Network: Comput. Neural Syst. **12** (2001) 131–140 www.iop.org/Journals/ne PII: S0954-898X(01)22484-1

Steady states in an iterative model for multiplicative spike-timing-dependent plasticity

Jonathan E Rubin

Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, USA

Received 18 August 2000, in final form 2 February 2001

Abstract

Recent experimental evidence suggests that synaptic plasticity depends on the precise timing of pre- and post-synaptic activity. In this paper, an iterative model for a multiplicative form of this spike-timing-dependent plasticity (mSTDP) is introduced. This model is incorporated into a neural network with many input cells coupled via excitation to a single output cell. Analysis of this network yields a criterion for the output cell to fire on every iteration, as well as general formulae for the steady-state output firing rate and the steady-state value to which all synaptic weights are driven by mSTDP. These characterize the basic state of network operation generated by mSTDP.

1. Introduction

A basic postulate of Hebbian learning is that synaptic modification in neural networks depends on correlations in the activity of pre- and post-synaptic cells (Hebb 1949). Recent experimental results indicate that such synaptic modification, in a variety of biological neuronal networks, depends on short-timescale details of the relative timing of pre- and post-synaptic action potentials (Debanne *et al* 1996, Bell *et al* 1997, Markram *et al* 1997, Bi and Poo 1998, Zhang *et al* 1998). In particular, presynaptic spikes that precede post-synaptic firing lead to synaptic potentiation, while those that follow post-synaptic firing elicit synaptic depression, with a sharp ($\sim$5 ms) transition zone (Markram *et al* 1997, Bi and Poo 1998, Zhang *et al* 1998). Moreover, the magnitude of synaptic modification decays exponentially with the time interval between pre- and post-synaptic spikes, yielding a window for significant synaptic plasticity of only about 20 ms on either side of a post-synaptic spike. This form of synaptic modification has been labelled as spike-timing-dependent plasticity (STDP) (Song *et al* 2000).

Given these results, it is clearly desirable to understand the functional role of STDP. Previous authors (Abbott and Song 1999, Song *et al* 2000, Levy *et al* 2000) have concluded that STDP represents a mechanism for synaptic competition. This interpretation is based on numerical simulations of STDP in networks of integrate-and-fire neurons with excitatory synapses that undergo STDP and inhibitory synapses that do not. In appropriate parameter regimes, STDP was observed to drive the networks into an irregular firing regime, in which output rate depends at most weakly on input rate and individual presynaptic spikes can significantly affect timing of post-synaptic spikes. This occurred through adjustment of synaptic weights towards a bimodal distribution, in which individual synapse weights approached preset upper or lower bounds built into the system.

In these works, STDP adjusts synaptic weights by a constant step, modulated by the time between pre- and post-synaptic spikes, with fixed barriers limiting the possible ranges of synaptic strengths. However, Bi and Poo (1998) found that the percentage changes in weights due to a spike-timing-dependent potentiation event significantly depend on the weights before the event, while the percentage changes due to spike-timing-dependent depression do not. This suggests that synaptic enhancements by STDP should be scaled multiplicatively by a term representing the difference between the current synaptic strength and some maximum. Similarly, the magnitude of synaptic reduction should be given by a fixed percentage of current synaptic strength.

The aim of this paper is to theoretically determine the basic properties of STDP in a model neural network that incorporates such multiplicative scaling (see also Kistler and van Hemmen (2000), van Rossum *et al* (2000), Rubin *et al* (2001)). We introduce a discrete, iterative model for this multiplicative STDP (mSTDP) that can be treated analytically. While discretization eliminates precise spike times, our model does incorporate time steps and allows for a temporal ordering of events. This is of general interest as a representation of a discrete learning process in which short-timescale details of correlations in activity of coupled elements determine future coupling strength. This paper represents an initial step to lead to future analysis both of iterative processes with more complex input structures and of a continuous model including mSTDP.

In the model featured here, a population of N independent presynaptic cells feeds excitation, via synapses that undergo mSTDP, into a single post-synaptic cell. The firing of each presynaptic cell is governed by a Poisson process. Throughout the paper, we use the term firing rate to refer to a cell's mean firing rate, which is the cell's probability of firing on any fixed time step. For fixed parameter values, as input firing rate changes, the post-synaptic cell undergoes a transition from silence to a repeated firing regime, with output firing rate rising monotonically between the two limiting states. For any fixed input firing rate, synaptic weights all approach the same steady level. This contrasts completely with the competitive form of STDP seen in the barrier model and sets the system into a baseline mode of operation with a regular output rate that reflects input rate. Thus, mSTDP represents a means of establishing a baseline state, in which a system transmits rate information and will be ready to respond to structured inputs.

2. Model and steady states

To consider the behaviour of the N presynaptic cells, the single post-synaptic cell, and the synaptic weights between them in our model, we treat time as a sequence of discrete time steps. At each time step, each presynaptic cell may fire, with probability r, or remain silent; the firing is all-or-none. The length of a time step thus represents the minimum time between two firings of any cell in the network, which can be selected depending on the application of interest. The weight of the synapse that a presynaptic cell makes with the post-synaptic cell is then updated according to the recent firing history of the two cells. When a presynaptic cell does fire, its excitatory influence on the post-synaptic cell is modulated by the weight of the corresponding synapse. To start the next time step, the post-synaptic cell either fires or does not fire, based on the total strength of input it receives from the presynaptic cells from the previous time step; then, the process repeats, restarting with the presynaptic cells.

Mathematically, the firing of the output cell at step $n + 1$, $n = 1, 2, \ldots$, is cast as

$$\sigma_{\mathrm{o}}(n+1) = g\left(\sum_{i=1}^{N} \sigma_i(n) J_i(n)\right). \tag{1}$$

Here each $\sigma_i(n)$ is 1 with probability r (representing a firing of presynaptic cell i) and 0 with probability $1 - r$. The function $g(x) = 1$ if $x > NT$ or 0 if $x \leqslant NT$ for some threshold T. Finally, the synaptic weights are updated according to the STDP rule

$$J_i(n) = J_i(n-1) + a\sigma_i(n-1)\sigma_{\mathrm{o}}(n)(1 - J_i(n-1)) - b\sigma_i(n)\sigma_{\mathrm{o}}(n)J_i(n-1), \qquad n \geqslant 2 \tag{2}$$

for fixed constants $a, b \in (0, 1)$, where $J_i(1)$ are preset initial weights. Note from (1) that $\sigma_{\mathrm{o}}(n)$ is calculated from the inputs at step $n - 1$, and can be thought of as preceding $\sigma_i(n)$. In this way the negative term in (2) really does encode the depressive side of mSTDP.

There are two steady states to which this system could, in principle, converge, both of which are considered here. First, if the inputs $\sum_{i=1}^{N} \sigma_i(n) J_i(n)$ were always (i.e. with probability 1) above threshold, then the post-synaptic cell would fire at every time step for an output rate of one; likewise, if the inputs were always below threshold, then the post-synaptic cell would never fire, giving an output rate of zero. We give conditions for the output cell to fire at every time step in section 3.

Alternatively, the synaptic weights could remain at some intermediate level, while the post-synaptic cell would fire only on those iterations when it received sufficient input, yielding an output rate between zero and one. In both classes of states, it turns out that the synaptic weights converge to a steady value, $J_i = J^*$ for all i, where J^* depends on a, b, r, T, N. We compute this value, and estimate the output rate σ_{o} when it lies in (0,1), in section 4.

The solid curves in figures 1 and 2 show the output firing rates and averaged steady synaptic weights for a range of input values $r = r_{\mathrm{in}}$ and four different thresholds T. These come from simulations of (2) with $a = 0.1$, $b = 0.15$, and $N = 250$, starting with weights $J_i(1) = 1$ for all i, run until the system reached a steady regime. These are compared with predictions from our analytical computations, shown as dashed curves; this is discussed further in section 4.

3. Steady-state output firing rates

Suppose first that the output firing rate satisfies $\sigma_{\mathrm{o}}(n) = \sigma^*$. For our model, $\sigma^* \in \{0, 1\}$, corresponding to no output firing or output firing on every time step, but we keep σ^* as a parameter for greater generality. In this section, we will show that in the limiting value of the average input to the post-synaptic cell, taken as the number of time steps executed goes to infinity, is given by

$$\lim_{n\to\infty} \langle \sigma(n) J(n) \rangle = ra\left[\frac{1 - \sigma^*(1-r)b}{a + b - \sigma^*(1-r)ab}\right] \tag{3}$$

where $\langle\cdot\rangle$ denotes averaging over the N input cells. This will allow us to show that, for large N, if

$$2r(a + b + 2(1-r)ab) < 3 \qquad \text{and} \qquad T < ra\left[\frac{1 + (1-r)b}{a + b - (1-r)ab}\right] < r \tag{4}$$

then $\sigma_{\mathrm{o}} \equiv 1$ is a steady state of the system and the output cell fires at every time step.

To derive (3) and (4), we substitute $\sigma_{\mathrm{o}} = \sigma^*$ into the synaptic weight update rule (2). This insertion allows us to rewrite (2), omitting reference to the cell number i, as

$$J(n) = h(n) + J(n-1)(1 - \lambda(n)), \qquad n \geqslant 2 \tag{5}$$

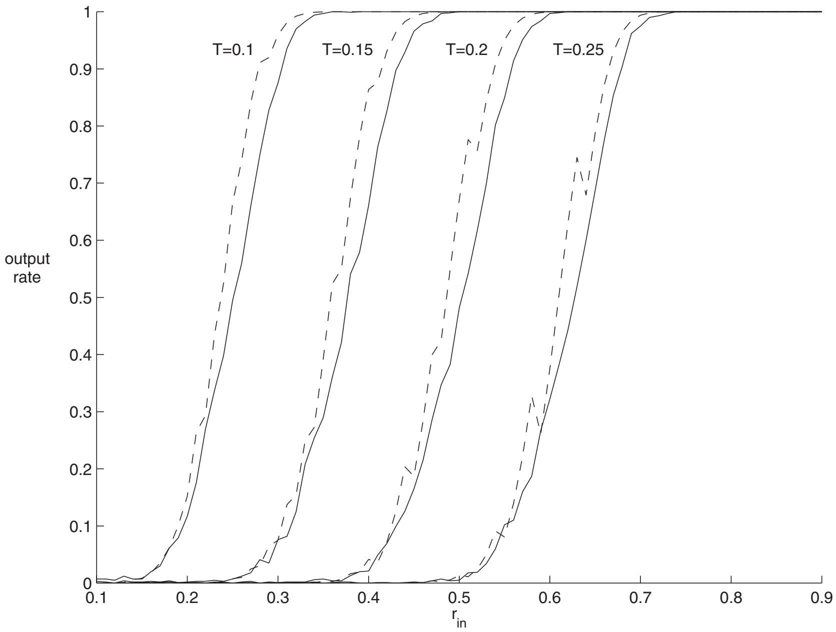

Figure 1. Output rates for the iterative process (2). The solid curves show results from steady states reached by simulation of (2) in MATLAB with $a = 0.1$, $b = 0.15$, $N = 250$, and the values of $r = r_{\text{in}}$ and thresholds T shown. The dashed curves show estimates from (16) in section 4.

where $h(n) = a\sigma(n-1)\sigma^*$ and $\lambda(n) = \sigma^*(a\sigma(n-1) + b\sigma(n))$. Recursively, assuming that the system was initialized with synaptic weight $J(1) = 1$, this implies that

$$\begin{aligned} J(n) &= h(n) + (h(n-1) + J(n-2)(1-\lambda(n-1)))(1-\lambda(n)) \\ &= \cdots \\ &= \sum_{l=1}^{n} \left(h(l) \prod_{k=l+1}^{n} (1-\lambda(k)) \right) \end{aligned}$$

where we set $\sigma(0) = 1/a\sigma^*$ such that $h(1) = J(1) = 1$ and we also define $\prod_{k=n+1}^{n} = 1$. From this, substituting back the full expressions for $h(n)$, $\lambda(n)$, it follows that

$$\begin{aligned} \langle \sigma(n) J(n) \rangle &= \left\langle \sigma(n) \sum_{l=1}^{n} a\sigma^* \sigma(l-1) \prod_{k=l+1}^{n} (1 - \sigma^*(a\sigma(k-1) + b\sigma(k))) \right\rangle \\ &= a\sigma^* \sum_{l=1}^{n} \langle \sigma(n)\sigma(l-1) P \rangle \end{aligned} \tag{6}$$

where we define $P = \prod_{k=l+1}^{n} (1 - \sigma^*(a\sigma(k-1) + b\sigma(k)))$.

For any nonzero x and y, define the quantity $t(k)$ by $t(k) = 1 - x\sigma(k-1) - y\sigma(k)$. Note that $t(k) = 1$ if $\sigma(k-1) = \sigma(k) = 0$, which occurs with probability $(1-r)^2$; $t(k) = 1-x$ if $\sigma(k-1) = 1$ and $\sigma(k) = 0$, which occurs with probability $r(1-r)$; and so on. To encode this information, let

$$M = \begin{pmatrix} 1 & 1-y \\ 1-x & 1-x-y \end{pmatrix}, \qquad D = \begin{pmatrix} 1-r & 0 \\ 0 & r \end{pmatrix}$$

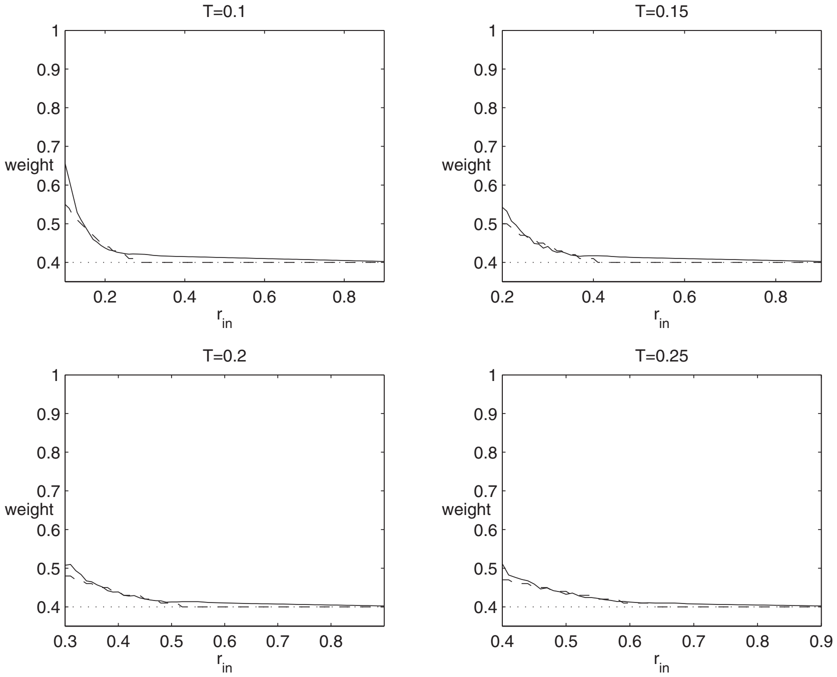

Figure 2. Steady-state synaptic weights for (2). These plots compare results from the same simulations used in figure 1 (solid curves), averaged over all 250 input cells, with estimates (dashed curves) from numerical solution of the condition (13) derived analytically in section 4. The ranges of input rates r_{in} shown have been chosen to avoid an output rate very near zero, where weights in simulations remain at their initial levels.

and let $[MDM]_{ij}$ refer to the element in the ith row and jth column of MDM, counting up from 0 such that $i, j \in \{0, 1\}$. Then averaging $t(k)$ over possible values of $\sigma(k)$ yields

$$\langle(1 - x\sigma(k-1) - y\sigma(k))(1 - x\sigma(k) - y\sigma(k+1))\rangle_{\sigma(k)} = [MDM]_{\sigma(k-1),\sigma(k+1)}. \qquad (7)$$

In particular, let $x = a\sigma^*$ and $y = b\sigma^*$ so that we can apply (7) to the product P from (6).

Notice that this product P contains $n - (l+1) + 1$ terms. There are $(n-1) - (l+1) + 1$ arguments of σ that appear in two terms of P (once each with a coefficient of x, once each with a y), namely $l+1, \ldots, n-1$. Further, the arguments l and n each appear in only one term of P, although, since we are averaging $\sigma(n)J(n)$, the coefficient $\sigma(n)$ also appears in each summand in (6). Finally, $\sigma(l-1)$ does not appear in the product P but also appears in each summand in (6). Pulling this all together, we find that

$$\begin{aligned}\langle\sigma(n)J(n)\rangle = ra\sigma^* &\left[\sum_{l=2}^{n-1} r(1-r)[(MD)^{n-(l+1)}M]_{01} + r^2[(MD)^{n-(l+1)}M]_{11}\right] \\ &+ r^2 a\sigma^* + r(1-r)[(MD)^{n-2}M]_{01} + r^2[(MD)^{n-2}M]_{11}\end{aligned} \qquad (8)$$

where we have separated out the $l = 1$ term to the second line since $\sigma(0) = \frac{1}{a\sigma^*}$. In (8), $[A]_{01}$ refers to the upper right element of a 2×2 matrix A and $[A]_{11}$ refers to the lower right element, as previously.

In expression (8), the indices n, l now appear only in the exponents of matrices. For a positive integer m, basic linear algebra shows that we can find matrices K_1, K_2 such that

$(MD)^m M = \lambda_1^m K_1 + \lambda_2^m K_2$, where λ_1, λ_2 are the eigenvalues of MD. Further, for $\lambda > 0$,

$$\sum_{l=2}^{n-1} \lambda^{n-(l+1)} = \lambda^{n-3} \sum_{l=0}^{n-3} \lambda^{-l} = \lambda^{n-3} \left(\frac{1 - \lambda^{2-n}}{1 - \lambda^{-1}} \right) = \frac{\lambda^{n-2} - 1}{\lambda - 1}. \tag{9}$$

The eigenvalues λ_1, λ_2 of MD are

$$\lambda_{1,2} = \tfrac{1}{2} \left[1 - r\sigma^*(a+b) \pm \sqrt{(1 - r\sigma^*(a+b))^2 + 4(\sigma^*)^2 r(1-r)ab} \right].$$

Correspondingly, one eigenvalue is positive and O(1) and the other is negative (or zero) and O(r). If $\sigma^* = 0$, then these are 1 and 0, respectively. Otherwise, the condition for the positive eigenvalue, call it λ_2, to be less than one is

$$\sigma^*(1-r)ab < a + b \tag{10}$$

and the condition for the negative eigenvalue, call it λ_1, to be greater than -1 (always true for small r) is

$$2\sigma^* r(a + b + 2\sigma^*(1-r)ab) < 3. \tag{11}$$

For $\sigma^* \leqslant 1$ and $a, b < 1$, (10) always holds, since

$$a + b > a^2 + b^2 > 2ab > \sigma^*(1-r)ab,$$

while condition (11) represents the first constraint on parameters given in (4).

Since (10) holds, we know that $\lambda_2 < 1$ and hence (9) can be approximated by $1/(1 - \lambda_2)$ for large n. Similarly, under (11), $\sum_{l=2}^{n-1} \lambda_1^{n-(l+1)} = 1 + \lambda_1 + \lambda_1^2 + \cdots \approx \frac{1}{1-\lambda_1}$ for large n. Finally, we can neglect the last two terms in (8) for large n since they each split into factors multiplied by λ_i^{n-2} for appropriate i.

When we actually compute K_1, K_2, using the relation $\lambda_1 = 1 - \lambda_2 - r(x + y)$, the above approximation yields that for large n

$$\langle \sigma(n) J(n) \rangle \approx ra\sigma^* \left[\frac{r}{\lambda_2 - \lambda_1} \left(\frac{\lambda_1(y(1-r) - \lambda_1)}{1 - \lambda_1} + \frac{\lambda_2(\lambda_2 - y(1-r))}{1 - \lambda_2} \right) + r \right].$$

This expression for mean input simplifies, after some algebra, to (3), where the denominator is positive since condition (10) holds. Note from (3) that $\langle \sigma(n) J(n) \rangle = \mathrm{O}(r)$: the mean input scales linearly with the presynaptic firing rate.

When (11) holds with $\sigma^* = 1$, we can insert $\sigma^* = 1$ into the mean input (3) to determine a criterion for $\sigma^* = 1$ to be realized as a steady state of the system. That is, for the mean input level in (3) to be attainable, it must correspond to a mean synaptic weight below 1. This translates to the requirement $\langle \sigma(n) J(n) \rangle < r$. Moreover, for the post-synaptic cell to fire at each iteration when this input strength is provided, the quantity in (3) must be above threshold. Together, these give the second constraint in (4).

In summary, for fixed parameters a, b, r, T and for large N such that averaging is justified, the steady state $\sigma_0 = 1$ is attained by this system, with mean inputs approximated by (3), when the conditions (4) hold. For $r = 1$, the second inequality in the second condition in (4) always holds. Thus, the output firing rate approaches 1 as $r \to 1$ unless the first inequality is violated for $r = 1$, which amounts to $\frac{a}{a+b} < T$.

4. Steady-state synaptic weight

In the previous section, we assumed that the firing of the post-synaptic cell occurs at a constant rate in order to derive conditions for such a steady state to occur. Now, we drop that assumption and compute the steady-state values to which all synaptic weights may converge under (2).

In general, the output firing rate could have large fluctuations over trials, especially if the inputs that the output cell receives tend to be near its firing threshold. In such a regime, the synaptic weights will evolve slowly relative to the jumps in output firing. To gain insight into how this occurs, we suppose that all synaptic weights reach a steady state J and average the update rule (2) over discrete time iterations. This yields

$$J = J + a\langle\sigma_o(n)\sigma_i(n-1)\rangle(1-J) - b\langle\sigma_o(n)\sigma_i(n)\rangle J \tag{12}$$

where $\langle\cdot\rangle$ now denotes averaging over n. We will show that all solutions to (12) are stable and will use a combination of analysis and numerics to compute solutions to this equation. These results imply that for at least some initial weight distributions, rule (2) drives all synaptic weights in the system to the same steady level. Simulations suggest that this convergence in fact holds for any initial weights.

Denote the mean post-synaptic firing rate, for a system with N presynaptic cells, by $\langle\sigma_o\rangle = p_N$. Since $\sigma_o(n)$, $\sigma_i(n)$ are independent, it follows that for each i, (12) becomes

$$0 = ar[p_N + \delta_N](1-J) - brp_N J$$

where $r\delta_N$ denotes the covariance of $\sigma_o(n)$, $\sigma_i(n-1)$ for N inputs and any n. That is, any steady-state synaptic weight must satisfy

$$J = 1 - \frac{bp_N}{(a+b)p_N + a\delta_N} \tag{13}$$

where p_N, δ_N depend on J; δ_N quantifies the influence of a single input on the next output.

For any solution J^* of (13), plugging $J = J^* + \epsilon(n)$ into (12) yields

$$\epsilon(n) = \epsilon(n-1)(1 - ar(p_N + \delta_N) - brp_N)$$

so the stability condition for such a state is

$$|1 - ar(p_N + \delta_N) - brp_N| < 1. \tag{14}$$

The term on the left-hand side of (14) is clearly bounded above by 1. Further, $ar(p_N + \delta_N) < 1$ and $brp_N < 1$, since by definition, $0 < p_N < p_N + \delta_N < 1$. Thus, every steady-state synaptic weight that solves (13) is stable, and hence the system can have at most one steady-state synaptic weight for any fixed set of parameters.

To make use of the steady-state equation (13), we compute formulae for p_N and δ_N when all synaptic strengths are fixed at some level J. Since $p_N = \langle\sigma_o\rangle$ with N inputs

$$\begin{aligned} p_N &= \left\langle g\left(J\sum_{i=1}^{N}\sigma_i\right)\right\rangle \\ &= \mathrm{Prob}\left(\sum_{i=1}^{N}\sigma_i > \frac{NT}{J}\right). \end{aligned} \tag{15}$$

Using a binomial expansion, this yields the formula for p_N:

$$p_N \equiv p_N^N = \sum_{k=[NT/J]^+}^{N} \binom{N}{k} r^k (1-r)^{N-k} \tag{16}$$

where $[\cdot]^+$ denotes rounding up to the next integer, the superscript N refers to the constant in the threshold $[NT/J]^+$, and the subscript N refers to the total number of input cells.

It is convenient also to define

$$\begin{aligned} q_N^N &= \left\langle g\left(J\sum_{i=1}^{N}\sigma_i + J\right)\right\rangle \\ &= \mathrm{Prob}\left(\sum_{i=1}^{N}\sigma_i > \frac{NT}{J} - 1\right). \end{aligned} \tag{17}$$

Since any one input cell fires with probability r, (15) and (17) yield the relation

$$p_N = p_N^N = rq_{N-1}^N + (1-r)p_{N-1}^N. \tag{18}$$

Direct examination of (12), (13) reveals that for fixed presynaptic cell i, at any fixed time step, $r(p_N + \delta_N) = 0$ with probability $(1-r)$ and $r(p_N + \delta_N) = \langle g(J + \sum_{j \neq i} J\sigma_j(n-1))\rangle$ with probability r; that is,

$$r(p_N + \delta_N) = rq_{N-1}^N.$$

Thus, plugging in from (18), we obtain a formula for δ_N, namely

$$\begin{aligned} \delta_N &= q_{N-1}^N - p_N^N = (1-r)(q_{N-1}^N - p_{N-1}^N) \\ &= \begin{pmatrix} N-1 \\ \left[\dfrac{NT}{J} - 1\right]^+ \end{pmatrix} r^{[\frac{NT}{J}-1]^+}(1-r)^{N-[\frac{NT}{J}-1]^+}. \end{aligned} \tag{19}$$

That is, δ_N is the difference between the probability of obtaining $[\frac{NT}{J}]^+$ firings from N input cells and that of obtaining $[\frac{NT}{J} - 1]^+$ firings from $N-1$ cells, which in some sense measures the influence of an individual input cell when the total of the inputs is close to threshold. As $N \to \infty$, $\delta_N \to 0$, such that in the limit of infinitely many cells, the steady synaptic weight becomes $a/(a+b)$ and the output firing rate tends to a step function (from 0 to 1) as a function of r (assuming $a/(a+b) > T$).

Formulae (16), (19) can be used in (13) to solve numerically for steady-state synaptic weights. Note that a particularly simple case arises when the output firing rate is 1, which was seen in section 3 to occur under (4); in particular, this occurs when input rate r is sufficiently large and a, b are correspondingly small. In this case, (12) simplifies to $ar(1-J^*) - brJ^* = 0$, or $J^* = a/(a+b)$.

The dashed curves in figure 1 show estimates of output rates from (16). These compare quite well to simulations of (2), shown as solid curves. Figure 2 compares actual steady synaptic weights from the same simulations (solid curves), averaged over all 250 input cells, with those (dashed curves) estimated from numerical solution of (13) using (16) and (19). It is important to note that all weights approached the average values shown, independent of initial conditions selected. Also, notice that steady-state weight values approach a constant level as r increases, as predicted. Figure 3 compares the actual (solid) and estimated (dashed) weights for ranges of a, b values with $r = 0.4$. These figures show that our weight estimates work quite well as long as input rates r are not too low. For smaller r values, the probability of output firing is extremely low, such that p_N, δ_N are extremely small, but a single output firing can have a large effect on J. The asymptotic weight estimate (13) takes this effect into account, whereas in actual simulations, an output spike is never observed and synaptic weights remain at their initial values.

5. Discussion

We have proposed a simple model (2) for an iterative form of mSTDP. This represents a discrete learning process in which changes in weights between coupled elements are determined by short-timescale relationships in their past activity. Analysis of this model shows that the output firing rate increases monotonically with the input firing rate and then saturates, with output firing at each time step. Moreover, for fixed parameter values, there exists a stable steady state in which all synaptic weights take on the same value; as shown in figure 2, this value depends only weakly on input firing rate. These findings are consistent with recent results in networks in which Poisson excitatory inputs feed into an integrate-and-fire output cell through

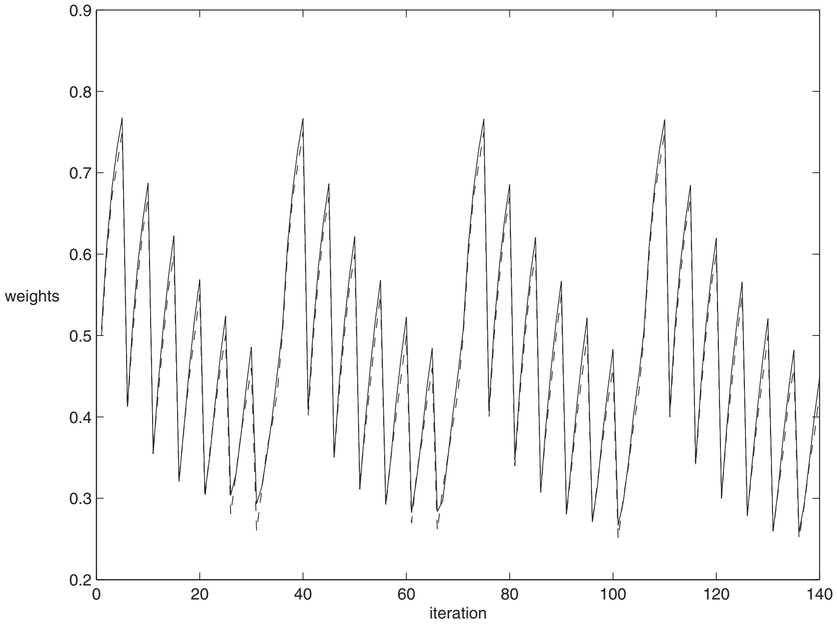

Figure 3. Simulated (solid) and estimated (dashed) steady-state synaptic weights from (13). The iterations are ordered sequentially such that the first 35 have $r = 0.4$, the next have $r = 0.425$, and so on up to $r = 0.475$. Within each set of 35, b cycles from 0.05 to 0.2 in steps of 0.025; for each fixed b, a cycles from 0.05 to 0.15 in steps of 0.025. $N = 250$ for all iterations.

synapses featuring mSTDP (van Rossum *et al* 2000, Rubin *et al* 2001). In that setting, the independence of steady-state weights from input rates yields increases in output firing rate as input rates increase. On the other hand, our results contrast with the effects of STDP in numerical simulations of non-multiplicative barrier models (Abbott and Song 1999, Song *et al* 2000, Levy *et al* 2000) which do not take into account recent experimental data (Bi and Poo 1998) on STDP. Analysis shows that there, synaptic weights tend to a bimodal distribution, clustered near built-in barrier levels, scaling such that output firing rate is approximately constant over a certain range of input rates (Rubin *et al* 2001). No such scaling arises in the model that we have considered.

We have ignored experimentally observed delays between co-activity of pre- and postsynaptic cells and the resultant adjustment of synaptic weights (Markram *et al* 1997, Bi and Poo 1998, Zhang *et al* 1998). Such delays can be included in our model by adjusting the arguments of σ_o, σ_i on the right-hand side of (2) by a delay τ to obtain

$$\begin{aligned} J_i(n) = J_i(n-1) &+ a\sigma_i(n-\tau-1)\sigma_o(n-\tau)(1-J_i(n-1)) \\ &- b\sigma_i(n-\tau)\sigma_o(n-\tau)J_i(n-1). \end{aligned}$$

As long as the same delay is included in each term, the conclusions of our analysis hold.

Our results indicate that the key to the utility of mSTDP in a discrete process must lie in the transmission of rate information and of information about input structure, such as correlations among inputs. Future work should address how this system responds to changes in inputs and to spatially or temporally structured inputs. These issues also remain to be considered in networks with other coupling architectures, such as all-to-all coupling, and in continuous models, such as networks of integrate-and-fire cells.

Acknowledgments

I gratefully acknowledge the hospitality of the Woods Hole Marine Biological Laboratory, where much of this work was completed during the Methods in Computational Neuroscience course. I thank Haim Sompolinsky of Hebrew University for guidance throughout this work and Kenneth Miller of UCSF for significant contributions to the calculations in section 4. I also thank Carson Chow and Bard Ermentrout for comments on the revised version of this paper. The author is partially supported by NSF grant DMS-9804447.

References

Abbott L F and Sen Song 1999 Temporally asymmetric Hebbian learning, spike timing and neuronal response variability *Advances in Neural Information Processing Systems* vol 11, ed M Kearns, S Solla and D Cohn (Cambridge, MA: MIT Press)

Bell C C, Han V Z, Sugawara Y and Grant K 1997 Synaptic plasticity in a cerebellum-like structure depends on temporal order *Nature* **387** 278–81

Bi G Q and Poo M M 1998 Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and post-synaptic cell type *J. Neurosci.* **18** 10464–72

Debanne D, Gähwiler B H and Thompson S M 1996 Synaptic and non-synaptic plasticity between individual pyramidal cells in the rat hippocampus *in vitro J. Physiol. (Paris)* **90** 307–9

Hebb D O 1949 *The Organization of Behavior: A Neuropsychological Theory* (New York: Wiley)

Kistler W M and van Hemmen J L 2000 Modeling synaptic plasticity in conjunction with the timing of pre- and post-synaptic action potentials *Neural Comput.* 385–405

Levy N, Horn D, Meilijson I and Ruppin E 2000 Distributed synchrony in a cell assembly of spiking neurons *Neural Networks* submitted

Markram H, Lübke J, Frotscher M and Sakmann B 1997 Regulation of synaptic efficacy by coincidence of post-synaptic APs and EPSPs *Science* **275** 213–5

Rubin J, Lee D D and Sompolinsky H 2001 Equilibrium properties of temporally asymmetric Hebbian plasticity *Phys. Rev. Lett.* **86** 364–7

Song S, Miller K D and Abbott L F 2000 Competitive Hebbian learning through spike-timing-dependent synaptic plasticity *Nature Neurosci.* **3** 919–26

van Rossum M C W, Bi G Q and Turrigiano G G 2000 Stable Hebbian learning from spike timing dependent plasticity *J. Neurosci.* **20** 8812–21

Zhang L I, Tao H W, Holt C E, Harris W A and Poo M M 1998 A critical window for cooperation and competition among developing retinotectal synapses *Nature* **395** 37–44