Biol. Cybern. 81, 211–225 (1999)

Biological Cybernetics

Emergence of symmetric, modular, and reciprocal connections in recurrent networks with Hebbian learning

Sherwin E. Hua, James C. Houk, Ferdinando A. Mussa-Ivaldi

Department of Physiology M211, Northwestern University Medical School, 303 East Chicago Avenue, Chicago, IL 60611 3008, USA

Received: 27 April 1998 / Accepted in revised form: 16 March 1999

Abstract. While learning and development are well characterized in feedforward networks, these features are more difficult to analyze in recurrent networks due to the increased complexity of dual dynamics – the rapid dynamics arising from activation states and the slow dynamics arising from learning or developmental plasticity. We present analytical and numerical results that consider dual dynamics in a recurrent network undergoing Hebbian learning with either constant weight decay or weight normalization. Starting from initially random connections, the recurrent network develops symmetric or near-symmetric connections through Hebbian learning. Reciprocity and modularity arise naturally through correlations in the activation states. Additionally, weight normalization may be better than constant weight decay for the development of multiple attractor states that allow a diverse representation of the inputs. These results suggest a natural mechanism by which synaptic plasticity in recurrent networks such as cortical and brainstem premotor circuits could enhance neural computation and the generation of motor programs.

Key words: Hebbian learning rule – attractor dynamics – symmetric connections – multiplicative normalization – self-organization – stability

1 Introduction

Recurrent networks that have stable equilibrium points, called point attractors, can perform associative memory tasks and are able to find approximate solutions to important classes of computational problems (Hopfield and Tank 1985). Such networks are often proposed as models of information processing in the cerebral cortex, since this structure is characterized by multiple recurrent pathways (Anderson et al. 1977; Ballard 1986; Rumelhart et al. 1986; Goldman-Rakic 1988; Amit 1995). Recurrent networks are also considered to be important in the generation of distributed motor commands in premotor nuclei of the hindbrain and motor cortex (Allen and Tsukahara 1974; Eisenman et al. 1991; Houk et al. 1993).

Although attractor properties concern the dynamical behavior of activation states, recurrent networks that learn involve a second dynamical system, namely that of learning or the modification of synaptic efficacy. Dual dynamics make the study of learning in recurrent networks quite complex. For this reason, earlier studies have implemented dual dynamics separately in two phases during simulations, i.e. a learning phase and an activation phase (Hopfield 1982; Parisi 1986; Shinomoto 1987). In contrast, when both dynamics evolve concurrently in an initially asymmetric network, different results may occur (Dong and Hopfield 1992).

A second issue in learning attractor networks is the method used to stabilize learning. Hebbian learning is generally unstable because it only specifies a weight increment. Most models of learning attractor networks include a constant weight decay to achieve a balance between the increment and decrement of synaptic strengths (Shinomoto 1987; Dong and Hopfield 1992). This approach is problematic because weights decay to zero if no learning occurs, unless a weight refresh mechanism is implemented (Amit and Brunel 1995). In contrast, most models of Hebbian learning in feedforward networks utilize weight normalization, a type of weight decay in which weights compete for limited resources (Whitelaw and Cowan 1981; Miller et al. 1989; Miller 1994). These models can simulate the development of feedforward pathways such as topographic maps, ocular dominance columns, and orientation domains. Furthermore, forms of weight normalization exist biologically at the level of nuclei (Hayes and Meyer 1988a,b) and at the level of neurons (Markram and Tsodyks 1996).

In this study, we compare learning and development in recurrent networks under either constant weight decay or weight normalization. Both learning and activation dynamics are allowed to evolve simultaneously. Our simulations show that weight normalization provides an

Correspondence to: F.A. Mussa-Ivaldi
(e-mail: sandro@nwu.edu,
Fax: +-312-5035101, Tel.: +-312-5035173)

effective means of learning in recurrent networks with dual dynamics, and normalization may be a more plausible mechanism to stabilize Hebbian learning than a constant decay.

In particular, this work focuses on the formation of three distinctive features of recurrent networks, namely symmetry, modularity, and reciprocity. Symmetric connections have been previously thought to be too strict a condition to be biologically plausible (see Schürmann et al. 1990). In contrast, modularity has been widely demonstrated in the central nervous system (CNS) in both sensory and motor systems (Leise 1990; Lund et al. 1993; Malach 1994). Reciprocity of connections on a neuronal level has been postulated but has been difficult to demonstrate by current experimental methods (Ma and Juliano 1991; Ghosh et al. 1994; Keifer 1996). All three of these patterns of connectivity have functional implications for learning and information processing in the motor system. Our analytic and numerical results predict that all three patterns develop naturally under Hebbian learning with plausible forms of weight decrement.

2 Hebbian learning in a recurrent network, formulation and model

2.1 Formulation

In this investigation, we study the development of recurrent connections while allowing both activation and learning dynamics to evolve simultaneously. We assume that the dynamics of the activation states are governed by the additive model,

$$C\frac{d}{dt}\mathbf{x}(t) = -\frac{1}{R}\mathbf{x}(t) + \mathbf{W}(t)\sigma[\mathbf{x}(t)] + \mathbf{b}(t) \ . \tag{1}$$

$\mathbf{x}$ is a vector of real values representing the instantaneous activation states, $\mathbf{W}$ is the weight matrix composed of real valued elements W_{ji}, t is the time variable, RC is the time constant for activation dynamics, and $\mathbf{b}$ represents the time-varying inputs that may arise from external inputs or from spontaneous activity similar to that seen in the developing CNS (Cook 1991). $\sigma[\ ]$ is the activation function whose value is comparable to the firing rate, and $\sigma[\mathbf{x}(t)] = (\sigma[x_1(t)], \ldots, \sigma[x_N(t)])$ is a column vector. For simplicity, we use the convention that small bold letters represent vectors while capital bold letters represent matrices.

Learning in our model occurs as a function of the time-varying instantaneous firing rate by the continuous Hebb rule,

$$\tau_W \frac{d}{dt}\mathbf{W}(t) = \alpha_0 \sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)] \ , \tag{2}$$

where α_0 is the learning rate, τ_W is the time constant for continuous learning. Although both learning and activation dynamics evolve simultaneously, the dynamics for weight update are slower than that for changes in the instantaneous firing rate, i.e. $RC \ll \tau_W$.

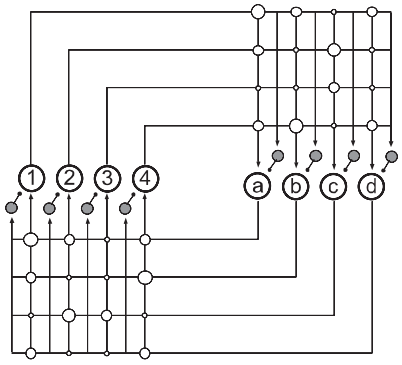

Fig. 1. Schematic diagram of the two-layered learning recurrent network. Excitatory projection neurons project to all neurons on the next layer. Inhibitory interneurons on each layer receive fixed excitatory connections from all projection neurons on the previous layer equal in magnitude to the mean of all excitatory connections to projection neurons. Interneurons inhibit a single projection neuron with fast dynamics and with weight -1

The analytical results on Hebbian learning presented here do not depend on network architecture and are not limited to units having additive activation dynamics. The learning results apply to a general recurrent network. For the sake of the computational simulations, we have used a network with additive dynamics and a specific two-layered recurrent network architecture as described in Fig. 1 and in Sect. 4.2.

2.2 Definitions

It is important to define the widely used terms, symmetry, reciprocity, and modularity on the neuronal as well as the network level.

Definition 1: A network is said to be symmetric when for every connection from neuron A to neuron B in the network, (1) there exists a connection from neuron B back to neuron A and (2) this connection from B to A has the same strength or weight as the connection from A to B, i.e. $\mathbf{W} = \mathbf{W}^{\mathrm{T}}$. The degree of asymmetry can be measured by the cosine of the angle between the vectorized weight matrix and the vectorized transpose matrix[1]; symmetry exists when the cosine equals one.

Definition 2: A unit is said to possess a reciprocal or reentrant pathway if a set of connections exists such that activity initiated in a unit will return to that unit. The returning activity can be excitatory or inhibitory; however, for simplicity we will limit our discussion to excitatory reciprocal feedback. In addition, we can define the order of reciprocal connections. If n synapses are passed before the activity returns to the unit, then the unit possesses an nth-order reciprocal connection. A first order reciprocal connection is a self connection.

[1] we are referring to $\cos\theta = \frac{(\mathrm{vec}\ \mathbf{W})^{\mathrm{T}}(\mathrm{vec}\ \mathbf{W}^{\mathrm{T}})}{\|\mathrm{vec}\ \mathbf{W}\|\|\mathrm{vec}\ \mathbf{W}^{\mathrm{T}}\|}$, where the vec operation transforms a $m \times n$ matrix into a $mn \times 1$ column vector by stacking the columns of the matrix one underneath the other.

In a linear network, i.e. when $\sigma(x)$ is a linear function of x, first-order excitatory reciprocity exists when diagonal terms of the weight matrix are greater than zero. Similarly, nth-order positive reciprocity exists when the diagonal terms of the nth order composite matrix are positive. An nth order composite matrix is the product of the weight matrix multiplied by itself n times, or $\mathbf{W}^n$. In nonlinear networks, positive diagonal terms of the composite matrix do not guarantee that activity will return to a unit but instead provides a potential reciprocal pathway. For example, when (σx) is a threshold function, the inputs must exceed the threshold before activity is transmitted to the next unit.

Definition 3: In general, a module is a functionally or anatomically related group of units. A network is said to be modular if there exist connection pathways between units belonging to one group of units (or a module) but not between units in different groups. The most extreme example of a modular network is a network with only self connections where each unit is a module. At the opposite and trivial extreme is a fully connected network, where the whole network comprises one module.

Modularity in the strictest sense exists in a network with weight matrix $\mathbf{W}$ when there exists a permutation matrix $\mathbf{P}$ such that $\mathbf{P}^{-1}\mathbf{W}\mathbf{P}$ is block diagonal (Buckley and Harary 1990). According to this definition, modules cannot be connected with other modules. This mathematical definition is more stringent than the customary notion of modularity in which limited communication between modules is allowed. For the present study, we use the strict definition of modularity as a guide to measure degrees of modularity. Highly modular networks have weight matrices that can be transformed into nearly block diagonal form. Less modular networks have more distributed connections which cannot be transformed into a form resembling a block diagonal.

We also introduce two special types of modularity. Topographic modularity is a subset of modularity where neighboring units belong to the same module. Computational or graded modularity (Houk et al. 1993) exists when the weight matrix can be transformed into a banded diagonal matrix. In this case, not only does activity reverberate between a functionally related population of units, but divergent connections also allow activity to spread to related populations of units in a graded manner.

3 Symmetry in Hebbian learning with decay

The Hebbian learning rule, as expressed by Eq. (2), is a symmetric learning rule. Since continuous Hebbian learning as expressed by Eq. (2) is unbounded and generally unstable, it is common to include a decay term as described by

$$\frac{d}{dt}W_{ji}(t) = \alpha\sigma(x_j(t))\sigma(x_i(t)) - \rho_{ji}(t)W_{ji}(t) \ , \tag{3}$$

where $\alpha = \alpha_0/\tau_W$, and $\rho_{ji}(t) = \rho_{0,ji}(t)/\tau_W$ is the decay function and is generally greater than zero, $\rho_{ji}(t) > 0$. By standard methods for a linear first-order differential equation (shown in Appendix A.2), we can rewrite Eq. (3) in the following format,

$$W_{ji}(t) = e^{-R_{ji}(t')}\int_{t_0}^{t} e^{R_{ji}(t')}\alpha\sigma[x_j(t')]\sigma[x_i(t')]dt' + e^{-R_{ji}(t)}W_{ji}(t_0) \ , \tag{4}$$

where $R_{ji}(t') = \int_{t_0}^{t'} \rho_{ji}(t'')dt''$. Note that the existence of $W_{ji}(t)$ is contingent upon the integrability of $e^{R_{ji}(t')}\alpha\sigma[x_j(t')]\sigma[x_i(t')]dt'$, where the state function $x_j(t)$ is a solution of Eq. (1).

Theorem 1: Let $\mathbf{W}(t)$ be the weight matrix at time t of a recurrent network undergoing Hebbian learning with a decay function as described by Eq. (3). If (1) the decay function is bounded by $N \leqslant \rho_{ji}(t) \leqslant M$ with real valued $M, N > 0$, (2) the activation function $\sigma(\mathbf{x})$ is bounded, i.e. $L \leqslant \sigma(x_i(t)) \leqslant K$ with real values L and K, and (3) the decay function is symmetric, i.e. $\rho_{ji}(t) = \rho_{ij}(t)$, then

1. The weights are bounded, and as $t \to \infty$ the weights are bounded by $\alpha L^2/M \leqslant \lim_{t\to\infty} W_{ji}(t) \leqslant \alpha K^2/N$;
2. The weight matrix tends to become symmetric as $t \to \infty$, i.e. $\mathbf{W}(\infty) = \mathbf{W}^{\mathrm{T}}(\infty)$ with $\mathbf{W}(\infty) = \lim_{t\to\infty}\mathbf{W}(t)$.

The proof for Theorem 1 is shown in Appendix A.2.1. Since continuous Hebbian learning involves two coupled dynamical systems, i.e. weight dynamics and activation dynamics, it is difficult to provide an explicit solution for the weight matrix $\mathbf{W}(\infty)$. Instead, we show that regardless of the solution to the dynamical system, the weight matrix becomes symmetric. In addition, although the proof necessitates that the decay function is greater than zero, this requirement is not absolute. For short periods such as when no learning occurs, the decay term can equal zero. This point is further discussed in weight normalization.

4 Constant weight decay

4.1 Derivation

Hebbian learning is generally unstable because there are no bounds to learning (Miller and MacKay 1994). A constant exponential weight decay has been commonly used to limit the growth of weights. The Hebbian update rule with a constant weight decay is

$$\frac{d}{dt}\mathbf{W}(t) = \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)] - \gamma\mathbf{W}(t) \ , \tag{5}$$

where $\alpha = \alpha_0/\tau_W$ and $\gamma = \gamma_0/\tau_W$ (γ_0 is the decayterm, $0 < \gamma_0 < 1$). By standard methods for a linear first order differential equation (shown in Appendix A.2.2), Eq. (5) can be written as

$$\mathbf{W}(t) = \alpha\int_{t_0}^{t} e^{\gamma(t'-t)}\sigma[\mathbf{x}(t')]\sigma^{\mathrm{T}}[\mathbf{x}(t')]dt' + e^{-\gamma t}\mathbf{W}(t_0) \ . \tag{6}$$

As noted before the existence of $\mathbf{W}(t)$ is contingent upon the integrability of $e^{\gamma(t'-t)}\sigma[\mathbf{x}(t')]\sigma^{\mathrm{T}}[\mathbf{x}(t')]dt'$, where the state function $\mathbf{x}(t)$ is a solution of Eq. (1). Eq (6) shows

that starting from any initial weight matrix, the network develops symmetric weights as $t \to \infty$.

Corollary 1: Let $\mathbf{W}(t)$ be the weight matrix at time t of a recurrent network undergoing Hebbian learning with a constant weight decay, i.e. Eq. (5), with $0 < \gamma_0 < 1$. If the activation function $\sigma(\mathbf{x})$ is bounded, then

1. The weight matrix is bounded and remains bounded as $t \to \infty$;
2. The weight matrix tends to become symmetric as $t \to \infty$, i.e. $\mathbf{W}(\infty) = \mathbf{W}(\infty)^{\mathrm{T}}$ with $\mathbf{W}(\infty) = \lim_{t\to\infty} \mathbf{W}(t)$;
3. If the correlation matrix $\mathbf{C}(t) = \frac{1}{t}\int_0^t \sigma\ [\mathbf{x}(t')]\sigma^{\mathrm{T}}[(\mathbf{x}(t')]dt'$ converges to a limit as $t \to \infty$, that is if $\mathbf{C}(\infty) = \lim_{t\to\infty} \frac{1}{t}\int_0^t \sigma[\mathbf{x}(t')]\sigma^{\mathrm{T}}[\mathbf{x}(t')]dt'$ exists and is finite, then the averaged weight matrix converges to the matrix $\langle W(\infty)\rangle = \frac{\alpha_0}{\gamma_0}\mathbf{C}(\infty)$.

The proof for Corollary 1 is provided in Appendix A.2.3. Although Corollary 1 guarantees that with bounded activation the weight matrix becomes symmetric and that the weights remain bounded, it does not describe the pattern of weights that develop or how the weights are related to the set of inputs. Generally the pattern of connections that develop is difficult to derive analytically; therefore, we used numerical simulations to study these patterns.

4.2 Simulations

In the two layered model, schematized in Fig. 1 and detailed in Appendix A.1, excitatory weights were updated by Hebbian learning with a constant weight decay. In all simulations, units followed additive activation dynamics as specified by Eq. (1). Inputs to the network were supplied in the form of spatially correlated activation of random clusters of three neighboring units. This type of input gives simple correlated activity that is topographically organized. Only one cluster of units was activated by external inputs during each epoch (50 time steps), and only units on layer 1 received external inputs. Additionally, inputs were wrapped at the borders of the layer such that the first and last units were considered neighbors for the purpose of external inputs.

Since Hebbian learning has been most extensively studied in excitatory synapses, we only allow excitatory connections to adapt. To reduce the bias of fixed inhibitory connections on the plasticity in excitatory connections, inhibitory connections were uniform in strength and were equal in magnitude to the mean value of all excitatory connections.

Figure 2a and b show the weight matrix from layer 1 to 2, $\mathbf{W}_{21}$, and the transpose of the weight matrix from layer 2 to 1, $\mathbf{W}_{12}^{\mathrm{T}}$, after learning. These two matrices are identical, i.e. $\mathbf{W}_{21} = \mathbf{W}_{12}^{\mathrm{T}}$; thus, the network developed symmetric connections. Symmetry developed rapidly as shown in Fig. 3 (dashed line). Weight symmetry is also demonstrated by the global weight matrix of the network, $\mathbf{W}$, shown in Fig. 2c. The global weight matrix includes all weights in the network and represents the two layered network as a single layered network having two populations of units. Units 1–27 represent layer 1, and units 28–54 represent the second layer. Note that there are no intra-layer connections as shown by the zero block diagonal terms. Symmetric connections were formed by all simulations of Hebbian learning with

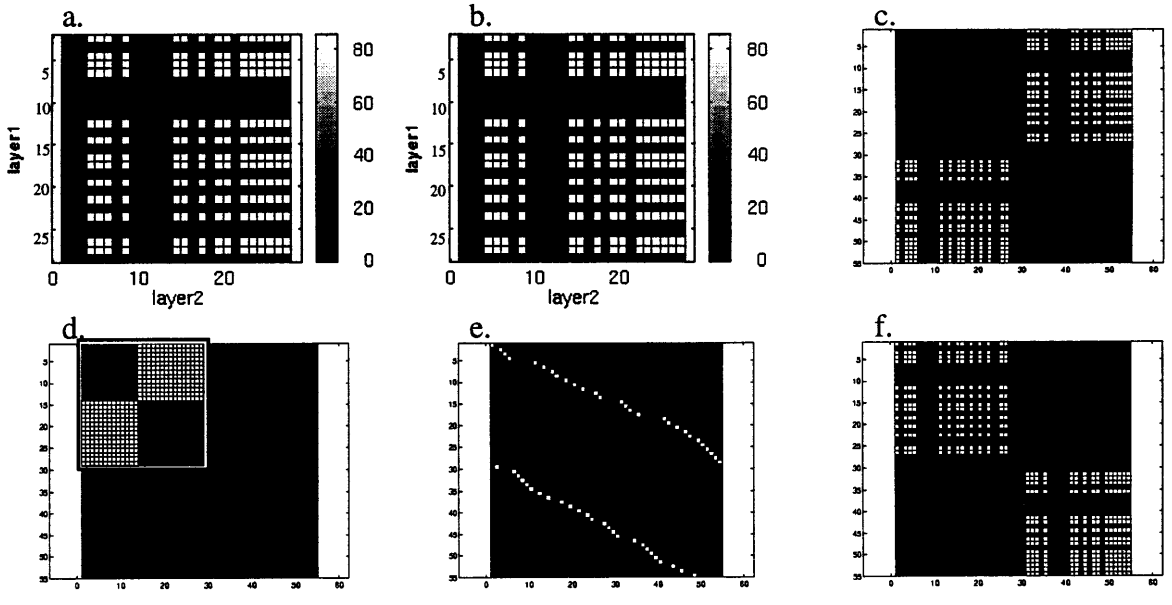

Fig. 2a–f. Weight maps for Hebbian learning with constant weight decay with $\gamma = 0.1$. The *bar* to the right hand side of **a** and **b** shows the mapping of weight values to gray levels. **a** The weight matrix $\mathbf{W}_{21}$ from layer 1 to layer 2. **b** The weight matrix $\mathbf{W}_{12}$ from layer 2 to layer 1 (transposed for direct comparison with **a**). **c** The two weight matrices can be combined into a overall weight matrix $\mathbf{W}$ showing all connections in the network. Units 1–27 represent units on layer 1, while units 28–54 represent layer 2 units. **d** The weight matrix $\mathbf{W}$ can be transformed into a matrix with a single block or module of connections by the permutation matrix shown in **e**. After permutation of the matrix $\mathbf{W}$, units 1 to 27 no longer represents units on layer 1 but instead represents a combination of layer 1 and layer 2 units. **f** The second-order composite matrix formed by the product of the overall weight matrix $\mathbf{W}$, i.e. $\mathbf{WW}$

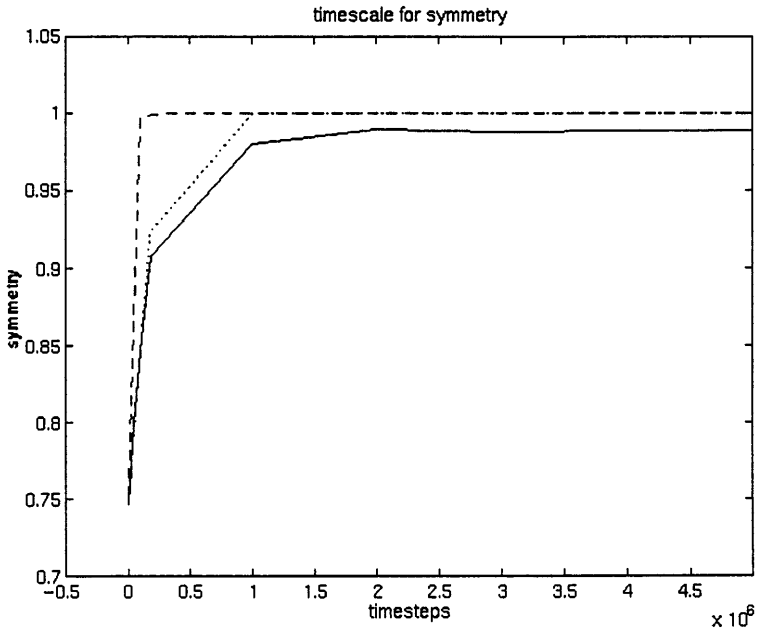

Fig. 3. The time courses for development of symmetric weights for constant weight decay (*dashed line*), dual normalization (*dotted line*), and postsynaptic normalization (*solid line*). Symmetry is calculated by the cosine of the angle between the vectorized weight matrix from layer 1 to 2 and vectorized weight matrix from layer 2 to 1

constant weight decay while varying the decay coefficient. Additionally, the values for synaptic weights remained stable, as shown in Fig. 4. Weights either decayed to zero or grew to an identical value.

The weights also evolved to become modular. Although the weights are not obviously modular, we can reorder the numbering of units to better visualize the modularity in the weights. By using the permutation matrix **P** shown in Fig. 2e, we can transform **W** into a permuted weight matrix, $\mathbf{W_P} = \mathbf{P}^{-1}\mathbf{WP}$, in which modularity is better visualized. The permuted weight matrix, $\mathbf{W_P}$, (Fig. 2d) demonstrates that the network develops only one module of interconnected units (outlined by the box). Half of the units in this module are on layer 1 while the other half belong to layer 2. Note that since the units have been renumbered, $\mathbf{W_P}$ no longer has a simple layered structure. Instead $\mathbf{W_P}$ has a modular organization in which units that belong to the same module are grouped together, regardless to which layer they belong. $\mathbf{W_P}$ merely serves as a tool to better visualize modularity inherent in the weights. A block diagonal $\mathbf{W_P}$ means that the network is modular regardless of how **P** is found and regardless of the uniqueness of **P**.

Reciprocal connections are visualized by the diagonal terms of the composite matrix. Figure 2f shows a second-order composite matrix constructed from the squared weight matrix **WW**. The composite matrix represents the projections that a unit makes to itself and to other units through multisynaptic connections. The diagonal terms of the second-order composite matrix in Fig. 2f show that each unit with non-zero weights projected to itself through disynaptic connections.

Although the weights became symmetric, modular, and reciprocal, the pattern of weights did not accurately

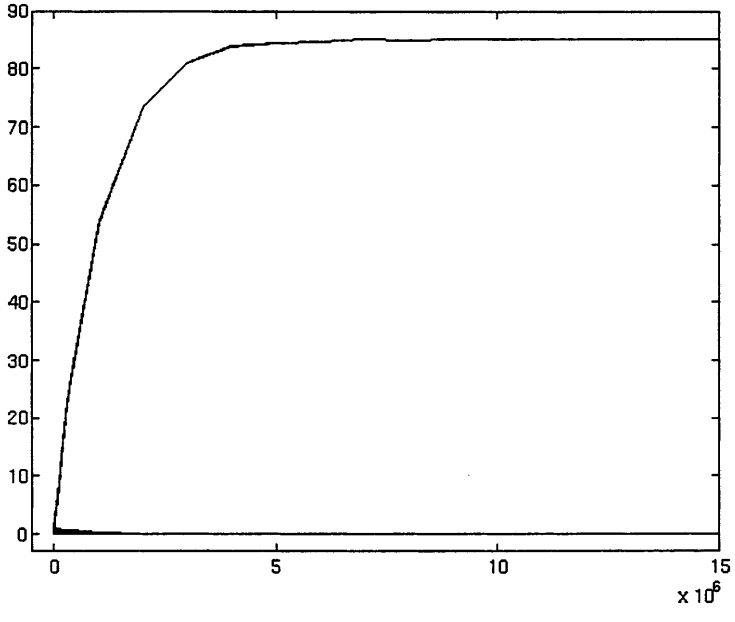

Fig. 4. Time course of weight values during a simulation of Hebbian learning with constant decay. All 2*27*27 (1,458) weights are shown. Because weights either decay to zero or assume the same identical value, only two weight-time tracings are visualized

capture the topographic nature of the inputs. In feedforward Hebbian networks, topographically correlated activity leads to the development of topographic receptive fields for cells efferent (downstream) to the correlated activity and the development of topographic projection fields for cells afferent (upstream) from the correlated activity. In our simulations, such topographic receptive and projection fields did not develop in recurrent networks. Instead a single non-topographic module involving about half of the units on each layer was formed. Thus, the 27 inputs used in the simulations could only be represented by two possible states in the network, an OFF-state and an ON-state where all units in the single module were activated.

We considered the possibility that the single module of connections resulted from high initial weights, thereby allowing internal feedback in the network to overshadow the inputs. In simulations with extremely low initial weights, though with the same pattern of random weights, the weights grew to similar magnitudes as in simulations with initially high weights. The weights in these simulations also developed a single module of weights (data not shown).

5 Concurrent presynaptic and postsynaptic normalization

5.1 Derivation

In contrast to constant weight decay, weight normalization provides a form of weight reduction in which synapses compete for limited resources (Miller and MacKay 1994). These resources may include metabolic limitations on synaptic maintenance, limitations on trophic substances, or limitations on synaptic surface area. Weight normalization limits either the total synaptic input converging onto (postsynaptic) or arising from (presynaptic) a particular neuron. Since there is evidence for both presynaptic and postsynaptic normalization in biological systems (Hayes and Meyer 1988a,b), some models using Hebbian learning have incorporated both types of normalization (Whitelaw and Cowan 1981; Hua et al. 1993).

Hebbian learning with dual normalization is defined as

$$W_{ji}(t+\Delta t) = \beta[W_{ji}(t)]_{\text{post}}\{W_{ji}(t) + \alpha_0\sigma[x_j(t)]\sigma[x_i(t)]\} \times \beta[W_{ji}(t)]_{\text{pre}} \quad , \tag{7}$$

where the coefficient for presynaptic normalization is defined by

$$\beta[W_{ji}(t)]_{\text{pre}} = \frac{\chi_{\text{pre}}}{\sum_h [W_{hi}(t) + \Delta W_{hi}(t)\Delta t]} \quad , \tag{8}$$

and the coefficient for postsynaptic normalization is defined by

$$\beta[W_{ji}(t)]_{\text{post}} = \frac{\chi_{\text{post}}}{\sum_h [W_{jh}(t) + \Delta W_{jh}(t)\Delta t]} \quad . \tag{9}$$

χ_{pre} is the limit on the sum of all weights from each presynaptic unit, χ_{post} is the limit on the sum of all weights onto each postsynaptic unit, and ΔW is a weight change obtained at time step Δt. Note that in order for both normalization conditions to be satisfied, the condition $\chi_{\text{pre}} = \chi_{\text{post}}$ must hold.[2]

Continuous Hebbian learning with dual normalization (derived in Appendix A.2.4) is specified by

$$\tau_w \frac{d\mathbf{W}(t)}{dt} = \alpha_0\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)] - \Gamma_{\text{post}}(t)\mathbf{W}(t) - \mathbf{W}(t)\Gamma_{\text{pre}}(t) \quad , \tag{10}$$

where

$$\Gamma_{\text{post}}(t) = \frac{1}{\chi}\text{diag}\left\{\alpha_0\sum_h \sigma[x_1(t)]\sigma[x_h(t)], \ldots, \alpha_0\sum_h \sigma[x_j(t)]\sigma[x_h(t)], \ldots, \alpha_0\sum_h \sigma[x_N(t)]\sigma[x_h(t)]\right\}$$

and

$$\Gamma_{\text{pre}}(t) = \frac{1}{\chi}\text{diag}\left\{\alpha_0\sum_h \sigma[x_h(t)]\sigma[x_1(t)], \ldots, \alpha_0\sum_h \sigma[x_1(t)], \ldots, \alpha_0\sum_h \sigma[x_h(t)]\sigma[x_N(t)]\right\}$$

As derived in Appendix A.2.4, $W_{ji}(t)$ can be expressed as

$$W_{ji}(t) = \alpha\int_0^t e^{R_{ji}(t')-R_{ji}(t)}\sigma[x_j(t')]\sigma[x_i(t')]dt' + e^{-R_{ji}(t)}W_{ji}(0) \quad , \tag{11}$$

where $R_{ji}(t) = \frac{\alpha}{\chi}\left(\sum_h tC_{jh}(t) + \sum_h tC_{hi}(t)\right)$, and $C_{ji}(t)$ is the correlation matrix of the activation states over time, i.e. $C_{ji}(t) = \frac{1}{t}\int \sigma[x_j(t')]\sigma[x_i(t')]dt'$. The existence of $\mathbf{W}(t)$ is contingent upon the integrability of $e^{R_{ji}(t')-R_{ji}(t)}\sigma[x_j(t')]\sigma[x_i(t')]]dt'$, where the state function $\mathbf{x}(t)$ is a solution of Eq. (1).

Corollary 2: Let $\mathbf{W}(t)$ be the weight matrix of a network which undergoes continuous Hebbian learning with concurrent presynaptic and postsynaptic normalization. If $\chi_{\text{pre}} = \chi_{\text{post}} = \chi$ and if dual normalization is implemented by Eq. (10) such that $\sum_h W_{jh}(t) = \sum_h W_{hi}(t) = \chi$ is satisfied, then $\mathbf{W}(t)$ tends toward a symmetric matrix as $t \to \infty$, i.e. $\mathbf{W}\infty = \mathbf{W}^{\mathrm{T}}(\infty)$.

Proof. By definition, weights are bounded in weight normalization. The solution to continuous dual normalization (11) shows that as $t \to \infty$, the weight matrix becomes

[2] For this work we assume that all units have the same χ_{pre} and χ_{post}. Thus, for presynaptic normalization, we have $\sum_{j=1}^N W_{ji} = \chi_{\text{pre}}$, and for postsynaptic normalization we have $\sum_{i=1}^N W_{ji} = \chi_{\text{post}}$. Since the double sums $\sum_{i=1}^N\sum_{j=1}^N W_{ji} = N\chi_{\text{pre}}$ and $\sum_{j=1}^N\sum_{i=1}^N W_{ji} = N\chi_{\text{post}}$ are the same, it follows that $\chi_{\text{post}} = \chi_{\text{pre}} = \chi$ must hold if both normalization conditions are to be satisfied.

$$\lim_{t\to\infty} W_{ji}(t) = \alpha \lim_{t\to\infty} \int_0^t e^{R_{ji}(t')-R_{ji}(t)} \sigma[x_j(t')]\sigma[x_i(t')]dt' \quad . \quad (12)$$

Since $R_{ji}(t) = \alpha/\chi\left(\sum_h tC_{jh}(t) + \sum_h tC_{hi}(t)\right)$ is symmetric, $\lim_{t\to\infty} W_{ji}(t)$ also becomes symmetric.

5.2 Simulations

In simulations of the model with dual normalization, the weights became symmetric, reciprocal, and modular. Additionally, the weights allowed a better representation of the inputs when compared with constant weight decay. The dynamics and inputs used in these simulations were the same as those used for constant decay; the only difference was the use of weight normalization instead of a constant weight decay. Weights were normalized by Eqs. (8) and (9) at all times.

Starting from initially random connections, the network developed symmetric connections as shown by identical weight matrices between the two layers shown in Fig. 5a and b, i.e. $\mathbf{W}_{21} = \mathbf{W}_{12}^{\mathrm{T}}$ and by the symmetric global weight matrix $\mathbf{W}$ shown in Fig. 5c. The time course for the development of symmetry is shown in Fig. 3 (dotted line).

The network also developed modular connections, as shown by the block diagonal permuted global matrix $\mathbf{W_P}$ in Fig. 5d. The modules are outlined and overlap with neighboring modules. Thus, the network develops graded modularity rather than strict modularity. The permutation matrix that performs the transformation $\mathbf{W_P} = \mathbf{P}^{-1}\mathbf{WP}$ is shown in Fig. 5e.

Additionally, the positive diagonal terms of the composite matrix $\mathbf{WW}$, shown in Fig. 5f, show that each unit makes second-order reciprocal connections with itself. These reciprocal connections were a robust feature of the model and were seen in all simulations. The off diagonal terms of $\mathbf{WW}$ show that units in layer 1 (units 1–16) also developed disynaptic connections with topographically neighboring units in layer 1, while units in layer 2 (units 17–32) developed disynaptic connections with non-neighboring units in layer 2. Thus, units in layer 1 belonging to the same module are organized topographically, while units on layer 2 belonging to the same module are spread out discontinuously over a large territory. This difference in development between the two layers is consistent with the difference in correlated activity in the two layers.

In contrast to constant weight decay which only allowed two values of weights, Fig. 6 shows that normalization with a small χ allowed a wide range of stable weight values to develop. Although the numerical value of weights showed minor fluctuations, the overall pattern of weights remained stable over time.

The size and number of modules were dependent on the value of the normalization constant χ. For larger χ, weights no longer obeyed topographic boundaries, as shown in Fig. 7a and b. Permutation of the global weight matrix shows that two modules developed, as shown in Fig. 7c. Two modules are demonstrated in Fig. 7c by the block diagonal segregation of the connections. These modules are larger and more distinct than the modules with a smaller χ. Additionally, diagonal terms in the composite matrix in Fig. 7d show the development of reciprocity.

These results show that weight normalization allowed the development of symmetry, modularity, and reciprocity. Multiple modules developed with weight normalization whereas, in weight decay, only a single module developed. Additionally, in weight normaliza-

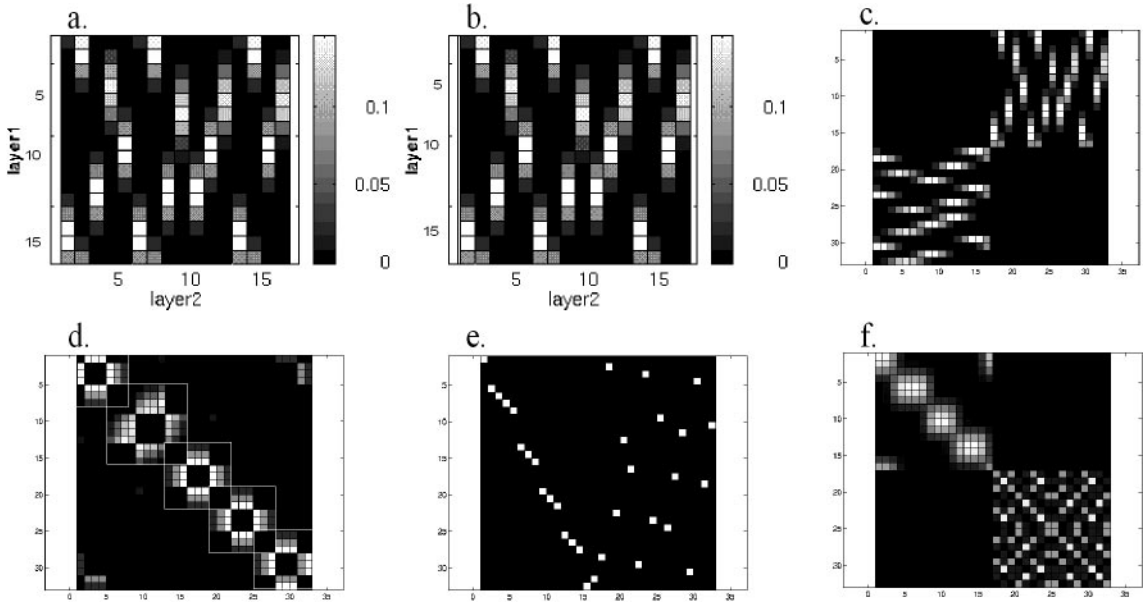

Fig. 5a–f. Weight maps and composite weight matrices for Hebbian learning with dual normalization and with $\chi = 0.5$. The *bar* to the right hand side of **a** and **b** shows the mapping of weight values to gray levels. **a** The weight matrix $\mathbf{W}_{21}$ from layer 1 to layer 2. **b** The weight matrix $\mathbf{W}_{12}$ from layer 2 to layer 1 (transposed for direct comparison with **a**). **c** The overall weight matrix $\mathbf{W}$. **d** The weight matrix $\mathbf{W}$ transformed by the permutation matrix shown in **e**. **f** The second-order composite matrix, $\mathbf{WW}$

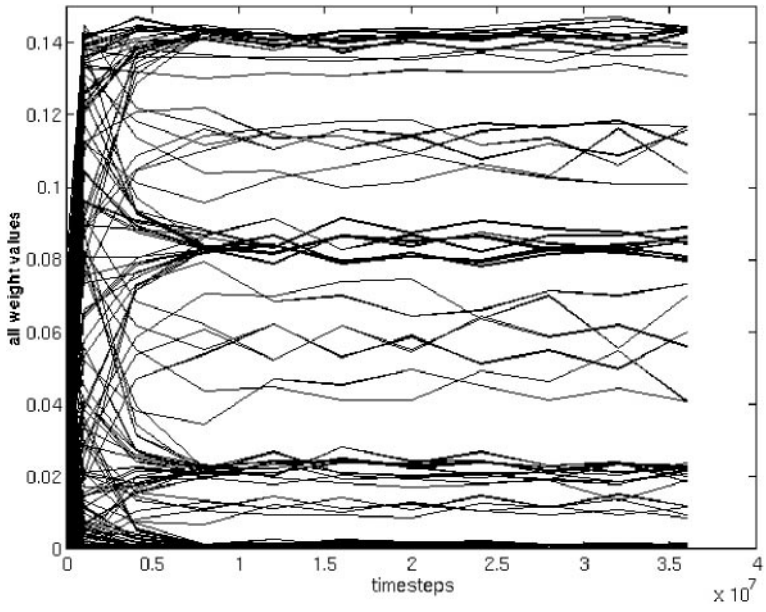

Fig. 6. Time course of weight values during a simulation of Hebbian learning with dual normalization and with $\chi = 0.5$. All weights are shown

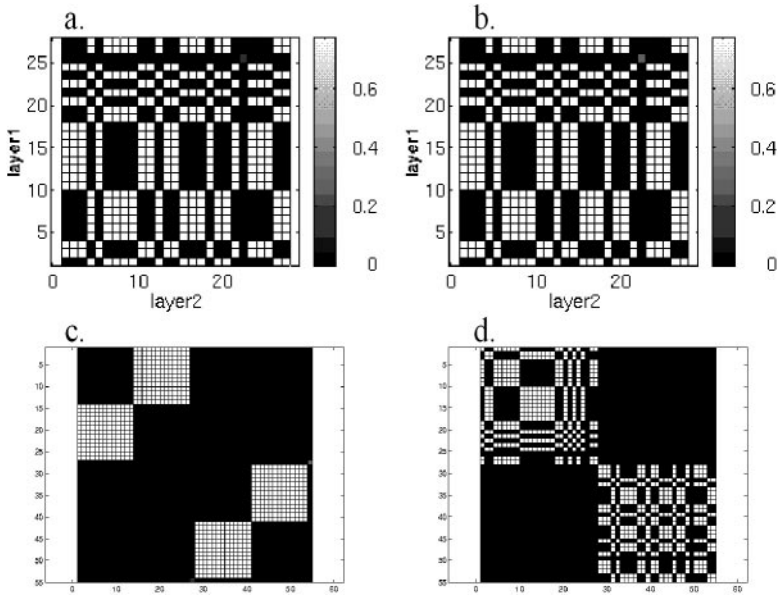

Fig. 7a–d. Weight maps and composite weight matrices for Hebbian learning with dual normalization and a high normalization constant χ. The *bar* to the right hand side of **a** and **b** shows the mapping of weight values to gray levels. **a** The weight matrix $\mathbf{W}_{21}$ from layer 1 to layer 2. **b** The weight matrix $\mathbf{W}_{12}$ from layer 2 to layer 1 (transposed for direct comparison with **a**). **c** The transformed weight matrix **W**. **d** The second-order composite matrix, **WW**

tion with a low normalization constant, the modular weights on layer 1 were organized topographically, reflecting the topographic quality of the inputs. The topographic modules were not exact images of specific inputs. Inputs consisted of overlapping clusters of three units, while modules on layer 1 were more distinct and were 4 or 5 units wide. Thus, modularity formed even though the inputs did not have inherently modular distributions. Since modules without topographic relationships developed on layer 2, which did not receive external inputs, these results suggest that modularity is an inherent quality of the network. In contrast, the development of topography is related to the external inputs as well as the size of the weights. In simulations with high χ, modularity without topography developed in both layers. High weight values effectively increased network gain, and high levels of internal feedback in the reciprocal network may overshadow the relatively smaller external inputs.

6 Postsynaptic normalization

6.1 Formulation

Some computational models of self-organization have used only presynaptic or only postsynaptic normalization. We provide the derivation of postsynaptic normalization, which is easily adaptable to the presynaptic case. The Hebbian learning rule with postsynaptic normalization (derived in Appendix A.2.5) is,

$$\tau_W \frac{dW_{ji}(t)}{dt} = \alpha_0 \sigma[x_j(t)]\sigma[x_i(t)] - \Gamma[x_j(t)]_{\text{post}} W_{ji}(t) \ . \qquad (13)$$

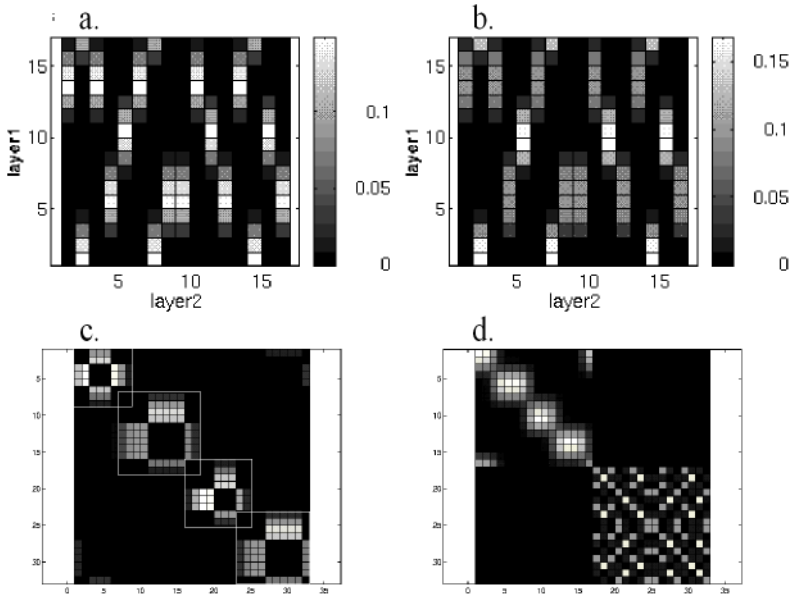

Fig. 8a–d. Weight maps and composite matrices for Hebbian learning with postsynaptic normalization with $\chi = 1.0$. The *bar* to the right hand side of **a** and **b** shows the mapping of weight values to gray levels. **a** The weight matrix $\mathbf{W}_{21}$ from layer 1 to layer 2. **b** The weight matrix $\mathbf{W}_{12}$ from layer 2 to layer 1 (transposed for direct comparison with **a**). **c** The transformed weight matrix **W**. **d** The second-order composite matrix, **WW**

Similar to the dual normalization case, we can express $W_{ji}(t)$ as

$$W_{ji}(t) = \alpha \int_0^t e^{\frac{\alpha}{\chi}\left[t' \sum_h C_{ih}(t') - t \sum_h C_{jh}(t)\right]} \sigma[x_j(t')]\sigma[x_i(t')]dt' + e^{-\frac{\alpha}{\chi}\sum_h C_{jh}(t)} W_{ji}(0) \quad . \tag{14}$$

As can be inferred from Eq. (14), Hebbian learning with postsynaptic normalization does not generally lead to symmetric connections.

6.2 Simulations

As shown in Fig. 8a and b, the weights of the connections from layer 1 to layer 2 became very similar to the weights from layer 2 to layer 1. The time course of weight development shows that weights tended towards more symmetric values during development (Fig. 3). In spite of the lack of perfect symmetry, the weights did become reciprocal and modular. Figure 8c shows that the global weight matrix can be transformed into a block diagonal matrix by a permutation matrix. Additionally, the composite matrix in Fig. 8d shows that layer 1 units formed topographic, modular, and reciprocal connections, while layer 2 units developed reciprocal and modular connections. These simulations demonstrate that the development of reciprocity and modularity are robust events that do not depend on the development of perfect symmetry.

7 Comparison of weight decay

The equations for weight dynamics in Hebbian learning with constant decay, dual, and single normalization all can be expressed in a common format, i.e.

$$W_{ji}(t) = \alpha \int_{t_0}^t e^{R_{ji}(t') - R_{ji}(t)} \sigma[x_j(t')]\sigma[x_i(t')]dt' + e^{-R_{ji}(t)} W_{ji}(t_0) \quad .$$

The weight matrix is the integral of a weight increment term $\sigma[x_j(t')]\sigma[x_i(t')]dt'$ multiplied by the corresponding decay term $e^{R_{ji}(t') - R_{ji}(t)}$. More recent weight increment terms do not undergo significant decay because $t' \approx t$ and $e^{R_{ji}(t') - R_{ji}(t)} \approx 1$. However, weight increment terms added more distant in the past do undergo significant decay because $t' \ll t$ and the corresponding decay term approaches zero. Thus, the decay term favors more recently added weight increment terms, and older weight increment terms decay to zero. By examining the time course of the decay of previously added weight increment terms, we can compare the process of weight decay for constant decay, single, and dual normalization.

In constant decay, previously added weight increment terms decay over time with an exponential time course (Fig. 9). The time course for decay is identical for all weights, and the decay of previous weight increment terms occurs even if learning does not occur. Thus, if activity in the network were quiescent and learning ceased, the weights would decay towards zero.

In contrast, weight decay in weight normalization occurs only when learning occurs. Thus, if the network is silent, weights do not decay. The decay in normalization is also specific to synapses at units which have undergone weight increment. Figure 9 shows that the decay from normalization has generally a stepwise, exponential shape. Each step of decay in postsynaptic normalization occurs when the postsynaptic unit experiences a weight increment at one of its connections. Similarly, each step of decay in dual normalization occurs when either the presynaptic or postsynaptic unit experiences a weight increment at one of its connections. Weight decay that is dependent on an increment at other synapses is termed heterosynaptic depression (Lo and Poo 1991).

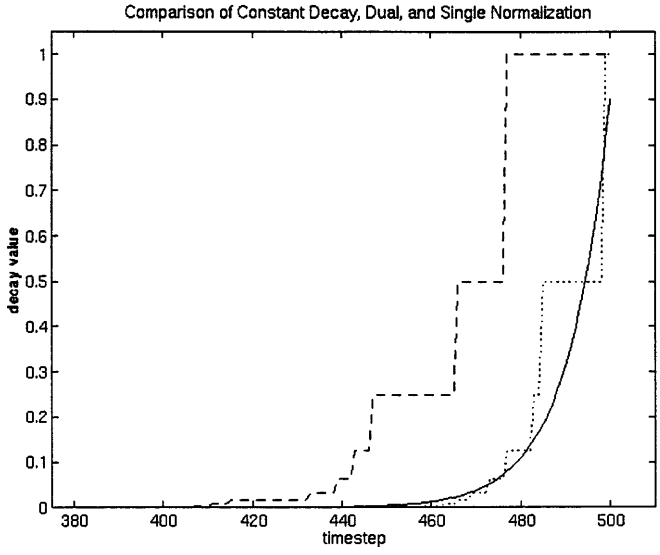

Fig. 9. A comparison of decay coefficients for constant weight decay (*solid line*), dual normalization (*dotted line*), and postsynaptic normalization (*dashed line*). Decay coefficients are simulated numerically for the term $e^{R_{ji}(t')-R_{ji}(t)}$ from Eqs. (6), (11), and (14) calculated after 500 time steps using randomly generated values for activation states

8 Discussion

In this study, we showed that symmetric, modular, and reciprocal connections developed naturally from Hebbian learning coupled with certain types of weight decay. Constant weight decay and dual normalization allowed perfectly symmetric weights to form. In postsynaptic normalization, the weights approached but did not usually attain perfect symmetry. Despite differences in symmetry formation, numerical simulations showed that all three types of decay led to the development of modular and reciprocal connectivity under Hebbian learning. Additionally, weights formed under normalization were more diversified in weight value and structure than weights formed under constant decay. The effect of this diversity on the number and variety of attractor states is beyond the scope of this paper but would be an important point for future investigations.

8.1 Symmetry

Weight symmetry is a requirement of the Lyapunov function that guarantees stability in attractor networks (Hopfield 1982; Cohen and Grossberg 1983). More recently, Amit (reviewed in 1989) has found that some asymmetry introduced into the weights as noise does not eliminate the attractor properties of the network but may in fact help attractor networks overcome local minima. Our analytic and numerical results showed that symmetric and near symmetric connections develop from Hebbian learning and various forms of weight decay. Forms of Hebbian learning have been found in various regions of the CNS, and they are likely associated with 1-methyl-D-aspartate (NMDA) channel activation. Also, experimental evidence supports the existence of weight normalization (Hayes and Meyer 1988a,b; Markram and Tsodyks 1996). Thus, symmetry should not be seen as an artificial condition instilled by physical models; instead, symmetric connections result naturally from Hebbian learning with decay in networks with recurrent connections.

The existence of symmetric weights does not in itself give a network attractor properties. Oscillations arise if the time delay in synaptic connections is long (Marcus and Westervelt 1989). The type of activation function and activation dynamics are also important. However, our analysis shows that the development of symmetric connections is independent of the activation function, provided this function is bounded. Symmetry develops because Hebbian learning is implemented mathematically by the outer-product of the vector of firing rates in the network with a symmetric decay. Our simulations used a piecewise linear activation function. In future work, we plan to implement simulations with sigmoid units and to study the attractor landscape of the networks that develop through Hebbian learning.

8.2 Dual dynamics

In palimpsest learning schemes which implement the dual dynamics into two separate phases, the network learns the set of input patterns given to the system (Mézard et al. 1986; Parisi 1986; Shinomoto 1987). In the present model with simultaneous implementation of the dual dynamics, weights did not simply reflect the set of inputs. In a recurrent network, activation states are not only determined by the inputs but also by internal feedback. If the level of feedback is low, the activation states are largely determined by the inputs. However, if the weights grow large, the feedback becomes strong and dominates the value of the activation states, overshadowing the contribution from inputs. Dong and Hopfield (1992) have found that in a recurrent network with dual dynamics and weight decay, large internal feedback allows only a single attractor to develop. However, if feedback is forced to remain weak, their network is able to learn multiple input patterns.

Our simulations used a linear-piecemeal function similar to that used in Anderson's Brain-State-In-A-Box (Anderson et al. 1977). In Anderson's model, a recurrent network with large weight values will amplify any initial activation state in the direction of the eigenvector of the weight matrix with the largest eigenvalue. The activation will grow until it reaches the bounds of the activation function. Thus, inputs given to the network will amplify in the direction of the principal eigenvector, and the weight update terms will be dominated by outer product terms of the principal eigenvector. Thus only one module develops.

In weight normalization, the weights have a hard upper limit because the sum of weights is limited. This upper limit generally prevents any one set of weights from dominating the network. Once the weight limit for these weights has been reached, other modules can develop. Thus, in contrast to constant weight decay,

multiple modules can develop with weight normalization, even if the internal feedback overshadows the input.

In both constant and normalized types of decay, if the internal feedback is sufficiently high, the modules do not take the shape of inputs presented to the network. There may be biological mechanisms to overcome this difficulty. If the inputs were inhibitory, such as the output from the cerebellum or basal ganglia, the inhibitory inputs could shape the attractor landscape even if the internal feedback is high. Additionally, in thalamocortical connections, plasticity only occurs if corticocortical connections are concurrently active (Iriki et al. 1991). This associative learning mechanism may limit overlearning of strong feedback connections in thalamocortical connections. Lastly, low coding levels in feedback activity may help networks differentiate between feedback activity and sensory inputs arriving at higher firing rates (Amit and Brunel 1995). Thus, in these models, connections are modified only when the higher firing rates provided by external inputs overcome the threshold for learning. With these mechanisms to limit overlearning, networks can shape their landscapes to accommodate different patterns of sensory inputs.

8.3 Reciprocity

Reciprocal connections are a robust feature of our model. Although reciprocity at the level of individual neurons has been difficult to verify experimentally, reciprocity between groups of neurons have been demonstrated in the CNS (Jones 1985; Goldman-Rakic 1988; Ma and Juliano 1991; Ghosh et al. 1994). Additionally, reverberatory activity has been demonstrated in premotor circuits in the brainstem (Tsukahara et al. 1983; Keifer 1996). Such reverberations are thought to result from reciprocal connections allowing activity to amplify and propagate in specific circuits (Houk et al. 1993). In contrast to symmetry, reciprocity does not require direct feedback connections between two units and can be applied to higher order recurrent networks such as long range cortico-cortical circuits as well as premotor circuits involving the cortex, brainstem, and cerebellum. Positive feedback in premotor circuits has been postulated to underlie the generation of limb motor commands (Houk et al. 1993). In previous modeling efforts, we have shown that such reciprocal circuits not only allow the generation of motor activity but also allow cerebellar learning to be exported to cortical sites (Hua and Houk 1997). The current simulations suggest that these reciprocal connections are a natural consequence of Hebbian learning.

8.4 Modularity

Modularity is widespread in the CNS on a variety of scales. In models of feedforward networks, modularity can arise as a result of correlations in the inputs (Pearson et al. 1987; Miller 1992; Miller and MacKay 1994) or correlations from lateral connections (Chernjavsky and Moody 1990). Our results show that modularity arises in recurrent networks from correlations in the activation states. We predict that correlation in activations from any number of sources including correlated inputs, lateral connections, diffusible substances such as NO, or correlated spontaneous activity will result in modular connections.

Even though modularity is widespread, its function in neural information processing has been debated (Swindale 1990; Purves et al. 1992; Malach 1994). Nelson and Bower (1990) suggest that modularity may alleviate load imbalance in parallel processing similar to parallel computing. Furthermore, Jacobs et al. (1991) proposed that modularity gives a network the ability to perform function decomposition, eliminate both spatial and temporal crosstalk, perform local generalization, and allow more efficient representation of the task to be performed.

These ideas are directly applicable to modular premotor networks in brainstem and thalamocortical areas (Houk et al. 1993). These networks receive inputs from sensory and higher premotor areas and project directly to spinal motor circuits. Houk and Wise (1995) have proposed that sensory or higher premotor inputs initiate activity in these modular attractor networks, and this activity is further shaped by cerebellar inhibition. With training, the attractor landscape is shaped to perform routine motor tasks such as sensory-motor associations (Hua and Houk 1997). Our simulation results suggest that correlated activity patterns from sensory and cerebellar inputs during different movements drive the formation of modules, each subserving an elemental motor command. Such commands could control synergistic muscles in the movement towards a particular direction or towards a particular target. In this manner, movements in three-dimensional space are decomposed into elemental movements or primitives. If movement were not decomposed, only one particular movement could be performed at a time because of spatial crosstalk. Fitting each movement into distinct modules in effect orthogonalizes the movements, thereby removing temporal and spatial interference. In this way, elemental motor commands can be programmed independently and executed simultaneously to build more complex movements and movement sequences.

Additionally, modularity also simplifies and speeds learning while reducing temporal crosstalk. Temporal crosstalk occurs when two different tasks are being learned, and learning of the second task interferes with that of the first (Jacobs et al. 1991). Since modularity divides the task to distinct groups of units, learning in one module does not interfere with learning the second module. The elimination of crosstalk is apparent in weight normalization but not in constant weight decay. In constant weight decay, all weights experience the same decay coefficient regardless of learning; however, in weight normalization, weight decay is confined to the module in which learning occurs. Thus in premotor networks, modification of different elemental movement commands can occur independently without affecting unrelated movements.

9 Conclusions

Using analytical methods and numerical simulations, we have shown that Hebbian learning with either constant decay or weight normalization leads to the natural development of symmetric, reciprocal, and modular connections. The development of symmetric or near-symmetric connections provides an important component to the development of attractor dynamics in recurrent networks. The formation of reciprocal and modular connections allow information processing to occur in specific channels while reducing both temporal and spatial crosstalk. These properties may be important in the acquisition and execution of motor commands by recurrent premotor networks.

10 Appendix

A.1 The model

Our numerical simulations consist of a two layered recurrent network with initially full and random connectivity between the two layers, schematized in Fig. 1. The differential equations that govern the activation dynamics and Hebbian learning dynamics are approximated by Euler's method. The activation update Eq. (1) is simulated by

$$x_j(t+dt) = x_j(t) + \left[-\frac{1}{R}x_j(t) + \sum_i W_{ji}(t)\sigma[x_i(t)] + b_j(t)\right]\frac{dt}{C}$$

and is updated asynchronously, while the equation for continuous weight updating is simulated by

$$W_{ji}(t+dt) = W_{ji}(t) + \left\{\alpha\sigma[x_j(t)]\sigma[x_i(t)] - \Gamma_{ji}(t)W_{ji}(t)\right\}\frac{dt}{\tau_W} \ .$$

Typically, $RC = 1,\ \tau_W = 1000,\ dt = 0.1,\ \alpha = 1$. The activation rule used in the simulations is piecewise linear:

$$\sigma(x) = \left\{\begin{array}{ll} 1 & \text{if } x \geq 1 \\ x & \text{if } 0 < x < 1 \\ 0 & \text{if } x \leqslant 0 \end{array}\right\}$$

Because all interlayer weights are excitatory, we also include a population of linear inhibitory interneurons as a general means of stabilizing the network. Each interneuron inhibits only one projection (i.e. non-interneuron) unit with a weight of -1 and receives excitatory input from all projection units on the previous layer with a weight equal in magnitude to the mean of all excitatory weights between projection units. We assume that inhibitory interneurons have fast dynamics such that the time delay through the interneurons is negligible compared with the direct excitatory connections.

In addition to internal feedback activity, external activity is also supplied to the network. External activity is spatially correlated and supplied to a cluster of three neighboring units, $b_{i-1},\ b_i,\ b_{i+1}$. Input to the center unit of a cluster equals 1.0 while that to the side units equals 0.5. The clustered inputs are chosen randomly and are held constant for 50 time steps. All activations are reset to zero before the next set of inputs are activated; this resetting of activations can be achieved biologically by cellular fatigue or some active inhibitory process that shuts off the activations. Initial weights are randomized between 0 and 1, and when weight normalization is used, the initially random weights are normalized before simulations are begun. Noise is added to the weights at each time step at $\pm 10\%$ of the maximum weight value. This noise represents spontaneous synaptic sprouting and elimination.

A.2 Analysis of Hebbian learning

In order to derive the dynamical equations for Hebbian learning, we first review the standard solution for a linear first-order differential equation. Since the activation and learning dynamics are coupled, we cannot find an explicit solution to Hebbian learning in recurrent networks using the standard solution. However, we can use this solution to express the dual dynamical system in a format that permits the analysis of stability and symmetry. The standard solution for a linear first-order differential equation is given by the following theorem (Braun 1983):

On a given interval (a, b), let $f(x)$ and $r(x)$ be continuous, and let $a < x_0 < b$. Then equation $y' + ry = f$ has one and only one solution such that $y(x_0) = y_0$ and that solution is

$$e^{R(x)}y(x) = \int_{x_0}^{x} e^{R(t)}f(t)dt + y_0 \text{ where}$$

$$R(x) = \int_{x_0}^{x} r(s)ds \ .$$

A.2.1 Proof of theorem 1

We consider the neural network with learning dynamics as specified by Eq. (3)

$$\frac{dW_{ji}(t)}{dt} = \alpha\sigma(x_j(t))\sigma(x_i(t)) - \rho_{ji}(t)W_{ji}(t) \ , \tag{3}$$

with activation dynamics specified by a second dynamical equation such as Eq. (1). By a simple transformation similar to that used for the standard solution of a linear first order differential equation, Eq. (3) is equivalent to

$$W_{ji}(t) = e^{-R_{ji}(t)}\int_{t_0}^{t} e^{R_{ji}(t')}\alpha\sigma\left[x_j(t')\right]\sigma[x_i(t')]dt' + e^{-R_{ji}(t)}W_{ji}(t_0) \ , \tag{4}$$

where $R_{ji}(t') = \int_{t_0}^{t'}\rho_{ji}(t'')dt''$ and $\rho_{ji}(t) > 0$. Because this neural network is specified by both learning and activation dynamics, Eq. (4) is not a solution to the dynamical system. However, by expressing the learning dynamics in this form, we can show that the weights are bounded and that the weights become symmetric regardless of the behavior of the activation dynamics.

To show that the weight value $W_{ji}(t)$ is bounded, we evaluate Eq. (4) at the bounds of the activation function, i.e. $L \leqslant \sigma[x_i(t)] \leqslant K$. At the upper bounds of the activation function, we have

$$w_{ji}(t)_{\max} = \alpha K^2 e^{-\int_{t_0}^{t}\rho_{ji}(t'')dt''}\int_{t_0}^{t} e^{\int_{t_0}^{t'}\rho_{ji}(t'')dt''}dt' + e^{-\int_{t_0}^{t}\rho_{ji}(t'')dt''}W_{ji}(t_0) \ .$$

Similarly, at the lower bounds, we have

$$W_{ji}(t)_{\min} = \alpha L^2 e^{-\int_{t_0}^{t'}\rho_{ji}(t'')dt''}\int_{t_0}^{t} e^{\int_{t_0}^{t'}\rho_{ji}(t'')dt''}dt' + e^{-\int_{t_0}^{t}\rho_{ji}(t'')dt''}W_{ji}(t_0) \ .$$

If the decay function is bounded by $N \leqslant \rho_{ji}(t) \leqslant M$ with $M, N > 0$, we see that $W_{ji}(t)$ is bounded by

$$\alpha L^2 e^{-\int_{t_0}^{t}Mdt''}\int_{t_0}^{t} e^{\int_{t_0}^{t''}Mdt''}dt' + e^{-\int^{t}Mdt''}{}_{t_0}W_{ji}(t_0)$$
$$\leqslant W_{ji}(t)\alpha K^2 e^{-\int_{t_0}^{t}Ndt''}\int_{t_0}^{t} e^{\int_{t_0}^{t''}Ndt''}dt' + e^{-\int_{t_0}^{t}Ndt''}W_{ji}(t_0)$$

and

$$\frac{\alpha L^2}{M} - \frac{\alpha L^2 e^{Mt_0}}{Me^{Mt}} + \frac{e^{Mt_0}}{e^{Mt}}W_{ji}(t_0)$$
$$\leqslant W_{ji}(t) \leqslant \frac{\alpha K^2}{N} - \frac{\alpha K^2 e^{Nt_0}}{Ne^{Nt}} + \frac{e^{Nt_0}}{e^{Nt}}W_{ji}(t_0) \ .$$

In the limit as $t \to \infty$, $W_{ji}(t)$ is bounded by

$$\frac{\alpha L^2}{M} \leqslant \lim_{t\to\infty} W_{ji}(t) \leqslant \frac{\alpha K^2}{N} \ .$$

Let us prove that symmetric weights are a natural consequence of the learning algorithm. From Eq. (4) we see that in the limit as $t \to \infty$

$$\lim_{t\to\infty} W_{ji}(t) = \lim_{t\to\infty} e^{-R_{ji}(t)} \int_{t_0}^{t} e^{R_{ji}(t')} \alpha\sigma[x_j(t')]\sigma[x_i(t')]dt' \ , \tag{15}$$

again with $R_{ji}(t') = \int_{t_0}^{t'} \rho_{ji}(t'')\ dt''$ and $\rho_{ji}(t) > 0$.

From Eq. (15) we see that if the decay function is symmetric, i.e. $\rho_{ji}(t) = \rho_{ij}(t)$, then the weight values become symmetric as $t \to \infty$, $\lim_{t\to\infty} W_{ji}(t) = \lim_{t\to\infty} W_{ij}(t)$.

A.2.2 Constant weight decay

In simple weight decay, we have

$$\frac{dW_{ji}(t)}{dt} = \alpha\sigma[x_j(t)]\sigma[x_i(t)] - \gamma W_{ji}(t) \ ,$$

therefore $\rho_{ji}(t) = \gamma$ and $R_{ji}(t') = \int_{t_0}^{t'} \gamma\ dt'' = \gamma(t' - t_0)$. With $t_0 = 0$,

$$W_{ji}(t) = \alpha \int_0^t \sigma\big[x_j(t')\big]\sigma[x_i(t')]e^{\gamma(t'-t)}\, dt' + e^{-\gamma t} W_{ji}(0) \ . \tag{16}$$

A.2.3 Proof of corollary 1

As proved in A.2.1, the weights in Hebbian learning with constant decay are bounded, and as $t \to \infty$, the weights are bounded by $\frac{\alpha}{\gamma} L^2 < W_{ji}(\infty) < \frac{\alpha}{\gamma} K^2$. From Eq. (16) we immediately see that as $t \to \infty$, $\lim_{t\to\infty} e^{-\gamma t} W_{ji}(t_0) = 0$, and $\lim_{t\to\infty} W_{ji}(t)$ is symmetric. In fact, $\mathbf{W}(\infty)$ is the finite sum of the outer products of the activation states, weighted by the function $e^{\gamma(t'-t)}$.

Alternatively, we can describe the averaged weight matrix $\langle \mathbf{W}(t) \rangle$ in terms of the correlation matrix $\mathbf{C}(t)$ of the firing rates $\sigma(\mathbf{x}(t))$. The correlation matrix is

$$\mathbf{C}(t) = \frac{1}{t}\int_0^t \sigma[\mathbf{x}(t')]\sigma\big[\mathbf{x}^{\mathrm{T}}(t')\big]dt' = \langle \sigma[\mathbf{x}(t')]\sigma[\mathbf{x}^{\mathrm{T}}(t')] \rangle_t \ .$$

Additionally, the averaged change in weight is defined as

$$\left\langle \frac{d\mathbf{W}(t')}{dt'} \right\rangle_t = \frac{1}{t}\int_0^t \frac{d\mathbf{W}(t')}{dt'} dt'' = \frac{\mathbf{W}(t) - \mathbf{W}(0)}{t}$$

which implies that if $\mathbf{W}(t)$ is bounded, then $\left\langle \frac{d\mathbf{W}(t')}{dt'} \right\rangle_t \to 0$ as $t \to \infty$.

From Eq. (5),

$$\frac{d\mathbf{W}(t)}{dt} + \gamma\mathbf{W}(t) = \alpha\sigma[\mathbf{x}(t)]\sigma\big[\mathbf{x}^{\mathrm{T}}(t)\big] \ .$$

Averaging over time, we obtain

$$\begin{aligned} \langle \mathbf{W}(t') \rangle_t &= \frac{\alpha}{\gamma}\mathbf{C}(t) - \gamma^{-1}\left\langle \frac{d\mathbf{W}(t')}{dt'} \right\rangle_t \\ &= \frac{\alpha}{\gamma}\mathbf{C}(t) - \gamma^{-1}\frac{\mathbf{W}(t) - \mathbf{W}(0)}{t} \ . \end{aligned}$$

Thus,

$$\lim_{t\to\infty} \langle \mathbf{W}(t') \rangle_t = \frac{\alpha}{\gamma} \lim_{t\to\infty} \mathbf{C}(t) \ . \tag{17}$$

Eq. (17) shows that the averaged weight becomes symmetric as the length of training, t, becomes large. In fact the averaged weight matrix approaches the correlation matrix of the activation states multiplied by the constant, $\frac{\alpha}{\gamma}$.

A.2.4 Concurrent presynaptic and postsynaptic normalization

We derive the continuous Hebbian learning rule with dual normalization from the discrete case.

$$\mathbf{W}(t + \Delta t) = \mathrm{B}[\mathbf{W}(t)]_{\mathrm{post}}\big[\mathbf{W}(t) + \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}(\mathbf{x}(t))\Delta t\big]\mathrm{B}[\mathbf{W}(t)]_{\mathrm{pre}} \tag{18}$$

where $\mathrm{B}[\mathbf{W}(t)]_{\mathrm{pre}} = \mathrm{diag}\{\beta[w_{j1}(t)]_{\mathrm{pre}}, \ldots, \beta[w_{ji}(t)]_{\mathrm{pre}}, \ldots, \beta[w_{jn(t)]_{\mathrm{pre}}}\}$. It follows that

$$\begin{aligned} &\mathrm{B}[\mathbf{W}(t)]_{\mathrm{post}}^{-1}\Big(\mathrm{I} - \mathrm{B}[\mathbf{W}(t)]_{\mathrm{post}} + \mathrm{B}[\mathbf{W}(t)]_{\mathrm{post}}\Big)\mathbf{W}(t + \Delta t) \\ &\quad \times \Big(\mathrm{I} - \mathrm{B}[\mathbf{W}(t)]_{\mathrm{pre}} + \mathrm{B}[\mathbf{W}(t)]_{\mathrm{pre}}\Big)\mathrm{B}[\mathbf{W}(t)]_{\mathrm{pre}}^{-1} \\ &\quad = \mathbf{W}(t) + \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)]\Delta t \end{aligned} \tag{19}$$

and

$$\begin{aligned} \frac{\mathbf{W}(t + \Delta t) - \mathbf{W}(t)}{\Delta t} &= \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)] - \frac{1}{\Delta t}\Gamma'_{\mathrm{post}}(t)\mathbf{W}(t + \Delta t)\Gamma'_{\mathrm{pre}}(t) \\ &\quad - \frac{1}{\Delta t}\Gamma'_{\mathrm{post}}(t)\mathbf{W}(t + \Delta t) - \frac{1}{\Delta t}\mathbf{W}(t + \Delta t)\Gamma'_{\mathrm{pre}}(t) \end{aligned} \tag{20}$$

where

$$\begin{aligned} \Gamma'_{\mathrm{post}}(t) &= \mathrm{B}[\mathbf{W}(t)]_{\mathrm{post}}^{-1}\Big(\mathrm{I} - \mathrm{B}[\mathbf{W}(t)]_{\mathrm{post}}\Big) \\ &= \frac{1}{\chi}\mathrm{diag}\Bigg[\sum_h W_{jh}(t) + \alpha\sum_h \sigma[x_1(t)]\sigma[x_h(t)]\Delta t - \chi, \ldots, \sum_h W_{jh}(t) \\ &\qquad + \alpha\sum_h \sigma\big[x_j(t)\big]\sigma[x_h(t)]\Delta t - \chi, \ldots, \sum_h W_{Nh}(t) \\ &\qquad + \alpha\sum_h \sigma[x_N(t)]\sigma[x_h(t)]\Delta t - \chi\Bigg] \end{aligned}$$

and

$$\begin{aligned} \Gamma'_{\mathrm{pre}}(t) &= \Big(\mathrm{I} - \mathrm{B}[\mathbf{W}(t)]_{\mathrm{pre}}\Big)\mathrm{B}[\mathbf{W}(t)]_{\mathrm{pre}}^{-1} \\ &= \frac{1}{\chi}\mathrm{diag}\Bigg[\sum_h W_{ht}(t) + \alpha\sum_h \sigma[x_h(t)]\sigma[x_1(t)]\Delta t - \chi, \ldots, \sum_h W_{hi}(t) \\ &\qquad + \alpha\sum_h \sigma[x_h(t)]\sigma[x_i(t)]\Delta t - \chi, \ldots, \sum_h W_{hN}(t) \\ &\qquad + \alpha\sum_h \sigma[x_h(t)]\sigma[x_N(t)]\Delta t - \chi\Bigg] \end{aligned}$$

In the numerical implementation of dual normalization, the presynaptic or postsynaptic sum of weights is approximately equal but sometimes not exactly equal to χ, i.e. $\sum_h W_{jh}(t) \approx \sum_h W_{hi}(t) \approx \chi$. In numerical simulations, we implement dual normalization sequentially and use the pre-normalized weights to sum for both normalization steps. Analytically, we assume for simplicity that dual normalization occurs simultaneously, and both normalization conditions are satisfied, i.e. $\sum_h W_{jh}(t) = \sum_h W_{hi}(t) = \chi$. Under this assumption, it follows that

$$\begin{aligned} \frac{\mathbf{W}(t + \Delta t) - \mathbf{W}(t)}{\Delta t} &= \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)] - \Gamma_{\mathrm{post}}(t)\mathbf{W}(t + \Delta t)\Gamma_{\mathrm{pre}}(t)\Delta t \\ &\quad - \Gamma_{\mathrm{post}}(t)\mathbf{W}(t + \Delta t) - \mathbf{W}(t + \Delta t)\Gamma_{\mathrm{pre}}(t) \end{aligned} \tag{21}$$

where

$$\begin{aligned} \Gamma_{\mathrm{post}}(t) = \frac{1}{\chi}\mathrm{diag}\Bigg\{ &\alpha\sum_h \sigma[x_t(t)]\sigma[x_h(t)], \ldots, \\ &\alpha\sum_h \sigma\big[x_j(t)\big]\sigma[x_h(t)], \ldots, \alpha\sum_h \sigma[x_N(t)]\sigma[x_h(t)]\Bigg\} \ , \end{aligned}$$

and

$$\Gamma_{\text{pre}}(t) = \frac{1}{\chi}\text{diag}\Big[\alpha\sum_h \sigma[x_h(t)]\sigma[x_1(t)],\ldots,$$
$$\alpha\sum_h \sigma[x_h(t)]\sigma[x_i(t)],\ldots,\alpha\sum_h \sigma[x_h(t)]\sigma[x_n(t)]\Big] \ .$$

Thus,

$$\frac{d\mathbf{W}(t)}{dt} = \lim_{\Delta t\to 0}\frac{\mathbf{W}(t+\Delta t)-\mathbf{W}(t)}{\Delta t}$$
$$= \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)] - \Gamma_{\text{post}}(t)\mathbf{W}(t) - \mathbf{W}(t)\Gamma_{\text{pre}}(t) \ , \qquad (22)$$

or

$$\frac{dW_{ji}(t)}{dt} = \alpha\sigma\big[x_j(t)\big]\sigma[x_i(t)] - \frac{\alpha}{\chi}\Big\{\sum_h \sigma\big[x_j(t)\big]\sigma[x_h(t)]$$
$$+\sum_h \sigma[x_h(t)]\sigma[x_i(t)]\Big\}W_{ji}(t) \ . \qquad (23)$$

Thus, in dual normalization, comparing Eqs. (23) and (3), we get for the decay term $\rho_{ji}(t) = \frac{\alpha}{\chi}\big\{\sum_h \sigma\big[x_j(t)\big]\sigma[x_h(t)] + \sum_h \sigma\ [x_h(t)]\ \sigma[x_i(t)]\big\}$.

If $C_{ji}(t-t_0) = (1/t-t_0)\int_{t_0}^{t}\sigma\big[x_j(t')\big]\sigma[x_i(t')]dt'$ and $t_0 = 0$, we arrive at Eq. (11)

$$W_{ji}(t) = \alpha\int_0^t e^{R_{ji}(t')-R_{ji}(t)}\sigma\big[x_j(t')\big]\sigma[x_i(t')]dt' + e^{-R_{ji}(t)}W_{ji}(0) \ ,$$

with $R_{ji}(t) = \frac{\alpha}{\chi}\big(\sum_h tC_{jh}(t) + \sum_h tC_{hi}(t)\big)$.

A.2.5 Single pre- or postsynaptic normalization

We derive the continuous Hebbian learning rule with postsynaptic normalization from the discrete case (the presynaptic case is derived similarly),

$$\mathbf{W}(t+\Delta t) = \mathrm{B}[\mathbf{W}(t)]_{\text{post}}\big\{\mathbf{W}(t) + \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)]\Delta t\big\} \ , \qquad (24)$$

where $\mathrm{B}[\mathbf{W}(t)]_{\text{post}} = \text{diag}\ \big\{\beta[W_{1i}(t)]_{\text{post}},\ldots,\ \beta\big[W_{ji}(t)\big]_{\text{post}},\ \ldots,\ \beta[W_{ni}(t)]_{\text{post}}\big\}$.

By rearranging, we get

$$\mathrm{B}[\mathbf{W}(t)]_{\text{post}}^{-1}\Big\{\mathbf{I} - \mathrm{B}[\mathbf{W}(t)]_{\text{post}} + \mathrm{B}[\mathbf{W}(t)]_{\text{post}}\Big\}\mathbf{W}(t+\Delta t)$$
$$= \mathbf{W}(t) + \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)]\Delta t \qquad (25)$$

and

$$\frac{d\mathbf{W}(t)}{dt} = \lim_{\Delta t\to 0}\frac{\mathbf{W}(t+\Delta t)-\mathbf{W}(t)}{\Delta t}$$
$$= \alpha\sigma[\mathbf{x}(t)]\sigma^{\mathrm{T}}[\mathbf{x}(t)] - \Gamma_{\text{post}}(t)\mathbf{W}(t) \ , \qquad (26)$$

where

$$\Gamma_{\text{post}}(t) = \mathrm{B}[\mathbf{W}(t)]_{\text{post}}^{-1}\Big(\mathbf{I} - \mathrm{B}[\mathbf{W}(t)]_{\text{post}}\Big)\frac{1}{\Delta t}$$
$$= \frac{\alpha}{\chi_{\text{post}}}\text{diag}\Big\{\sum_h \sigma[x_1(t)]\sigma[x_h(t)],\ldots,$$
$$\sum_h \sigma\big[x_j(t)\big]\sigma[x_h(t)],\ldots,\sum_h \sigma[x_N(t)]\sigma[x_h(t)]\Big\}$$

We can similarly express Eq. (26) as

$$\frac{dW_{ji}(t)}{dt} = \alpha\sigma\big[x_j(t)\big]\sigma[x_i(t)] - \Big\{\frac{\alpha}{\chi_{\text{post}}}\sum_h \sigma\big[x_j(t)\big]\sigma[x_h(t)]\Big\}W_{ji}(t) \ . \qquad (27)$$

Thus, in postsynaptic normalization, comparing Eqs. (27) and (3), we get for the decay term $\rho_{ji}(t) = \frac{\alpha}{\chi}\sum_h \sigma\big[x_j(t)\big]\sigma[x_h(t)]$.

If $C_{ji}(t-t_0) = \frac{1}{t-t_0}\int_{t_0}^{t}\sigma\big[x_j(t')\big]\sigma[x_i(t')]dt'$, then

$$R_{ji}(t') = \int_{t_0}^{t'}\frac{\alpha}{\chi}\sum_h \sigma\big[x_j(t'')\big]\sigma[x_h(t'')]dt''$$
$$= \frac{\alpha}{\chi}\sum_h (t'-t_0)C_{jh}(t'-t_0) \ . \qquad (28)$$

If we let $t_0 = 0$, then we arrive at equation Eq. (14)

$$W_{ji}(t) = \alpha\int_0^t e^{\frac{\alpha}{\chi}\left(t'\sum_h C_{jh}(t') - t\sum_h C_{jh}(t)\right)}$$
$$\times\ \sigma\big[x_j(t')\big]\sigma[x_i(t')]dt' + e^{\frac{\alpha}{\chi}\sum_h C_{jh}(t)}W_{ji}(0) \ .$$

In single normalization, the condition, $\sum_h W_{jh}(t) = \chi$, is satisfied at each time step.

In all three cases of A.2.2, A.2.4, and A.2.5, as $t\to\infty$, the contribution from the initial weight matrix vanishes, i.e. $\lim_{t\to\infty} e^{-R(t)}W_{ji}(0) = 0$. In Hebbian learning with a constant decay and dual normalization, $R_{ji}(t)$ is symmetric and therefore, the weight matrix becomes symmetric as $t\to\infty$, i.e. $\lim_{t\to\infty} W_{ji}(t) = \lim_{t\to\infty} W_{ij}(t)$. However, with single postsynaptic normalization, the weight matrix is not guaranteed to become symmetric because $R_{ji}(t)$ is generally not symmetric.

Acknowledgements. We thank Andrew Barto and Sara Solla for helpful discussions. This work was supported by NIH/NIMH grant 5-P50-MH48185 to James C. Houk and Ferdinando A. Mussa-Ivaldi.

References

Allen GI, Tsukahara N (1974) Cerebro-cerebellar communication systems. Psychol Rev 54:957–1006

Amit DJ (1989) Modeling brain function. Cambridge University Press, Cambridge

Amit DJ (1995) The Hebbian paradigm reintergrated: local reverberations as internal representations. Behav Brain Sci 18:617–657

Amit DJ, Brunel N (1995) Learning internal representations in an attractor neural network with analogue neurons. Network 6:359–388

Anderson JA, Silverstein JW, Ritz SA, Jones RS (1977) Distinctive features, categorical perception, and probability learning: Some applications of a neural model. Psychol Rev 84:413–451

Ballard DH (1986) Cortical connections and parallel processing: structure and function. Behav Brain Sci 9:67–120

Braun M (1983) Differential equations and their applications. Springer, Berlin Heidelberg New York

Buckley F, Harary F (1990) Distance in graphs. Addison-Wesley, Redwood City, Calif

Chernjavsky A, Moody J (1990) Spontaneous development of modularity in simple cortical models. Neural Comput 2:334–354

Cohen MA, Grossberg S (1983) Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. IEEE Trans Syst Man Cybern SMC-13:815–826

Cook JE (1991) Correlated activity in the CNS: a role on every timescale? Trends Neurosci 14:397–401

Dong DW, Hopfield JJ (1992) Dynamic properties of neural networks with adapting synapses. Network 3:267–283

Eisenman LN, Keifer J, Houk JC (1991) Positive feedback in the cerebro-cerebellar recurrent network may explain rotation of population vectors. In: Eeckman F (ed) Analysis and modeling of neural systems, Kluwer, Norwell, Mass, pp 371–376

Ghosh S, Murray GM, Turman AB, Rowe MJ (1994) Cortico-thalamic influences on transmission of tactile information in the ventro-posterolateral thalamus of the cat: effect of revers-

ible inactivation of somatosensory cortical areas I and II. Exp Brain Res 100:276–286
Goldman-Rakic PS (1988) Topography of cognition: parallel distributed networks in primate association cortex. Annu Rev Neurosci 11:137–156
Hayes WP, Meyer RL (1988a) Optic synapse number but not density is maintained during regeneration onto surgically-halved tectum in goldfish: HRP-EM evidence that optic fibers compete for fixed numbers of postsynaptic sites on tectum. J Comp Neuro 274:539–559
Hayes WP, Meyer RL (1988b) Retinotopically inappropriate synapses of subnormal density formed by surgically misdirected optic fibers in goldfish tectum. Brain Res Dev Brain Res 38:304–312
Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA 79:2554–2558
Hopfield JJ, Tank DW (1985) "Neural" computation of decisions in optimization problems. Biol Cybern 52:141–152
Houk JC, Keifer J, Barto AG (1993) Distributed motor commands in the limb premotor network. Trends Neurosci 16:27–33
Houk JC, Wise SP (1995) Distributed modular architectures linking basal ganglia, cerebellum and cerebral cortex: their role in planning and controlling action. Cereb Cortex 5:95–111
Hua SE, Houk JC (1997) Cerebellar guidance of premotor network development and sensorimotor learning. Lear Mem 4:63–76
Hua SE, Massone LL, Houk JC (1993) Model of topographic map development guided by a transiently expressed repulsion molecule. Neuroreport 4:1319–1322
Iriki A, Pavlides C, Keller A, Asanuma H (1991) Long term potentiation of thalamic input to the motor cortex induced by coactivation of thalamocortical and corticocortical afferents. J Neurophysiol 65:1435–1441
Jacobs RA, Jordan MI, Barto AG (1991) Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. Cogn Sci 15:219–250
Jones EG (1985) The thalamus. Plenum, New York
Keifer J (1996) Effects of red nucleus inactivation on burst discharge in turtle cerebellum: evidence for positive feedback. J Neurophysiol 76:2200–2210
Leise EM (1990) Modular construction of nervous systems: a basic principle of design for invertebrates and vertebrates. Brain Res Brain Res Rev 15:1–23
Lo YJ, Poo MM (1991) Activity-dependent synaptic competition in vitro: heterosynaptic suppression of developing synapses. Science 254:1019–1022
Lund JS, Yoshioka T, Levitt JB (1993) Comparison of intrinsic connectivity in different areas of Macaque monkey cerebral cortex. Cereb Cortex 3:148–162
Ma W, Juliano SL (1991) The relationship between thalamocortical connections and stimulus-evoked metabolic activity in the ventroposterior nucleus of the monkey. Somatosens Mot Res 8:77–86
Malach R (1994) Cortical columns as devices for maximizing neuronal diversity. Trends Neurosci 17:101–104
Marcus CM, Westervelt RM (1989) Stability of analog neural networks with delay. Phys Rev 39:347–359
Markram H, Tsodyks M (1996) Redistribution of synaptic efficacy between neocortical pyramidal neurons. Nature 382:807–810
Mézard M, Nadal JP, Toulouse G (1986) Solvable models of working memories. J Phys 47:1457–1462
Miller KD (1992) Development of orientation columns via competition between ON- and OFF-center inputs. Neuroreport 3:73–76
Miller KD (1994) A model for the development of simple cell receptive fields and the ordered arrangement of orientation columns through activity-dependent competition between ON- and OFF-center inputs. J Neurosci 14:409–441
Miller KD, Keller JB, Stryker MP (1989) Ocular dominance column development: analysis and simulation. Science 245:605–615
Miller KD, MacKay DJC (1994) The role of constraints in Hebbian learning. Neural Comput 6:100–126
Nelson ME, Bower JM (1990) Brain maps and parallel computers. Trends Neurosci 13:403–408
Parisi G (1986) Asymmetric neural networks and the process of learning. J Phys A: Math Gen 19:L675–L680
Pearson JC, Finkel LH, Edelman GM (1987) Plasticity in the organization of adult cerebral cortical maps: A computer simulation based on neuronal group selection. J Neurosci 7:4209–4223
Purves D, Riddle DR, LaMantia A-S (1992) Iterated patterns of brain circuitry (or how the cortex gets its spots). Trends Neurosci 15:362–368
Rumelhart DE, Hinton GE, McClelland JL (1986) A general framework for parallel distributed processing. In: Rumelhart DE, McClelland JL (eds) Parallel distributed processing, MIT Press, Cambridge, mass, pp 45–76
Schürmann B, Hollatz J, Ramacher U (1990) Adaptive recurrent neural networks and dynamic stability. In: Garrido L (ed) Statistical mechanics of neural networks Springer, Berlin Heidelberg New York, pp 49–63
Shinomoto S (1987) Memory maintenance in neural networks. J Phys A: Math Gen 20:L1305–L1309
Swindale NV (1990) Is the cerebral cortex modular? Trends Neurosci 13:487–492
Tsukahara N, Bando N, Murakami T, Oda Y (1983) Properties of cerebello-precerebellar reverberating circuits. Brain Res 274:249–259
Whitelaw VA, Cowan JD (1981) Specificity and plasticity of retinotectal connections: a computational model. J Neurosci 1:1369–1387