A Biologically Plausible and Locally Optimal Learning Algorithm for Spiking Neurons

Peter L. Bartlett and Jonathan Baxter
Research School of Information Sciences and Engineering
Australian National University
Peter.Bartlett@anu.edu.au, Jonathan.Baxter@anu.edu.au

Abstract

We derive a new model of synaptic plasticity, based on recent algorithms for Direct Reinforcement Learning (DRL). We show that these direct reinforcement learning algorithms also give locally optimal performance for the problem of reinforcement learning with multiple agents, without any explicit communication between agents. By considering a network of spiking neurons as a collection of agents attempting to maximize the long-term average of a reward signal, we derive a synaptic update rule that is qualitatively similar to Hebb's postulate, requiring only simple computations, such as addition and leaky integration, and involving only quantities that are available in the vicinity of the synapse. Furthermore, it leads to synaptic connection strengths that give locally optimal values of the long term average reward. The approach has proved effective for simple pattern classification and motor learning tasks.

1 What is a good synaptic update rule?

It is widely accepted that the functions performed by neural circuits are modified by adjustments to the strength of the synaptic connections between neurons. In the 1940s, Donald Hebb speculated that such adjustments are associated with simultaneous (or nearly simultaneous) firing of the presynaptic and postsynaptic neurons [16]: "*When an axon of cell A ... persistently takes part in firing [cell B], some growth process or metabolic change takes place [to increase] A's efficacy as one of the cells firing B.*" Although this postulate is rather vague, it provides the important suggestion that the computations performed by neural circuits could be modified by a simple cellular mechanism. Many candidates for Hebbian synaptic update rules have been suggested, and there is considerable experimental evidence for such mechanisms [11, 26, 18, 19, 21, 24].

Hebbian modifications to synaptic strengths seem intuitively reasonable as a mechanism for modifying the function of a neural circuit. However, it is not clear that these synaptic updates actually improve the performance of a neural circuit in any useful sense. Indeed, simulation studies of specific Hebbian update rules have illustrated some serious shortcomings [22].

In contrast with the "plausibility of cellular mechanisms" approach, most artificial neural network research has emphasized performance in practical applications. Synaptic update

rules such as backpropagation are designed to minimize a suitable cost function [25]. Unfortunately, there is little evidence that backpropagatation's computations can be performed in biological neural circuits.

This paper presents a synaptic update rule that provably optimizes the performance of a neural network, but requires only simple computations involving signals that are readily available in biological neurons. The synaptic update rule is consistent with Hebb's postulate. Precursors to our proposed rule include [7, 5, 4, 6, 28, 29].

2 Direct reinforcement learning

Our setting is that of an agent taking actions in an environment according to a parameterized policy. The agent seeks to adjust its parameters in order to maximize the long-term average reward. Formally, the most natural model for this problem is that of Partially Observable Markov Decision Processes or POMDPs. For ease of exposition we consider finite POMDPs. Specifically, assume that there are n_s states $\mathcal{S} = \{1, \ldots, n_s\}$ of the world, n_c controls (or actions) $\mathcal{U} = \{1, \ldots, n_c\}$ and n_o observations $\mathcal{Y} = \{1, \ldots, n_o\}$. For each state $i \in \mathcal{S}$ there is a corresponding reward $r(i)$. Each $u \in \mathcal{U}$ determines a stochastic matrix $P(u) = [p_{ij}(u)]$ where $p_{ij}(u)$ is the probability of making a transition from state i to state j given control u. For each state $i \in \mathcal{S}$, an observation $y \in \mathcal{Y}$ is generated independently according to a probability distribution $\nu(i)$ over observations in $\mathcal{Y}$. We denote the probability of observation y by $\nu_y(i)$. A *randomized policy* is simply a function μ mapping observations $y \in \mathcal{Y}$ into probability distributions over the controls $\mathcal{U}$. That is, for each observation y, $\mu(y)$ is a distribution over the controls in $\mathcal{U}$. Denote the probability under μ of control u given observation y by $\mu_u(y)$. We parameterize the policies, so that μ now becomes a function $\mu(\theta, y)$ of a set of n_p real parameters $\theta \in \mathbb{R}^{n_p}$ as well as the observation y. In general, to perform optimally, the policy has to be a function of the entire history of observations, but this can be achieved by concatenating observations and treating the vector of observations as input to the policy. One could also consider policies that have memory, such as parameterized finite automata [23].

Our goal is to find a $\theta \in \mathbb{R}^{n_p}$ maximizing the *long-term average reward*: $\eta(\theta) := \lim_{T\to\infty} \frac{1}{T}\mathbf{E}_\theta \left[\sum_{t=1}^{T} r(i_t)\right]$ where $\mathbf{E}_\theta$ denotes the expectation over all state sequences $i_0, i_1, \ldots$, when the agent uses policy $\mu(\theta, \cdot)$. We pursue a local approach: get the agent to compute the gradient of the average reward with respect to its parameters, $\nabla\eta(\theta)$, and then adjust the parameters in the gradient direction.

The *direct reinforcement learning* approach presented in [8, 10, 9, 3], building on ideas due to a number of authors [6, 29, 12, 13, 17, 20, 1], adjusts the parameters θ of the policy in the direction of the long-term average of $r(i_t)z_t$, where $z_t \in \mathbb{R}^{n_p}$ is an *eligibility trace* that is updated according to

$$z_{t+1} = \beta z_t + \frac{\nabla\mu_{u_t}(y_t, \theta)}{\mu_{u_t}(y_t, \theta)}, \tag{1}$$

where β is a discount factor between 0 and 1.

Under general ergodicity assumptions, if θ remains constant, the long-term average of the product $r(i_t)z_t$ converges (with probability 1) to a vector $\nabla_\beta\eta(\theta)$ [8, Theorem 5]. In addition, $\lim_{\beta\to 1} \nabla_\beta\eta(\theta) = \nabla\eta(\theta)$ [8, Theorem 3]. Thus, $\nabla_\beta\eta(\theta)$ is a good approximation to $\nabla\eta(\theta)$ provided β is sufficiently close to 1. The factor preventing setting $\beta = 1$ is that the variance of the average of $r(i_t)z_t$ after a finite number of steps scales as $1/(1-\beta)$ [3, Corollary 16]. Fortunately, $\nabla_\beta\eta(\theta)$ is guaranteed to be a good approximation to $\nabla\eta(\theta)$ provided $1/(1-\beta)$ exceeds a certain *mixing time* associated with the POMDP[8, Theorem 4], [3, Theorem 21]. It is useful, although not quite correct, to think of the mixing time as the time from the occurrence of an action until the effects of that action have died away.

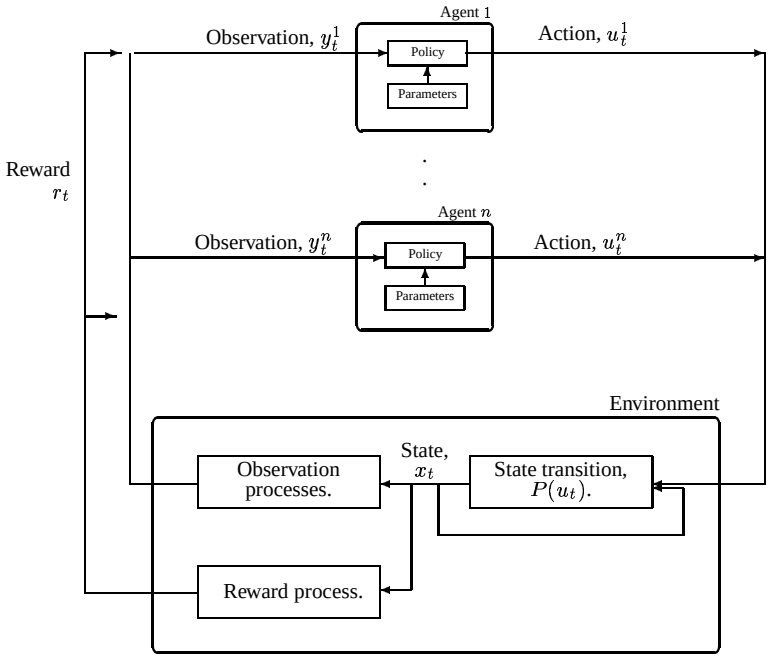

Figure 1: POMDP controlled by n independent agents.

To summarize, provided β is sufficiently close to 1, the long-term average of $r(i_t)z_t$ will converge to a vector that closely approximates the true gradient direction $\nabla\eta(\theta)$. This gives a simple way to compute an appropriate direction to update the parameters θ. An on-line algorithm (OLPOMDP) was presented in [10] that updates the parameters θ according to

$$\theta_t = \theta_{t-1} + \gamma r_t z_t, \tag{2}$$

where the small positive real number γ is the size of the steps taken in parameter space. If these steps are sufficiently small, so that the parameters change slowly, this update rule modifies the parameters in the direction that maximally increases the long-term average of the reward.

3 Direct reinforcement learning with independent agents

Suppose that, instead of a single agent, there are n independent agents, all cooperating to maximize the average reward (see Figure 1). Suppose that each of these agents sees a distinct observation vector, and has a distinct parameterized randomized policy that depends on its own set of parameters. This multi-agent reinforcement learning problem can also be modelled as a POMDP by considering the collection of agents as a single agent, with an observation vector that consists of the n observation vectors of each independent agent, and similarly for the parameter vector and action vector. The following decomposition theorem follows from a simple calculation.

Theorem 1. *For a* POMDP *controlled by multiple independent agents, the direct reinforcement learning update equations (1) and (2) for the combined agent are equivalent to those that would be used by each agent if it ignored the existence of the other agents. That is, if we let y_t^i denote the observation vector for agent i, u_t^i denote the action it takes, and θ^i denote its parameter vector, then the update equation (2) is equivalent to the system of*

n update equations,

$$\theta_t^i = \theta_{t-1}^i + \gamma r_t z_t^i, \tag{3}$$

where the vectors $z_t^1, \ldots, z_t^n \in \mathbb{R}^k$ are updated according to

$$z_{t+1}^i = \beta z_t^i + \frac{\nabla \mu_{u_t^i}\left(y_t^i, \theta^i\right)}{\mu_{u_t^i}\left(y_t^i, \theta^i\right)}. \tag{4}$$

Here, ∇ denotes the gradient with respect to the agent's parameters θ^i.

Effectively, each agent treats the other agents as a part of the environment, and can update its own behaviour while remaining oblivious to the existence of the other agents. The only communication that occurs between these cooperating agents is via the globally distributed reward, and via whatever influence agents' actions have on other agents' observations. Nonetheless, in the space of parameters of all n agents, the updates (3) adjust the complete parameter vector (the concatenation of the vectors θ^i) in the gradient direction of the average reward. We shall see in the next section that this convenient property leads to a synaptic update rule for spiking neurons that involves only local quantities, plus a global reward signal.

4 Direct reinforcement learning in spiking neural networks

The networks we consider contain simple models of spiking neurons, operate in discrete time, and we assume that each neuron in the network can choose one of two actions at time step t: to fire, or not to fire. We represent these actions with the notation $u_t = 1$ and $u_t = 0$, respectively. We use the following simple probabilistic model for the behaviour of the neuron. Define the *potential* v_t in the neuron at time t as $v_t = \sum_j w_j u_{t-1}^j$, where w_j is the connection strength of the jth synapse and u_{t-1}^j is the activity at the previous time step of the presynaptic neuron at the jth synapse. Then,

$$\Pr\left(\text{neuron fires at time } t\right) = \Pr\left(u_t = 1\right) = \sigma\left(v_t\right), \tag{5}$$

where $\sigma(\alpha) = 1/(1 + e^{-\alpha})$.

A real-valued global reward signal r_t is broadcast to every neuron in the network at time t. We view each (non-input) neuron as an independent agent in a reinforcement learning problem. The agent's (neuron's) policy is simply how it chooses to fire given the activities on its presynaptic inputs. The synaptic strengths (w_j) are the adjustable parameters of this policy. Theorem 1 shows how to update the synaptic strengths in the direction that maximally increases the long-term average of the reward. A simple calculation results in an update rule for the j-th synaptic strength of

$$w_{j,t+1} = w_{j,t} + \gamma r_{t+1} z_{j,t+1}, \tag{6}$$

where the $z_{j,t}$ are updated according to

$$z_{j,t+1} = \beta z_{j,t} + \left(u_t - \sigma\left(v_t\right)\right) u_{t-1}^j. \tag{7}$$

These equations describe the updates for the parameters in a single neuron. The pseudocode in Algorithm 1 gives a complete description of the steps involved in computing neuron activities and synaptic modifications for a network of such neurons. Suitable values for the quantities β and γ required by Algorithm 1 depend on the mixing time of the controlled POMDP. The coefficient β sets the decay rate of the variable z_t. For the algorithm to accurately approximate the gradient direction, the corresponding time constant, $1/(1-\beta)$, should be large compared with the mixing time of the environment. The step size γ affects the rate of change of the parameters. When the parameters are constant, the long term average of $r_t z_t$ approximates the gradient. Thus, the step size γ should be sufficiently small so that the parameters are approximately constant over a time scale that allows an accurate estimate. Again, this depends on the mixing time. Loosely speaking, both $1/(1-\beta)$ and $1/\gamma$ should be significantly larger than the mixing time.

Algorithm 1 Model of neural network activity and synaptic modification.

Given:

Coefficient $\beta \in [0, 1)$,
Step size γ,
Initial synaptic connection strengths of the i-th neuron $w^i_{j,0}$.

for time $t = 0, 1, \ldots$ **do**
  Set activities u^j_t of input neurons.
  for non-input neurons i **do**
    Calculate potential $v^i_{t+1} = \sum_j w^i_{j,t} u^j_t$.
    Generate activity $u^i_{t+1} \in \{0, 1\}$ using $\Pr\left(u^i_{t+1} = 1\right) = \sigma\left(v^i_{t+1}\right)$.
  end for
  Observe reward r_{t+1} (which depends on network outputs).
  for non-input neurons i **do**
    Set $z^i_{j,t+1} = \beta z^i_{j,t+1} + \left(u^i_t - \sigma\left(v^i_t\right)\right) u^j_{t-1}$.
    Set $w^i_{j,t+1} = w^i_{j,t} + \gamma r_{t+1} z^i_{j,t+1}$.
  end for
end for

5 Biological Considerations

In modifying the strength of a synaptic connection, the update rule described by Equations (6) and (7) involves two components. There is a Hebbian component ($u_t u^j_{t-1}$) that helps to increase the synaptic connection strength when firing of the postsynaptic neuron follows firing of the presynaptic neuron. When the firing of the presynaptic neuron is not followed by postsynaptic firing, this component is 0, while the second component ($-\sigma(v_t)\, u^j_{t-1}$) helps to decrease the synaptic connection strength. The update rule has several attractive properties:

Locality. The modification of a particular synapse w_j involves the postsynaptic potential v, the postsynaptic activity u, and the presynaptic activity u^j at the previous time step. Certainly the postsynaptic potential is available at the synapse. Action potentials in neurons are transmitted back up the dendritic tree [27], so that (after some delay) the postsynaptic activity is also available at the synapse. Since the influence of presynaptic activity on the postsynaptic potential is mediated by receptors at the synapse, evidence of presynaptic activity is also available at the synapse. While Equation (7) requires information about the *history* of presynaptic activity, there is some evidence for mechanisms that allow recent receptor activation to be remembered [21, 24]. Hence, all of the quantities required for the computation of the variable z_j are likely to be available in the postsynaptic region.

Simplicity. The computation of z_j in (7) involves only additions and subtractions modulated by the presynaptic and postsynaptic activities, and combined in a simple first order filter. This filter is a leaky integrator which models, for instance, such common features as the concentration of ions in some region of a cell or the potential across a membrane. Similarly, the connection strength updates described by Equation (6) involve simply the addition of a term that is modulated by the reward signal.

Optimality. The results from [8, 3], together with Theorem 1, show that this simple update rule modifies the network parameters in the direction that maximally increases the average reward, so it leads to parameter values that locally optimize the performance of the network.

There are some experimental results that are consistent with the involvement of the correlation component (the term $(u_t - \sigma(v_t))u^j_{t-1}$) in the parameter updates. For instance,

a large body of literature on long-term potentiation (beginning with [11]) describes the enhancement of synaptic efficacy following association of presynaptic and postsynaptic activities. More recently, the importance of the relative timing of the EPSPs and APs has been demonstrated [21, 24]. In particular, the postsynaptic firing must occur after the EPSP for enhancement to occur. The backpropagation of the action potential up the dendritic tree appears to be crucial for this [19].

There is also experimental evidence that presynaptic activity without the generation of an action potential in the postsynaptic cell can lead to a decrease in the connection strength [26]. The recent finding [21, 24] that an EPSP occurring shortly *after* an AP can lead to depression is also consistent with this aspect of Hebbian learning. However, in the experiments reported in [21, 24], the presence of the AP appeared to be important. It is not clear if the significance of the relative timings of the EPSPs and APs is related to learning or to maintaining stability in bidirectionally coupled cells. Finally, some experiments have demonstrated a decrease in synaptic efficacy when the synapses were not involved in the production of an action potential [18].

The update rule also requires a reward signal that is broadcast to all neurons in the network. In all of the experiments mentioned above, the synaptic modifications were observed without any evidence of the presence of a plausible reward signal. However, there is some limited evidence for such a signal in brains. It could be delivered in the form of particular neurotransmitters, such as serotonin or nor-adrenaline, to all neurons in a circuit. Both of these neurotransmitters are delivered to the cortex by small cell assemblies (the raphe nucleus and the locus coeruleus, respectively) that innervate large regions of the cortex. The fact that these assemblies contain few cell bodies suggests that they carry only limited information. It may be that the reward signal is transmitted first electrically from one of these cell assemblies, and then by diffusion of the neurotransmitter to all of the plastic synaptic connections in a neural circuit. This would save the expense of a synapse delivering the reward signal to every plastic connection, but could be significantly slower. This need not be a disadvantage; for the purposes of parameter optimization, the required rate of delivery of the reward signal depends on the time constants of the task, and can be substantially slower than cell signalling times. There is evidence that the local application of serotonin immediately after limited synaptic activity can lead to long term facilitation [14].

6 Simulation Results

We simulated Algorithm 1 on the sonar signal classification problem studied by Gorman and Sejnowski [15] and a 2-D inverted pendulum control problem, in both cases using a feedforward architecture with one hidden layer of neurons. In both cases the network achieved substantial performance improvement. For more details of these experiments see the full technical report [2].

References

[1] L. Baird and A. Moore. Gradient descent for general reinforcement learning. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

[2] P. L. Bartlett and J. Baxter. Hebbian Synaptic Modifications in Spiking Neurons that Learn. Technical report, Australian National University, November 1999.

[3] P. L. Bartlett and J. Baxter. Estimation and approximation bounds for gradient-based reinforcement learning. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, 2000. To Appear.

[4] A. G. Barto, C. W. Anderson, and R. S. Sutton. Synthesis of nonlinear control surfaces by a layered associative search network. *Biological Cybernetics*, 43:175–185, 1982.

[5] A. G. Barto and R. S. Sutton. Landmark learning: An illustration of associative search. *Biological Cybernetics*, 42:1–8, 1981.

[6] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13:834–846, 1983.

[7] A. G. Barto, R. S. Sutton, and P. S. Brouwer. Associative search network: A reinforcement learning associative memory. *Biological Cybernetics*, 40:201–211, 1981.

[8] J. Baxter and P. L. Bartlett. Direct Gradient-Based Reinforcement Learning: I. Gradient Estimation Algorithms. Technical report, Research School of Information Sciences and Engineering, Australian National University, July 1999.

[9] J. Baxter and P. L. Bartlett. Reinforcement learning in pomdp's via direct gradient ascent. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[10] J. Baxter, L. Weaver, and P. L. Bartlett. Direct Gradient-Based Reinforcement Learning: II. Gradient Descent Algorithms and Experiments. Technical report, Research School of Information Sciences and Engineering, Australian National University, September 1999.

[11] T. V. Bliss and T. Lomo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology (London)*, 232:331–356, 1973.

[12] X.-R. Cao and H.-F. Chen. Perturbation Realization, Potentials, and Sensitivity Analysis of Markov Processes. *IEEE Transactions on Automatic Control*, 42:1382–1393, 1997.

[13] X.-R. Cao and Y.-W. Wan. Algorithms for Sensitivity Analysis of Markov Chains Through Potentials and Perturbation Realization. *IEEE Transactions on Control Systems Technology*, 6:482–492, 1998.

[14] G. A. Clark and E. R. Kandel. Induction of long-term facilitation in *Aplysia* sensory neurons by local application of serotonin to remote synapses. *Proc. Natl. Acad. Sci. USA*, 90:11411–11415, 1993.

[15] R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[16] D. O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.

[17] H. Kimura, K. Miyazaki, and S. Kobayashi. Reinforcement learning in POMDPs with function approximation. In D. H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 152–160, 1997.

[18] Y. Lo and M. ming Poo. Activity-dependent synaptic competition *in vitro*: Heterosynaptic suppression of developing synapses. *Science*, 254:1019–1022, 1991.

[19] J. C. Magee and D. Johnston. A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons. *Science*, 275:209–213, 1997.

[20] P. Marbach and J. N. Tsitsiklis. Simulation-Based Optimization of Markov Reward Processes. Technical report, MIT, 1998.

[21] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–215, 1997.

[22] J. F. Medina and M. D. Mauk. Simulations of cerebellar motor learning: Computational analysis of plasticity at the mossy fiber to deep nucleus synapse. *The Journal of Neuroscience*, 19(16):7140–7151, 1999.

[23] N. M. L. Peshkin, K.-E. Kim, and L. P. Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence*, 1999.

[24] G. qiang Bi and M. ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience*, 18(24):10464–10472, 1998.

[25] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[26] P. K. Stanton and T. J. Sejnowski. Associative long-term depression in the hippocampus induced by Hebbian covariance. *Nature*, 339:215–218, 1989.

[27] G. J. Stuart and B. Sakmann. Active propagation of somatic action potentials into neocortical pyramidal cell dendrites. *Nature*, 367:69–72, 1994.

[28] G. Tesauro. Simple neural models of classical conditioning. *Biological Cybernetics*, 55:187–200, 1986.

[29] R. J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 1992.