Graduate Training Centre of Neuroscience
Computational Neuroscience
University of Tübingen

Reinforcement Learning in Biological Neural Networks

Author:
Akif Erdem Sağtekin

Supervised by:
Georgy Antonov, Peter Dayan
Max Planck Institute for Biological Cybernetics

Abstract

Reinforcement learning (RL) framework provides experimentally justified explanations for animal learning. On the other hand, animal learning has been extensively associated with synaptic plasticity, which is commonly studied by employing spiking neural networks in computational neuroscience literature. In this report, I investigate the literature that explores the relationship between synaptic plasticity rules and the RL framework. The literature can be divided into three parts: 1) using policy-gradient methods to analytically derive plasticity rules, 2) modulating the STDP rule by a global reward signal, and 3) using temporal-difference learning to find plasticity rules.

Contents

1 Introduction

Alongside supervised and unsupervised learning, reinforcement learning is one of the fundamental paradigms for machine learning and animal learning. In the reinforcement learning paradigm, the system's output elicits evaluative feedback, which can manifest as a reward, absence of reward, or punishment -I will refer to this evaluative feedback as "reward." Reinforcement learning is characterized by an agent's interaction with an environment, receiving rewards, and trying to accomplish an objective through improvement, for example trying to maximize its expected cumulative reward [1].

In supervised learning, the system's output is compared to the true output based on a distance metric, and the system parameters are adjusted accordingly. In reinforcement learning however, instead of directly providing the true output and comparing it to the system's output, the system is informed whether its output was true or not —where the term 'evaluative' term comes from. This characteristic renders the reinforcement learning (RL) paradigm more versatile across various tasks compared to supervised learning, which typically necessitates a true label for each input. RL, on the other hand, allows for evaluative feedback to the system.

Animals can learn about the given stimuli or their actions by evaluative feedback. Reinforcement learning framework may provide a robust and experimentally justified explanations for animal learning, whether it be playing table tennis or simply walking [1]. The main goal of this report is to provide possible explanations on how reinforcement learning framework might be implemented by the brain's wetware.

It has been commonly postulated that the changes of synaptic strength between neurons, i.e. plasticity, is the neurophysiological basis of learning [2]. Spike patterns of the neurons and neuromodulation (e.g. relative concentration of a neuromodulatory molecule such as dopamine) are some examples for plasticity factors. One of the ways of studying synaptic plasticity is via spiking neural networks (SNNs) by imitating the behaviours of biological neurons mathematically, or with simulations. In this report, I attempt to review the literature that investigates synaptic plasticity rules in spiking neural networks that are inspired by the reinforcement learning framework.

The structure of the essay is as follows: In section 2, I briefly summarize some of the different RL paradigms that are the subject of this essay. In section 3, I outline few plasticity rules for biological neural networks. In section 4, I present the papers that implement RL paradigms in (spiking) neural networks. Section 4 is the core part of this report. Lastly, I conclude by section 5, discussion.

2 Reinforcement Learning Framework

The true state value ($v_\pi(s)$) under policy π is defined by the expected value of the return, G_t, where policy is defined by $\pi(a|s,\theta) = Pr\{A_t = a|S_t = s, \theta_t = \theta\}$ for the probability of an action a taken at time t given that the environment is in state s at time t with parameter θ. And the return is defined as expected value of the (discounted) rewards. Discount factor is denoted as γ. In episodic tasks, the last reward value is denoted as R_T, where T is the episode length. One of the goals of the RL algorithms is to find a policy that maximizes the return [1].

$$v_\pi(s) = \mathbb{E}[G_t|S_t = s] \tag{1}$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} \cdots \tag{2}$$
$$= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} \cdots) \tag{3}$$
$$= R_{t+1} + \gamma G_{t+1} \tag{4}$$

2.1 Value-function Approximation

Value-function approximation methods map the tabular values of states to a parameterized function, reducing the dimensionality of the environment. The approximated (by the $\mathbf{w}$) value of a state s can be denoted by $\hat{v}(s, \mathbf{w}) \approx v_\pi(s)$. The approximation can be done by a linear function, or by a neural network. For the linear case, $\hat{v}(s, \mathbf{w}) = \mathbf{w}^T \mathbf{x}(s) = \sum_i w_i x_i(s)$.

Reducing the dimensionality allows this algorithm to be implemented when the states are partially observed as well [1]. Also, since a change of a single weight would change the value of more than one states, the algorithm also enables generalization. But this comes with a cost. Notice that for RL algorithms that uses values, the state values are updated towards their targets. For example for Monte Carlo update, the target value prediction is G_t. It is apparent from the approximation that finding the exact target values for each state is not possible. Bringing a particular state closer to its target value will inevitably result in another state moving farther away. For that reason, introducing a parameter which assigns an importance to each state is a good strategy. I refer the reader to the textbook [1] for a mathematically rigorous explanation and derivation.

Assume that the target value U_t for each state is given. Updating the weights should reduce the difference between each observed state's value and its true value. This can be done by the stochastic-gradient method.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{1}{2}\alpha \nabla [U_t - \hat{v}(S_t, \mathbf{w}_t)]^2 \tag{5}$$
$$= \mathbf{w}_t + \alpha [U_t - \hat{v}(S_t, \mathbf{w}_t)] \nabla \hat{v}(S_t, \mathbf{w}_t). \tag{6}$$

Setting the U_t to the return (G_t) would yield the Monte Carlo algorithm for value-approximation. After approximating the value function, a greedy policy can be chosen in which the agent prefers to take actions that leads to higher-valued states.

2.2 Policy Gradient Methods

Another approach is to directly parameterize the policy, and changing the policy parameters in order to increase the reward.

Policy-gradient methods try to improve the performance of the agent by learning the optimum policy parameters. Assume that the action space is discrete and not too large, then one can parameterize the preferences of actions, denoted by $h(s, a, \theta)$. One possible policy is the exponential soft-max.

$$\pi(a|s, \theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}} \tag{7}$$

The parameters, θ, can be linear in features or can be computed by a neural network. They are updated

by using gradient ascent in a performance measure $J(\theta)$. The performance measure can be chosen in different ways, but an intuitive way of thinking about it is defining the return as a performance measure for episodic tasks.

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_\mathbf{t})} \tag{8}$$

The exact derivation of the *policy-gradient theorem* is out of the scope of this report. Here, I will only provide the final REINFORCE update rule.

$$\theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \tag{9}$$

This algorithm, along with others, will be used to derive plasticity rules for (spiking) neural networks in Section 4.

2.3 Temporal-Difference Learning

Temporal-difference (TD) learning is one of the ways of combining the dynamic-programming (DP) and Monte Carlo (MC) methods. DP assumes that agent has full accessibility to the environment, allowing replacing the expected value of the return at time $t+1$ with expected value of $v_\pi(S_{t+1})$.

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] && (10)\\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s] && (11)\\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s] && (12) \end{aligned}$$

DP algorithm does not need to wait until the end of the episode since it can use $v_\pi(S_{t+1})$.

On the other hand, MC methods assumes that the agent learns about the environment only after experiencing it. Therefore, the return is only available to the agent after it finishes the episode. A reinforcement learning agent has no full access to the environment, it needs to experience, which makes MC method appealing. But having to wait until the episode finished makes the MC method unappealing. TD learning combines part of the DP with MC, by updating the value function at each time using the sampled (not expected) value of $v_\pi(S_{t+1})$. But since it will not hold for each iteration, we can define an error.

$$\delta(t) = R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}_t) - \hat{v}(S_t,\mathbf{w}_t) \tag{13}$$

TD error can be used for value-function approximation. Note that U_t is denoted as target before, in Equation 5 and 6. Here, U_t is replaced with the $R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}_t)$. Therefore Equation 6 becomes:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\delta(t)\nabla\hat{v}(S_t,\mathbf{w}_t). \tag{14}$$

Using linear features, since $\nabla\hat{v}(S,\mathbf{w}) = \mathbf{x}(s)$:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\delta(t)\mathbf{x}(s). \tag{15}$$

I refer the reader to the textbook [1] for more explanation.

2.4 Actor-Critic Methods

Consider the REINFORCE update given in Section 2.2, and consider the Equation 9. Replacing G_t by $\delta(t)$ simply incorporates the TD learning with policy-gradient methods. By doing that, the return is

replaced with a 'critic'. Before, the return was informing the algorithm for action updates. Now, it is the 'critic', criticising the action taken. With this replacement, algorithm does not need to wait until the end of the episode, but it needs to update the critic as well, which can be done by updating the values. Replacing U_t by $R_{t+1} + \gamma\hat{v}(S_{t+1}, \mathbf{w})$ (which gives $\delta(t)$) in Equation 6 simply incorporates TD learning in value-approximation method.

Therefore TD error is used to both update the parameters of the critic and the actor. The updates for each state for each iteration, after taking action A and observing S' and R is as follows:

$$\delta \leftarrow R + \gamma\hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w}) \tag{16}$$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha^w \delta \nabla\hat{v}(S, \mathbf{w}) \tag{17}$$

$$\theta \leftarrow \theta + \alpha^\theta \delta \frac{\nabla\pi(A|S, \theta)}{\pi(A|S, \theta)} \tag{18}$$

2.5 Eligibility Traces

In value-approximation, weights ($\mathbf{w}$) can be updated proportionally to the multiplication of TD error (δ) and derivative of the estimated value function with respect to its weights ($\nabla\hat{v}(S, \mathbf{w})$). As noted before, for the linear case, $\hat{v}(S, \mathbf{w}) \doteq \mathbf{w}^T\mathbf{x}(s)$, therefore the $\nabla\hat{v}(S, \mathbf{w})$ simply becomes $\mathbf{x}(s)$. Consequently, in the linear case, the weights are updated proportionally to the current TD error and current input vectors, when the state is visited (or when the action is taken, depending on the algorithm). But the agent can also benefit from by updating its weights considering the past input vectors as well. The idea behind eligibility traces closely related with λ-return, but that perspective is not mentioned in this report. In order to include the past input effects to the weight update, a new variable can be introduced which keeps the track of the input vectors by applying a low-pass filter.

$$\mathbf{z}_{-1} = \mathbf{0}, \tag{19}$$

$$\mathbf{z}_t = \mathbf{z}_{t-1} + \gamma\lambda\nabla\hat{v}(S_t, \mathbf{w}_t), \quad 0 \leq t \leq T. \tag{20}$$

For policy-gradient methods, rather than value-function approximation, the eligibility trace would be as following:

$$\mathbf{z}_{-1} = \mathbf{0}, \tag{21}$$

$$\mathbf{z}_t = \mathbf{z}_{t-1} + \gamma\lambda \frac{\nabla\pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}, \quad 0 \leq t \leq T. \tag{22}$$

Corresponding eligibility traces can be replaced with the gradient of the value or gradient of the policy (normalized) to get the eligibility trace-incorporated version of the algorithms presented above.

3 Plasticity Rules

In order to see the connections between RL algorithms and some plasticity rules that are widely used in computational neuroscience, I will first simply introduce few plasticity rules. Although the relationship between spike-timing dependent plasticity (STDP) and RL algorithms is more widely mentioned in this report, I believe that mentioning other plasticity rules is also helpful to provide more insight about Hebbian plasticity rules.

3.1 Basic Hebbian Rule

One can construct a plasticity rule that directly captures the Donald Hebb's speculation, "When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some

growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." [3].

$$\triangle w_{ij} = f_1(pre_j) f_2(post_i) \tag{23}$$

The weight change captures the correlation between the function of pre-synaptic neuron activity ($f_1(pre_j)$) and post-synaptic neuron activity ($f_2(post_i)$) [2]. The form of synaptic change in Equation 22 will be referred as a local plasticity rule since it only depends on the pre-synaptic and post-synaptic neuron. See [4] for more generalized definition of the locality of synaptic rules.

If these two functions are linear with the activity of pre- and post-synaptic neurons, since the activity of the neurons are positive by its definition, the weight can not decrease by time which makes the plasticity rule unstable. A threshold factor for either pre- or post-synaptic neuron can be introduced to the rule, which would allow the decrease of weight when the activity of the neuron is less then the threshold.

3.2 BCM Rule

One famous form of Hebbian learning is BCM (Bienenstock, Cooper, and Munro) rule [5]. Note that in case of including a threshold factor for pre-synaptic neuron, long-term depression (LTD) does not require pre-synaptic activity, since LTD would also occur when pre-synaptic activity is 0.

$$\tau_w \frac{d\mathbf{w}}{dt} = v(\mathbf{u} - \theta_{\mathbf{u}}), \tag{24}$$

where τ_w is the time scale of the weight vector ($\mathbf{w}$) change, v is post-synaptic neuron activity, $\mathbf{u}$ is the vector for pre-synaptic neuron activities, and $\theta_{\mathbf{u}}$ is pre-synaptic threshold.

Introducing a threshold only for the post-synaptic neuron leads to a similar problem. The BCM rule requires both, and has substantial amount of experimental support [5]:

$$\tau_w \frac{d\mathbf{w}}{dt} = v(v - \theta_v)\mathbf{u} \tag{25}$$

$$\tau_\theta \frac{d\theta_v}{dt} = v^2 - \theta_v \tag{26}$$

where τ_θ is the time-scale of the sliding-threshold, and $\theta_{\mathbf{v}}$ post-synaptic threshold.

In the BCM rule, if the pre-synaptic activity is non-zero, and the post-synaptic activity exceeds the threshold, it results in long-term potentiation (LTP); otherwise, if it is below the threshold, it leads to LTD (Figure 1). However, the threshold itself also depends on the activity of the post-synaptic neuron. If the post-synaptic neuron activity has been increased by the pre-synaptic activity, then the sliding-threshold increases. Therefore when post-synaptic neuron activity is small, the area for LTP is more than LTD (Figure 1, middle). Conversely, when the post-synaptic neuron activity is strong, then the area of LTD is more than LTP (Figure 1, right). Therefore, the plasticity rule is intrinsically stable as a result of the sliding threshold. I refer the reader to the BCM paper [BCM] for mathematically rigorous explanation.

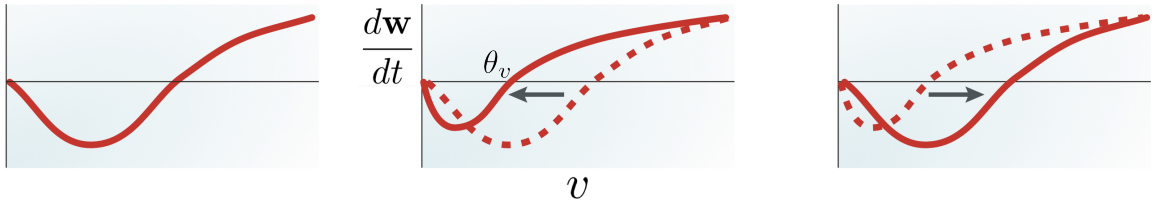


Figure 1: BCM rule. y-axis determines the synaptic change with respect to v, post-synaptic neuron activity. LTP for $v > \theta_v$, LTD for $v < \theta_v$. θ_v is also modulated by the post-synaptic activity v. The figure demonstrates that different curves during threshold shifting. Figure edited from [REWBCM].

On the other hand, the sliding-threshold also introduces synaptic competition between the synapses. When the post-synaptic neuron activity increases due to a specific pre-synaptic activity, the heightened threshold makes it more challenging for the weights of other synapses to increase. There are various ways for introducing synaptic normalization and competition, I have only introduced BCM since there exist a relationship between a RL algorithm as explained in Section 4.

3.3 Spike-Timing Dependent Plasticity (STDP)

Previously mentioned family of plasticity rules does not require time-dependency on pre- and post-synaptic neurons. STDP rule states that when the pre-synaptic spike precedes post-synaptic spike, the synaptic strength between these two neurons are increases [6], and vice versa (Figure 2). The magnitude of the strength change depends on the relative timing of the spikes. The weight change in given STDP rule is determined by multiplying $F(\triangle t)$ by a maximum weight change value.

$$F(\triangle t) = \begin{cases} A_+ e^{(\triangle t/\tau_+)} & \text{if } \triangle t < 0 \\ -A_- e^{(-\triangle t/\tau_-)} & \text{if } \triangle t \geq 0 \end{cases} \tag{27}$$

where $F(\triangle t)$ determines the synaptic modification amount, normalized by the current synaptic weight. A_+ and A_- determines the synaptic update when the time interval between the spikes ($\triangle t$) is equal to zero, while t_+ and t_- determines the time-scales for which synaptic-change occur. There are more than one types of STDP rules in the literature, additive and multiplicative [7] are some examples.

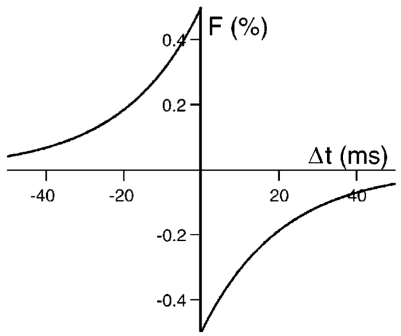


Figure 2: STDP rule. $\triangle t$ denotes pre-synaptic spike time minus post-synaptic spike time. LTP for $\triangle t < 0$ (causal part), LTD for $\triangle t \geq 0$ (anti-causal part). $F(\triangle t)$ determines the synaptic modification amount, normalized by the current synaptic weight. Figure from [6].

4 Bridging the Gap

The literature that uses RL framework to implement plasticity rules for (spiking) neural networks can be divided into 3 parts [8]. 1-) Top-down approach of directly applying the policy-gradient methods in (spiking) neural networks. 2-) Modulating the STDP rule with a globally broadcasted reward function. 3-) Top-down approach of applying TD learning algorithm, especially using actor-critic methods. In this report, I will be broadly mentioning the first two parts while just briefly touching the third part.

Starting with policy-gradient methods, I will initially provide a thorough introduction to two papers, as they serve as a backbone for subsequent works, and present the derivation of the plasticity rules outlined in these papers. Subsequently, I will discuss the connections between the reward-modulated STDP rules and the derived (using a top-down approach) plasticity rules. Finally, I will address the findings regarding the learning efficacy of the reward-modulated STDP rules. The BCM rule will also be very briefly mentioned in the context of reinforcement learning. Afterwards, I will also briefly mention the papers that used TD learning algorithms.

4.1 Applying policy-gradient methods in (spiking) neural networks

In 1999, P. L. Bartlett and J. Baxter published a paper [9] that led to a chain of publications attempting to directly adopt a reinforcement learning algorithm, REINFORCE, for biological neural networks. Machine learning rules are usually non-local, which is thought to be not easily biologically implementable by the brain [10]. However, REINFORCE naturally leads to a local plasticity rule combined with a global reward. Global reward refers to the fact that the reward signal is broadcasted to all neurons. Local part of the plasticity rule is dependent on the activity of certain pre-synaptic and post-synaptic neuron pair, denoted by subscripts j and i, and denoted by functions f_1 and f_2, and the function for global reward is shown by f_3.

$$\triangle w_{ij} = \underbrace{f_1(pre_j) f_2(post_i)}_{\text{Local}} \underbrace{f_3(reward)}_{\text{Global}} \tag{28}$$

The authors considered each neuron as an individual agent, where each neuron treats other neurons as a part of its environment. Pre-synaptic neurons (input neurons) connects to post-synaptic neurons (output neurons). The effect of each pre-synaptic spike on the certain post-synaptic neuron is determined via the connection strengths, w_j (Figure 3). Post-synaptic potential at time t is weighted sum of arrived pre-synaptic spikes at time $t-1$.

$$v_t = \sum_j w_j u_{t-1}^j, \tag{29}$$

u_{t-1}^j is a binary value of the pre-synaptic neuron j at time $t-1$ denoting the existence of a spike, and v_t is the membrane potential of the certain post-synaptic neuron at time t.

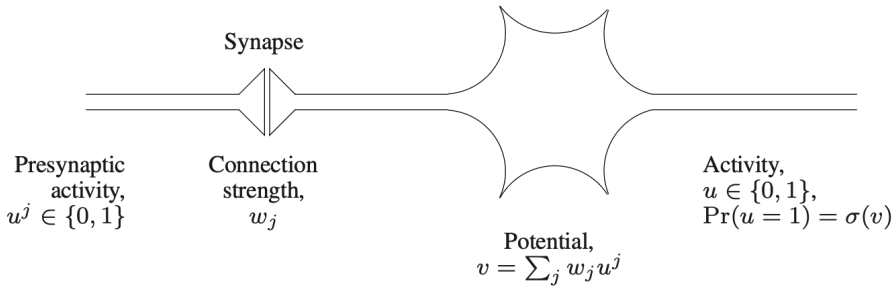


Figure 3: The activity of the post-synaptic neuron is determined by its potential, which is weighted sum of pre-synaptic spikes. Note that in the figure time subscript is dropped. Figure from [11].

Action space is binary, where each post-synaptic neuron can take two different actions, fire or not to fire, denoted by $u_t = 1$ and $u_t = 0$, respectively (Figure 3). The activity of the post-synaptic neuron (to spike or not to spike) is determined by its potential. The policy of the neuron is parameterized by:

$$\pi(u_t = 1 \mid v_t) = Pr(u_t = 1 \mid v_t) = \sigma(v_t) \tag{30}$$

$$\pi(u_t = 0 \mid v_t) = Pr(u_t = 0 \mid v_t) = 1 - \sigma(v_t) \tag{31}$$

where $\sigma(x) = 1/(1 + e^x)$. Although the authors referred to it as a spiking neuron, in current computational neuroscience literature, it usually would not be considered as an actual spiking neuron since there is no threshold (adaptive or non-adaptive) that determines the spike time.

Combining the REINFORCE update (Eq. 8) with eligibility traces (Eq. 21) and referring the parameters of the system as w, we get:

$$w_{j,t+1} = w_{j,t} + \alpha R_{t+1} z_{j,t+1}, \tag{32}$$

$$z_{j,t+1} = \beta z_{j,t} + \frac{\frac{\partial}{\partial w_j}\mu_{u_t}}{\mu_{u_t}} \tag{33}$$

Taking the derivative of the policy with respect to the weights, and dividing it by the policy itself yields:

$$z_{j,t+1} = \beta z_{j,t} + (u_t - \sigma(v_t))u^j_{t-1}. \tag{34}$$

Notice that the learning rule consist of two parts: a scalar global reward signal R which is assumed to be broadcasting to every neuron at time t, and the local Hebbian term of $(u_t - \sigma(v_t))u^j_{t-1}$. The first factor of the Hebbian term is only dependent on the post-synaptic neuron (whether it spiked or not minus the probability of the spike), while the second factor is only dependent on the pre-synaptic neuron.

This learning rule predicts that employing a global broadcasting signal to modulate the local Hebbian plasticity rule can be implemented for learning purposes. A closer examination of the equation reveals that the rule predicts synaptic strengthening when a pre-synaptic neuron precedes a post-synaptic neuron and if a positive reward is given. This contradicts with conventional STDP experiments [STDP], as the derived rule also necessitates a reward signal for synaptic change to occur.

Interestingly, the rule enables LTD even when the pre-synaptic neuron precedes the post-synaptic neuron, when the reward is negative. Moreover, if the pre-synaptic neuron fires, in the absence of post-synaptic spikes (indicating a negative value for the first factor of local Hebbian term), LTD occurs

with a positive reward, and LTP occurs with a negative reward. In the paper, the authors dismissed the possibility of a negative reward. However, as proposed by other papers mentioned below, it is biologically plausible to consider negative rewards. Theoretically, if the reward signal has a non-zero baseline, values below the baseline can be regarded as negative rewards.

The authors provided examples from some experimental works aligning with their plasticity rule. They noted that in those experiments, the discovered Hebbian rules were not experimentally gated by any neurotransmitter, which would be one of the candidates for global reward broadcasting.

The update rule aligns with the direction of steepest ascent only in two distinct situations, which will also serve as assumptions for the upcoming learning rules derived using the policy-gradient method:

1-) The actions of agents (spikes of neurons) do not have an effect on their inputs, and the reward only depends on the current input. However, when considering a neural network consisting of three parts—input layer, hidden network, and output layer—there are no restrictions on the hidden network. The hidden network can be a recurrent network if it receives inputs from input neurons and outputs to output neurons, provided that the output layer does not project to input neurons.
2-) Or, when the current reward only depends on the inputs after the last reward.

It is worth noting that in the original paper of REINFORCE [12], derived plasticity rule (Eq. 28 and 30) was already mentioned. However, Bartlett and Baxter's paper [9] introduced the idea from a biological perspective, along with its other contributions.

In the paper, the neuron model that considered (Bernoulli unit) does not incorporate the membrane dynamics, such as leak current. In the following paragraphs, I will discuss the papers that build upon the previously mentioned paper, which tries to implement membrane dynamics as well, leading to slights changes on the plasticity rule.

In 2003, X. Xie and H. S. Seung published a paper [11], which can be considered a follow-up to the previously mentioned paper. In short, they concluded that the correlation between reward and spiking fluctuations can be utilized as a biologically plausible learning algorithm. In another paper by H. S. Seung in 2003 [13], similar methods were applied to derive a plasticity rule, but synapses, rather than neurons, were considered as agents. However, since relating the plasticity rule derived in H. S. Seung's paper [13] to STDP is not straightforward, the findings of this paper will not be mentioned in this report (see [14] for yet another plasticity rule involving perturbation of conductances, similar to the node perturbation method in machine learning).

X. Xie and H. S. Seung expanded the Bartlett and Baxter's work by including the membrane dynamics into their neuron model, using Poisson neurons instead of Bernoulli units. They assumed [11] that the neurons produce Poisson spike trains with instantaneous firing rate which was determined by the total synaptic input, which is the weighted sum of low-pass filtered pre-synaptic spikes in the network.

$$\lambda_i(t) = f_i(I_i(t)), \tag{35}$$

$$I_i(t) = \sum_j w_{ij} h_{ij}(t), \tag{36}$$

$$\tau_s \frac{dh_{ij}}{dt} = -h_{ij}(t) + \sum_a \delta(t - f_j^a)\xi_{ij}^a, \tag{37}$$

where $\lambda_i(t)$ is instantaneous firing rate of the ith neuron, f_i is *f-I* curve (Figure 4, bottom), I_i is total synaptic current, w_{ij} is synaptic strength between neuron i and j, and $h_{ij}(t)$ is low-pass filtered incoming spikes (Figure 4, middle), denoted by $\sum_a \delta(t - f_j^a)$ (Figure 4, top), f_j^a being the time of the ath spike of neuron j. ξ_{ij}^a allows model to include dynamic synapses such as short-term plasticity but will not be mentioned in this report for simplicity reasons.

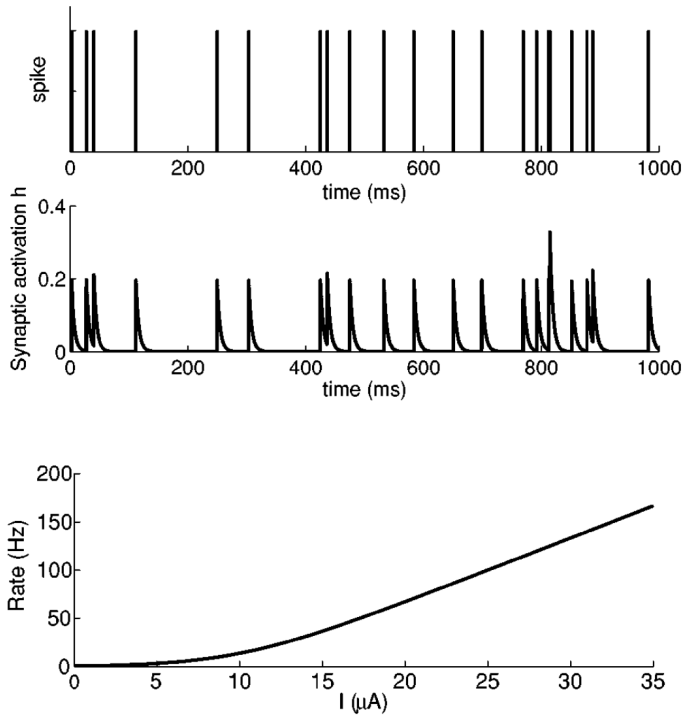


Figure 4: Top: Incoming spikes to neuron i from neuron j. Middle: Corresponded synaptic activation, denoted by $h_{ij}(t)$, which is low-pass filtered version of incoming spikes. Bottom: *f-I* curve. Figure from [11].

The authors show for episodic case that how their synaptic rule updates the weights for maximizing the reward. For the assumptions they made, details of the derivation, and for dynamic synapses, I refer the reader to the original paper [11]. Here, I will assume that we know the REINFORCE update rule (Eq. 9), and directly find the term $\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$.

The probability of a neuron to spike or not to spike, $\sigma_i(t)$, can be written for each time interval $[t, t + \triangle t)$ as:

$$\sigma_i(t) = \begin{cases} 1 \text{ with probability} & p_i(t) = \lambda_i(t)\triangle t \\ 0 \text{ with probability} & 1 - p_i(t) \end{cases} \tag{38}$$

Now, by taking the derivative of the policy and diving it by itself we get (denoted by e_{ij} in the paper):

$$e_{ij} = \sum_{t=0}^{T} \left[\frac{\sigma_i(t)}{p_i(t)} - \frac{1 - \sigma_i(t)}{1 - p_i(t)}\right] \frac{\partial p_i(t)}{\partial w_{ij}} \tag{39}$$

Notice that we concatenated the two different cases (to spike or not to spike): when $\sigma_i(t) = 0$ first term in the first factor becomes zero and when $\sigma_i(t) = 1$ second term in the first factor becomes zero.

Taking the derivative of $p_i(t)$ depends on $\lambda_i(t)$, which in turn is a function of f_i, itself dependent on $I_i(t)$, and ultimately linked to w_{ij}. After rearranging the terms and taking the limit of $\triangle t$ to zero, that can be followed in the paper [11], the final rule that is to be gated by the reward signal is:

$$e_{ij} = \int_0^T \left[\frac{(s_i(t) - f_i(t))}{f_i(t)} f_i'(t) \right] h_{ij}(t)\, dt, \tag{40}$$

where T is the episode length and $s_i(t)$ is the post-synaptic spike train denoted by $s_i(t) = \sum_a \delta(t - f_i^a)$. Investigating the update rule more closely, first, we notice that it is Hebbian. The terms encapsulated in the brackets is only dependent on post-synaptic neuron, while $h_{ij}(t)$ is only dependent on pre-synaptic neuron.

Second, the update makes use of spike fluctuations. The term $s_i(t) - f_i(t)$, integrated over the time period T, compares the number of spikes emitted by the post-synaptic neuron ($s_i(t)$) and the average firing rate ($f_i(t)$). Although the expected value of this difference is zero, the model exploits the trial-by-trial fluctuations. Note that when number of neurons in the network increase, fluctuations will decrease which would deteriorate the capability of the rule.

This fluctuation is scaled by the term $f_i'(t)/f_i(t)$ (Figure 5). The authors noted that since the scaling term decreases for large synaptic inputs, the term acts as a stabilizer. I would also like to note that since the variance of the Poisson process is equal to its mean, the scaling term penalizes potential large fluctuations while amplifying the impact of possible small fluctuations.

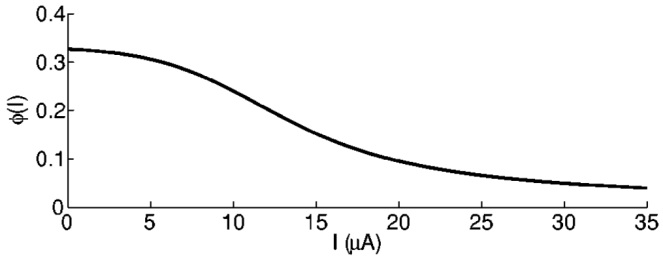


Figure 5: Scaling term, $f_i'(t)/f_i(t)$. Figure from [11].

The authors used the online version of their algorithm to show the capabilities of their update rule, using two different tasks: XOR problem and learning direction selectivity. For the XOR problem, the input data is {[1,0],[0,1],[1,1],[0,0]} with {1,1,0,0} being their desired outputs. They constructed their network with 2 input neurons, 10 hidden neurons, and 1 output neuron for the XOR problem. Input data 1 was encoded by a Poisson spike train with a 200Hz rate, and 0 was encoded by a 5Hz rate. For instance, for the input [1,1], both input neurons received a Poisson spike train with a 200Hz rate. The network received a negative reward of R = -1 each time the output neuron fired a spike for the inputs [1,1] and [0,0], and a positive reward of $R = 2$ each time the output neuron fired a spike for the inputs [0,1] and [1,0].

They found out that the network learns the XOR task (Figure 6) by balancing the excitation and inhibition of the hidden neurons (Figure 7, b), i.e. each hidden neuron receives one excitatory input from one input neuron and one inhibitory input from the other input neuron. However since there are no constrictions on any input neuron's projections, one neuron can excite one hidden neuron while inhibiting the other hidden neuron (Figure 7, b).

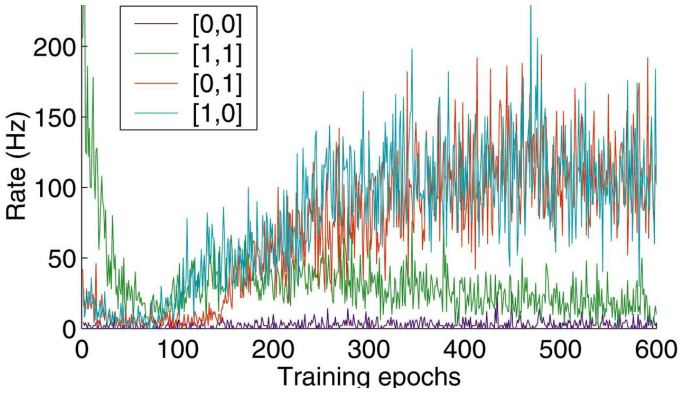


Figure 6: Firing rate of the output neuron given the input, during the learning process. The activity of the output neuron becomes stronger for the inputs [0,1] and [1,0]. Figure from [11].

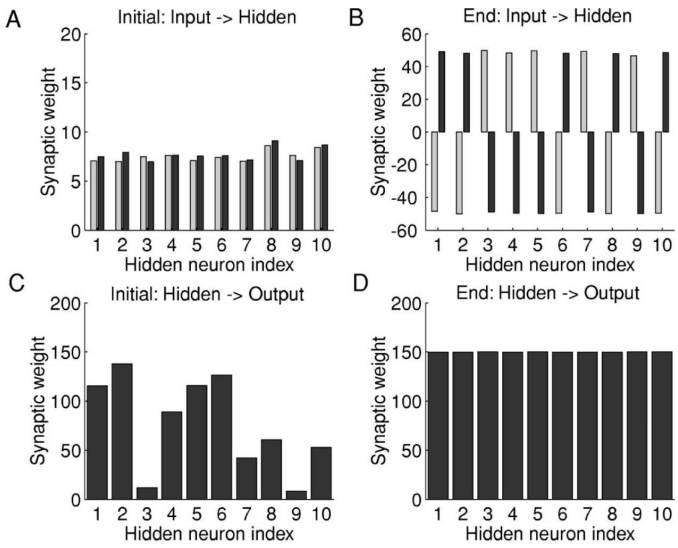


Figure 7: Initialized weights before training (A and C) and learned weights after training (B and D). After training, each hidden neuron receives one excitatory and one inhibitory input, and each hidden neuron uniformly excites the output neuron. Figure from [11].

For the input [0,0], intuitively, the output neuron naturally produces lowest firing rate even before learning. The input [1,1], after the learning, since each hidden neuron receives one excitatory and one inhibitory input, they cancel out, leads to a low firing rate of output neuron.

Additionally, to test whether their update rule also works when the network deviates from perfect Poisson behavior, the authors implemented a spiking neural network consisting of leaky integrate-and-fire (LIF) neurons. They demonstrated that LIF neurons were able to learn the solution to the XOR problem. Interestingly, it appears from their figures (comparing Fig. 7, c and Fig. 2, a from the paper [11]) that learning with the LIF network was even faster than with Poisson neurons, but this might be due to factors such as learning rate parameters of the network, which are not mentioned in the paper.

Lastly, the authors provided examples from two different experimental studies [15, 16]. The first paper [15] suggests an increase of synaptic strength when the pre-synaptic spike precedes the post-synaptic spike, while the second paper [16] suggests a decrease in the same case (the latter, known as anti-Hebbian behavior, has much less evidence than the findings of the former work). The authors mentioned that if the reward is fixed at a positive value, their rule would replicate the findings of the first paper. Conversely, if the reward is fixed at a negative value, their rule would replicate the findings of the second paper. Additionally, from a biophysical standpoint, for this plasticity rule to be implemented by neurons, each neuron should be able to estimate its average firing rate, $f_i(t)$. This is more feasible for a neuron if its input is slowly varying and less likely if the input changes instantly.

Notice the similarities between X. Xie and H. S. Seung's update rule (Eq. 40) and Bartlett & Baxter's update rule (Eq. 34). The concept of updating synaptic strengths based on the correlation of a global reward and a local Hebbian plasticity rule remains the same for both rules, as both are derived using policy-gradient methods. X. Xie and H. S. Seung's rule includes more biological dynamics, but it is based on the assumption that neurons are producing Poisson spike trains, which is not always true for biological neurons.

Also notice that Bartlett & Baxter's and Xie & Seung's rules have one more common behavior: They both increase the synaptic strength if pre-synaptic spike precedes post-synaptic spike in the presence of positive reward. On the other hand, Xie & Seung's update rule also depends on the relative time interval between the spikes due to the $h_{ij}(t)$ function, therefore predicting the so-called causal part of the STDP (LTP part, assuming positive reward): If the pre-synaptic spike is immediately followed by a post-synaptic spike, the synaptic change is higher than compared to if the post-synaptic spike occurs slightly later. In contrast, Bartlett & Baxter's rule does not account for the relative time interval between spikes, as it only implements the pre-synaptic neuron effect as a spike (u^j_{t-1}, Eq. 34), rather than a synaptic activation that spans time ($h_{ij}(t)$, Eq. 40) as in Xie & Seung's rule. Neither of the papers found plasticity rules that account for the anti-causal part of the STDP rule.
In 2005 and 2006, R. V. Florian published two papers [17, 18] demonstrating that a similar update rule can be mathematically derived for spiking neural networks without assuming Poisson statistics. R. V. Florian also investigated the conditions under which the so-called anti-causal part of the STDP might occur. I will primarily discuss the 2006 paper since it is more extensive and also delves into the relationship between the learning rule and STDP more broadly.

In the 2006 paper [18], the author first employed the spike-response model (SRM) [19] with escape noise [20] to model the spiking neurons. Given that the derivations closely resemble those in X. Xie and H. S. Seung's paper [11], with the main change being the neuron model (SRM instead of Poisson neurons), and not the reinforcement learning algorithm (policy-gradient method), I will mention the eligibility trace of the learning rule without providing its derivation. The eligibility trace is determined to be gated by the reward as before. The learning rule below is not an offline rule as given in Equation 40 but an online rule.

$$\frac{de_{ij}}{dt} = -\frac{e_{ij}}{\tau_e} + \frac{g'(u_i(t))}{g(u_i(t))}[Y_i(t) - \rho_i(t)] \sum_{f_i^a \in x_{j,t}} \epsilon(t - f_i^a) \tag{41}$$

where $g(u_i)$ is a function that increases with the membrane voltage u_i, similar with $f(I_i(t)$ in Eq. 40. $Y_i(t)$ is the spike train of post-synaptic neuron, exactly the same as $s_i(t)$ in Eq. 40. $\sum_{f_i^a \in x_{j,t}} \epsilon(t - f_i^a)$ is the kernel for pre-synaptic effect on post-synaptic neuron, similar with $h_{ij}(t)$, where $x_{j,t}$ denotes

the pre-synaptic spike times before the present time t. Finally, τ_e is the time-scale for the eligibility trace.

Notice that this derived rule is also not accounting for the anti-causal part of the STDP, like Bartlett & Baxter's [9] and Xie & Seung's rules [11]. Consider this case to see why so far derived rules cannot include the anti-causal part of the STDP by their construction: Assume one post-synaptic neuron that fired at time t_{post}. Also consider one pre-synaptic neuron that connects to the post-synaptic neuron and fires at times t_{pre1} and t_{pre2}, where $t_{pre1} < t_{post} < t_{pre2}$. The modeled neurons, a Bernoulli unit for Bartlett & Baxter's paper, a Poisson neuron for Xie & Seung's paper, and an SRM model for R. V. Florian's model [18], only model the voltage membrane of a neuron. Considering a post-synaptic neuron, only the pre-synaptic inputs that arrived before the present time can affect the voltage membrane. Since we derive the rule by taking the derivative of the expected reward with respect to the synaptic weights between pre- and post-synaptic neurons, and since only the effects of pre-synaptic neuron spikes before the present time are multiplied by the weight, the only contribution of pre-synaptic inputs to the learning rule is the inputs that arrive before the present time. By construction of the models, pre-synaptic neurons that occur after the present time cannot have an effect on the post-synaptic neuron membrane potential. In other words, the spike at t_{pre2} can not affect the post-synaptic neuron at time t_{post}, but only spikes before the t_{post}, so, in our example, t_{pre}, may effect the post-synaptic neuron. However, STDP curves that are fitted to data, and different types of implementations such as all-to-all, nearest-neighbor, post-synaptic centric, pre-synaptic centric STDP rules [21], they allow so-called anti-causal effects of pre-synaptic neuron.

After deriving a plasticity rule that does not rely on Poisson statistics, R. V. Florian also demonstrated that if the spiking model is extended to account for homeostatic plasticity, the derived rule results in the anti-causal part of the STDP as well. Homeostatic plasticity can be defined as a regulation mechanism that acts to stabilize the neuronal activity. There are different mechanisms for homeostatic plasticity [22, 23]. In the paper, he used the version in which pre-synaptic neurons adapt their spiking thresholds to maintain the mean rate of the post-synaptic neuron constant. This implies that a pre-synaptic neuron increases (decreases) its firing threshold if the firing rate of the post-synaptic neuron increases (decreases), thereby trying to keep the post-synaptic neuron activity constant.

In that case, the policy does not only depend on the firing probability of the post-synaptic neuron (notice that in Eq. 39, we only take the derivative of $p_i(t)$, which is the post-synaptic firing probability, with respect to synaptic strength) but also on the firing probability of the pre-synaptic neuron, since the activity of pre-synaptic neurons also depend on synaptic strength in the extended spiking model. For an analytical derivation of the rule, I refer the reader to the original paper [18]. Simply, the derived plasticity rule can still be interpreted as correlations between the global reward and (a combination of different) local plasticity rules, as shown in the general formula in Equation 28: $\triangle w_{ij} = f_1(pre_j) f_2(post_i) f_3(reward)$.

4.2 Reward-modulated STDP

As mentioned, incorporating the homeostatic plasticity to the spiking neuron model enables both the causal and anti-causal parts of the STDP. However, unlike the commonly used STDP rules [6], the derived rule also depends on the firing rate intensity parameter $g(u_i(t))$ of the SRM model. The author claims that the dependence on this intensity parameter is not experimentally justified. For this reason, among others, the author also proposed a learning rule in which the global reward directly modulates the eligibility trace determined by the STDP rule. Note that, different from all the other previously mentioned learning rules, this rule is not derived with a top-down approach and not analytically derived.

The author claimed that reward-modulated STDP rule can be seen as a simplified version of the analytically derived rule.

In reward-modulated STDP, the eligibility trace increases if pre-synaptic spike precedes post-synaptic spike, and decreases if post-synaptic neuron precedes the pre-synaptic spike. The increase and decrease amount is determined by the time interval of the two spikes, obeying the STDP rule. The eligibility trace is gated by the reward to induce changes in synaptic weights.

$$w_{j,t+1} = w_{j,t} + \alpha R_{t+1} z_{j,t+1}, \tag{42}$$

$$z_{j,t+1} = \beta z_{j,t} + s_i(t) A_+ \sum_{j,a} e^{(-\frac{t-f_j^a}{t_+})} + s_j(t) A_- \sum_{i,a} e^{(-\frac{t-f_i^a}{t_-})} \tag{43}$$

The second and third term in the right-hand side of the eligibility trace equation is one of the ways of implementing the STDP rule in an efficient way. A_+ and A_- determines the maximum synaptic update when the time interval is equal to zero, while t_+ and t_- determines the time-scales for which synaptic-change occur. Effects of pre-then-post pairs are accumulated in the first sum, and post-then-pre pairs are in the second sum. As before, $s_i(t)$ and $s_j(t)$ are denoting the spike trains of post- and pre-synaptic neurons, respectively.

Notice that in reward-modulated STDP, when the reward is positive the STDP curve is what is called Hebbian STDP (Figure 2), while when the reward is negative it is called Anti-Hebbian STDP. It is shown that [24] the Hebbian STDP rule minimizes the variability of post-synaptic firing, while anti-Hebbian STDP maximizes the variability [25]. The author claimed that reward-modulated STDP rule decreases the variability of the post-synaptic neuron when the reward is positive, allowing the network to exploit a particular configuration that facilitates learning, while enabling exploration of other configurations by increasing the variability of the post-synaptic neuron when the reward is negative.

R. V. Florian simulated the reward-modulated STDP rule for the XOR problem with rate-coded input (similar to the previously mentioned XOR problem setting) and demonstrated that the network was able to solve the XOR problem. The author also applied the algorithm to the XOR problem with temporally coded input, coding 0 and 1 as two distinct spike patterns while maintaining equal firing rates, and showed that algorithm successful solved the temporal-XOR problem. Moreover, the author illustrated that the network can also learn a target firing-rate pattern coded by individual firing rates of each output neuron. In this case, the reward was provided to the network for each time step by comparing the distance between the output neurons' pattern and the target pattern.

To investigate the contributions of causal and anti-causal parts of STDP for learning firing-rate patterns, the author systematically varied the parameters A_+ and A_-. Interestingly, when eligibility traces were used, the network was only able to learn the desired firing pattern when A_+ and A_- was equal, causal and anti-causal parts contributing equally. However, the author also tried the algorithm without using eligibility traces (i.e. $\beta = 0$). In that case, the network was able to learn the patterns irrespective of A_- (whether it's equal to 0, A_+, $-A_+$), given that the A_+ is positive, suggesting the importance of the causal part of STDP (i.e., pre-synaptic spike preceding post-synaptic spike increases the weights if the reward is positive and decreases the weights if the reward is negative).

The effect of delayed rewards has also been investigated by the author. If the reward was provided to the network with delays, the rule with eligibility traces was still able to learn the task, but with

decreasing learning efficacy for increasing reward delays between 0ms and approximately 10ms. It is crucial, however, that the decay rate of the eligibility trace should be large compared to the mixing time of the system. Mixing time of the system can be loosely interpreted as the time interval between the initiation and termination of the effect of action initiation [9, 18]. Nevertheless, increasing the decay rate too much would make the gradient estimate more noisy, which deteriorates learning efficacy [11].

Lastly, the author also discussed the effect of network size on learning efficacy. It is demonstrated that learning efficacy decreases as the network size increases, which can be considered an intrinsic property of the learning rule since the rule exploits the fluctuations of spikes, and these fluctuations become less pronounced with larger network sizes. It is also shown that the learning efficacy is even more affected by the network size with the rule that uses eligibility traces (Figure 8).

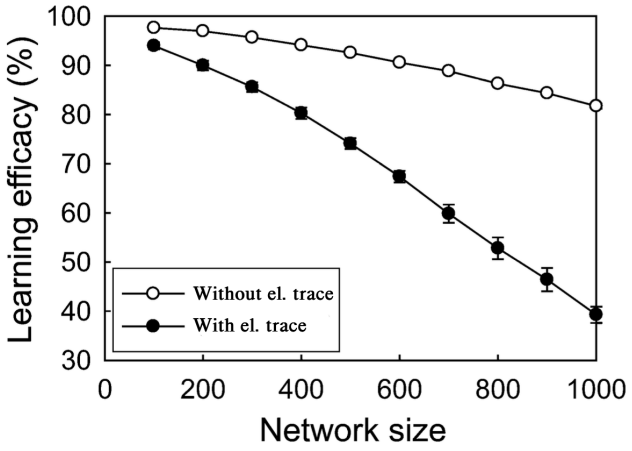


Figure 8: Learning efficacy with respect to network size for the algorithms with or without eligibility trace. Figure from [18], edited.

To conclude, R. V. Florian extended the previously mentioned rules by using the SRM model. Importantly, the author drew attention to the similarities between the classical STDP rule and the local Hebbian part of the analytically derived rule. It is claimed that using the STDP rule, which has direct experimental evidence, instead of the derived local Hebbian rule, can also be effective for learning. Finally, they reiterated the speculations of Xie & Seung's, hypothesizing that in classical STDP experimental setups, the reward signal might have been fixed during the experiments. They suggested more experimental work to investigate the biological plausibility of the reward-modulated STDP.

Similar to the findings of R. V. Florian, homeostatic mechanism giving rise to anti-causal part of STDP, J. P. Pfister et al. [26] found that the anti-causal part of STDP emerges if the post-synaptic neuron activity is optimized for inducing a spike at a desired time given the constraint of keeping its firing rate constant. The authors defined the desired firing time by inducing a reward signal for spikes that fired at that time.

In 2007, E. M. Izhikevich also introduced a model that uses reward-modulated STDP [27]. In previous works, the weights of the pre-synaptic neurons were not constrained; that is, they were able to change their sign during the learning procedure, and one pre-synaptic neuron was able to excite one post-synaptic neuron while inhibiting another, thereby violating Dale's law. By explicitly using excitatory

and inhibitory neurons, E. M. Izhikevich showed that the reward-modulated STDP also works for the networks that obey Dale's law. The author used a recurrent spiking neural network consisting of excitatory and inhibitory neurons, and only used the STDP rule for excitatory neurons as suggested by experiments [28]. The idea was the same with previously mentioned papers, modulating the STDP with a global broadcasting reward (Figure 9).

The author selected 100 different groups of neurons, each consisting of 50 neurons. All the groups were simulated consecutively, but the delayed reward was administered only after stimulating the a priori chosen group. Stimulating the groups increases the activity of neurons (pre-synaptic), which, in turn, increases the activity of projected neurons (post-synaptic). Since the dopamine level is only increased (with a random delay between 1 and 3 seconds) after stimulating the chosen group, the synaptic strength between the neurons of the chosen group and their projections increases due to the pre-then-post order. When the reward is not administered after the stimulation of other groups, since dopamine has a tonic level, these connections also slightly increase. However, it's important to note that the area of LTD is greater than LTP (Figure 9, b) in the used STDP rule, which decreases the synaptic strength of uncorrelated pairs of two neurons [29, 6], compensating for the slight potentiation of other neurons.

Differently from previously mentioned papers, the author explicitly related the reward with extracellular dopamine, and he also claimed that the autophosphorylation of CaMK-II or oxidation of PKC or PKA may implement the eligibility trace [30].

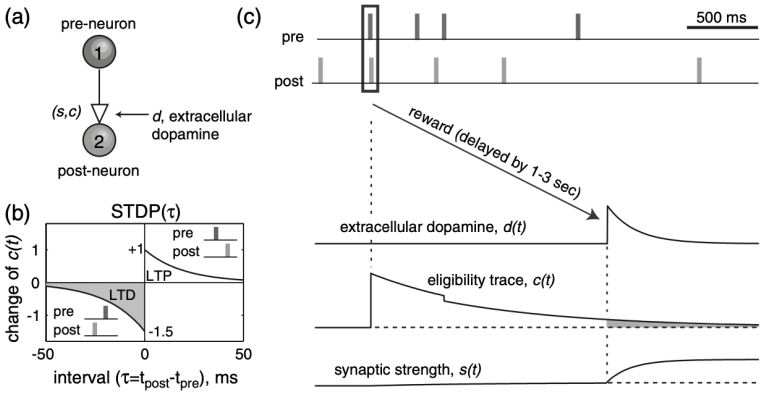


Figure 9: a) The synaptic strength shown by s and the eligbility trace shown by c determines the dynamics of a synapse. b) Used STDP rule. c) Coincident firing of pre-then-post neurons sets the eligibility trace together with the delayed reward determines the synaptic strength change.

In 2007, M. A. Farries and A. L. Fairhall [31] also investigated the capabilities of reward-modulated additive and multiplicative STDP models. The authors demonstrated that using a reward-modulated STDP rule together with homeostatic plasticity rule has a better learning efficacy compared to when homeostatic plasticity rule is not used. The authors tried different homeostatic plasticity rules, one similar to what R. V. Florian used and another that was different. They concluded that both of the distinct homeostatic plasticity has similar effects on learning efficacy. They also claimed, with simulations, that the network is able to learn the previously mentioned tasks even when only the causal part of the STDP is used. In fact, they also showed that the anti-causal part is deteriorating the learning efficacy. They

claimed that there are no reasons for both the causal and anti-causal parts of the STDP to have the same functional roles. They briefly touched on the experimental findings that the biological mechanism of different parts of STDP is different [32], suggesting the possibility that they might be modulated separately.

The authors also discussed the potential circuitry and mechanisms that would enable the global reward broadcasting. The midbrain neurons that releases dopamine could be a good candidate (note that they do not encode the reward itself but the TD signal), but these neurons mostly innervates striatum. Although it is still yet debated how, it is known that dopamine indeed modulates the plasticity in striatum [31]. But it is also known that the isocortex receives much less dopaminergic input. The authors claimed that the another source of dopamine response for reward-prediction could be the basal ganglia, which widely outputs to isocortex, which would enable global reward broadcasting to the cortex.

In 2008, R. Legenstein et al. analytically investigated the reward-modulated additive STDP rule, assuming Poisson neurons [33], and also demonstrated the capability of the reward-modulated additive and multiplicative STDP through simulations with LIF neurons. One interesting modeling attempt in this paper is the 'biofeedback' experiment. In this experiment, the authors recorded the firing rate of a monkey's single neuron, and showed it to the monkey, and the monkey is rewarded when it was able to increase the firing rate of that particular neuron [34]. The monkey was able to learn the task in a few minutes. The authors also showed that the monkey is also able to learn to separately increase and decrease the activity of two different neurons that are spatially separated by a few hundred microns [34]. The authors showed that the network endowed with reward-modulated STDP, when the reward is a function of the firing rate of a chosen neuron (Figure 10), was indeed able to learn (Figure 11) the biofeedback task.

The reward signal $d(t)$ is provided to the whole network (Figure 10, a) using the reward kernel (Figure 10, c) according to the filtered spike train of the chosen neuron (Figure 10, b).

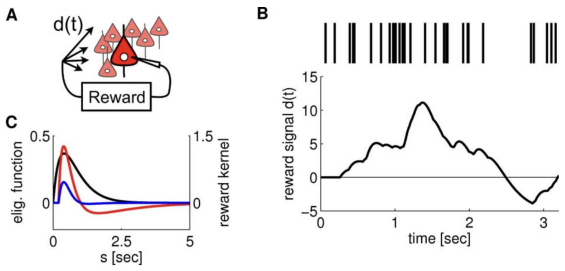


Figure 10: a) The reward signal given to whole neuron population is dependent on the chosen neuron. b) Spike train of the chosen neuron determines the reward signal. c) A demonstration of the eligibility trace (black line) multiplied by the reward kernel (red line) which gives the synaptic strength change (blue line). Figure from [33].

Figure 11 shows the raster plot of a part of the network before (left panel) and after (right panel) learning. The neuron whose activity was desired to be reduced is shown by green, and another neuron activity to be increased is shown by blue.

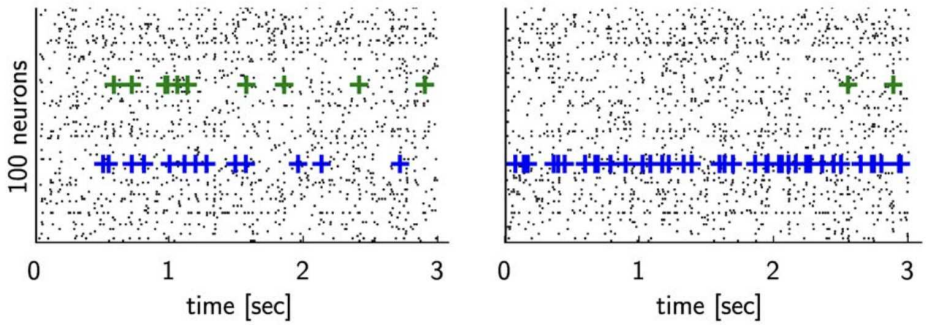
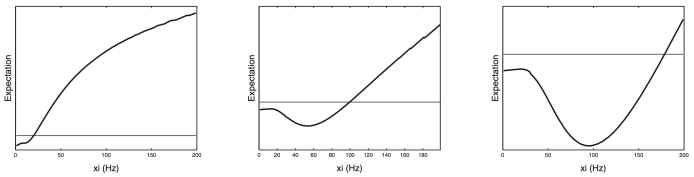


Figure 11: Left: Raster plot of a subsampled network for 3 seconds before learning. Right: Raster plot of a subsampled network for 3 seconds after learning. The activity of neuron shown in blue is increased while green is decreased as the learning rule dictates. Figure from [33].

The authors claimed that since reward-modulated STDP rule requires spontaneous activity of the network together with trial-to-trial variability, it may provide a functional explanation for these two commonly observed features of cortical neurons.

Up to this point, I have attempted to summarize papers that: 1) utilize policy-gradient methods to analytically derive a learning rule for various neuron models (including the Bernoulli unit, Poisson neuron, and SRM model), 2) explore the relationship between analytically derived rules and reward-modulated STDP rules, and 3) investigate the efficacy of reward-modulated STDP.

One more interesting connection between a plasticity rule and RL framework was studied by D. Baras and R. Meir in 2007 [35]. Similarly to the other papers mentioned, they derived a plasticity rule by applying a policy-gradient method to a spiking neuron model. They considered the average behavior of their plasticity rule by taking the expectation over a temporal window to convert the individual spikes to rates. To do that, they assumed two statistical properties for a pre- and post-synaptic neuron. 1-) The pre-synaptic neuron fires according to a homogeneous Poisson process. 2-) The post-synaptic neuron fires according to an inhomogeneous Poisson process that is modulated by the pre-synaptic effect. Under these conditions, they showed that based on a threshold, LTD and LTP may occur. They found that the threshold increases linearly with the pre-synaptic neuron activity, and the expected weight change is similar to the BCM rule qualitatively (Figure 12). It is noteworthy to mention that the BCM and STDP relationship has been also studied [30].



(a) $x_j = 20$ (b) $x_j = 100$ Hz (c) $x_j = 180$ Hz

Figure 12: Weight change (shown by the y-axis) with respect to post-synaptic firing rate. Three different threshold cases are shown for different pre-synaptic rates. Figure from [35].

In 2009, E. Vasilaki showed the situations in which policy-gradient methods fail [8]. They considered the Morris Water maze task, in which the agent is placed in opaque water and tries to find the hidden

platform just below the water, which would be rewarding since it ends an inconvenient experience. They assumed that the pre-synaptic neurons are place cells, which encode the spatial location of the agent. These place cells project to hypothetical action cells and the action cells are also connected to other action cells via lateral projections (Figure 13). Due to the lateral connections, an activity bump occurs in action cells, which would determine the next place the agent goes. The agent is re-initiated from a random place in the maze if it reaches the hidden platform (which is fixed in different trials), or if it hits the wall or exceeds the predetermined trial duration. The authors showed that policy-gradient methods are unable to learn the location of the hidden platform and claimed that one of the reasons is that the action taken in their model is chosen by a rate code of the action cells but not the temporal patterns, which is a suitable learning paradigm for policy-gradient methods.

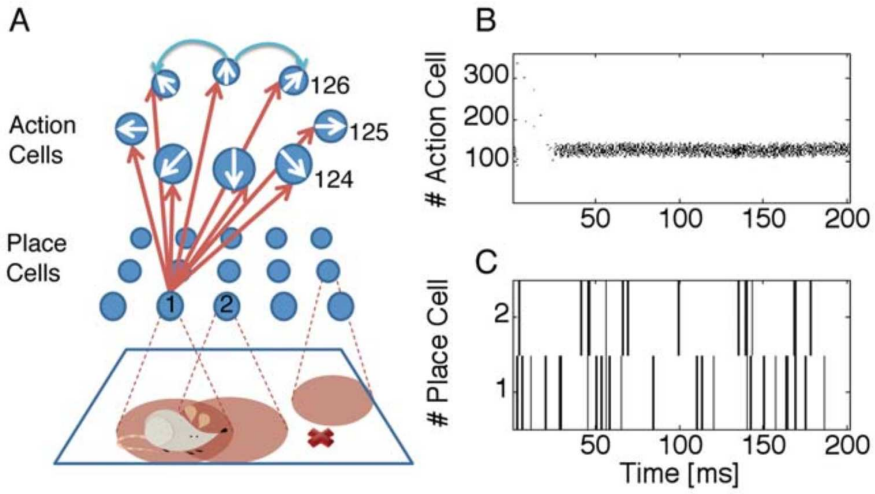


Figure 13: a) Overview of the network. Location of the agent algorithmically determines the firing rates of place cells, which have an all-to-all connections to action cells. Action cells determines the direction of the next movement. b) Activity of action cell population, notice the activity bump after 25ms. c) Firing pattern of place cells denoted by 1 and 2.

The authors showed that a simpler Hebbian rule (hard bounded from above to avoid unstability) that gated by the reward is able to learn the task, while policy-gradient methods fail as expected. They introduced a new variable τ_c to the Equation 39, extending the rule shown in Equation 41.

$$\frac{de_{ij}}{dt} = -\frac{e_{ij}}{\tau_e} + \frac{g'(u_i(t))}{g(u_i(t))}\left[Y_i(t) - \frac{\rho_i(t)}{1+\tau_c\rho_i(t)}\right]\sum_{f_i^a \in x_{j,t}} \epsilon(t - f_i^a) \tag{44}$$

Notice that when $\tau_c = 0$, the learning rule becomes the policy-gradient rule, and when τ_c goes to infinity, the rule becomes a reward-modulated simpler Hebbian plasticity. The authors showed that setting $\tau_c = 5ms$ makes the rule able to learn the task since it introduces a bias toward the Hebbian rule. They showed that the agent was able to learn the task in approximately 20 trials, which they claimed is also the case in experiments for the water maze task [36]. In Figure 14, they showed the decrease in escape latency, defined as the time until the agent reaches the hidden platform in each trial, for different configurations of the network.

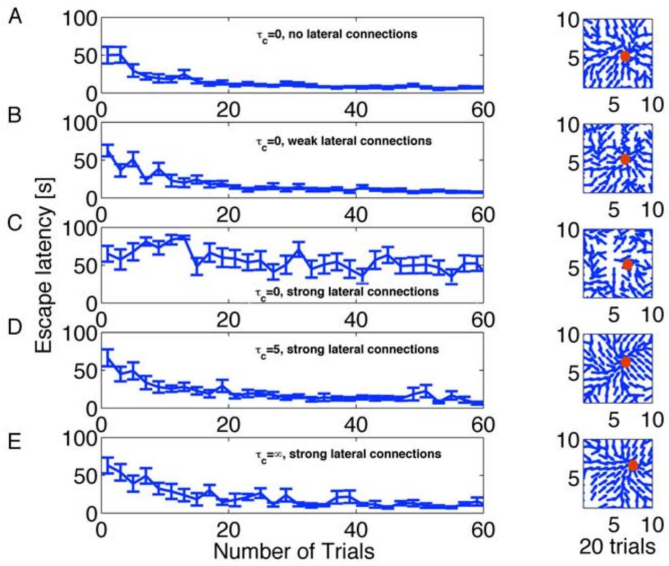


Figure 14: Left panels are showing the escape latency with respect to number of trials. Right panels are showing the formed navigation map after 20 trials.

The authors also pointed out two properties of their model. First, increasing the number of action cells does not significantly reduces the learning efficacy, which is intuitive since the rate coding is used as a readout. Second, increasing the area of the maze would decrease the efficacy since the correlation between eligibility trace and reward would diminish if the agent takes too much time before receiving the reward.

In conclusion, gating the STDP rule by a reward signal in theory can be implemented by spiking neural networks for learning purposes. It is not an analytically derived rule as the rules mentioned in Section 4.1, therefore it is not guaranteed that the update rule is in the direction of reward maximization, but it appears to be more biologically plausible. The rule enables networks to learn specific temporal patterns, and solve simple tasks such as XOR problem. These papers suggests for further experimental plasticity studies including neuromodulators, which would be one of the potential candidates for reward signaling. For a review study that shows some evidence for eligibility traces and neuromodulator-gated plasticity, see [37].

4.3 Applying temporal-difference learning in spiking neural networks

As written in Section 2.3, temporal-difference learning uses an error, $\delta(s)$, to update the weights (Eq. 16). In 2001, R. P. N. Rao and T. J. Sejnowski used a two-compartment model, including the dendrite compartment and soma-axon compartment [38]. The pre-synaptic spike input is given to the dendrite compartment of the neuron with different time points, before or after the post-synaptic spike. In the model, the post-synaptic neuron is denoted as $v_t = \sum_i w(i)x_t(i)$, where $x_t(i)$ is a pre-synaptic input and $w(i)$ is a synaptic weight. Notice that it corresponds to defining the states with linear features, as denoted in Section 2.3. The synaptic weight is changed according to the temporal-difference learning rule. By setting the $\gamma = 1$ and $R_{t+1} = 0$ in Equation 15, the synaptic update becomes:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha(v_{t+1} - v_t)\mathbf{x}_t. \tag{45}$$

By fixing the pre-synaptic spike time and changing the post-synaptic spike time (Figure 15, a and b), thus, changing the time interval between the spike pairs, they measured the weight change curve governed by the TD learning, and showed that it is qualitatively similar with the STDP rule (Figure 15, c).

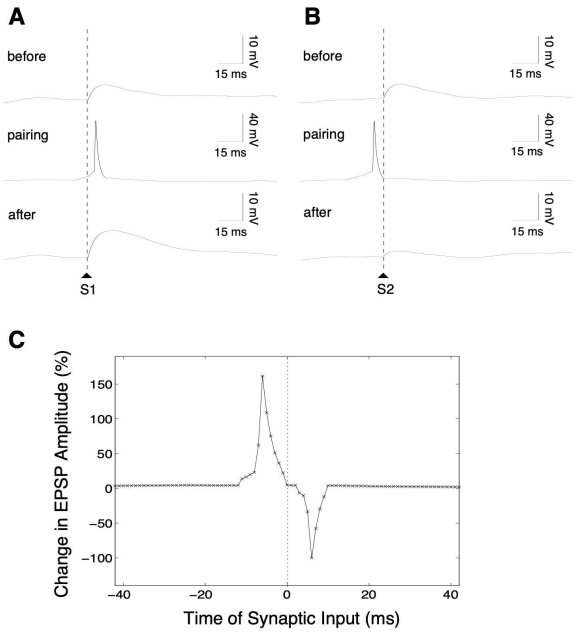


Figure 15: a) Pre-synaptic spike is shown by S1. The post-synaptic spike elicited 5 ms after the pre-synaptic spike. The weight change is shown before (top) and after (bottom) the pairing procedure. b) Same as a, but the post-synaptic spike elicited 5ms before the pre-synaptic spike. Pre-synaptic spike is shown by S2. c) By changing the time interval (in a and b it was 5ms) they calculated the weight change curve. Observed curve is qualitatively similar with the STDP rule. Figure from [38].

To conclude, the authors demonstrated that the STDP rule can be understood as a type of TD learning for prediction. They also implemented a network architecture to show how a network using TD learning rule can learn to predict input patterns.

The TD learning can also be used for the control problem, i.e., finding the optimal policy, rather than finding the value, which is be the prediction problem. Using actor-critic temporal-difference learning, in 2008, D. D. Castro et al. published a paper [39] in which they derived a plasticity rule using SRM model, similarly with R. V. Florian's paper [18]. Different from other works that mentioned in Section 4.1, they used the TD error to gate the local Hebbian plasticity rule for actor neurons, instead of the reward itself. Critic neurons also received TD error which they use it for a simple linear function approximation method for updating the states (membrane voltage) of the neurons (identical to Equation 17). They demonstrated the learning capability of their plasticiy rule using a navigation task, I refer the reader to the original paper for further details. This top-down approach is quite similar with using policy-gradient methods to derive a plasticity rule, instead, they used actor-critic temporal-difference framework.

W. Potjans and A. Morrison also suggested plasticity rules that implements the actor-critic temporal-difference (AC-TD) framework in spiking neural networks [40]. Differently from the paper by D. D.

Castro et al., they did not derive the model using a top-down approach, but they suggested engineered plasticity mechanisms that would correspond to AC-TD framework. The authors named their plasticity rules as state-critic plasticity rule and state-actor plasticity rule and showed that their plasticity rule reaches similar learning efficacy with the non-spiking models of the AC-TD algorithm.

5 Discussion

In this report, I have investigated some plasticity rules that are inspired by the RL framework. The application of policy-gradient methods has resulted in the formulation of reward-modulated Hebbian plasticity rules. Since the derived Hebbian rule shared similar properties with the experimentally observed STDP rules, it paved the way of investigation of reward-modulated STDP rules. Moreover, it has been also shown that using TD error to modulate Hebbian rules can also be implemented by spiking neural networks. I believe that studying such top-down approaches to derive plasticity rules may facilitate the experimental studies by suggesting different experimental setups, which, in turn, re-iterates the theoretical studies, closing the theory-experiment loop.

On the other hand, in this report, all the RL-inspired plasticity rules that is discussed shares similar properties, and they can be shown in a general formula:

$$\triangle w_{ij} = \underbrace{f_1(pre_j)f_2(post_i)}_{\text{Local}}\underbrace{f_3(reward)}_{\text{Global}}$$

This form is also called three-factor learning rule [37], since there exist three different factors that determine the weight change: pre-synaptic neuron activity, post-synaptic neuron activity, and a global signal. However, this general formula is not the only possible way of incorporating the reward signal into the established Hebbian rules. There is a piece of experimental evidence which show that dopamine increases the excitability of D1 receptor-expressing striatal projection neurons [41]. This case can be shown in another general formula:

$$\triangle w_{ij} = f_1(pre_j)f_2(post_i, reward) \tag{46}$$

where a reward signal directly effects the activity of the post-synaptic neuron, rather than gating the Hebbian rule governed by pre- and post-synaptic activities. There may also exist different ways for reward signal to act on plasticity, and I believe it is more likely that the brain recruits various mechanisms for distinct learning paradigms, rather than only exploiting a universal learning mechanism. For example, it is known that pure Hebbian learning rules (not effected by any global reward signal) are able to successfully implement various unsupervised learning settings, such as novelty responses [42], principal component learning [43], orientation specificity [5]. I believe that studying the interactions of pure Hebbian rules and reward modulated Hebbian rules, coupled with an investigation into how different brain areas are recruiting distinct learning paradigms, may further facilitate our understanding of animal learning.

References

[1] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 1998.

[2] Peter Dayan and L. F. Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational neuroscience. Massachusetts Institute of Technology Press, Cambridge, Mass, 2001.

[3] Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, June 1949.

[4] Colin Bredenberg, Ezekiel Williams, Cristina Savin, Blake Aaron Richards, and Guillaume Lajoie. Formalizing locality for normative synaptic plasticity models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[5] L Bienenstock, N Cooper, and W Munro. THEORY FOR THE DEVELOPMENT OF NEURON SELECTIVITY: ORIENTATION SPECIFICITY AND BINOCULAR INTERACTION IN VISUAL CORTEX.

[6] Sen Song, Kenneth D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, September 2000.

[7] Robert C. Froemke and Yang Dan. Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature*, 416(6879):433–438, March 2002.

[8] Eleni Vasilaki, Nicolas Frémaux, Robert Urbanczik, Walter Senn, and Wulfram Gerstner. Spike-Based Reinforcement Learning in Continuous State and Action Space: When Policy Gradient Methods Fail. *PLoS Computational Biology*, 5(12):e1000586, December 2009.

[9] Peter L. Bartlett and Jonathan Baxter. Hebbian Synaptic Modifications in Spiking Neurons that Learn, November 2019. arXiv:1911.07247 [cs, stat].

[10] Colin Bredenberg, Ezekiel Williams, Cristina Savin, Blake Richards, and Guillaume Lajoie. Formalizing locality for normative synaptic plasticity models.

[11] Xiaohui Xie and H. Sebastian Seung. Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, 69(4):041909, April 2004.

[12] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning.

[13] H.Sebastian Seung. Learning in Spiking Neural Networks by Reinforcement of Stochastic Synaptic Transmission. *Neuron*, 40(6):1063–1073, December 2003.

[14] Ila R. Fiete and H. Sebastian Seung. Gradient Learning in Spiking Neural Networks by Dynamic Perturbation of Conductances. *Physical Review Letters*, 97(4):048104, July 2006.

[15] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–215, 1997.

[16] Curtis C. Bell, Victor Z. Han, Yoshiko Sugawara, and Kirsty Grant. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 387(6630):278–281, May 1997.

[17] R.V. Florian. A reinforcement learning algorithm for spiking neural networks. In *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05)*, pages 8 pp.–, 2005.

[18] Răzvan V. Florian. Reinforcement Learning Through Modulation of Spike-Timing-Dependent Synaptic Plasticity. *Neural Computation*, 19(6):1468–1502, June 2007.

[19] W. Gerstner. Chapter 12 a framework for spiking neuron models: The spike response model. In F. Moss and S. Gielen, editors, *Neuro-Informatics and Neural Modelling*, volume 4 of *Handbook of Biological Physics*, pages 469–516. North-Holland, 2001.

[20] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[21] Guo-Qiang Bi. Spatiotemporal specificity of synaptic plasticity: cellular rules and mechanisms. *Biological Cybernetics*, 87(5):319–332, December 2002.

[22] Karunesh Ganguly, Laszlo Kiss, and Mu-ming Poo. Enhancement of presynaptic neuronal excitability by correlated presynaptic and postsynaptic spiking. *Nature Neuroscience*, 3(10):1018–1026, October 2000.

[23] Teresa A. Nick and Angeles B. Ribera. Synaptic activity modulates presynaptic excitability. *Nature Neuroscience*, 3(2):142–149, February 2000.

[24] Sander Bohte and Michael C Mozer. Reducing spike train variability: A computational theory of spike-timing dependent plasticity. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

[25] Emmanuel Daucé, Hédi A. Soula, and Guillaume Beslon. Learning methods for dynamic neural networks. 2005.

[26] Jean-Pascal Pfister, Taro Toyoizumi, David Barber, and Wulfram Gerstner. Optimal Spike-Timing-Dependent Plasticity for Precise Action Potential Firing in Supervised Learning. *Neural Computation*, 18(6):1318–1348, June 2006.

[27] E. M. Izhikevich. Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling. *Cerebral Cortex*, 17(10):2443–2452, October 2007.

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

[29] Richard Kempter, Wulfram Gerstner, and J. Leo van Hemmen. Hebbian learning and spiking neurons. *Phys. Rev. E*, 59:4498–4514, Apr 1999.

[30] Eugene M. Izhikevich and Niraj S. Desai. Relating STDP to BCM. *Neural Computation*, 15(7):1511–1523, July 2003.

[31] Michael A. Farries and Adrienne L. Fairhall. Reinforcement Learning With Modulated Spike Timing–Dependent Synaptic Plasticity. *Journal of Neurophysiology*, 98(6):3648–3665, December 2007.

[32] Gayle M. Wittenberg and Samuel S.-H. Wang. Malleability of Spike-Timing-Dependent Plasticity at the CA3–CA1 Synapse. *Journal of Neuroscience*, 26(24):6610–6617, 2006. Publisher: Society for Neuroscience _eprint: https://www.jneurosci.org/content/26/24/6610.full.pdf.

[33] Robert Legenstein, Dejan Pecevski, and Wolfgang Maass. A Learning Theory for Reward-Modulated Spike-Timing-Dependent Plasticity with Application to Biofeedback. *PLoS Computational Biology*, 4(10):e1000180, October 2008.

[34] E E Fetz and M A Baker. Operantly conditioned patterns on precentral unit activity and correlated responses in adjacent cells and contralateral muscles. *Journal of Neurophysiology*, 36(2):179–204, 1973. PMID: 4196269.

[35] Dorit Baras and Ron Meir. Reinforcement Learning, Spike-Time-Dependent Plasticity, and the BCM Rule. *Neural Computation*, 19(8):2245–2279, August 2007.

[36] D.J. Foster, R.G.M. Morris, and Peter Dayan. A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16, 2000.

[37] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules. *Frontiers in Neural Circuits*, 12:53, July 2018.

[38] Rajesh P. N. Rao and Terrence J. Sejnowski. Spike-Timing-Dependent Hebbian Plasticity as Temporal Difference Learning. *Neural Computation*, 13(10):2221–2237, October 2001.

[39] Dotan D Castro, Dmitry Volkinshtein, and Ron Meir. Temporal Difference Based Actor Critic Learning - Convergence and Neural Implementation.

[40] Wiebke Potjans, Abigail Morrison, and Markus Diesmann. A Spiking Neural Network Model of an Actor-Critic Learning Agent. *Neural Computation*, 21(2):301–339, February 2009.

[41] Asha K. Lahiri and Mark D. Bevan. Dopaminergic transmission rapidly and persistently enhances excitability of d1 receptor-expressing striatal projection neurons. *Neuron*, 106(2):277–290.e6, 2020.

[42] Auguste Schulz, Christoph Miehl, II Berry, Michael J, and Julijana Gjorgjieva. The generation of cortical novelty responses through inhibitory plasticity. *eLife*, 10:e65309, oct 2021.

[43] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, November 1982.