Learning in neural networks by reinforcement of irregular spiking

Xiaohui Xie[1,*] and H. Sebastian Seung[1,2]
[1]*Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*
[2]*Howard Hughes Medical Institute, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*
(Received 21 October 2003; published 30 April 2004)

Artificial neural networks are often trained by using the back propagation algorithm to compute the gradient of an objective function with respect to the synaptic strengths. For a biological neural network, such a gradient computation would be difficult to implement, because of the complex dynamics of intrinsic and synaptic conductances in neurons. Here we show that irregular spiking similar to that observed in biological neurons could be used as the basis for a learning rule that calculates a stochastic approximation to the gradient. The learning rule is derived based on a special class of model networks in which neurons fire spike trains with Poisson statistics. The learning is compatible with forms of synaptic dynamics such as short-term facilitation and depression. By correlating the fluctuations in irregular spiking with a reward signal, the learning rule performs stochastic gradient ascent on the expected reward. It is applied to two examples, learning the XOR computation and learning direction selectivity using depressing synapses. We also show in simulation that the learning rule is applicable to a network of noisy integrate-and-fire neurons.

DOI: 10.1103/PhysRevE.69.041909 PACS number(s): 87.19.La, 87.18.Sn, 87.10.+e, 05.45.−a

I. INTRODUCTION

In engineering applications, the parameters of a learning machine are often trained by optimizing an objective function that quantifies performance at a desired computational task. A popular method of optimization is to iteratively change the parameters in the direction of the gradient of the objective function. For example, back propagation learning is a gradient-following algorithm for artificial neural networks [1].

Animals can also learn to optimize performance at certain tasks, and it is widely believed that such optimization occurs at least in part through changes in synaptic strengths. However, it is unclear what mechanism could perform such an optimization. For biological neural networks, computing the gradient of an objective function with respect to synaptic strengths is difficult, because the dynamics of intrinsic and synaptic conductances is so complex. An alternative is to compute a stochastic approximation to the gradient by correlating fluctuations in the network dynamics with fluctuations in its performance [2–6].

A prominent source of fluctuations in neural systems is the irregular firing of action potentials. For example, cortical neurons recorded in vivo often produce interspike interval distributions that are roughly exponential, with a coefficient of variation that is close to 1 [7]. This suggests that the spike trains of cortical neurons are roughly Poisson. Such irregular spiking could be used as a mechanism for learning. Suppose there exists a global reward signal that assesses the overall performance of the network. Then the correlation between reward and spiking fluctuations can serve as an error signal for training the network.

Deriving a learning rule with mathematically proved convergence property in realistic neuronal networks is difficult, since it is hard to to describe the precise statistics on the state of the whole network. Previous work has been concentrated on much simplified network models with Bernoulli logistic units in which each neuron is a memory-less device and current state of neurons completely describes the whole network [4–6,8,9]. In this work, we consider a more realistic class of network models in which two fundamental properties of biological synapses are taken into account: First, neurons interact with each other through synaptic currents. The time scale of synaptic currents is much longer then action potentials. Therefore, the state of the network has to be described by both the state of neurons and the state of synaptic currents. Second, synapses are allowed to be dynamic with short-term plastic effects such as facilitation and depression.

In order to derive a learning rule in such networks, we simplify the spike generation process of neurons, and model it with a Poisson process with firing rates of each neuron determined by its synaptic inputs. The synaptic update rule we derived, on average, is in the direction of the gradient of the expected reward with respect to the synaptic strengths. The algorithm is compatible with dynamic properties of synapses such as short-term facilitation and depression. We illustrate the algorithm with two simple examples, to learn the exclusive OR (XOR) computation and to learn direction selectivity using depressing synapses [10].

For real neurons, the spike trains will not exactly follow Poisson statistics. We show in simulation that the learning rule can still be applied in a network of noisy integrate-and-fire neurons to learn XOR computation.

II. BASIC DEFINITIONS

We consider a model network in which each neuron receives a total synaptic current $I_i(t)$ and produces a Poisson spike train with instantaneous firing rate

*Email address: xhxie@mit.edu

$$\lambda_i(t) = f_i(I_i(t)), \tag{1}$$

where f_i is the current-discharge relationship, or *f-I* curve, of the *i*th neuron. The total synaptic current received by neuron *i* is given by the sum of the contributions over presynaptic neurons *j*:

$$I_i(t) = \sum_{j=1}^{n} W_{ij} h_{ij}(t). \tag{2}$$

W_{ij} is the synaptic strength from neuron *j* to neuron *i*. The synaptic current $W_{ij}h_{ij}(t)$ in neuron *i* due to the spiking of neuron *j* behaves with a time course given by

$$\tau_s \frac{dh_{ij}}{dt} + h_{ij}(t) = \sum_a \delta(t - T_j^a)\zeta_{ij}^a, \tag{3}$$

where T_j^a is the time of the *a*th spike of neuron *j*. ζ_{ij}^a is a binary random variable modeling the stochastic release of neurotransmitters in response to a presynaptic spike. $\zeta_{ij}^a = 1$ denotes a release event of neurotransmitters from neuron *j* to neuron *i* in response to the *a*th spike of neuron *j*, and $\zeta_{ij}^a = 0$ denotes a nonrelease event. A synapse can be dynamic, in which case the release variable ζ_{ij}^a depends on the history of neurotransmitter release experiencing short-term dynamic effect such as depression or facilitation [11–13].

In our synaptic transmission model, the postsynaptic current jumps instantaneously in response to a presynaptic release event and otherwise decays exponentially with time constant τ_s [Fig. 1(a)]. The amplitude of the postsynaptic current is determined by the synaptic strength W_{ij}.

Equation (3) is a simplistic model of a synapse, but there is no barrier to replacing it with more complex dynamical equations. A more serious limitation is Eq. (1), which models spike generation statistically, without reference to biophysics.

Strictly speaking, the validity of the learning rule to be introduced shortly is dependent on the Poisson assumption. However, we expect the learning rule to be valid for more realistic networks of conductance-based model neurons under conditions when spiking is approximately Poisson. For example, such a network can be made to approximate Poisson firing at low rates by injecting white noise into each model neuron. Alternatively, approximate Poisson spiking can arise from internally generated fluctuations of synaptic input in a network with balanced excitation and inhibition [14].

III. LEARNING RULE

We first consider the episodic form of the learning rule. Suppose that the learning process falls into distinct episodes. The network is reinitialized at the beginning of each episode at a fixed initial condition or one drawn at random from some probability distribution. For each episode, the performance of the network is evaluated by a reward function *R*, which depends on the spike trains.

At the end of each episode, the synaptic weights are updated by

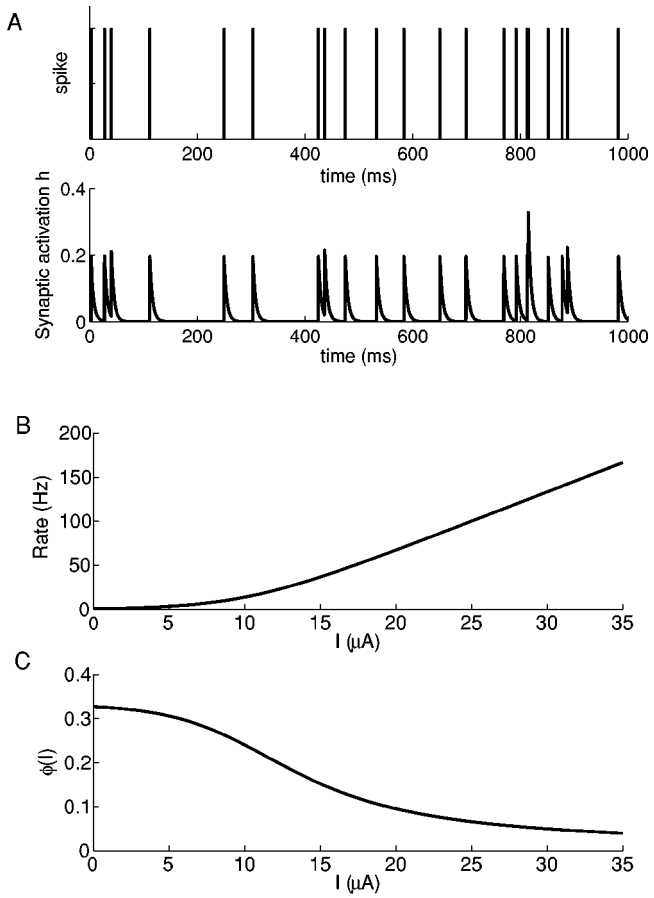

FIG. 1. Poisson model of spiking neurons. (a) An example neuron which fires Poisson spike train with the average rate of 20 Hz (top panel), and the corresponding change in synaptic activation variable with $\tau_s = 10$ ms (second panel). In this example, every presynaptic spike is successfully transmitted. (b) The transfer function *f* used in the simulation, which takes the form $f(x) = 20[x/3 - 3.3 + \ln(1 + \exp(-x/3 + 3.3))]$ where the parameters are chosen to imitate the shape of the *f-I* curve of real neurons. (c) The shape of the function ϕ_i with the transfer function in panel (b).

$$\Delta W_{ij} = \eta R e_{ij}, \tag{4}$$

where $\eta > 0$ is a learning rate, *R* is a reward signal, and e_{ij} is the eligibility trace of the synapse from neuron *j* to *i*. The eligibility trace is defined by

$$e_{ij} = \int_0^T \phi_i(I_i)[s_i(t) - f_i(I_i)]h_{ij}(t)dt, \tag{5}$$

where $s_i(t) = \Sigma_a \delta(t - T_i^a)$ is the spike train of neuron *i*, and *T* is the length of an episode. The function $\phi_i(I_i) = f_i'(I_i)/f_i(I_i)$ is a positive factor, assuming that f_i is monotonically increasing.

In the following section, it is shown that expectation value of the right-hand side of Eq. (4) is proportional to the gradient of the expected reward with respect to W_{ij}. This means that the learning rule performs stochastic gradient ascent on the expected reward. The learning rule can be applied to either feedforward or recurrent networks.

The eligibility trace, Eq. (5), depends on both presynaptic and postsynaptic activities. Presynaptic activity contributes

through the synaptic activation $h_{ij}(t)$, which is elevated upon successful transmission of each spike from neuron j. The postsynaptic neuron contributes through $s_i(t)-f_i(t)$, scaled by the positive factor ϕ_i.

In this learning rule, synaptic change depends on the correlation between the global reward signal and the fluctuations in neural spiking. The learning rule can be understood intuitively as follows. When greater than expected activity in a neuron leads to greater reward, the total inputs to this neuron should be increased. Therefore, we increase excitatory synaptic weights and decrease inhibitory synaptic weights to this neuron. Conversely, if greater than expected activity leads to reduced reward, we decrease excitatory synaptic weights and increase inhibitory synaptic weights to this neuron.

The eligibility trace in learning rule Eq. (4) depends on the fluctuation of the postsynaptic activity away from its average activity, its expected value is zero (proof in the next section). Hence, if the reward function is constant, or uncorrelated to the eligibility trace, there will be no change in average synaptic strength. Therefore, in the learning rule of Eq. (4), we can substitute R by $R-R_0$ as long as R_0 is uncorrelated with the Poisson randomness of the spike trains. Using R_0 will not change the expected value of the synaptic strength, but a carefully chosen R_0 could be helpful in reducing the variance of each update.

IV. DERIVATION OF THE LEARNING RULE

In this section, we derive the episodic version of the algorithm. Suppose the network is run between time 0 and T. At the end of each episode, the overall performance of the network is evaluated by a reward function R that depends on the output states of the network. The episodic version of the algorithm performs stochastic gradient ascent on the expected reward R, which we prove next.

The spike generation is a continuous process. To facilitate derivation of the algorithm, we discretize the time into sufficiently small intervals Δt such that the probability for producing two spikes in the interval is close to zero. Let the binary variable $\sigma_i(t)$ denote the state of the ith neuron in the time interval $[t,t+\Delta t)$. $\sigma_i(t)=1$ denotes spiking of the neuron, and $\sigma_i(t)=0$ denotes nonspiking. For a network of n neurons, we will also use vector $\boldsymbol{\sigma}(t)=[\sigma_1(t),\sigma_2(t),...,\sigma_n(t)]$ to denote the state of all neurons. The state of neurons in the network between time 0 and T can be completely described by $\boldsymbol{\Omega}\equiv[\boldsymbol{\sigma}(0),\boldsymbol{\sigma}(\Delta t),...,\boldsymbol{\sigma}(T)]$. Besides randomness in the generation of spikes, we also consider stochasticity in the release of neurotransmitters with the state variable denoted by $\boldsymbol{\Gamma}\equiv[\boldsymbol{\zeta}(0),\boldsymbol{\zeta}(\Delta t),...,\boldsymbol{\zeta}(T)]$, where $\boldsymbol{\zeta}(t)$ is a binary vector with its component representing whether there is a synaptic transmission from neuron i to neuron j for all i and j at the time interval $[t,t+\Delta t)$. In the case of dynamic synapses with depression or facilitation, the release probability will change depending on the activities of presynaptic neurons. The overall state of the network in one episode is fully determined by the random variables $\boldsymbol{\Omega}$ and $\boldsymbol{\Gamma}$, and the expected reward is

$$\langle R\rangle=\sum_{\boldsymbol{\Omega},\boldsymbol{\Gamma}}P(\boldsymbol{\Omega},\boldsymbol{\Gamma})R(\boldsymbol{\Omega}), \tag{6}$$

where $P(\boldsymbol{\Omega},\boldsymbol{\Gamma})$ is the probability for the state $\boldsymbol{\Omega}$ and $\boldsymbol{\Gamma}$, and the summation is over all possible states.

The gradient of $\langle R\rangle$ with respect to weight W_{ij} is

$$\frac{\partial\langle R\rangle}{\partial W_{ij}}=\sum_{\boldsymbol{\Omega},\boldsymbol{\Gamma}}\frac{\partial P(\boldsymbol{\Omega},\boldsymbol{\Gamma})}{\partial W_{ij}}R(\boldsymbol{\Omega})=\langle e_{ij}R(\boldsymbol{\Omega})\rangle, \tag{7}$$

where $e_{ij}\equiv\partial\ln P(\boldsymbol{\Omega},\boldsymbol{\Gamma})/\partial W_{ij}$ is called the eligibility trace, which determines the direction on updating different synapses. The expected value of the eligibility trace is zero. This can be proved by setting R to be constant in Eq. (7), in which case the left hand side of the equation is $\langle e_{ij}\rangle$ and the right-hand side is zero.

Using the eligibility trace, we can write down the stochastic gradient ascent algorithm to gradually increase the expected reward function,

$$\Delta W_{ij}=\eta R(\boldsymbol{\Omega})e_{ij}, \tag{8}$$

where η controls the learning rate.

To further calculate the eligibility trace, we first factorize the probability $P(\boldsymbol{\Omega},\boldsymbol{\Gamma})$ of the state into

$$P(\boldsymbol{\Omega},\boldsymbol{\Gamma})=\prod_{t=0}^{T}P(\{\boldsymbol{\sigma}(t),\boldsymbol{\zeta}(t)\}|\{\boldsymbol{\sigma}(t'),\boldsymbol{\zeta}(t'),\ \forall\, t'<t\}), \tag{9}$$

which is the product of the probability at each time step conditioned on all previous states. Because at any one time the spike generation and the neurotransmitter release between different neurons are conditionally independent of each other, the logarithm of the probability $P(\boldsymbol{\Omega},\boldsymbol{\Gamma})$ can be written as

$$\ln P(\boldsymbol{\Omega},\boldsymbol{\Gamma})=\sum_{t=0}^{T}\sum_{i=1}^{n}\ln P(\{\sigma_i(t),\zeta_{ij}(t),\ \forall\, j\}|\{\boldsymbol{\sigma}(t'),\boldsymbol{\zeta}(t'),\ \forall\, t'<t\}). \tag{10}$$

Hence, the eligibility trace is

$$e_{ij}=\sum_{t=0}^{T}\frac{\partial}{\partial W_{ij}}\ln P(\{\sigma_i(t),\zeta_{ij}(t),\ \forall\, j\}|\{\boldsymbol{\sigma}(t'),\boldsymbol{\zeta}(t'),\ \forall\, t'<t\}). \tag{11}$$

Let us assume that the neurotransmitter release is independent of the spike generation process at any one particular time. Then we have

$$e_{ij}=\sum_{t=0}^{T}\frac{\partial}{\partial W_{ij}}\ln P(\{\sigma_i(t)\}|\{\boldsymbol{\sigma}(t'),\boldsymbol{\zeta}(t'),\ \forall\, t'<t\}). \tag{12}$$

Given the Poisson assumption on the generation of the spike train of each neuron, the probability for neuron i to fire a spike or not during the interval $[t,t+\Delta t)$ is determined by

$$\sigma_i(t) = \begin{cases} 1 \text{ with probability } p_i(t) = \lambda_i(t)\Delta t, \\ 0 \text{ with probability } 1 - p_i(t), \end{cases}$$

which holds when Δt is sufficiently small. Given this, the eligibility trace e_{ij} can be further simplified to

$$e_{ij} = \sum_{t=0}^{T} \sigma_i(t) \frac{\partial \ln p_i(t)}{\partial W_{ij}} + [1 - \sigma_i(t)] \frac{\partial \ln(1 - p_i(t))}{\partial W_{ij}}$$

$$= \sum_{t=0}^{T} \left[\frac{\sigma_i(t)}{p_i(t)} - \frac{1 - \sigma_i(t)}{1 - p_i(t)} \right] \frac{\partial p_i(t)}{\partial W_{ij}} \tag{13}$$

$$= \sum_{t=0}^{T} \frac{\sigma_i(t) - \lambda_i(t)\Delta t}{\lambda_i(t)\Delta t(1 - \lambda_i(t)\Delta t)} \frac{\partial \lambda_i(t)}{\partial W_{ij}} \Delta t \tag{14}$$

$$= \int_0^T f_i'(t)/f_i(t)[s_i(t) - f_i(t)]h_{ij}(t)dt, \tag{15}$$

where $s_i(t) = \Sigma_a \delta(t - T_i^a)$ is a series of delta functions representing spiking of neuron i in continuous time. The last equation is derived in the limit of Δt being zero. Substituting the eligibility trace e_{ij} into Eq. (8), we derive the episodic version of the learning rule Eq. (4) for Poisson spiking neurons.

From the derivation of the algorithm, we see that the algorithm can be applied to stochastic and/or dynamic synapses, as long as the random process for generating synaptic transmission is independent of the process for generating spike trains. The differences only reflect on the way for calculating synaptic activation variables. For stochastic and/or dynamic synapses, the right-hand side of Eq. (3) includes only the successfully transmitted spikes.

V. ONLINE LEARNING

In some contexts it is not possible to segment the learning process into discrete episodes, and reward is administered continuously in time. Then one can apply the online learning rule

$$\frac{dW_{ij}}{dt} = \eta R(t) \overline{e_{ij}}(t), \tag{16}$$

where the weights are updated continuously in time. The eligibility trace $\overline{e_{ij}}$ is defined by

$$\tau_e \frac{d\overline{e_{ij}}}{dt} + \overline{e_{ij}} = \phi_i(t)[s_i(t) - f_i(t)]h_{ij}(t). \tag{17}$$

The convergence of online learning rules like Eq. (16) is mathematically far more complex than that of episodic learning. Equation (16) can be viewed as an approximation to stochastic gradient ascent. The time constant τ_e of the eligibility trace should be set with reference to the correlation between spiking fluctuations $s_i(t) - f_i(t)$ and reward $R(t')$ for $t' > t$. Suppose that this correlation decays over some characteristic time scale τ_c. Roughly speaking, the approximation to stochastic gradient ascent is expected to be good if τ_e is longer than τ_c. However, it is bad to make τ_e too long, as that will increase the noisiness of the gradient estimate. Next we demonstrate the application of the online learning rules in two examples.

VI. APPLICATION OF THE LEARNING RULE

Next, we illustrate the application of the learning rule in two examples. The first one is to learn XOR computation, which is known to require hidden layer representation in feedforward networks. The second example is to learn direction selectivity using dynamic synapses. We use this example primarily to illustrate the applicability of the algorithm in networks with dynamic synapses.

A. Learning XOR computation

The learning rule Eq. (16) is used to train a three-layer feedforward network of Poisson neurons to learn the XOR computation. The feedforward architecture consists of two input neurons, ten hidden neurons and one output neuron. The training data are four binary patterns $\{[1,0],[0,1],[1,1],[0,0]\}$ with desired outputs of $\{1,1,0,0\}$, respectively. For input neurons, we model the input data 1 with Poisson spike trains of high rate (200 Hz), and 0 with Poisson spike trains of low rate (5 Hz). Lower rates can be used if each binary variable is represented by a neural population, rather than by a single neuron.

At each training epoch, a randomly chosen input pattern is presented for 500 ms. The reward is evaluated based on activities of the output neuron. If the input pattern is $[1,0]$ or $[0,1]$, we give a positive reward of $R=2$ whenever the output neuron fires a spike. For the input patterns $[1,1]$ and $[0,0]$, we administer a negative reward of $R=-1$ when the output neuron fires a spike. No reward is administered when the output neuron fires no spikes. Other reward schemes are possible, and may be better or worse for the speed of convergence or for escaping from local optima. For example, the algorithm often gets stuck in local optima if instead we use $R=1$ for output spiking in response to input patterns $[1,0]$ and $[0,1]$. The problem of escaping from local optima is a potential problem for all gradient algorithms.

We use the same transfer function f_i for all neurons; f_i has a shape similar to the f-I (firing rate vs input current) curve of real neurons [Fig. 1(b)]. Under this transfer function, the corresponding scale function $\phi_i(I_i)$ in Eq. (5) decreases for large neural activities [Fig. 1(c)], and therefore acts to stabilize the algorithm.

The online learning algorithm is used to learn both layers of synaptic weights (input to hidden, and hidden to output), starting from random initial conditions for the weights. An example of the learning process is shown in Fig. 2. Initially, the output neuron fires at high rates only when both inputs are 1. By the end of training, the output neuron fires at high rates only when one input neuron receives 1, but not both [Fig. 2(a)]. The total reward administered during one epoch gradually increases on average, although it fluctuates up and down on short time scales [Fig. 2(b)].

The network learns to represent XOR computation by balancing excitation and inhibition (Fig. 3). After learning, each

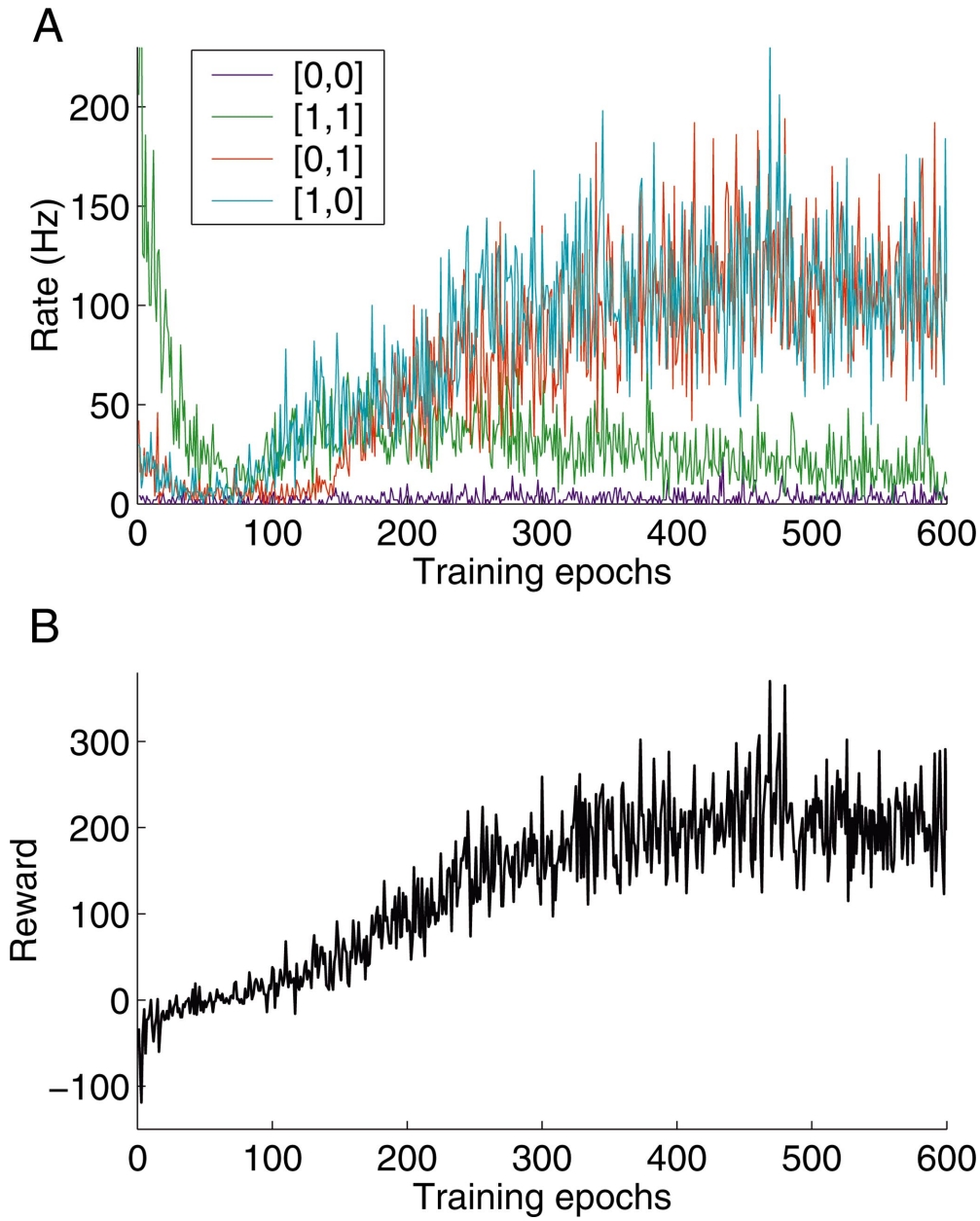

FIG. 2. (Color) Learning XOR computation. (a) The change in the firing rates of the output neurons corresponding to the four input patterns plotted as a function of training epochs. During training, the activities of the output neuron to pattern $[1,1]$ gradually decrease, whereas those corresponding to pattern $[0,1]$ and $[1,0]$ increase. (b) The total reward administered during one training epoch drawn as a function of training epochs. The total reward in one training epoch is the sum of the reward administered during four training episodes for each pattern.

hidden neuron is excited by one input neuron and inhibited by the other one. Therefore, if both input neurons are activated, the hidden neurons are inactive because their total synaptic inputs roughly cancel out.

In the above simulation, the reward signal is administered following each spike of the output neuron. This constraint can be relaxed under the condition that the correlation between the reward signal and the eligibility trace is large enough. For example, in simulation we find that with the time constant of eligibility trace $\tau_e = 10$ ms, the XOR can still be learned even if the reward signal is delivered after a delay of 10 ms. To compensate much longer delay, we have to choose correspondingly a larger τ_e, but at the price of slowing down the learning process. For a delay of reward in the scale of seconds, the system needs to solve the temporal credit assignment problem of estimating current reward. Algorithms such as temporal difference learning may provide a solution to such problems [15].

B. Learning direction selectivity using dynamic synapses

Many theories have been proposed for the computational functions of dynamic synapses, such as gain control, temporal information processing, and sequence recognition

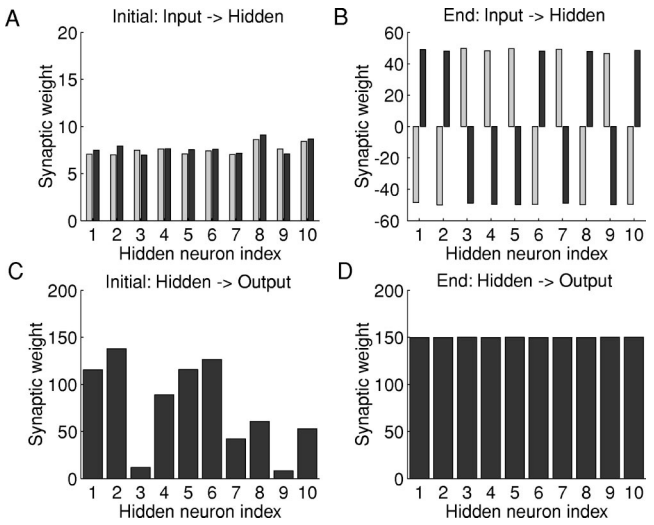

FIG. 3. Synaptic weights before and after learning. The top two panels represent synaptic connections from two input neurons to 10 hidden neurons before [panel (a)] and after [panel (b)] learning. Dark and gray bars represent synapses emanating from the two input neurons separately. The bottom two panels plot the synaptic weights from the hidden to the output neuron before [panel (c)] and after [panel (d)] learning. To limit the firing rate of the output neuron not being too high, we have bounded the synaptic strength from the input neurons to the hidden neurons to be less than 50 and from the hidden neurons to the output neuron to be less than 150.

[13,16–19]. In particular, one study showed that a neural circuit of depressing synapses can possess the property of direction selectivity [10]. Here we show that such a circuit can be learned through the synaptic update rules introduced earlier.

Direction selectivity can be produced by mechanisms other other dynamic synapses. Our main goal here is not to argue against other models, but to illustrate the applicability of the algorithm to dynamic synapses.

A simple model for producing a direction selective neuron is shown in Fig. 4. The neuron receives 40 presynaptic inputs, which are separated into two groups. Each group contains 20 neurons representing inputs from a region in visual space. The regions of the two groups are spatially displaced. When a moving stimulus is presented, the two groups see the same input, except for a temporal shift [Figs. 6(a) and 6(b)].

Half of the synapses in each group are nondepressing, whereas the other half are depressing. In response to a presynaptic spike, a nondepressing synapse always releases a vesicle with a fixed probability p_0, and can release at arbitrarily high rates. In contrast, a depressing synapse enters a refractory state after vesicle release, during which it cannot release again. It recovers after a time that is exponentially distributed with mean τ_r. In response to a presynaptic spike, a depressing synapse releases a vesicle with probability p_0, but only while nonrefractory. The synaptic activation variable is calculated using Eq. (3), except that the right-hand side of the equation consists of the release events, rather than the whole spike train.

All neurons in the network fire Poisson spike trains. For the input neurons, the instantaneous rates are specified by the

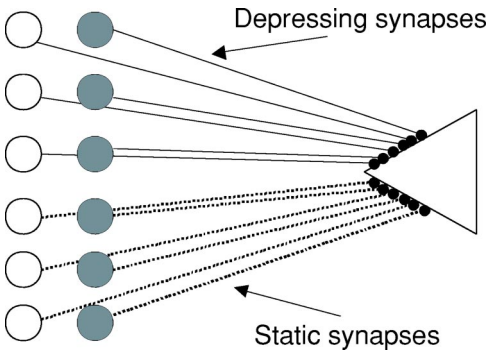

FIG. 4. Diagram of the depressing synapse model for direction selectivity. The postsynaptic neuron receives synaptic inputs from two groups with 20 neurons in each. Firing rates of the neurons in each group are the same, and are coded as the rectified sinusoidal function with peak amplitude 200 Hz and period 300 ms. Between groups, the firing rates are temporally shifted from each other with a delay or advance of 60 ms in the second group (filled circles) representing inputs in the preferred or nonpreferred directions, respectively. The synapses are all excitatory. From each group, half synapses are depressing synapses (solid lines) with the mean recovery time τ_r=200 ms, and the other half are nondepressing synapses (dashed lines). The stochastic release probability p_0=0.8. The synaptic weights are randomly initialized.

functions shown in Figs. 6(a) and 6(b). The rate of the output neuron is determined by the total synaptic inputs using the transfer function given in Fig. 1.

The synapses are all excitatory with the synaptic weights randomly initialized. In each trial of training, both directions are presented, with each held for 600 ms. One direction is trained to be the preferred direction by administering a reward of 1 if the output neuron fires a spike. The other direction is trained to be the nonpreferred direction by administering a reward of −1 if the output neuron fires a spike. We use the online learning rule Eq. (16) to update synaptic weights.

The learned synaptic weights are shown in Fig. 5(c). At the end of learning, only nondepressing synapses remain for one group, and depressing synapses for the other group. The synaptic weights of all other synapses decay to near zero. The total reward is plotted in Fig. 5(d) as a function of the training epoch number.

By using nondepressing synapses from one group and depressing synapses from the other group, the postsynaptic neuron is able to achieve direction selectivity [10]. The preferred direction stimulates the nondepressing synapses first, and the depressing synapses second. Because the depressing synapses introduce a phase shift forward in time, the postsynaptic neuron is coactivated by the synaptic input from both groups [Fig. 6(a), 6(c), and 6(e). On the other hand, the nonpreferred direction stimulates the depressing synapses first, and the nondepressing synapses second, which separates the synaptic inputs from both groups in time, leading to weak response in the postsynaptic neuron [Fig. 6(b), 6(d), and 6(f)].

VII. NETWORKS OF INTEGRATE-AND-FIRE NEURONS

The learning rule is derived with the assumption of Poisson spike trains. The spike trains of biological neurons are

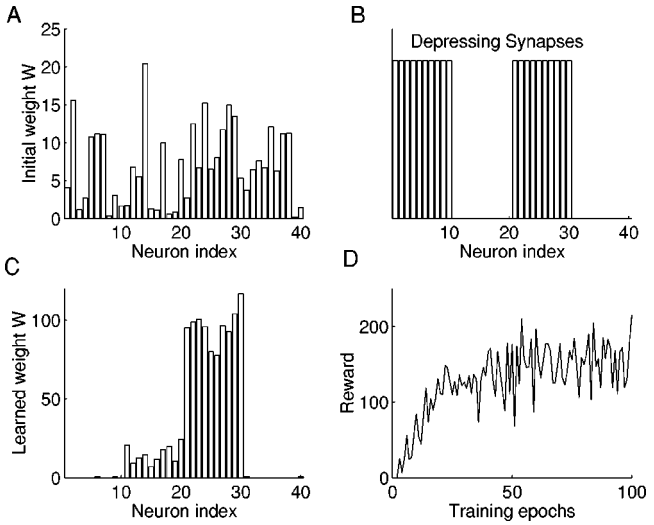

FIG. 5. Learning to be direction selective using synapses with depression. (a) Initial synaptic weights from all input neurons. (b) The depressing synapses from the input neurons are indicated by bars. Neurons from 1 to 20 belong to the first group and neurons from 21 to 40 belong to the second group, which is activated with a delay after the first group when the stimulus is moving in the preferred direction. (c) The synaptic weights after training. Only nondepressing synapses from the first group and the depressing synapses from the second group are selected, whereas the strength of the rest synapses decays to zero. (d) The total reward administered during each trial as a function of the training epochs.

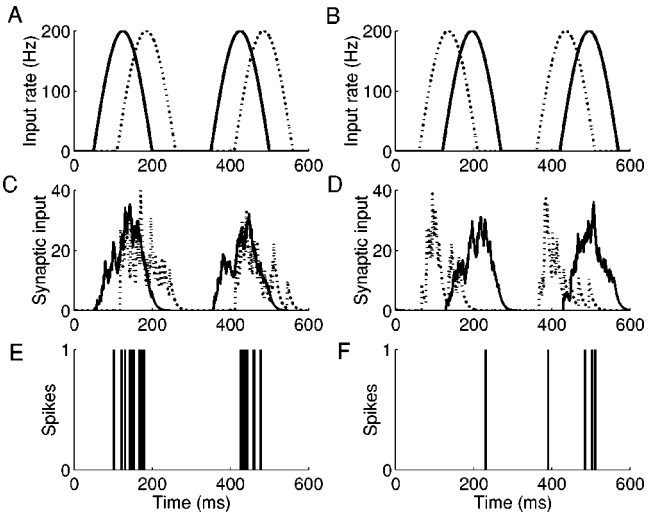

FIG. 6. Responses of the postsynaptic neuron to the preferred (left three panels) and nonpreferred (right three panels) directions after training. (a) and (b) show the input firing rates from the two groups. The first group is indicated by the solid lines, and the second group by the dotted lines. (c) and (d) plot the total synaptic inputs from the two groups separately. The solid line indicates the total input from the first group and the dotted line indicates the total input from the second group. Because after training the synapses from the first group are mostly nondepressing synapses, their total synaptic input follows the input firing rate without significant phase shift. However, the synapses from the second group are mostly depressing synapses. Their total synaptic input is phase-shifted forward, which leads to overlapping synaptic inputs in (c) and nonoverlapping synaptic inputs in (d). (e) and (f) show the spike trains of the postsynaptic neuron corresponding to the preferred and nonpreferred directional inputs, respectively.

not truly Poisson. However, the learning rule may still work in the presence of deviations from Poisson behavior. In this section, we test application of the algorithm to a network of integrate-and-fire neurons. We inject white noise to those neurons to emulate random inputs neurons receive. The model neuron is described by

$$\tau_m \frac{dV_i}{dt} = -V_i + V_{rest} + I_i(t) + \xi_i(t), \tag{18}$$

where V_i is the membrane potential for neuron i, τ_m is the membrane time constant, V_{rest} is the resting potential, and $I_i(t)$ is the total synaptic input. $\xi_i(t)$ is the white noise:

$$\langle \xi_i(t) \rangle = 0, \quad \langle \xi_i(t)\xi_j(t') \rangle = \sigma^2/\tau_m \delta_{ij}\delta(t-t'), \tag{19}$$

for all $i,j=1,\ldots,n$. When membrane potential V_i reaches a threshold V_{th}, a spike is generated and V_i is reset to V_r. The synaptic input $I_i(t)=\Sigma_j W_{ij}h_{ij}(t)$, where $h_{ij}(t)$ is the synaptic activation variable, which is modeled by Eq. (3).

The firing rate vs current relationship can be calculated explicitly when white noise is injected. The firing rate is described by [20]

$$f_i(I_i) = \left[\tau_m \int_0^\infty e^{-u^2}(e^{2y_{th}u} - e^{2y_r u})/u\,du\right]^{-1}, \tag{20}$$

where $y_{th}=(V_{th}-V_{rest}-I)/\sigma$ and $y_r=(V_r-V_{rest}-I_i)/\sigma$.

We apply the learning rule Eq. (16) to learn the same XOR problem described in Sec. VI A. The result is shown in Fig. 7, which demonstrates that the learning rule could still be used for learning XOR computation. The learning curve and the resulting synaptic strength are also similar to those in the preceding section with Poisson spiking neurons.

The learning rule Eq. (16) uses the correlation between reward and variations in neural activities to direct the change of synaptic strength. If spike trains are not Poisson, the learning rule may still, on average, perform hill-climbing on the expected reward function, though not necessarily in the gradient ascent direction.

VIII. DISCUSSION

We have proposed a synaptic update rule for learning in networks of spiking neurons. We show that the learning rule is on average performing gradient ascend on an expected reward function. The algorithm itself does not compute gradient information explicitly, but estimates the gradient using the correlation between the global reward signal and the fluctuations in neural activities.

The learning rule depends on the spiking of presynaptic and postsynaptic neurons. Recent experiments have demonstrated types of synaptic plasticity that depends on the temporal ordering of presynaptic and postsynaptic spiking. At cortical and hippocampal synapses, long-term potentiation is induced by repeated pairing of a presynaptic spike and a

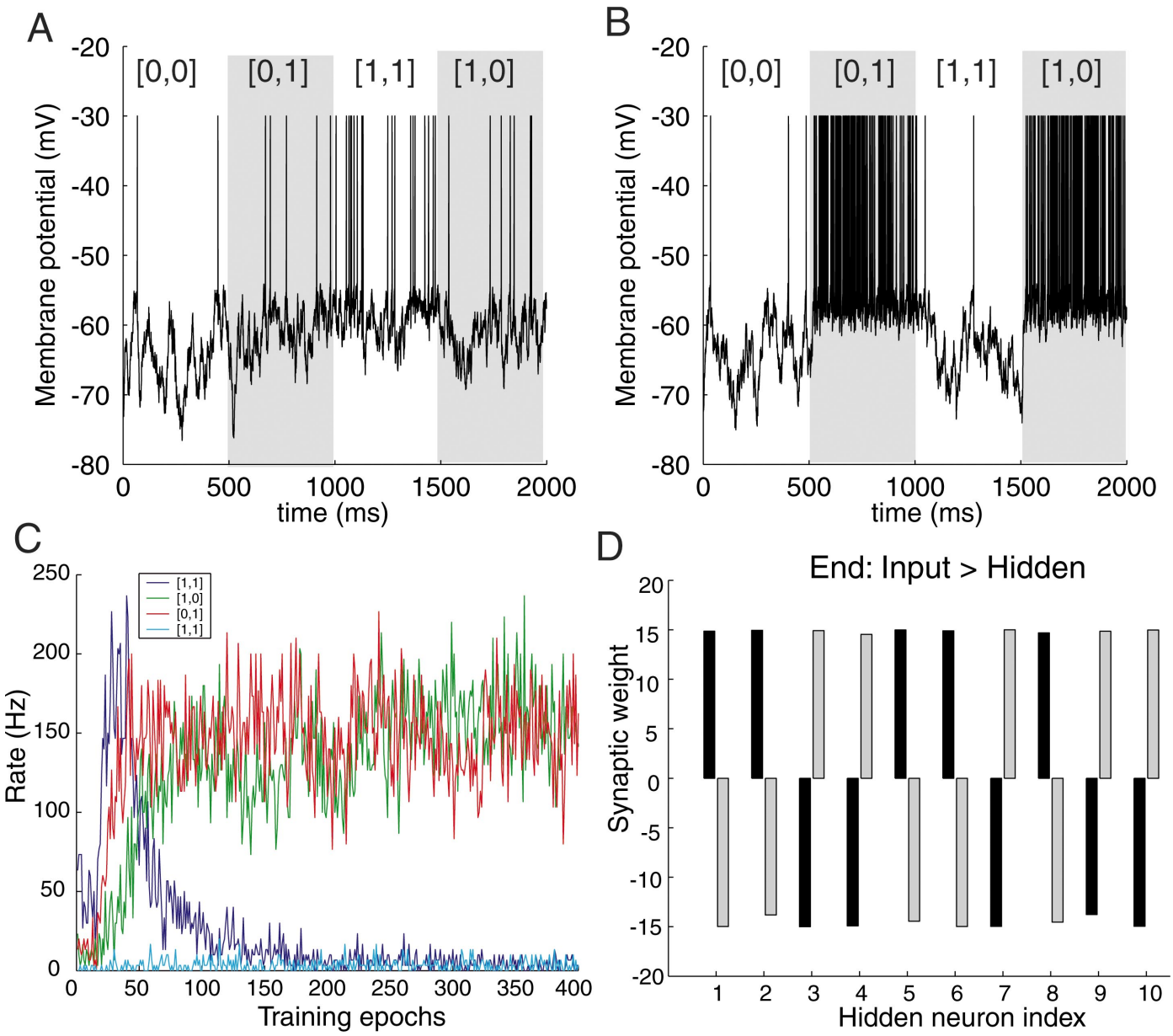

FIG. 7. (Color) Learning XOR in a network of integrate-and-fire neurons. Panels (a) and (b) plot the membrane potentials of the output neuron over four input patterns before and after learning, respectively. Each input pattern is held for 500 ms. Panel (c) plots the change on the firing rates of the output neurons over different input patterns during training. Panel (d) plots the synaptic weight from the input to the hidden neurons after training. Dark and gray bars represent synapses emanating from the two input neurons separately. The learned synaptic weight from the hidden to the output neurons saturates at maximum value as in Fig. 3 and is not shown here.

succeeding postsynaptic spike, while long-term depression results when the order is reversed [21,22]. To induce either type of plasticity, the time difference between presynaptic and postsynaptic spikes has to be within a short time window.

Relating our learning rule to such experiments is not rigorously possible, because spiking in vitro is far from Poisson. To properly make the connection, it is necessary to extend the learning rule to a model of spiking that is more biophysically realistic than the Poisson model, a task that is outside the scope of the present work. However, the following heuristic arguments can be made. The eligibility trace is determined by filtering $\phi_i(s_i - f_i)h_{ij}$, and therefore its sign is governed by the term, $s_i h_{ij}$ or $f_i h_{ij}$, that dominates the difference. A lone presynaptic spike in neuron j sets up an exponentially decaying trace in h_{ij}. If it is closely followed by a postsynaptic spike, then $s_i h_{ij}$ will dominate over $f_i h_{ij}$. If it is closely preceded by a postsynaptic spike, then $s_i h_{ij} = 0$, but $f_i h_{ij}$ could be nonzero (whether this is true depends on how the learning rule is extended to the non-Poisson case). Therefore it is possible for the eligibility trace to change sign, depending on the temporal ordering of presynaptic and postsynaptic spiking.

Experiments on spike-timing dependent plasticity have not tried to control any reward signal. If any reward circuitry exists, it is not clear whether it would be operative in vitro

anyway. If we suppose that the reward signal is frozen at some positive value in vitro, then the above argument about the sign of the eligibility trace implies the form of spike-timing-dependent plasticity observed by Bi and Poo, and Markram *et al.* [21,22]. On the other hand, if the reward signal is frozen at some negative value, the learning rule would lead to potentiation for the opposite temporal ordering, as observed by Bell *et al.* [23]. According to this interpretation, the time window for plasticity induced by presynaptic followed by postsynaptic spiking is determined by the time constant of the postsynaptic currents.

In this paper, we derived the algorithm from a simple Poisson spiking model. We speculate that the idea of updating synaptic strength based on the correlation between local neural activity fluctuation and a global reward signal could be applicable to networks of biophysical neurons. However, for the algorithm to work properly, each neuron should have a mechanism for estimating its mean activity, that is, $f_i(t)$ in the learning rule. If the input to a postsynaptic neuron is slowly variant, the neuron could estimate its mean activity based on previous spikes, for example by low-pass filtering. However, this mechanism will fail to work if the input changes rapidly. One possible way to solve this problem is to do noise injection into neurons, and modify the learning rule to depend on the correlation between injected noise and the global reward signal. This way, each neuron only needs to detect the statistics of injected noise, independent of its own activity. If the injected noise is close to be stationary, estimating its variance by the postsynaptic neuron becomes relatively easy. Another possible way to solve the problem is to use the temporal variance rather than statistical variance of neural activities. For example, we can modify the algorithm to update synaptic strength based on the correlation between the temporal variance of neural activity and the temporal variance of reward function.

The learning update is a noisy estimate of the gradient of the expected reward function. The signal-to-noise ratio of this estimate tends to deteriorate with increased network size, because the correlation of reward with the fluctuations in spiking of any single neuron becomes weaker. Therefore, similar to other types of reinforcement learning [2,3,15], the learning rule suffers from slow convergence in large networks. However, the learning rule proposed here should be faster than algorithms that correlate reward with fluctuations in synaptic efficacy [2,3]. One way to speed up reinforcement learning is to dissolve a large learning problem into smaller subproblems, each learned by a module trained by a separate reward signal [24]. How to construct such a hierarchical organization is a challenging issue that needs to be resolved.

The learning rule relies critically on the modulation of reward signals. Investigating the existence of such reward signals and manipulating them to change the course of synaptic plasticity is an interesting topic for future experimental studies.

ACKNOWLEDGMENTS

We thank Sam Roweis, Mark Goldman, and Ila Fiete for helpful discussions.

APPENDIX: METHODS

In simulation, we discretize the continuous dynamics using Euler's method with a step size of $\Delta t = 0.1$ ms. At each time interval, a spike is generated with probability $\lambda_i(t)\Delta t$ for neuron i, where $\lambda_i(t)$ is its firing rate at discretized time t. The spike train generated this way would be an approximation to the Poisson statistics and is exact in the limit when Δt goes to zero. The synaptic time constant $\tau_s = 10$ ms, the time constant for eligibility trace $\tau_e = 10$ ms, and the transfer function used is shown in Fig. 1(b).

In the first example of learning XOR computation, we use a three-layer feedforward network with two input neurons, ten hidden neurons and one output neuron. Each one of the four input patterns is presented for a fixed period of 500 ms with random orders in each training epoch. When the output neuron fires a spike, we give a positive reward of 2 if the desired output is 1 and a negative reward of -1 if the desired output is 0. In this example, to simplify the problem we use static and deterministic synapses, therefore, every presynaptic spikes are faithfully transmitted to postsynaptic neurons. In the second example of learning direction selectivity using depressing synapses, the preferred and nonpreferred direction inputs are presented for 600 ms in alternation. For depressing synapses, the refractory time after each release event is modeled by an exponential distribution with mean time $\tau_r = 200$ ms. In simulation, we model this distribution using $-\tau_r \ln(x)$, where x is a random variable uniformly distributed between 0 and 1. For both depressing and nondepressing synapses, we choose the stochastic release probability $p_0 = 0.8$.

For the integrate-and-fire neuron model, the parameters we use are $\tau_m = 20$ ms, $V_{th} = -54$ mV, $V_r = -60$ mV, $V_{rest} = -74$ mV, and $\sigma = 5.6$.

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature (London) **323**, 533 (1986).

[2] M. Jabri and B. Flower, IEEE Trans. Neural Netw. **3**, 154 (1992).

[3] G. Cauwenberghs, in *A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization*, edited by S. J. Hanson, J. D. Cowan, and C. L. Giles (Morgan Kaufmann, San Mateo, CA, 1993), Vol. 5, pp. 244–251.

[4] R. J. Williams, Mach. Learn. **8**, 229 (1992).

[5] J. Baxter and P. L. Bartlett, J. Artif. Intell. Res. **15**, 319 (2001).

[6] P. Mazzoni, R. A. Andersen, and M. I. Jordan, Proc. Natl. Acad. Sci. U.S.A. **88**, 4433 (1991).

[7] W. Softky and C. Koch, J. Neurosci. **13**, 334 (1993).

[8] A. G. Barto and P. Anandan, IEEE Trans. Syst. Man Cybern. **15**, 360 (1985).

[9] A. G. Barto and M. I. Jordan, in *IEEE First International*

Conference on Neural Networks, San Diego, 1987, edited by M. Caudill and C. Butler (IEEE, New York, 1987), Vol. 2, pp. 629–636.
[10] F. S. Chance, S. B. Nelson, and L. F. Abbott, J. Neurosci. **18**, 4785 (1998).
[11] R. S. Zucker, Annu. Rev. Neurosci. **12**, 13 (1989).
[12] M. Tsodyks, K. Pawelzik, and H. Markram, Neural Comput. **10**, 821 (1998).
[13] L. F. Abbott, J. A. Varela, K. Sen, and S. B. Nelson, Science **275**, 220 (1997).
[14] C. van Vreeswijk and H. Sompolinsky, Science **274**, 1724 (1996).
[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 1998).
[16] W. Maass and A. M. Zador, Neural Comput. **11**, 903 (1999).
[17] T. Natschlager, W. Maass, and A. Zador, Network **12**, 75 (2001).
[18] D. V. Buonomano and M. M. Merzenich, Science **267**, 1028 (1995).
[19] J.-S. Liaw *et al.*, Hippocampus **6**, 591 (1996).
[20] N. Brunel and V. Hakim, Neural Comput. **11**, 1621 (1999).
[21] G.-Q. Bi and M.-M. Poo, J. Neurosci. **18**, 10464 (1998).
[22] H. Markram, J. Lubke, M. Frotscher, and B. Sakmann, Science **275**, 213 (1997).
[23] C. C. Bell, V. Z. Han, Y. Sugawara, and K. Grant, Nature (London) **387**, 278 (1997).
[24] P. Dayan and G. E. Hinton, in *Feudal Reinforcement Learning*, edited by S. J. Hanson, J. D. Cowan, and C. L. Giles (Morgan Kaufmann, San Mateo, CA, 1993), Vol. 5, pp. 271–278.