit normalization, by an integrated dynamics without an objective function, by penalty terms, or even implicitly by a coordinate transformation depends on the system as well as the methods applied to analyze it. Table 1 shows several constraint functions and their corresponding normalization rules as derived in different coordinate systems and by the three different methods discussed above. Not shown is normalization implicit in a coordinate transformation. It is interesting that there are only two types of constraints. All variations arise from using different coordinate systems and different methods by which the normalization rules are implemented. The first type is a *limitation constraint* I, which limits the range of individual weights. The second type is a *normalization constraint* N, which affects a group of weights, usually the sum, very rarely the sum of squares as indicated by Z. In the next section we show how to use Table 1 for analyzing models of neural map formation and give some examples from the literature.

6 Examples and Applications

6.1 How to use Table 1

The aim of Table 1 is to provide an overview of the different objective functions and derived growth terms as well as the constraint functions and derived normalization rules/terms discussed in this paper. The terms and rules are ordered in columns belonging to a particular coordinate transformation $\mathcal{C}$. Only entries in the same column may be combined to obtain a consistent constrained optimization formulation for a system. However, some terms can be derived under different coordinate transformations. For instance, the normalization rule $\mathrm{I}_=$ is the same for all coordinate transformations and term $\mathrm{L}^{\alpha w}$ with $\beta_i = 1/\alpha_i$ is the same as term L^w with $\beta_i = 1$.

To analyze a model of neural map formation, first identify possible candidates in Table 1 representing the different terms of the desired dynamics. Notice that the average activity of output neurons is represented by $\langle a_\tau \rangle = \sum_j A_{ij} w_j$ and that the correlation between input and output neurons is represented by $\langle a_\tau, a_\rho \rangle = \sum_j D_{ij} w_j$. Usually both terms will only be an approximation of the actual mean activities and correlations of the system under consideration (cf. Sec. 2.1). Notice also that normalization rules $\mathrm{N}^w_=$, $\mathrm{N}^{\alpha w}_=$, $\mathrm{Z}^1_=$, and $\mathrm{Z}^\alpha_=$ are actually multiplicative normalization rules and not subtractive ones as might be suggested by the special form in which they are written in Table 1.

Next identify the column in which all terms of the weight dynamics can be represented. This then gives the coordinate transformation under which the model can be analyzed through the objective functions and constraint or penalty functions listed on the left side of the table. Equivalent models (cf. Sec. 6.4) can be derived by moving from one column to another and by using normalization rules derived by a different method. Thus Table 1 provides a convenient tool for checking whether a system can be analyzed within the

		Coordinate Transformations			
		$\mathcal{C}^1$	$\mathcal{C}^\alpha$	$\mathcal{C}^w$	$\mathcal{C}^{\alpha w}$
		$w_i = v_i$	$w_i = \sqrt{\alpha_i} v_i$	$w_i = \frac{1}{4} v_i^2$	$w_i = \frac{1}{4} \alpha_i v_i^2$
		$\left(\frac{\mathrm{d}w_i}{\mathrm{d}v_i}\right)^2 = 1$	$\left(\frac{\mathrm{d}w_i}{\mathrm{d}v_i}\right)^2 = \alpha_i$	$\left(\frac{\mathrm{d}w_i}{\mathrm{d}v_i}\right)^2 = w_i$	$\left(\frac{\mathrm{d}w_i}{\mathrm{d}v_i}\right)^2 = \alpha_i w_i$
Objective Functions $H(\mathbf{w})$		Growth Terms: $\dot{w}_i = ... + ...$ or $\tilde{w}_i = w_i + \Delta t(... + ...)$			
L	$\sum_i \beta_i w_i$	β_i	$\alpha_i \beta_i$	$\beta_i w_i$	$\alpha_i \beta_i w_i$
Q	$\frac{1}{2} \sum_{ij} w_i D_{ij} w_j$	$\sum_j D_{ij} w_j$	$\alpha_i \sum_j D_{ij} w_j$	$w_i \sum_j D_{ij} w_j$	$\alpha_i w_i \sum_j D_{ij} w_j$
Constraint Functions $g(\mathbf{w})$		Normalization Rules (if constraint is violated): $w_i = ... \quad \forall i \in I_n$			
$\mathrm{I}_=$, $\mathrm{I}_\geq$	$\theta_i - w_i$	θ_i	θ_i	θ_i	θ_i
$\mathrm{N}_=$, $\mathrm{N}_\geq$	$\theta_n - \sum_{j \in I_n} \beta_j w_j$	$\tilde{w}_i + \lambda_n \beta_i$	$\tilde{w}_i + \lambda_n \alpha_i \beta_i$	$\tilde{w}_i + \lambda_n \beta_i \tilde{w}_i$	$\tilde{w}_i + \lambda_n \alpha_i \beta_i \tilde{w}_i$
$\mathrm{Z}_=$, $\mathrm{Z}_\geq$	$\theta_n - \sum_{j \in I_n} \beta_j w_j^2$	$\tilde{w}_i + \lambda_n \beta_i \tilde{w}_i$	$\tilde{w}_i + \lambda_n \alpha_i \beta_i \tilde{w}_i$	$\tilde{w}_i + \lambda_n \beta_i \tilde{w}_i^2$	$\tilde{w}_i + \lambda_n \alpha_i \beta_i \tilde{w}_i^2$
Constraint Functions $g(\mathbf{w})$		Normalization Terms: $\dot{w}_i = ...$ or $\tilde{w}_i = w_i + \Delta t(...)$			
$\mathrm{N}_\simeq$	$\theta_n - \sum_{j \in I_n} w_j$			$f_i - \frac{w_i}{\theta_n} \sum_j f_j$	
Penalty Functions $H(\mathbf{w})$		Normalization Terms: $\dot{w}_i = ... + ...$ or $\tilde{w}_i = w_i + \Delta t(... + ...)$			
$\mathrm{I}_\approx$	$-\frac{1}{2}\gamma_i(\theta_i - w_i)^2$	$\gamma_i(\theta_i - w_i)$	$\alpha_i \gamma_i(\theta_i - w_i)$	$\gamma_i w_i(\theta_i - w_i)$	$\alpha_i \gamma_i w_i(\theta_i - w_i)$
$\mathrm{I}_>$	$\gamma_i \ln \lvert\theta_i - w_i\rvert$	$-\frac{\gamma_i}{\theta_i - w_i}$	$-\frac{\alpha_i \gamma_i}{\theta_i - w_i}$	$-\frac{\gamma_i w_i}{\theta_i - w_i}$	$-\frac{\alpha_i \gamma_i w_i}{\theta_i - w_i}$
$\mathrm{N}_\approx$	$-\frac{1}{2}\gamma_n(\theta_n - \sum_{j \in I_n} \beta_j w_j)^2$	$\beta_i \gamma_n \times$ $(\theta_n - \sum_j \beta_j w_j)$	$\alpha_i \beta_i \gamma_n \times$ $(\theta_n - \sum_j \beta_j w_j)$	$\beta_i \gamma_n w_i \times$ $(\theta_n - \sum_j \beta_j w_j)$	$\alpha_i \beta_i \gamma_n w_i \times$ $(\theta_n - \sum_j \beta_j w_j)$

Table 1: Objective functions, constraint functions, and the dynamics terms they induce in different coordinate systems: $\mathcal{C}$ indicates a coordinate transformation that is specified by a superscript. L indicates a *linear term*. Q indicates a *quadratic term* that is usually induced by correlations $\langle a_\tau, a_\rho \rangle = \sum_j D_{ij} w_j$. But it can also account for mean activities $\langle a_\tau \rangle = \sum_j A_{ij} w_j$. I indicates a *limitation constraint* that limits the range for individual weights (I may stand for 'interval'). N indicates a *normalization constraint* that limits the sum over a set of weights. Z is a rarely used variation of N (The symbol Z can be thought of as a rotated N). Subscript signs distinguish between the different ways in which constraints can be enforced. $\mathrm{I}_\approx^w$, for instance, indicates the normalization term $\gamma_i w_i(\theta_i - w_i)$ induced by the penalty function $-\frac{1}{2}\gamma_i(\theta_i - w_i)^2$ under the coordinate transformation $\mathcal{C}^w$. Subscripts n and i for θ, λ, and γ denote different constraints of the same type, e.g. the same constraint applied to different output neurons. Normalization terms are integrated into the dynamics directly while normalization rules are applied iteratively to the dynamics of the growth rule. f_i denotes a *fitness* by which a weight would grow without any normalization (cf. Sec. 5.2.2).

constrained optimization framework presented here and for identifying the equivalent models. The function of each term can be coherently interpreted with respect to the objective, constraint, and penalty functions on the left side. The table can be extended with respect to additional objective, constraint, and penalty functions as well as additional coordinate transformations. Although the table is compact, it suffices to explain a wide range of representative examples from the literature, as discussed in the next section.

6.2 Examples from the Literature

Table 2 shows representative models from the literature. The original equations are listed as well as the classification in terms of growth rules and normalization rules listed in Table 1. Detailed comments for these models and the model in (Amari, 1980) follow below. The latter is not listed in Table 2 because it cannot be interpreted within our constrained optimization framework. The dynamics of the introductory example of Section 2 can be classified as Q^1 (Eq. 3), $I^1_{\geq}$ (Eq. 4), and $N^1_{\geq}$ (Eq. 5, 6).

The models are discussed here mainly with respect to whether or not they can be consistently described within the constrained optimization framework, i.e. whether or not growth rules and normalization rules can be derived from objective functions and constraint functions under one coordinate transformation (that does not imply anything about the quality of a model). Another important issue is whether the linear correlation model introduced in Section 3 is an appropriate description for the activity dynamics of these models. It is an accurate description for some of them but others are based on non-linear models and the approximations discussed in Section 2.1 and Appendix A have to be made.

Models typically contain three components: the quadratic term Q to induce neighborhood preserving maps, a limitation constraint I to keep synaptic weights positive, and a normalization constraint N (or Z) to induce competition between weights and to keep weights limited. The limitation constraint can be waived for systems with positive weights and multiplicative normalization rules (Konen & von der Malsburg, 1993; Obermayer et al., 1990; von der Malsburg, 1973) (cf. Sec. 5.2.4). A presynaptic normalization rule can be introduced implicitly by the activity dynamics (cf. Appendix A.2). In that case it may be necessary to use an explicit presynaptic normalization constraint in the constrained optimization formulation. Otherwise the system may have a tendency to collapse on the input layer (see 6.3), a tendency it does not have in the original formulation as a dynamical system. Only few systems contain the linear term L, which can be used for dynamic link matching. In (Häussler & von der Malsburg, 1983) the linear term was introduced for analytical convenience and does not differentiate between different links. The two models of dynamic link matching (Bienenstock & von der Malsburg, 1987; Konen & von der Malsburg, 1993) introduce similarity values implicitly and not through the linear term. The models are now discussed individually in chronological order.

von der Malsburg (1973): The activity dynamics of this model is non-linear and based on hexagon patterns in the output layer. Thus the applicability of the linear correlation model is not certain (cf. Sec. 2.1). The weight dynamics is inconsistent in its original formulation. However, Miller & MacKay (1994) have shown that constraints $N^w_=$ and $Z^1_=$ have a very similar effect on the dynamics, so that the weight dynamics could be made consistent by using $Z^1_=$ instead of $N^w_=$. No limitation constraint is necessary because neither the growth rule nor the multiplicative normalization rule can lead to negative weights and the normalization rule limits the growth of positive weights.

Amari (1980): This is a particularly interesting model not listed in Table 2. It is based on a blob dynamics, but no explicit normalization rules are applied, so that the derivation of correlations and mean activities as discussed in Section 3 cannot be used. Weights are prevented from growing infinitely by a simple decay term, which is possible because correlations induced by the blob model are finite and do not grow with the total strength of the synapses. Additional inhibitory inputs received by the output neurons from a constantly active neuron ensure that the average activity is evenly distributed in the output layer, which also leads to expanding maps. In this respect the architecture deviates from Figure 2. Thus this model cannot be formulated within our framework.

Whitelaw & Cowan (1981): The activity dynamics is non-linear and based on blobs. Thus the linear correlation model is only an approximation (cf. Sec. 2.1). The weight dynamics is difficult to interpret in the constrained optimization framework. The normalization rule is not specified precisely, but it is probably multiplicative because a subtractive one would lead to negative weights and possibly infinite weight growth. The quadratic term $-Q^1$ is based on mean activities and would lead by itself to zero weights. The Ω term

Reference	Weight Dynamics	Eq. (#)	Classification
VON DER MALSBURG (1973)	$\tilde{w}_{\tau\rho} = w_{\tau\rho} + h a_\rho a_\tau$ $w_{\tau\rho} = \tilde{w}_{\tau\rho} \cdot 19 \cdot \frac{w}{2} / \tilde{w}_\tau, \quad \tilde{w}_\tau = \sum_{\rho=1}^{19} \tilde{w}_{\tau\rho}$		Q^1 $\mathrm{N}^{w}_{=}$
WHITELAW & COWAN (1981)	$\dot{w}_{\tau\rho} = \alpha_{\tau\rho} a_\rho a_\tau - \alpha a_\tau + \Omega \quad$ (Ω: small noise term) $\sum_{\rho'} w_{\tau\rho'} = 1, \ \sum_{\tau'} w_{\tau'\rho} = 1$	(2) (5)	Q^α - Q^1 + ? $\mathrm{N}^{?}_{=}$
HÄUSSLER & VON DER MALSBURG (1983)	$\dot{w}_{\tau\rho} = f_{\tau\rho} - \frac{1}{2N} w_{\tau\rho} \left(\sum_{\tau'} f_{\tau'\rho} + \sum_{\rho'} f_{\tau\rho'} \right)$ $f_{\tau\rho} = \alpha + \beta w_{\tau\rho} C_{\tau\rho}$ $C_{\tau\rho} = \sum_{\tau'\rho'} D_{\tau\tau'} D_{\rho\rho'} w_{\tau'\rho'}$	(2.1) (2.2) (2.3)	$(\mathrm{I}^w_{>}+\mathrm{Q}^w)$-$(\mathrm{L}^w+\mathrm{N}^w_{\simeq})$
LINSKER (1986)	$\dot{w}_{\tau\rho} = k_1 + \frac{1}{N_G} \sum_{\rho'} \left(Q^F_{\rho\rho'} + k_2 \right) w_{\tau\rho'}$ $+ R_b \sum_{\tau'} f_{\tau\tau'} \left[k_{1a} + \frac{1}{N_G} \sum_{\rho'} \left(Q^F_{\rho\rho'} + k_2 \right) w_{\tau'\rho'} \right]$ $= k_1' - \frac{A_\rho - k_2}{N_G} \sum_{\tau'\rho'} D_{\tau\tau'} A_{\rho'} w_{\tau'\rho'} + \frac{1}{N_G} \sum_{\tau'\rho'} D_{\tau\tau'} D_{\rho\rho'} w_{\tau'\rho'}$ $(k_1' = k_1 + R_b k_{1a} \sum_{\tau'} f_{\tau\tau'}, \ D_{\tau\tau'} = R_b f_{\tau\tau'} + \delta_{\tau\tau'}$ ($\delta_{\tau\tau'}$ Kronecker), $D_{\rho\rho'} = \langle a_\rho a_{\rho'} \rangle, \ A_\rho = \langle a_\rho \rangle, \ k_2 < 0)$ some $w_{\tau\rho} \in [0,1]$ and some $w_{\tau\rho} \in [-1,0]$ or all $w_{\tau\rho} \in [-0.5, 0.5]$	(5)	L^1 + Q^1 $\mathrm{I}^1_{\geq}$
BIENENSTOCK & VON DER MALSBURG (1987)	$H = -\sum_{\tau\tau'\rho\rho'} D_{\tau\tau'} w_{\tau'\rho'} w_{\tau\rho} D_{\rho\rho'}$ $+ \gamma' \sum_\tau \left(\sum_\rho w_{\tau\rho} - p' \right)^2 + \gamma' \sum_\rho \left(\sum_\tau w_{\tau\rho} - p' \right)^2$ $w_{\tau\rho} \in [0, T_{\tau\rho}]$	(2)	Q^1 + $\mathrm{N}^1_{\approx}$ $\mathrm{I}^1_{\geq}$
MILLER ET AL. (1989)	$\dot{w}^L_{\tau\rho} = \lambda \alpha_{\tau\rho} \sum_{\tau'\rho'} D_{\tau\tau'} \left[D^{LL}_{\rho\rho'} w^L_{\tau'\rho'} + D^{LR}_{\rho\rho'} w^R_{\tau'\rho'} \right] - \left[\gamma w^L_{\tau\rho} + \epsilon \alpha_{\tau\rho} \right]$ a) $\sum_{\rho'} (w^L_{\tau\rho'} + w^R_{\tau\rho'}) = 2 \sum_{\rho'} \alpha_{\tau\rho'}, \quad w^L_{\tau\rho} = \tilde{w}^L_{\tau\rho} + \lambda_\tau \alpha_{\tau\rho}$ b) $\sum_{\tau'} w^L_{\tau'\rho} = \mathrm{const}, \quad w^L_{\tau\rho} = \tilde{w}^L_{\tau\rho} + \lambda_\tau \alpha_{\tau\rho}$ $w^L_{\tau\rho} \in [0, 8\,\alpha_{\tau\rho}]$ (If weights were cut due to $\mathrm{I}^\alpha_{\geq}$: $w^L_{\tau\rho} = \tilde{w}^L_{\tau\rho} + \lambda_\tau \tilde{w}^L_{\tau\rho}$) Interchanging L (left eye) and R (right eye) yields equations for $w^R_{\tau\rho}$.	(1) (Note 23)	Q^α - $\mathrm{I}^\alpha_{\approx}$ $\mathrm{N}^\alpha_{=}$ $\mathrm{N}^\alpha_{=}$ $\mathrm{I}^\alpha_{\geq}$ $(\mathrm{N}^w_{=})$
OBERMAYER ET AL. (1990)	$w_{\tau\rho}(t+1) = \frac{w_{\tau\rho}(t) + \epsilon(t) a_\tau(t) a_\rho}{\sqrt{\sum_{\rho'} \left(w_{\tau\rho'}(t) + \epsilon(t) a_\tau(t) a_{\rho'} \right)^2}}$	(4)	$\frac{\mathrm{Q}^1}{\mathrm{Z}^1_{=}}$
TANAKA (1990)	$\dot{w}_{\tau\rho} = w_{\tau\rho} \left[\kappa_0 - \kappa_1 \sum_{\rho'} \beta_{\rho'} w_{\tau\rho'} \right] + g m_\tau w_{\tau\rho} a_\rho + \gamma_{\tau\rho}$ (later in the paper $\beta_{\rho'} = 1$)	(2.1)	$\mathrm{N}^{\alpha w}_{\approx}$ + Q^w + $\mathrm{I}^w_{>}$ $(\mathrm{N}^{\alpha w}_{\approx} = \mathrm{N}^w_{\approx})$
GOODHILL (1993)	$w_{\tau\rho} = w_{\tau\rho} + \alpha a_\rho a_\tau$ a) $w_{\tau\rho} = \begin{cases} w_{\tau\rho} - t & \text{if } w_{\tau\rho} - t > 0 \\ 0 & \text{otherwise} \end{cases}, \ t = \frac{\sum_{\rho'} w_{\tau\rho'} - N_\tau}{n_\tau}, \ n_\tau = \sum_{\{\rho' \mid 0 < w_{\tau\rho'}\}} 1$ (If some weights have become zero due to $\mathrm{I}^1_{\geq}$: $w_{\tau\rho} = \frac{N_\tau w_{\tau\rho}}{\sum_{\rho'} w_{\tau\rho'}}$) b) $w_{\tau\rho} = \frac{N_\rho w_{\tau\rho}}{\sum_{\tau'} w_{\tau'\rho}}$		Q^1 $\left\{ \begin{matrix} \mathrm{N}^1_{=} \\ \mathrm{I}^1_{\geq} \end{matrix} \right.$ $(\mathrm{N}^w_{=})$ $\mathrm{N}^w_{=}$
KONEN & VON DER MALSBURG (1993)	$w_{\tau\rho} \rightarrow w_{\tau\rho} + \epsilon w_{\tau\rho} \alpha_{\tau\rho} a_\tau a_\rho$ $\rightarrow w_{\tau\rho} / \sum_{\rho'} \frac{w_{\tau\rho'}}{\alpha_{\tau\rho'}}$ $\rightarrow w_{\tau\rho} / \sum_{\tau'} \frac{w_{\tau'\rho}}{\alpha_{\tau'\rho}}$ ($w_{\tau\rho}$ are the "effective couplings" $J_{\tau\rho} T_{\tau\rho}$)	(3.5)	$\mathrm{Q}^{\alpha w}$ $\mathrm{N}^{\alpha w}_{=}$ $\mathrm{N}^{\alpha w}_{=}$

Table 2: Examples of weight dynamics from previous studies. The original equations are written in a form that uses the notation of this paper. The classification of the original equations by means of the terms and coordinate transformations listed in Table 1 are shown in the right column (the coordinate transformations are indicated by superscripts). See Section 6.2 for further comments on these models.

was introduced only to test the stability of the system.

Häussler & von der Malsburg (1983): This model is directly formulated in terms of weight dynamics, thus the linear correlation model is accurate. The weight dynamics is consistent; however, as argued in Section 5.2.2, there is usually no objective function for the normalization rule $\mathrm{N}^w_{\simeq}$, but by replacing $\mathrm{N}^w_{\simeq}$ by $\mathrm{N}^w_{=}$ or $\mathrm{N}^w_{\approx}$, the system can be expressed as a constrained optimization problem without qualitatively changing the model behavior. The limitation term $\mathrm{I}^w_{>}$ and the linear term L^w are induced by the constant α and were introduced for analytical reasons. The former is meant to allow weights to grow from zero strength and the latter limits this growth. α needs to be small for neural map formation and for a stable one-to-one mapping, α strictly should be zero. Thus, these two terms could be discarded if all weights would be initially larger than zero. Notice that the linear term does not differentiate between different links and thus does not have a function as suggested for dynamic link matching (cf. Sec. 4 and 6.5).

Linsker (1986): This model is also directly formulated in terms of weight dynamics, thus the linear correlation model is accurate. The weight dynamics is consistent. Since the model uses negative and positive weights and weights have a lower and an upper bound, no normalization rule is necessary. The weights converge to their upper or lower limit.

Bienenstock & von der Malsburg (1987): This is a model of dynamic link matching and originally formulated in terms of an energy function. Thus the classification is accurate. The energy function does not include the linear term. The features are binary, black vs. white, and the similarity values are therefore 0 and 1 and do not enter the dynamics as continuous similarity values. The $T_{\tau\rho}$ in the constraint $\mathrm{I}^1_{\geq}$ represent the stored patterns in the associative memory and not similarity values.

Miller et al. (1989): This model is directly formulated in terms of weight dynamics, thus the linear correlation model is accurate. One inconsistent part in the weight dynamics is the multiplicative normalization rule $\mathrm{N}^w_{=}$, which is applied when subtractive normalization leads to negative weights. But it is only an algorithmic shortcut to solve the problem of interfering constraints (limitation and subtractive normalization). A more systematic treatment of the normalization rules could replace this inconsistent rule (cf. Sec. 5.2.1). Another inconsistency is that weights that reach their upper or lower limit become frozen, i.e. fixed at the limit value. With some exception this seems to have little effect on the resulting maps (Miller et al., 1989, Note 23.). Thus this model has only two minor inconsistencies, which could be modified to make the system consistent. Limitation constraints enter the weight dynamics in two forms, $\mathrm{I}^\alpha_{\approx}$ and $\mathrm{I}^\alpha_{\geq}$. The former tends to keep $w^L_{\tau\rho} = -\frac{\epsilon}{\gamma}\alpha_{\tau\rho}$ while the latter keeps $w^L_{\tau\rho} \in [0, 8\alpha_{\tau\rho}]$, which can unnecessarily introduce conflicts. However, $\gamma = \epsilon = 0$, so that only the latter constraint applies and the $\mathrm{I}^\alpha_{\approx}$ term is discarded in later publications. The system can in principle be simplified by using coordinate transformation $\mathcal{C}^1$ instead of $\mathcal{C}^\alpha$. Thereby eliminating $\alpha_{\tau\rho}$ in the growth rule Q^α as well as in the normalization rule $\mathrm{N}^\alpha_{=}$, but not in the normalization rule $\mathrm{I}^\alpha_{\geq}$. This is different from setting $\alpha_{\tau\rho}$ to a constant in a certain region. Using coordinate transformation $\mathcal{C}^1$ would result in the same set of stable solutions, though the trajectories would differ. Changing $\alpha_{\tau\rho}$ generates a different set of solutions. However, the original formulation using $\mathcal{C}^\alpha$ is more intuitive and generates the 'correct' trajectories, i.e. those which correspond to the intuitive interpretation of the model.

Obermayer et al. (1990): This model is based on an algorithmic blob model and the linear correlation model is only an approximation (cf. Appendix A). The weight dynamics is consistent. It employs the rarely used normalization constraint Z, which induces a multiplicative normalization rule under the coordinate transformation $\mathcal{C}^1$. No limitation constraint is necessary because neither the growth rule nor the multiplicative normalization rule can lead to negative weights and positive weights are limited by the normalization rule.

Tanaka (1990): This model uses a non-linear input-output function for the neurons, which makes a clear distinction between membrane potential and firing rate. However, this non-linearity does not seem to play a specific functional role and is partially eliminated by linear approximations. Thus the linear correlation model seems to be justified. The weight dynamics includes parameters $\beta_{\rho'}$ (f_{SP} in the original notation), which make it inconsistent. The penalty term $\mathrm{N}^{\alpha w}_{\approx}$, which induces the first terms of the weight dynamics, is $-\frac{1}{2\kappa_1}\sum_{\tau'}(\kappa_0 - \kappa_1 \sum_{\rho'} \beta_{\rho'} w_{\tau'\rho'})^2$, which has to be evaluated under the coordinate transformation $\mathcal{C}^{\alpha w}$ with $\alpha_{\tau\rho} = 1/\beta_\rho$. Later in the paper the parameters $\beta_{\rho'}$ are set to 1, so that the system becomes consistent. Tanaka gives an objective function for the dynamics, employing a coordinate transformation for this purpose. The objective function is not listed here because it is derived under a different set of assumptions, including the non-linear input-output function of the output neurons and a mean field approximation.

Goodhill (1993): This model is based on an algorithmic blob model and the linear correlation model is only an approximation (cf. Appendix A). As the model in (Miller et al., 1989) this model uses an inconsistent normalization rule as a backup and it freezes weights that reach their upper or lower limit. In addition it uses an inconsistent normalization rule for the input neurons. But since this inconsistent multiplicative normalization for the input neurons is applied after a consistent subtractive normalization for the output neurons, its effect is relatively weak and substituting it by a subtractive one would make little difference (G.J. Goodhill, personal communication). To avoid *dead units*, i.e. neurons in the output layer that never become active, Goodhill (1993) divides each output activity by the number of times each output neuron has won the competition for the blob in the output layer. This guarantees a roughly equal average activity of the output neurons. With the probabilistic blob model (cf. Appendix A) dead units do not occur as long as output neurons have any input connections. The specific parameter setting of the model even guarantees a roughly equal average activity of the output neurons under the probabilistic blob model because the sum over the weights converging on an output neuron is roughly the same for all neurons in the output layer. Thus, despite some inconsistencies this model can probably be well approximated within the constrained optimization framework.

Konen & von der Malsburg (1993): The activity dynamics is non-linear and based on blobs. Thus the linear correlation model is only an approximation (cf. Sec. 2.1). The weight dynamics is consistent. Although this is a model of dynamic link matching it does not contain the linear term to bias the links. It introduces the similarity values in the constraints and through the coordinate transformation $\mathcal{C}^{\alpha w}$ (see Sec. 6.4). No limitation constraint is necessary because neither the growth rule nor the multiplicative normalization rule can lead to negative weights and positive weights are limited by the normalization rule.

6.3 Some Functional Aspects of Term Q

So far the focus of the considerations was only on formal aspects of models of neural map formation. In this section some remarks on functional aspects of the quadratic term Q are made.

Assume the effective lateral connectivities in the output layer and in the input layer are sums of positive and/or negative contributions. Each contribution can either be a constant, C, or a centered Gaussian-like function, G, which depends only on the distance of the neurons, e.g. $D_{\rho\rho'} = D_{|\rho-\rho'|}$ if ρ is a spatial coordinate. The contributions can be indicated by subscripts to the objective function Q. First index indicates the lateral connectivity of the input layer, the second index the one of the output layer. A negative Gaussian (constant) would have to be indicated by $-G$ $(-C)$. $\mathrm{Q}_{(-C)G}$, for instance, would indicate a negative constant $D_{\rho\rho'}$ and a positive Gaussian $D_{\tau\tau'}$. $\mathrm{Q}_{G(G-G')}$ would indicate a positive Gaussian $D_{\rho\rho'}$ and a $D_{\tau\tau'}$ that is a difference of Gaussians. Notice that negative signs can cancel each other, e.g. $\mathrm{Q}_{(G-C)G}$ = -$\mathrm{Q}_{(C-G)G}$ = -$\mathrm{Q}_{(G-C)(-G)}$. We thus discuss the terms only in their simplest form, i.e. -Q_{CG} instead of $\mathrm{Q}_{(-C)G}$. All feed-forward weights are assumed to be positive. Assuming all weights to be negative would lead to equivalent results because Q does not change if all weights change their sign. The situation becomes more complex if some weights were positive and others negative. A term Q is called positive if it can be written in a form where it has a positive sign and only positive contributions, e.g. -$\mathrm{Q}_{(-C)G}$ = Q_{CG} is positive while $\mathrm{Q}_{(G-C)G}$ is not. Since Q is symmetrical with respect to $D_{\rho\rho'}$ and $D_{\tau\tau'}$, a term such as $\mathrm{Q}_{(G-C)G}$ has the same effect as $\mathrm{Q}_{G(G-C)}$ with the role of input layer and output layer exchanged. A complicated term can be most easily analyzed by splitting it into its *elementary* components. For instance, the term $\mathrm{Q}_{G(G-C)}$ can be split into $\mathrm{Q}_{GG}-\mathrm{Q}_{GC}$ and analyzed as a combination of these two simpler terms.

Some elementary terms are now discussed in greater detail. The effect of the terms is considered under two types of constraints. Constraint A: The total sum of weights is constrained, $\sum_{\rho'\tau'} w_{\rho'\tau'} = 1$. Constraint B: The sums of weights originating from an input neuron, $\sum_{\tau'} w_{\rho\tau'} = 1/R$, or terminating on an output neuron, $\sum_{\rho'} w_{\rho'\tau} = 1/T$, are constrained, where R and T denote the number of input and output neurons, respectively. Without further constraints, a positive term always leads to infinite weight growth and a negative term to weight decay.

Terms $\pm\mathrm{Q}_{CC}$ simplify to $\pm\mathrm{Q}_{CC}$= $\pm D_{\rho\rho} D_{\tau\tau} (\sum_{\rho'\tau'} w_{\rho'\tau'})^2$ and depend only on the sum of weights. Thus, neither term has any effect under Constraints A or B.

Term $+\mathrm{Q}_{CG}$ takes its maximum value under Constraint A if all links terminate on one output neuron. The map has the tendency to *collapse*. This is because the lateral connections in the output layer are higher for smaller distances and maximal for zero distance between connected neurons. Under the constraint

$\sum_{\tau'} w_{\rho\tau'} \leq 1, \sum_{\rho'} w_{\rho'\tau} \leq 1$, for instance, the resulting map connects the input layer to a region in the output layer that is of the size of the input layer even if the output layer is much larger. No topography is taken into account because $D_{\rho\rho'}$ is constant and does not differentiate between different input neurons. Thus this term has no effect under Constraint B.

Term -Q_{CG} has the opposite effect of +Q_{CG}. Consider the induced growth term $\dot{w}_{\rho\tau} = -D_{\rho\rho} \sum_{\tau'} D_{\tau\tau'} \sum_{\rho'} w_{\tau'\rho'}$. This is a convolution of $D_{\tau\tau'}$ with $\sum_{\rho'} w_{\tau'\rho'}$ and induces the largest decay in regions where the weighted sum over terminating links is maximal. A stable solution would require equal decay for all weights because Constraint A can only compensate for equal decay. Thus the convolution of $D_{\tau\tau'}$ with $\sum_{\rho'} w_{\tau'\rho'}$ must be a constant. Since $D_{\tau\tau'}$ is a Gaussian, this is only possible if $\sum_{\rho'} w_{\tau'\rho'}$ is a constant, as can be easily seen in Fourier space. Thus the map *expands* over the output layer and each output neuron receives the same sum of weights. Constraint A could be substituted by a constant growth term L, in which case the expansion effect could be obtained without any explicit constraint. As +Q_{CG}, this term has no effect under Constraint B.

Term +Q_{GG} takes its maximum value under Constraint A if all but one weight are zero. The map collapses on the input and the output layer. Under Constraint B, the map becomes topographic because links that originate from neighboring neurons (high $D_{\rho\rho'}$-value) favorably terminate on neighboring neurons (high $D_{\tau\tau'}$-value). A more rigorous argument would require a definition of topography but as argued in Section 6.7, the term +Q_{GG} can be directly taken as a generalized measure for topography.

Term -Q_{GG} has the opposite effect of +Q_{GG}. Thus it leads under Constraint A to a map that is *expanded* over input and output layer. In addition the map becomes anti-topographic. Further analytical or numerical investigations are required to show whether the expansion is as even as for the term -Q_{CG} and how an anti-topographic map may look like. Constraint B also leads to an anti-topographic map.

6.4 Equivalent Models

The effect of coordinate transformations has been considered so far only for single growth terms and normalization rules. Coordinate transformations can be used to generate different models that are equivalent in terms of their constrained optimization problem. Consider the system in (Konen & von der Malsburg, 1993). Its objective function and constraint function are Q and $N_{\geq}$,

$$H(\mathbf{w}) = \frac{1}{2} \sum_{ij} w_i D_{ij} w_j \ , \qquad g_n(\mathbf{w}) = 1 - \sum_{j \in I_n} \frac{w_j}{\alpha_j} = 0 \ , \tag{42}$$

which must be evaluated under the coordinate transformation $\mathcal{C}^{\alpha w}$ to induce the original weight dynamics $Q^{\alpha w}$ and $N_{\geq}^{\alpha w}$,

$$\dot{w}_i = \alpha_i w_i \sum_j D_{ij} w_j \ , \qquad w_i = \frac{\tilde{w}_i}{\sum_{j \in I_n} \frac{\tilde{w}_j}{\alpha_j}} \ . \tag{43}$$

If evaluated directly, i.e. under the coordinate transformation $\mathcal{C}^1$, one would obtain

$$\dot{w}_i = \sum_j D_{ij} w_j \ , \qquad w_i = \tilde{w}_i + \frac{1}{\sum_{j \in I_n} \alpha_j^{-2}} (1 - \sum_{j \in I_n} \frac{\tilde{w}_j}{\alpha_j}) \frac{1}{\alpha_i} \ . \tag{44}$$

As argued in Section 5.2.4 an additional limitation constraint $I_{>}^1$ (or $I_{\geq}^1$) has to be added to this system to account for the limitation constraint implicitly introduced by the coordinate transformation $\mathcal{C}^{\alpha w}$ for the dynamics above (Eq. 43).

It follows from Equation (31) that the flow fields of the weight dynamics in Equations (43) and (44) differ, but since $\mathrm{d}w_i/\mathrm{d}v_i \neq 0$ for positive weights, the fixed points are the same. That means that the resulting maps to which the two systems converge, possibly from different initial states, are the same. In this sense these two dynamics are equivalent.

This also holds for other coordinate transformations within the defined region as long as $\mathrm{d}w_i/\mathrm{d}v_i$ is finite ($\mathrm{d}w_i/\mathrm{d}v_i = 0$ may introduce additional fixed points). Thus this method of generating equivalent models makes it possible to abstract the objective function from the dynamics. Different equivalent dynamics may have different convergence properties, their attractor basins may differ and some regions in state space may not be reachable under a particular coordinate transformation. In any case, within the reachable state space the fixed points are the same. Thus, coordinate transformations make it possible to optimize the dynamics without changing its objective function.

It should also be mentioned that normalization rules derived by different methods can substitute each other without changing the qualitative behavior of a system. For instance, $\mathrm{I}_{=}$ can be replaced by $\mathrm{I}_{\approx}$ or $\mathrm{N}_{\geq}$ can be replaced by $\mathrm{N}_{>}$ under any coordinate transformation. These replacements will also generate equivalent systems in a practical sense.

6.5 Dynamic Link Matching

In the previous section the similarity values α_i entered the weight dynamics in two places. In Equation (43) the differential effect of α_i enters only the growth rule, while in Equation (44) it enters only the normalization rule. Growth and normalization rules can to some extent be interchangeably used to incorporate feature information in dynamic link matching. However, the objective function (Eq. 42) shows that the similarity values are introduced through the constraints and that they are transfered to the growth rule only by the coordinate transformation $\mathcal{C}^{\alpha w}$. Similarity values can enter the growth rule more directly through the linear term L. An alternative objective function for dynamic link matching is

$$H(\mathbf{w}) = \sum_i \beta_i w_i + \frac{1}{2}\sum_{ij} w_i D_{ij} w_j \ , \qquad g_n(\mathbf{w}) = 1 - \sum_{j \in I_n} w_j = 0 \ , \tag{45}$$

with $\beta_i = \alpha_i$. The first term now directly favors links with high similarity values. This may be advantageous because it allows better control over the influence of the topography vs. the feature similarity term. Furthermore, this objective function is more closely related to the similarity function of elastic graph matching in (Lades et al., 1993), which has been developed as an algorithmic abstraction of dynamic link matching (see Sec. 6.7).

6.6 Soft vs. Hard Competitive Normalization

Miller & MacKay (1994) have analyzed the role of normalization rules for neural map formation. They consider a linear Hebbian growth rule Q^1 and investigate the dynamics under a subtractive normalization rule $\mathrm{N}^1_{=}$ (S1 in their notation) and two types of multiplicative normalization rules, $\mathrm{N}^w_{=}$ and $\mathrm{Z}^1_{=}$ (M1 and M2 in their notation, respectively). They show that when considering an isolated output neuron with the multiplicative normalization rules, the weight vector tends to the principal eigenvector of the matrix D, which means that many weights can maintain some finite value. Under the subtractive normalization rule, a winner-take-all behavior occurs and the weight vector tends to saturate with each single weight having either its minimal or maximal value producing a more compact receptive field. If no upper bound is imposed on individual weights, only one weight survives corresponding to a point receptive field.

von der Malsburg & Willshaw (1981) have performed a similar, though less comprehensive, analysis using a different approach. Instead of modifying the normalization rule they considered different growth rules with the same multiplicative normalization rule $\mathrm{N}^w_{\simeq}$. They also found two qualitatively different behaviors, a highly competitive case in which only one link survives (or several if single weights are limited in growth by individual bounds) (case μ=1 or μ=2 in their notation) and a less competitive case in which each weight is eventually proportional to the correlation between pre- and post-synaptic neuron (case μ=0).

Hence, one can either change the normalization rule and keep the growth rule or, *vice versa*, modify the growth rule and keep the normalization rule the same. Either choice generates the two different behaviors. As shown above, by changing both the growth and normalization rules consistently by a coordinate transformation, it is possible to obtain two different weight dynamics with qualitatively the same behavior. More precisely, the system (Q^w, N^w) is equivalent to (Q^1, N^1, I^1) and has the same fixed points; the former one uses a multiplicative normalization rule while the latter uses a subtractive one. This also explains why changing the growth rule or changing the normalization rule can be equivalent.

It may therefore be misleading to refer to the different cases by the specific normalization rules (subtractive vs. multiplicative), because that is valid only for the linear Hebbian growth rule Q^1. We suggest using a more generally applicable nomenclature that refers to the different behaviors rather than the specific mathematical formulation. Following the terminology of Nowlan (1990) in a similar context, the term *hard competitive* normalization could be used to denote the case where only one link survives (or a set of saturated links, which are limited by upper bounds); the term *soft competitive* normalization could be used to denote the case where each link has some strength proportional to its fitness.

6.7 Related Objective Functions

Objective functions also provide means for comparing weight dynamics with other algorithms or dynamics of a different origin for which an objective function exists.

First, it should be pointed out that maximizing the objective functions L and Q under linear constraints I and N is the quadratic programming problem, and finding an optimal one-to-one mapping between two layers of same size for objective function Q is the quadratic assignment problem. These problems are known to be NP-complete. However, there is a large literature on algorithms that efficiently solve special cases or that find good approximate solutions in polynomial time (e.g. Horst et al., 1995).

Many related objective functions are only defined for maps for which each input neuron terminates on exactly one output neuron with weight 1, which makes the index $\tau = \tau(\rho)$ a function of index ρ. An objective function of this kind may have the form

$$H = \sum_{\rho\rho'} G_{\tau\rho\tau'\rho'} \ , \tag{46}$$

where G encodes how well a pair of links from ρ to $\tau(\rho)$ and from ρ' to $\tau'(\rho')$ preserves topography. A pair of parallel links, for instance, would yield high G-values while others would yield lower values. Now define a particular family of weights $\mathbf{w}$ that realize one-to-one connectivities:

$$\bar{w}_{\tau\rho} = \begin{cases} 1 & \text{if } \tau = \tau(\rho) \\ 0 & \text{otherwise} \end{cases} \ . \tag{47}$$

$\bar{\mathbf{w}}$ is a subset of $\mathbf{w}$ with $\bar{w}_{\tau\rho} \in \{0, 1\}$ as opposed to $w_{\tau\rho} \in [0, 1]$. It indicates that an objective function was originally defined for a one-to-one map rather than the more general case of an all-to-all connectivity. Then objective functions of one-to-one maps can be written as

$$H(\bar{\mathbf{w}}) = \sum_{\tau\rho\tau'\rho'} \bar{w}_{\tau\rho} G_{\tau\rho\tau'\rho'} \bar{w}_{\tau'\rho'} = \sum_{ij} \bar{w}_i G_{ij} \bar{w}_j \tag{48}$$

with $i = \{\rho, \tau\}, j = \{\rho', \tau'\}$ as defined above. Simply replacing $\bar{\mathbf{w}}$ by $\mathbf{w}$ then yields a generalization of the original objective function to all-to-all connectivities.

Goodhill et al. (1996) have compared ten different objective functions for topographic maps and have proposed another, the C-measure. They show that for the case of an equal number of neurons in the input and the output layer, most other objective functions can be either reduced to the C-measure, or they represent a closely related objective function. This suggests that the C-measure is a good unifying measure for topography. The C-measure is equivalent to our objective function Q with $\bar{\mathbf{w}}$ instead of $\mathbf{w}$. Adapted to the notation of this paper the C-measure has the form

$$C(\bar{\mathbf{w}}) = \sum_{ij} \bar{w}_i G_{ij} \bar{w}_j \ , \tag{49}$$

with a separable G_{ij}, i.e. $G_{ij} = G_{\rho\tau\rho'\tau'} = G_{\tau\tau'} G_{\rho\rho'}$. Thus, the objective function Q is the typical term for topographic maps in other contexts as well.

Elastic graph matching is an algorithmic counterpart to dynamic link matching and has been used for applications such as object and face recognition (Lades et al., 1993). It is based on a similarity function that in its simplest version is

$$H(\bar{\mathbf{w}}) = \sum_i \beta_i \bar{w}_i + \frac{1}{2} \sum_{ij} \bar{w}_i G_{ij} \bar{w}_j \ , \tag{50}$$

where $G_{ij} = -[(\mathbf{p}_\rho - \mathbf{p}_{\rho'}) - (\mathbf{p}_\tau - \mathbf{p}_{\tau'})]^2$, and $\mathbf{p}_\rho$ and $\mathbf{p}_\tau$ are two-dimensional position vectors in the image plane. This similarity function corresponds formally to the objective function in Equation (45). The main difference between these two functions is hidden in G and D. The latter ought to be separable into two factors $D_{\rho\tau\rho'\tau'} = D_{\rho\rho'} D_{\tau\tau'}$ while the former is clearly not. G actually favors a metric map, which tends to preserve not only neighborhood relations but also distances, whereas with D the maps always tend to collapse.

6.8 Self-Organizing Map Algorithm

Models of the self-organizing map (SOM) algorithm can be high-dimensional or low-dimensional and two different learning rules, which we have called weight dynamics, are commonly used. The validity of the probabilistic blob model for the high-dimensional models is discussed in Appendix A. A classification of the high-dimensional model by Obermayer et al. (1990) is given in Table 2. The low-dimensional models do not fall into the class of one-to-one mappings considered in the previous section, because the input layer is represented as a continuous space and not as a discrete set of neurons.

One learning rule for the high-dimensional SOM-algorithm is given by

$$\tilde{w}_{\tau\rho}(t) = w_{\tau\rho}(t-1) + \epsilon B_{\tau\tau_0} B_{\rho\rho_0} \tag{51}$$

$$w_{\tau\rho}(t) = \frac{\tilde{w}_{\tau\rho}(t)}{\sqrt{\sum_{\rho'} \tilde{w}^2_{\tau\rho'}(t)}} , \tag{52}$$

as used, for example, in (Obermayer et al., 1990). $B_{\tau\tau_0}$ denotes the neighborhood function (commonly indicated by h) and $B_{\rho\rho_0}$ denotes the stimulus pattern (sometimes indicated by x) with index ρ_0. $B_{\rho\rho_0}$ does not need to have a blob shape, so that ρ_0 may be an arbitrary index. Output neuron τ_0 is the winner neuron in response to stimulus pattern ρ_0. This learning rule is a consistent combination of growth rule Q^1 and normalization rule $\mathrm{Z}^1_{=}$ and an objective function exists, which is a good approximation to the extent the probabilistic blob model is valid.

The second type of learning rule is given by

$$w_{\tau\rho}(t+1) = w_{\tau\rho}(t) + \epsilon B_{\tau\tau_0}(B_{\rho\rho_0} - w_{\tau\rho}(t)) . \tag{53}$$

as used, for example, in (Bauer et al., 1997). For this learning rule the weights and the input stimuli are assumed to be sum normalized, i.e. $\sum_\rho w_{\tau\rho} = 1$ and $\sum_\rho B_{\rho\rho_0} = 1$. For small ϵ this learning rule is equivalent to

$$\tilde{w}_{\tau\rho}(t) = w_{\tau\rho}(t-1) + \epsilon B_{\tau\tau_0} B_{\rho\rho_0} \tag{54}$$

$$w_{\tau\rho}(t) = \frac{\tilde{w}_{\tau\rho}(t)}{\sum_{\rho'} \tilde{w}_{\tau\rho'}(t)} , \tag{55}$$

which shows that it is a combination of growth rule Q^1 and normalization rule $\mathrm{N}^w_{=}$. Thus this system is inconsistent and to formulated it within our constrained optimization framework $\mathrm{N}^w_{=}$ would have to be approximated by $\mathrm{Z}^1_{=}$, which leads back to the learning rule in Equations (51, 52).

There are two ways of going from these high-dimensional models to the low-dimensional models. The first is simply to use fewer input neurons, e.g. two. A low-dimensional input vector is then represented by the activities of these few neurons. However, since the low-dimensional input vectors are usually not normalized to homogeneous mean activity of the input neurons and since the receptive and projective fields of the neurons do not co-develop in a homogeneous way, the probabilistic blob model is usually not valid.

A second way of going from a high-dimensional model to a low-dimensional model is by considering the low-dimensional input vectors and weight vectors as abstract representatives of the high-dimensional ones (Ritter et al., 1991; Behrmann, 1993). Consider, for example, the weight dynamics in Equation (53) and a two-dimensional input layer. Let $\mathbf{p}_\rho$ be a position vector of input neuron ρ. The center of the receptive field of neuron τ can be defined as

$$\mathbf{m}_\tau(\mathbf{w}) = \sum_\rho \mathbf{p}_\rho w_{\tau\rho} , \tag{56}$$

and the center of the input blob can be defined similarly,

$$\mathbf{x}(\mathbf{B}_{\rho_0}) = \sum_\rho \mathbf{p}_\rho B_{\rho\rho_0} . \tag{57}$$

Notice that the input blobs as well as the weights are normalized, i.e. $\sum_\rho B_{\rho\rho_0} = 1$ and $\sum_\rho w_{\tau\rho} = 1$. Using these definitions and given a pair of blobs at locations ρ_0 and τ_0, the high-dimensional learning rule (Eq. 53) yields the low-dimensional learning rule

$$\begin{aligned} \mathbf{m}_\tau(\mathbf{w}(t+1)) &= \sum_\rho \mathbf{p}_\rho \left(w_{\tau\rho}(t) + \epsilon B_{\tau\tau_0}(B_{\rho\rho_0} - w_{\tau\rho}(t)) \right) && (58) \\ &= \mathbf{m}_\tau(\mathbf{w}(t)) + \epsilon B_{\tau\tau_0} \left(\mathbf{x}(\mathbf{B}_{\rho_0}) - \mathbf{m}_\tau(\mathbf{w}(t)) \right) && (59) \\ \iff \quad \mathbf{m}_\tau(t+1) &= \mathbf{m}_\tau(t) + \epsilon B_{\tau\tau_0} \left(\mathbf{x}_{\rho_0} - \mathbf{m}_\tau(t) \right) \ . && (60) \end{aligned}$$

One can first calculate the centers of the receptive fields of the high-dimensional model and then apply the low-dimensional learning rule or one can first apply the high-dimensional learning rule and then calculate the centers of the receptive fields, the result is the same. Notice that the low-dimensional learning rule is even formally equivalent to the high-dimensional one and that it is the rule commonly used in low-dimensional models (Kohonen, 1990). Even though the high- and the low-dimensional learning rules are equivalent for a given pair of blobs, the overall behavior of the models is not. This is because the positioning of the output blobs is different in the two models (Behrmann, 1993). It is clear that many different high-dimensional weight configurations having different output blob positioning can lead to the same low-dimensional weight configuration. However, for a high-dimensional model that self-organizes a topographic map with point receptive fields, the positioning may be similar for the high- and the low-dimensional models, so that the stable maps may be similar as well.

These considerations show that only the high-dimensional model in Equations (51, 52) can be consistently described within our constrained optimization framework. The high-dimensional model of Equation (53) is inconsistent. The probabilistic blob model is in general not applicable to low-dimensional models, because some assumptions required for its derivation are not valid. The simple relation between the high- and the low-dimensional model sketched above holds only for the learning step but not for the blob positioning, though the positioning and thus the resulting maps may be very similar for topographic maps with point receptive fields.

7 Conclusions and Future Perspectives

The results presented here can be summarized as follows:

- A probabilistic non-linear blob model can behave like a linear correlation model under fairly general conditions (Section 2.1 and Appendix A). This clarifies the relationship between deterministic non-linear blob models and linear correlation models and provides an approximation of the former by the latter.

- Coordinate transformations can transform dynamics with curl into curl-free dynamics, allowing the otherwise impossible formulation of an objective function (Section 4). A similar effect exists for normalization rules. Coordinate transformations can transform non-orthogonal normalization rules into orthogonal ones, allowing the normalization rule to be formulated as a constraint (Section 5.1).

- Growth rules and normalization rules must have a special relationship in order to make a formulation of the system dynamics as a constrained optimization problem possible, namely the growth rule must be a gradient flow and the normalization rules must be orthogonal under the same coordinate transformation (Section 5.1).

- Constraints can be enforced by various types of normalization rules (Section 5.2) and they can even be implicitly introduced by coordinate transformations (Section 5.2.4) or the activity dynamics (Appendix A.2).

- Many all-to-all connected models from the literature can be classified within our constrained optimization framework based on only four terms, L, Q, I, and N (Z) (Section 6.2). The linear term L has rarely been used, but it can have a specific function that may be useful in future models (Section 6.5).

- Models may differ considerably in their weight dynamics and still solve the same optimization problem. This can be revealed by coordinate transformations and by comparing the different but possibly equivalent types of normalization rules (Section 6.4). Coordinate transformations make it in particular possible to optimize the dynamics without changing the stable fixed points.

- The constrained optimization framework provides a convenient formalism to analyze functional aspects of the models (Sections 6.3, 6.5, 6.6).
- The constrained optimization framework for all-to-all connected models presented here is closely related to approaches for finding optimal one-to-one maps (Section 6.7) but is not easily adapted to the self-organizing map algorithm (Section 6.8).
- Models of neural map formation formulated as constrained optimization problems provides a unifying framework. It abstracts from arbitrary differences in the design of models and leaves only those differences that are likely to be crucial for the different structures that emerge by self-organization.

It is important to note that our constrained optimization framework is unifying in the sense that it provides a canonical formulation independent of most arbitrary design decisions, e.g. due to different coordinate transformations or different types of normalization rules. This does not mean that most models are actually equivalent. But with the canonical formulation of the models as constrained optimization problems it should be possible to focus on the crucial differences and to understand better what the essentials of neural map formation are.

Based on the constrained optimization framework presented here, a next step would be to consider specific architectures with particular effective lateral connectivities and to investigate the structures that emerge. The role of parameters and effective lateral connectivities might be investigated analytically for a variety of models by means of objective functions, similar to the approach sketched in Section 6.3 or the one taken in (MacKay & Miller, 1990).

We have considered here only three levels of abstraction: detailed neural dynamics, abstract weight dynamics, and constrained optimization. There exist even higher levels of abstraction and the relationship between our constrained optimization framework and these more abstract models should be explored. For example, in Section 6.7 our objective functions were compared with other objective functions defined only for one-to-one connectivities. Another possible link is with Bienenstock & von der Malsburg (1987) and Tanaka (1990) who have proposed spin models for neural map formation. An interesting approach is that taken by Linsker (1986), who analyzed the receptive fields of the output neurons, which were oriented edge filters of arbitrary orientation. He derived an energy function to evaluate how the different orientations would be arranged in the output layer due to lateral interactions. The only variables of this energy function were the orientations of the receptive fields, an abstraction from the connectivity. Similar models were proposed earlier in (Swindale, 1980), though not derived from a receptive-field model, and more recently in (Tanaka, 1991). These approaches and their relationships to our constrained optimization framework need to be investigated more systematically.

A neural map formation model of (Amari, 1980) could not be formulated within the constrained optimization framework presented here (cf. Sec. 6.2). The weight growth in this model is limited by weight decay rather than explicit normalization rules, which is possible because the blob dynamics provides only limited correlation values even if the weights would grow large. This model is particularly elegant with respect to the way it indirectly introduces constraints and should be investigated further. Our discussion in Section 6.3 indicates that the system L+Q might also show map expansion and weight limitation without any explicit constraints, but further analysis is needed to confirm this.

The objective functions listed in Table 1 have a tendency to produce either collapsing or expanding maps. It is unlikely that the terms can be counterbalanced such that they have the tendency to preserve distances directly, independent of normalization rules and the size of the layers, as does the algorithmic objective function in Equation (50). A solution to this problem might be found by examining propagating activity patterns in the input as well as the output layer, such as traveling waves (Triesch, 1995) or running blobs (Wiskott & von der Malsburg, 1996). Waves and blobs of activity have been observed in the developing retina (Meister et al., 1991). If the waves or blobs have the same intrinsic velocity in the two layers, they would tend to generate metric maps, regardless of the scaling factor induced by the normalization rules. It would be interesting to investigate this idea further and to derive correlations for this class of models.

Another limitation of the framework discussed here is that it is confined to second-order correlations. As von der Malsburg (1995) has pointed out, this is appropriate only for a subset of phenomena of neural map formation, such as retinotopy and ocular dominance. Although orientation tuning can arise by spontaneous symmetry breaking (e.g. Linsker, 1986), a full understanding of the self-organization of orientation selectivity and other phenomena may require taking higher-order correlations into account. It would be interesting as a next step to consider third-order terms in the objective function and the conditions under which they can be derived from detailed neural dynamics. There may also be an interesting relationship to recent advances in algorithms for independent component analysis (Bell & Sejnowski, 1995), which can be derived from a maximum entropy method and is dominated by higher-order correlations.

Finally, it may be interesting to investigate the extent to which the techniques used in the analysis presented here can be applied to other types of neural dynamics, such as learning rules. The existence of objective functions for dynamics with curl may make it possible to formulate more learning rules within the constrained optimization framework, which could lead to new insights. Optimizing the dynamics of a learning rule without changing the set of stable fixed points may be an interesting application for coordinate transformations.

Appendix A Probabilistic Blob Model

A.1 Noise Model

Consider the activity model of Obermayer et al. (1990) as an abstraction of the neural activity dynamics in Section 2.1 (Eqs. 1, 2). Obermayer et al. use a high-dimensional version of the self-organizing map algorithm (Kohonen, 1982). A blob $B_{\rho'\rho_0}$ is located at a random position ρ_0 in the input layer and the input $i_{\tau'}(\rho_0)$ received by the output neurons is calculated as in Equation (7). A blob $\bar{B}_{\tau'\tau_0}$ in the output layer is located at the position τ_0 of highest input, i.e. $i_{\tau_0}(\rho_0) = \max_{\tau'} i_{\tau'}(\rho_0)$. Only the latter step differs in its outcome from the dynamics in Section 2, the maximal input instead of the maximal overlap determining the location of the output blob.

The transition to the probabilistic blob location can be done by assuming that the blob $\bar{B}_{\tau'\tau_0}$ in the output layer is located at τ_0 with probability

$$p(\tau_0|\rho_0) = i_{\tau_0}(\rho_0) = \sum_{\rho'} w_{\tau_0\rho'} B_{\rho'\rho_0} \ . \tag{61}$$

For the following considerations the same normalization assumptions as in Sections 2.1 are made, which leads to $\sum_{\tau'} i_{\tau'}(\rho_0) = 1$ and $\sum_{\tau_0} p(\tau_0|\rho_0) = 1$ and justifies the interpretation of $p(\tau_0|\rho_0)$ as a probability. The effect of different normalization rules, like those used by Obermayer et al. (1990), is discussed in the next section. The probabilistic blob location can be achieved by multiplicative noise η_τ with the cumulative density function $f(\eta) = \exp(-1/\eta)$, which leads to a modified input $l_\tau = \eta_\tau i_\tau$ with a cumulative density function

$$f_\tau(l_\tau) = \exp\left(-\frac{i_\tau(\rho_0)}{l_\tau}\right) , \tag{62}$$

and a probability density function

$$p_\tau(l_\tau) = \frac{\partial f_\tau}{\partial l_\tau} = \frac{i_\tau(\rho_0)}{l_\tau^2} \exp\left(-\frac{i_\tau(\rho_0)}{l_\tau}\right) . \tag{63}$$

Notice that the noise is different for each output neuron but always from the same distribution. The probability of neuron τ_0 having larger input l_{τ_0} than all other neurons τ', i.e. the probability of the output blob being located at τ_0, is

$$\begin{aligned}
p(\tau_0|\rho_0) &= p(l_{\tau_0} > l_{\tau'} \ \forall \tau' \neq \tau_0) && (64)\\
&= \int_0^\infty p_{\tau_0}(l_{\tau_0}) \prod_{\tau' \neq \tau_0} f_{\tau'}(l_{\tau_0}) \, \mathrm{d}l_{\tau_0} && (65)\\
&= \int_0^\infty \frac{i_{\tau_0}(\rho_0)}{l_{\tau_0}^2} \exp\left(-\frac{1}{l_{\tau_0}} \sum_{\tau'} i_{\tau'}(\rho_0)\right) \mathrm{d}l_{\tau_0} && (66)\\
&= \frac{i_{\tau_0}(\rho_0)}{\sum_{\tau'} i_{\tau'}(\rho_0)} && (67)
\end{aligned}$$

$$= \quad i_{\tau_0}(\rho_0)\ , \qquad \text{(since } \sum_{\tau'} i_{\tau'}(\rho_0) = 1) \tag{68}$$

which is the desired result. Thus, the model by OBERMAYER ET AL. (1990) can be modified by multiplicative noise to yield the probabilistic blob location behavior. A problem is that the modified input l_τ has an infinite mean value, but this can be corrected by consistently transforming the cumulative density functions by the substitution $l_\tau = k_\tau^2$ yielding

$$f_\tau(k_\tau) = \exp\left(-\frac{i_\tau(\rho_0)}{k_\tau^2}\right)\ , \tag{69}$$

for the new modified inputs k_τ, the means of which are finite. Due to the non-linear transformation $l_\tau = k_\tau^2$ the modified inputs k_τ are no longer a product of the original input i_τ with noise, whose distribution is the same for all neurons, but each input i_τ generates a modified input k_τ with a non-linearly distorted version of the cumulative density function in Equation (62).

The probability for a particular combination of blob locations is

$$p(\tau_0, \rho_0) = p(\tau_0|\rho_0)p(\rho_0) = \sum_{\rho'} w_{\tau_0\rho'} B_{\rho'\rho_0}\frac{1}{R}\ , \tag{70}$$

and the correlation between two neurons defined as the average product of their activities is

$$\begin{aligned}
\langle a_\tau a_\rho \rangle &= \sum_{\tau_0\rho_0} p(\tau_0,\rho_0)\bar{B}_{\tau\tau_0} B_{\rho\rho_0} && (71)\\
&= \sum_{\tau_0\rho_0}\sum_{\rho'} w_{\tau_0\rho'} B_{\rho'\rho_0}\frac{1}{R}\bar{B}_{\tau\tau_0} B_{\rho\rho_0} && (72)\\
&= \frac{1}{R}\sum_{\tau'\rho'} \bar{B}_{\tau\tau'} w_{\tau'\rho'}\left(\sum_{\rho_0} B_{\rho'\rho_0} B_{\rho\rho_0}\right) && (73)\\
&= \frac{1}{R}\sum_{\tau'\rho'} \bar{B}_{\tau\tau'} w_{\tau'\rho'} \bar{B}_{\rho'\rho}\ , \qquad \text{with} \quad \bar{B}_{\rho'\rho} = \sum_{\rho_0} B_{\rho'\rho_0} B_{\rho\rho_0}, && (74)
\end{aligned}$$

where the brackets $\langle\cdot\rangle$ indicate the ensemble average over a large number of blob presentations. This is equivalent to Equation (13) if $\bar{B}_{\tau'\tau} = \sum_{\tau_0} B_{\tau'\tau_0} B_{\tau\tau_0}$. Thus the two probabilistic dynamics are equivalent, though the blobs in the output layer must be different.

A.2 Different Normalization Rules

The derivation of correlations in the probabilistic blob model given above assumes explicit presynaptic normalization of the form $\sum_{\tau'} w_{\tau'\rho'} = 1$. This assumption is not valid for some models that use only postsynaptic normalization (e.g. VON DER MALSBURG, 1973). The model by OBERMAYER ET AL. (1990) postsynaptically normalizes the square sum, $\sum_{\rho'} w_{\tau'\rho'}^2 = 1$, instead of the sum, which may make the applicability of the probabilistic blob model even more questionable.

To investigate the effect of these different normalization rules on the probabilistic blob model, assume that the projective (or receptive) fields of the input (or output) neurons co-develop in such a way that, at any given moment, all neurons in a layer have the same *weight histogram*. Neuron ρ, for instance, would have the weight histogram $w_{\tau'\rho}$ taken over τ' and it would be the same as those of the other neurons ρ'. Two neurons of same weight histogram have the same number of non-zero weights and the square sums over their weights differ from the sums by the same factor c, e.g. $\sum_{\tau'} w_{\tau'\rho'}^2 = c\sum_{\tau'} w_{\tau'\rho'} = 1$ for all ρ' with $c \leq 1$. The weight histogram, and with it the factor c, may change over time. For instance, if point receptive fields develop from an initial all-to-all connectivity, the histogram has a single peak at $1/T$ in the beginning and has a peak at 0 and one entry at 1 at the end of the self-organization process and $c(t)$ grows from $1/T$ up to 1, where T is the number of output neurons.

Consider first the effect of the square sum normalization under the assumption of homogeneous co-development of receptive and projective fields. The square sum normalization differs from the sum normalization by a factor $c(t)$ common to all neurons in the layer. Since the non-linear blob model is insensitive to such a factor, the derived correlations and the learning rule are off by this factor c. Since this factor is common to all weights, the trajectories of the weight dynamics are identical though the time scales differ by c between the two types of normalization.

Consider now the effect of pure postsynaptic normalization under the assumption of homogeneous co-development of receptive and projective fields. Assume a pair of blobs is located at ρ_0 and τ_0. With a linear growth rule the sum over weights originating from an input neuron would change according to

$$\dot{W}_\rho = \sum_\tau \dot{w}_{\tau\rho} = \sum_\tau B_{\tau\tau_0} B_{\rho\rho_0} = B_{\rho\rho_0} \ , \tag{75}$$

since the blob $B_{\tau\tau_0}$ is normalized to one. Averaging over all input blob positions yields an average change of

$$\langle \dot{W}_\rho \rangle = \frac{1}{R} \sum_{\rho_0} B_{\rho\rho_0} = \frac{1}{R} \ , \tag{76}$$

since we assume a homogeneous average activity in the input layer, i.e. $\sum_{\rho_0} B_{\rho\rho_0} = 1$. A similar expression follows for the postsynaptic sum:

$$\begin{aligned}
\langle \dot{W}_\tau \rangle &= \sum_{\rho_0 \tau_0} p(\tau_0, \rho_0) \sum_\rho B_{\tau\tau_0} B_{\rho\rho_0} && (77)\\
&= \sum_{\rho_0 \tau_0} \left(\frac{1}{R} \sum_{\tau'\rho'} B_{\tau'\tau_0} w_{\tau'\rho'} B_{\rho'\rho_0} \right) \sum_\rho B_{\tau\tau_0} B_{\rho\rho_0} && (78)\\
&= \frac{1}{R} \sum_{\tau_0} B_{\tau\tau_0} \sum_{\tau'} B_{\tau'\tau_0} \sum_{\rho'} w_{\tau'\rho'} \sum_{\rho_0} B_{\rho'\rho_0} \sum_\rho B_{\rho\rho_0} && (79)\\
&= \frac{1}{T} \ , && (80)
\end{aligned}$$

where $\sum_{\rho'} w_{\tau'\rho'} = R/T$ is assumed due to the postsynaptic normalization rule and the blobs are normalized with respect to both of their indices. R and T are the number of neurons in the input and output layer, respectively. This equation shows that each output neuron has to normalize its sum of weights by the same amount and it has to do that by a subtractive normalization rule if the system is consistent. The amount by which each single weight $w_{\tau\rho}$ is changed depends on the number of non-zero weights an output neuron receives. Since we assume the weight histograms are the same, each output neuron has the same number of non-zero weights and each weight gets corrected by the same amount. Since we also assume same weight histograms for the projective fields, the sum over all weights originating from an input neuron is corrected by the same amount for each input neuron, namely by $1/R$ per time unit. Thus the postsynaptic normalization rule preserves presynaptic normalization.

It can even be argued that a postsynaptic normalization rule stabilizes presynaptic normalization. Assume an input neuron has a larger (or smaller) sum over its weights than the other input neurons. Then this neuron is likely to have more (fewer) non-zero weights than the other input neurons. This results in a larger (smaller) negative compensation by the postsynaptic normalization rule, since each weight is corrected by the same amount. This then reduces the difference between the input neuron under consideration and the others. It is important to notice that this effect of stabilizing the presynaptic normalization is not preserved in the constrained optimization formulation. It may be necessary to use explicit presynaptic normalization in the constrained optimization formulation to account for the implicit presynaptic normalization in the blob model.

If the postsynaptic constraint is based on the square sum, then the normalization rule is multiplicative and the projective fields of the input neurons need not have the same weight histograms. The system would still preserve the presynaptic normalization. Notice that the derivation given above does not hold for a non-linear Hebbian rule, e.g. $\dot{w}_{\tau\rho} = w_{\tau\rho} a_\tau a_\rho$.

These considerations show that the probabilistic blob model may be a good approximation even if the constraints are based on the square sum instead of the sum and if only the postsynaptic neurons are constrained and not the presynaptic neurons, as was required in the derivation of the probabilistic blob model above. The homogeneous co-development of receptive and projective fields is probably a reasonable assumption for high-dimensional models with a homogeneous architecture. For low-dimensional models, such as the low-dimensional self-organizing map algorithm (Kohonen, 1982), the assumption is less likely to be valid. However, numerical simulations or more detailed analytical considerations are needed to verify the assumption for any given concrete model.

Acknowledgment

We are grateful to Geoffrey J. Goodhill, Thomas Maurer, Jozsef Fiser, and two anonymous referees for carefully reading the manuscript and useful comments. Laurenz Wiskott has been supported by a Feodor-Lynen fellowship by the Alexander von Humboldt-Foundation, Bonn, Germany.

References

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybern.*, 27:77–87. 5

Amari, S. (1980). Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42:339–364. 2, 16, 25

Bauer, H.-U., Brockmann, D., and Geisel, T. (1997). Analysis of ocular dominance pattern formation in a high-dimensional self-organizing-map model. *Network: Computation in Neural Systems*, 8(1):17–33. 23

Behrmann, K. (1993). Leistungsuntersuchungen des "Dynamischen Link-Matchings" und Vergleich mit dem Kohonen-Algorithmus. Diploma thesis, internal report IR-INI 93–05, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum. 23, 24

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159. 26

Bienenstock, E. and von der Malsburg, C. (1987). A neural network for invariant pattern recognition. *Europhysics Letters*, 4(1):121–126. 2, 3, 13, 16, 17, 18, 25

Dirac, P. A. M. (1996). *General Theory of Relativity*. Princeton landmarks in physics. Princeton University Press, 41 William Street, Princeton, NJ 08540. 9

Ermentrout, G. B. and Cowan, J. D. (1979). A mathematical theory of visual hallucination patterns. *Biological Cybernetics*, 34(3):137–150. 6

Erwin, E., Obermayer, K., and Schulten, K. (1995). Models of orientation and ocular dominance columns in the visual cortex: A critical comparison. *Neural Computation*, 7:425–468. 2

Ginzburg, I. and Sompolinsky, H. (1994). Theory of correlations in stochastic neural networks. *Physical Review E*, 50(4):3171–3191. 7

Goodhill, G. J. (1993). Topography and ocular dominance: A model exploring positive correlations. *Biol. Cybern.*, 69:109–118. 17, 18, 19

Goodhill, G. J., Finch, S., and Sejnowski, T. J. (1996). Optimizing cortical mappings. In Touretzky, D., Mozer, M., and Hasselmo, M., editors, *Proc. Advances in Neural Information Processing Systems*, volume 8, pages 330–336, Cambridge, MA. MIT Press. 22

Häussler, A. F. and von der Malsburg, C. (1983). Development of retinotopic projections — An analytical treatment. *J. Theor. Neurobiol.*, 2:47–73. 2, 8, 13, 16, 17, 18

Horst, R., Pardalos, P. M., and Thoai, N. V. (1995). *Introduction to Global Optimization*, volume 3 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands. 22

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.*, 43:59–69. 2, 26, 29

Kohonen, T. (1990). The self-organizing map. *Proc. of the IEEE*, 78(9):1464–1480. 24

Konen, W., Maurer, T., and von der Malsburg, C. (1994). A fast dynamic link matching algorithm for invariant pattern recognition. *Neural Networks*, 7(6/7):1019–1030. 2, 5

Konen, W. and von der Malsburg, C. (1993). Learning to generalize from single examples in the dynamic link architecture. *Neural Computation*, 5(5):719–735. 16, 17, 19, 20

Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311. 21, 22

Linsker, R. (1986). From basic network principles to neural architecture: Emergence of orientation columns. *Ntl. Acad. Sci. USA*, 83:8779–8783. 3, 7, 17, 18, 25

MacKay, D. J. C. and Miller, K. D. (1990). Analysis of Linsker's simulations of hebbian rules. *Neural Computation*, 2:173–187. 3, 25

Meister, M., Wong, R. O. L., Baylor, D. A., and Shatz, C. J. (1991). Synchronous bursts of action potentials in ganglion cells of the developing mammalian retina. *Science*, 252:939–943. 25

Miller, K. D. (1990). Derivation of linear Hebbian equations from nonlinear Hebbian model of synaptic plasticity. *Neural Computation*, 2:321–333. 7

Miller, K. D., Keller, J. B., and Stryker, M. P. (1989). Ocular dominance column development: Analysis and simulation. *Science*, 245:605–245. 17, 18, 19

Miller, K. D. and MacKay, D. J. C. (1994). The role of constraints in Hebbian learning. *Neural Computation*, 6:100–126. 16, 21

Nowlan, S. J. (1990). Maximum likelihood competitive learning. In Touretzky, D. S., editor, *Proc. Advances in Neural Information Processing Systems*, volume 2, pages 574–582, San Mateo, CA 94403. Morgan Kaufmann Publishers. 6, 21

Obermayer, K., Ritter, H., and Schulten, K. (1990). Large-scale simulations of self-organizing neural networks on parallel computers: Application to biological modelling. *Parallel Computing*, 14:381–404. 5, 16, 17, 18, 23, 26, 27

Ritter, H., Martinetz, T., and Schulten, K. (1991). *Neuronale Netze*. Addison-Wesley, second edition. 23

Sejnowski, T. J. (1976). On the stochastic dynamics of neuronal interaction. *Biol. Cybern.*, 22:203–211. 7

Sejnowski, T. J. (1977). Storing covariance with nonlinearly interacting neurons. *J. Math. Biology*, 4:303–321. 7

Swindale, N. V. (1980). A model for the formation of ocular domance stripes. *Proc. R. Soc. Lond. B*, 208:243–264. 25

Swindale, N. V. (1996). The development of topography in the visual cortex: A review of models. *Network: Comput. in Neural Syst.*, 7(2):161–247. 2

Tanaka, S. (1990). Theory of self-organization of cortical maps: Mathematical framework. *Neural Networks*, 3:625–640. 17, 18, 25

Tanaka, S. (1991). Theory of ocular dominance column formation. *Biol. Cybern.*, 64:263–272. 25

Triesch, J. (1995). Metrik im visuellen System. Diploma thesis, internal report IR-INI 95-05, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum. 25

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100. 2, 6, 16, 17, 27

von der Malsburg, C. (1995). Network self-organization in the ontogenesis of the mammalian visual system. In Zornetzer, S. F., Davis, J., and Lau, C., editors, *An Introduction to Neural and Electronic Networks*, pages 447–463. Academic Press, second edition. 7, 25

von der Malsburg, C. and Willshaw, D. J. (1977). How to label nerve cells so that they can interconnect in an ordered fashion. *Proc. Natl. Acad. Sci. (USA)*, 74:5176–5178. 2

von der Malsburg, C. and Willshaw, D. J. (1981). Differential equations for the development of topological nerve fibre projections. *SIAM-AMS Proceedings*, 13:39–47. 13, 21

Whitelaw, D. J. and Cowan, J. D. (1981). Specificity and plasticity of retinotectal connections: A computational model. *J. Neuroscience*, 1(12):1369–1387. 16, 17

Willshaw, D. J. and von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proc. R. Soc. London*, B194:431–445. 4

Wiskott, L. and von der Malsburg, C. (1996). Face recognition by dynamic link matching. In Sirosh, J., Miikkulainen, R., and Choe, Y., editors, *Lateral Interactions in the Cortex: Structure and Function*, chapter 11. The UTCS Neural Networks Research Group, Austin, TX, http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/. Electronic book, ISBN 0-9647060-0-8. 11, 25