Modification of a Reward-Modulated Hebbian Learning Rule as a Model of Working Memory Emergence

Tammy Tran Brad Theilman Nuttida Rungratsameetaweemana

Neurosciences Graduate Program, University of California, San Diego

ttt075@ucsd.edu, btheilma@ucsd.edu, nrungrat@ucsd.edu

Abstract

In liquid state machines with generic cortical microcircuits, synaptic plasticity can be optimized by reward-modulated Hebbian learning, eliminating the need for supervised learning (Hoerzer, Legenstein, and Maass, 2014). Reward-modulated Hebbian learning can thus lead to autonomous emergence of task-specific working memory during the learning of computational rules. However, in Hoerzer, Legenstein, and Maass (2014), reward-modulated Hebbian learning was modeled with an all-or-none modulatory signal that permitted synaptic weight change only above a criterion level of learning. We use liquid computing models to investigate working memory emergence via Hebbian learning with non-binary modulatory signals. We implement a nonbinary, but discrete modulatory signal and an analog signal. In doing so, we model physiological conditions of tonic and phasic output of reward-mediating systems like the dopaminergic system. We find that the effects of analog modulatory signals on working memory emergence improve reward-modulated Hebbian learning in liquid state machines. We propose that reward-modulated Hebbian learning in generic microcircuits of neurons can abstractly model general cognitive processes.

1 Introduction

The liquid state machine (LSM) is a conceptual framework that assumes generic recurrent neural microcircuits where neurons are randomly connected to one another. Specifically, the vector of contributions of all the neurons in the microcircuit to the membrane potential at time *t* of a readout neuron is referred to as the liquid state *x(t)*, and this is all the information about the circuit state a readout neuron has access to [3]. The LSMs do not require task-dependent constructions and need not be engineered for a specific task, and hence this framework can be used to investigate a wide range of computational tasks. The liquid state of an LSM is assumed to vary continuously over time and to be sufficiently sensitive to information needed for specific tasks. In addition to this universal computing power, the LSM framework has a capability of turning time-varying circuit inputs into spatio-temporal activity pattern that represents the circuit dynamics. These characteristics make it possible for researchers to train LSMs to fulfill a large variety of complex computational tasks.

While past studies have shown that the LSMs can successfully be trained to learn several different tasks, the training paradigms generally provide the microcircuits with knowledge of the desired activity output (supervised learning). This type of learning also presupposes another neural network that is capable of a particular computational task used, and thus cannot explain how specialization first emerges. Additionally, the feedback provided in supervised learning represents global activity of the entire network rather than localized dynamical activity of specific neurons, resulting in physiologically inaccurate network outputs. Due to these pitfalls of supervised learning, the present study examines working memory emergence by training recurrent neural networks through unsupervised, reward-modulated Hebbian learning, where feedback provided to the neural network represents local activity between the pre- and postsynaptic neurons.

2 Methods

To test working memory emergence via reward-modulated Hebbian learning in recurrent neural networks, a generic network model was implemented as previously described [1]. Leaky integrator neurons (N = 1000) were sparsely, recurrently connected. Two external input streams $u_i(t)$ were provided to the recurrent neurons, and the recurrent neurons provided output to a single readout neuron, which in turn provided feedback to the recurrent network. All neurons were connected or received input or feedback with probability p of 0.1. Synaptic weights between recurrent neurons (W^{rec}) were randomly drawn from Gaussian distributions with zero mean and 1/(pN) variance. Input and feedback weights (W^{in} and W^{fb}) were drawn from uniform distributions in [-1, 1]. Output weights **w** to the readout neuron were initialized to zero and adjusted during training.

Membrane potentials $x_j(t)$ per recurrent neuron *j* were initialized to zero, and network dynamics per simulation time-step dt = 1ms is given by

$$\frac{\tau dx_i}{dt} = -x_i(t) + \lambda \sum_{i=1}^{N} W_{ij}^{rec} r_j(t) + \sum_{i=1}^{M} W_{ij}^{in} u_j(t) + \sum_{i=1}^{L} W_{ij}^{fb} z_j(t)$$

per recurrent neuron *i.* The chaoticity level λ was 1.2, and the network constant τ was 50ms.

The firing rate $r_i(t)$ per recurrent neuron *j* is given by $r_j(t) = \tanh[x_j(t)] + \zeta_j^{state}(t)$, with the noise $\zeta_j^{state}(t)$ drawn from uniform distributions in [-0.05, 0.05].

Output at the readout neuron is given by $z(t) = \boldsymbol{w^T r(t)} + \zeta(t)$, where **r(t)** is the column vector of firing rates $r_j(t)$ and the zero-mean exploration noise $\zeta(t)$ was drawn independently at each time step from uniform distributions in [-0.5, 0.5].

Weight change for w(t) for each dt is given by $\Delta w(t) = \eta(t)[z(t) - z_{avg}(t)]M(t)r(t)$.

$\eta(t) = \frac{\eta_{init}}{1+\frac{t}{T}}$ is a linearly decaying learning rate with η_{init}= 0.0005 and T = 20s. $z_{avg}(t)$ is the average readout output given by $z_{avg}(t) = \left(1 - \frac{dt}{\tau_{avg}}\right) z_{avg}(t - dt) + \left(\frac{dt}{\tau_{avg}}\right) z_i(t)$, with $\tau_{avg} = 5ms$. Initially, M(t) is a binary modulatory signal of 1 if performance P(t) > $P_{avg}(t)$, 0 otherwise. Performance P(t) is given by $P(t) = -\sum_{i=1}^{L}[z(t) - f(t)]^2$ where f(t) is the target output of the readout neuron. $P_{avg}(t)$ is given by $P_{avg}(t) = \left(1 - \frac{dt}{\tau_{avg}}\right) P_{avg}(t - dt) + \left(\frac{dt}{\tau_{avg}}\right) P(t)$.

For the working memory computational task described previously, input streams $\widehat{u_{on}}(t)$ and $\widehat{u_{on}}(t)$ were independently set to 1 with probability 0.0005, zero otherwise [1]. The pulses were then smoothed to amplitude 0.4 and duration 100ms. Thus, $u_{on}(t) = \frac{1}{\sigma_u}\left(\theta_0 \; o \; (\hat{u}_{on} * h)\right) * g$, and $u_{off}(t)$ is similarly constructed with $\widehat{u_{on}}(t)$. The functions g and h are $g(s) = \exp(-s/\tau_L)\theta_1(s)$ and $h(s) = \theta_1(s) - \theta_1(s - 100ms)$, with τ_L = 50ms and $\theta_x(s)$ being the Heaviside function equal to 0 for x < 0, x for s = 0, and 1 otherwise. The target function $f(t) = \frac{1}{\sigma_f}\hat{f}(t) * g$ with amplitude 0.5 for the readout neuron was then a smoothed version of $\hat{f}(t) =$ 0.5 $if\ \hat{u}_{on}(t) = 1$, 0.5 $if\ \hat{u}_{on}(t) = 1$, and $\hat{f}_i(t - dt)$ otherwise.

3 Results

3.1 Reproduction of Hoezer, Legenstein, and Maass's model

This study began by recreating some of the main results from [1], namely, the production of periodic signals and a simple working memory task. We created a 1000 neuron recurrent neural network with dynamics identical to the model in [1] and trained the network to recreate a periodic function consisting of the sum of five sine waves of differing frequencies. The network was trained using the binary reward modulation signal for 500 seconds of simulation time. The figure below compares the output of the network after training and the target function. It can be seen that the network reproduces the target function closely, but over time the phase drifts away from the perfectly periodic target function.

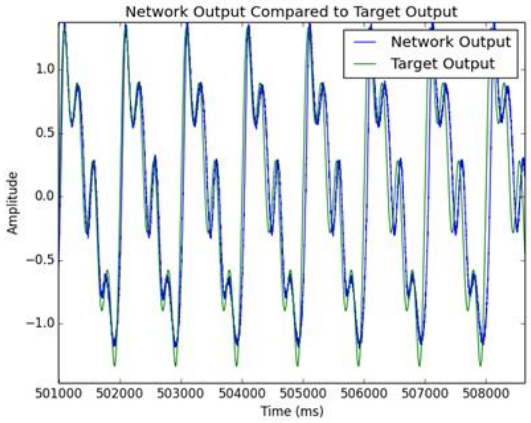

Figure 1. Comparison between the network output after training and the periodic target function when using a binary modulatory signal

Next, we trained the same network to perform the simple working memory task from the [1]. Two input streams were connected to the network with random, uniformly distributed weights to each neuron in the network. The readout output was trained to drive to a high value (0.5) when the most recent input stream that was active was input stream "on", and to drive to a low output (-0.5) when the most recent active input stream was stream "off". The figure below shows the output of the network compared to a target function that idealizes the desired outcome of the task:

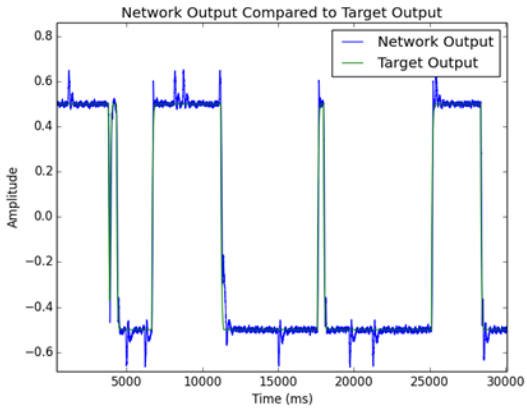

Figure 2. Comparison between the network output after training and the working memory target function when using a binary modulatory signal

While the state transitions are clear and correct, the network transiently deviates from the target function. These deviations are artifacts of the input stream activations and are present because network dynamics are not instantaneous. Taken together, these results show that our model successfully reproduces the key results of [1]. Thus, our model can be used as a platform upon which to investigate how dopamine-like reward signals modulating Hebbian learning affect learning in liquid state machines.

3.2 Modification of the modulatory signal

3.2.1 Implementing a nonbinary, but discrete modulatory signal

While M(t) = 0 or 1 distinguishes between absent and high transmission of a reward signal, a binary signal does not accurately reflect physiological transmission of reward signals like dopamine. In reality, dopamine transmission is both phasic and tonic, with some persistent, baseline dopamine tone at all times. In addition, phasic dopamine transmission can vary in amplitude, exhibiting characteristics of an analog signal. Therefore, to better represent physiological reward signals, we altered M(t) and tested the network's performance in the simple working memory task from [1].

First, M(t) was altered such that M(t) = 0.9 at performance greater than P_{avg} and 0.1 otherwise. This nonbinary, yet discrete signal accurately reflects baseline reward signal or tonic dopamine transmission, though without the continuous nature of an analog reward signal. With M(t) defined so, performance on the simple working memory task after 300s of training was measured. Performance was calculated as the percent of time after training in which the network output was within criterion 0.5 from the target function. With the nonbinary, yet discrete signal, the network achieved 85% ± 11% performance compared to 93% ± 3% performance with the binary M(t) modulatory signal (95% confidence level). While these performance levels are not significantly different, the larger standard error with the nonbinary signal indicates that with M(t) = 0.1 or 0.9, working memory emergence and performance are less consistent per trial and potentially more vulnerable to random network configurations, chaoticity, and noise (Figure 3). This is likely because the baseline discrete signal permits weight change even if performance is significantly poorer than average, leading to inappropriate weight change and failure to converge to the target output.

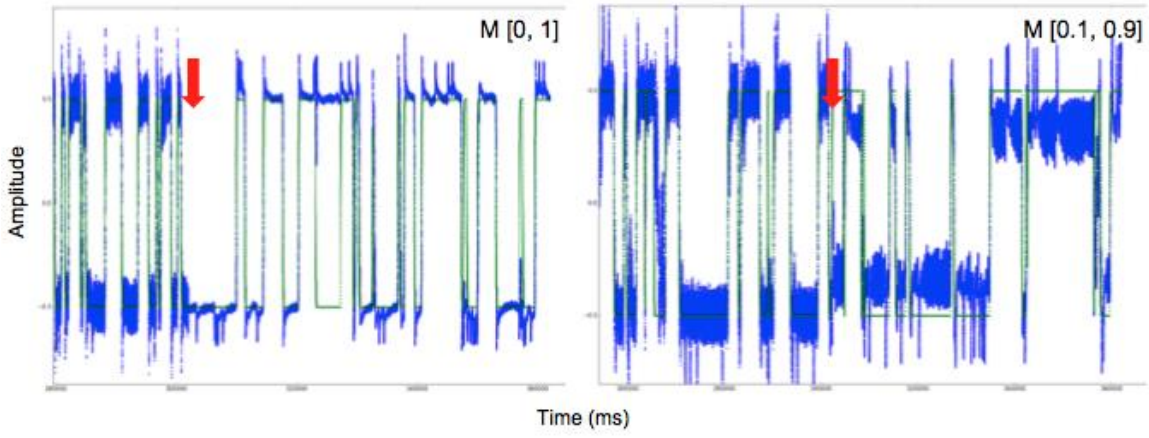

Figure 3. Comparison of network output (blue) and target function (green) with a binary modulatory signal (left panel) or a nonbinary, but discrete signal (right panel). The red arrows represent the end of training.

3.2.2 Implementing an analog modulatory signal

In the brain, dopaminergic neurons fire with tonic activity, and their activity has been observed to encode a sort of reward signal. More specifically, dopamine neurons have been recorded firing in ways that represent a sort of "reward prediction error" [2]. This means roughly that upon presentation of an unexpected reward, dopaminergic neurons will increase their firing rate above baseline, while when a predicted reward is not observed, they will decrease their firing rate below baseline. The model we investigated did not originally include an analog reward signal with similar dynamics to dopaminergic neurons. Thus, we modified the model by changing the reward signal to a function that approximated the behavior of the dopaminergic neurons.

In line with the original model, we thought of the average performance computed with a moving-average filter as being analogous to the predicted reward. Thus, at any time, the current "reward prediction error" would be given by the difference between the current performance P and the average performance P_{avg}. We then used this difference as the independent variable in a linear equation giving the reward signal. We set the intercept of this equation to be 0.1 to represent a constant level of dopamine from tonically active dopaminergic neurons. We set the slope of the linear function to be (0.9-0.1) = 0.8 to approximate the conditions imposed by the previous non-binary discrete reward signal. Finally, a nonlinearity was added in the form of a hard cutoff at 0: If the linear function produced values below 0, it was artificial set to 0. The combination of a strict lower bound, a nonzero baseline signal, and dynamics depending upon a difference between a stored and observed variable make this function a reasonable approximation to the observe behavior of dopaminergic neurons.

The network was successfully able to learn the working memory task using the "dopamine-like" reward signal (Figure 4). Interestingly, the network showed a slight increase in performance during the testing period compared to the original model. The model trained with the "dopamine-like" reward signal achieved a performance of (96.2 ± 0.99)%. This contrasts with the original model which achieved a (93 ± 3)% performance. Thus, for the modified model, the performance increased and the variation in performance decreased.

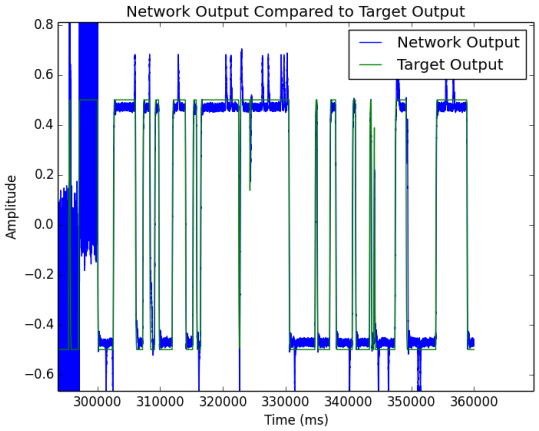

Figure 4. Comparison between the network output after training and the working memory target function when using an analog "dopamine-like" modulatory signal

The analog reward signal we have described is not the only analog reward signal that could be constructed. One obvious choice is the raw difference between the current performance and the average (predicted) performance. For comparison, we used this raw difference to train a different model to perform the working memory task. This new network was also able to successfully learn the task, but a comparison of the network output of this raw performance difference network to the output of the "dopamine-like" network show drastic differences. Namely, the raw performance

difference network consistently failed to follow the target function as precisely as the other models. While the transitions were correct, the amplitudes and variances of the output signals were very different and led to a poorer approximation. The figure below illustrates the differences seen between the "dopamine-like" network and the raw performance difference network.

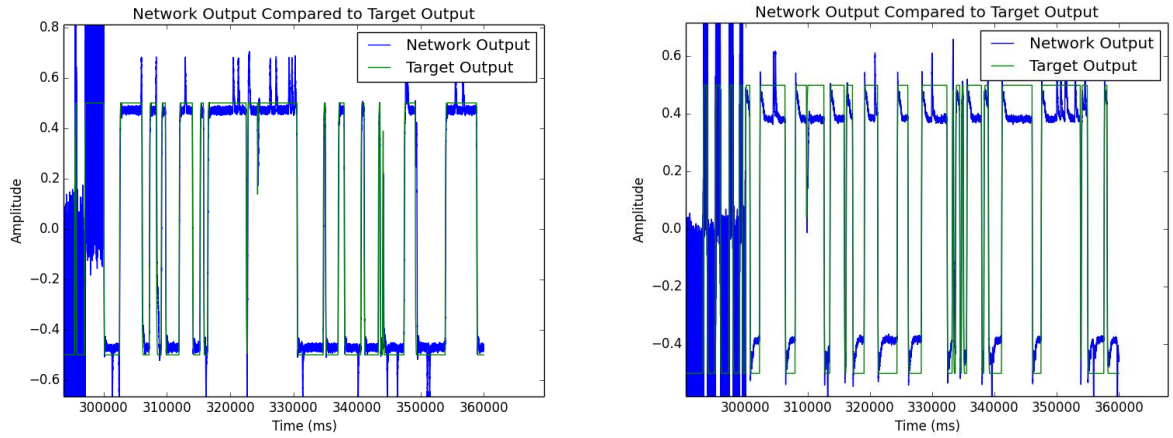

Figure 5. Comparison of working memory emergence with a "dopamine-like" modulatory signal (left panel) and a raw performance difference signal (right panel)

One possible explanation for the apparent improvement in performance for the "dopamine-like" reward signal could be based on the fact that a baseline reward signal implies that the weights are changing if performance is not significantly worse than average. In this way, even when the network performance dips slightly below average, the weights still change to explore new regions of weight space. This contrasts with using the raw performance difference signal, in which the weight change can be anti-Hebbian. This might imply that if training finds a set of weights that lead to a relatively constant performance, then if the network tries to explore new regions of weight space with poorer performance, anti-Hebbian learning will prevent the weights from leaving that region. Thus, it is conceivable that using the raw performance difference signal increases the probability of the network to fall into local minima. However, the constant weight change of the "dopamine-like" reward signal allows the network to find more global minima so long as performance is not significantly poorer than average, in which case the hard cutoff signal of zero prevents further exploration.

3.3 Development of a delayed non-matching to sample task

In order to examine biological relevance of our model with modified modulatory signal, we conceptualized a behavioral working memory test called delayed non-matching to sample task. The test consists of three phases: sample, delay, and test phase. During the sample phase, the rodent is presented with a sample stimulus (e.g. a left lever). After the rodent presses the sample lever, a food pallet is presented at the opposite side of the chamber (delay phase). After a short delay, the sample stimulus (i.e. left lever) is shown again along with a novel alternative (i.e. right lever). In this paradigm, the rodent is rewarded with food pallet for selecting the novel stimulus (i.e. right lever). This delayed non-matching to sample task is a good behavioral assessment for working memory emergence as it requires the animal to hold information about the sample stimulus in the online workspace throughout the delay phase in order to correctly select the novel stimulus during test phase (Figure 6). In our model, the sample stimulus and the novel stimulus can be thought of as input 1 and input 2 respectively. During the sample phase, the network is presented only with input 1, which has to be retained in the network's working memory throughout the delay phase. In the test phase, the network is presented with both input 1 and input

2. In order for network output to match the target function, the network has to correctly reject input 1 and select input 2 during the test phase.

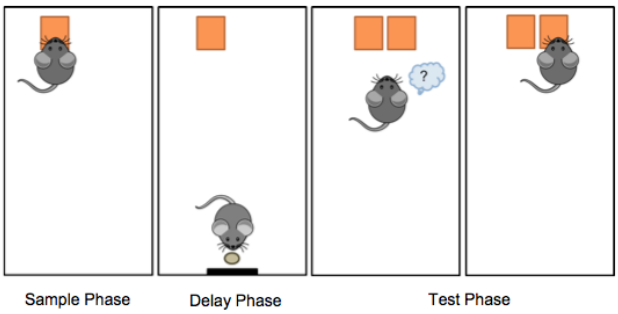

Figure 6. Schematic of delayed non-matching to sample task

In the delayed non-matching to sample task, we first tested working memory using both the binary and nonbinary, yet discrete modulatory signals M(t). With these signals, performance levels were 71% ± 5% and 72% ± 10%, respectively (confidence level 95%). These performance levels were significantly worse than those with either discrete modulatory signal in the original working memory task ($p < 0.001$). Thus, the network performed more poorly on the delayed non-matching to sample task than on the original task. In particular, analysis of individual runs indicates that the network was able to achieve correct output when one input stream was nonzero, but not when both input streams were nonzero. The network was not able to retain information about the original stimulus and switch its output when presented with the original stimulus and the novel stimulus. It is possible that different types of network architecture or non-Hebbian learning rules could lead to improved performance with the delayed non-matching to sample task.

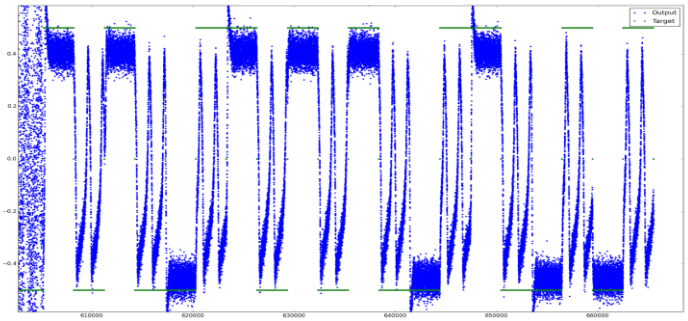

Figure 7. Failure to converge to target function in the delay non-matching to sample task

3.3.1 Implementing a delayed non-matching to sample task with a "go signal"

In an attempt to get the network to learn some version of the delayed non-matching to sample task, we added a third input stream to serve as a "go signal." The rationale for its inclusion was the possibility that since in the original implementation of this new task, the individual input streams require different outputs at different times, and that the network does not implement an explicit temporal difference learning rule, the network was unable to differentiate the meanings of the original input signals. We redesigned the task to present one pulse on one of the input streams, followed by a pulse on the "go signal" stream. Upon this pulse, we directed the network to choose the opposite value compared to the input stream. Thus, for every trial of this task, either original input stream requires only one output at any time. A binary modulatory signal was used.

Unfortunately, the network performed even worse on this task. The figure below shows the network output during and after training compared to the target function.

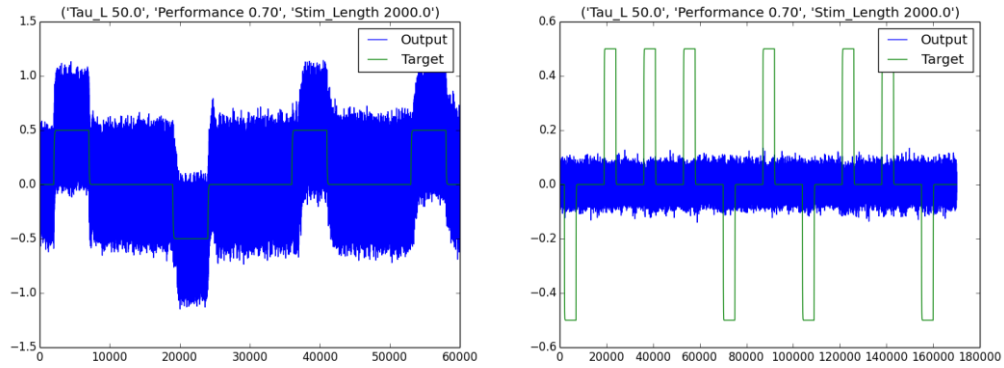

Figure 8. Comparison of network output and target function before (left panel) and after (right panel) training with a go-signal delay non-matching to sample task

Following 500 seconds of training, the network showed what appeared to be random noise throughout the testing period. No evidence was seen for any state transitions that would be indicative of the network trying to accomplish the task. Future work will have to investigate whether there is a fundamental reason the network was unable to learn this task.

4 Conclusions

This study examines working memory emergence using LSM framework. While several other studies trained their recurrent neural networks using supervised paradigms, the current study used unsupervised, reward-modulated Hebbian learning to examine working memory emergence in recurrent neural networks. We first reproduced a model proposed in [1] and showed matching target output in recurrent neural networks via reward-modulated Hebbian learning.

In addition, we replaced the binary modulatory signal used in to the original model [1] with a nonbinary, but discrete signal that represented the dynamics of the networks more accurately. We further modified the reward signal to reflect dopaminergic neurons' "reward prediction error," where dopaminergic neurons increase their firing rate above baseline when an unexpected reward is presented, and decrease their firing rate below baseline when a predicted reward is not observed. To incorporate this behavior of dopaminergic neurons into our model, we implemented an analog modulatory signal where, at any time, the current "reward prediction error" was given by the difference between the current performance and the average performance on the task, though with a lower limit of zero. We demonstrated that the our modified model with the "dopamine-like" reward signal, compared to the discrete signals, led to increased performance and decreased variation in performance on the original working memory task.

Finally, in order to further examine working memory emergence, we designed a behavioral working memory test (delayed non-matching to sample task) and trained the network to learn it. This task required the network to hold information regarding a sample stimulus (input 1) through the delay period and select a novel stimulus (input 2) when both novel and sample stimuli are presented during the test phase. However, we found that the neural network performed more poorly on the delayed non-matching to sample task than on the original task: the network could not achieve output matching the target function when two input streams were presented (i.e., nonzero). Future studies will be aimed at determining what network architectures, modulatory signals, or learning rules will facilitate the learning of the non-matching to sample task.

References

[1] Hoerzer, G.M., Legenstein, R., & Maass, W. (2014) Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning. *Cerebral Cortex* **3**: 677-690.

[2] Lak, A., Stauffer, W. R., & Schultz, W. (2014) Dopamine prediction error responses integrate subjective value from different reward dimensions. *PNAS* **111**(6): 2343-2348.

[3] Sussillo, D., and Abbott, L.F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**: 544-557.