Improving Learning Efficiency of Recurrent Neural Network through Adjusting Weights of All Layers in a Biologically-inspired Framework

Xiao Huang*, Wei Wu*, Peijie Yin† and Hong Qiao*‡
*Institute of Automation, Chinese Academy of Sciences, Beijing, China,
Email: huangxiao2015@ia.ac.cn, wei.wu@ia.ac.cn
†Institute of Applied Mathematics, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, China,
Email: yinpeijie@amss.ac.cn
‡State Key Lab of Management and Control for Complex Systems, Institute of Automation,
Chinese Academy of Sciences, Beijing, China,
CAS Centre for Excellence in Brain Science and Intelligence Technology (CEBSIT), Shanghai, China
University of Chinese Academy of Sciences, Beijing, China
Email: hong.qiao@ia.ac.cn

Abstract—**Brain-inspired models have become a focus in artificial intelligence field. As a biologically plausible network, the recurrent neural network in reservoir computing framework has been proposed as a popular model of cortical computation because of its complicated dynamics and highly recurrent connections. To train this network, unlike adjusting only readout weights in liquid computing theory or changing only internal recurrent weights, inspired by global modulation of human emotions on cognition and motion control, we introduce a novel reward-modulated Hebbian learning rule to train the network by adjusting not only the internal recurrent weights but also the input connected weights and readout weights together, with solely delayed, phasic rewards. Experiment results show that the proposed method can train a recurrent neural network in near-chaotic regime to complete the motion control and working-memory tasks with higher accuracy and learning efficiency.**

I. Introduction

In recent years, researches on biologically-inspired models have become a hotspot. Many different types of brain-inspired neural networks have been developed to solve complex cognition, decision making and motor control problems [1]–[3]. In our brain, neural networks in different parts of the cerebral cortex generally perform a large variety of different computations and pattern generation tasks in visual recognition and motor control [4]. But it is still challenging for computers to perform these tasks as effectively as human.

Many artificial neural networks have been designed to mimic human behavior. However, many of them are not biologically plausible, because (1) human brain is a highly dynamical system rather than a feedforward neural network with stable activation of neurons; (2) the connections between neurons are very complicated and highly recurrent, but many artificial neural networks don't have many recurrent connections, which greatly influence the performance. On the contrary, many recurrent neural networks (RNNs) are able to produce a wide range of dynamical behaviors because their recurrent connectivity gives them more internal states along time to generate more complex activities of neurons. So far, these neural networks could mimic activities of the brain to some extent.

However, training these recurrent neural networks is generally difficult because of the encoded time information. To train some simple networks, backpropagation through time is a popular way to tune the weights in supervised learning regime. But it generally requires a derivable cost function and a constant supervisory signal, which is absent in most actual cognitive and decision making tasks, especially in some delayed-reinforcement tasks.

How to solve these problems may depend on the research findings of neuroscience. In macroscopic view, according to the related work in neuroscience, emotions are powerful determinant factors of people's perception, cognition, decision-making and other behaviors. Emotion appraisal is considered as one of the most important mechanisms that can modulate cognition, motion control and decision making [5]–[7]. Meanwhile, in macroscopic view, it is well known that populations of neurons play an important role in cognitive computing and motor learning. Many recent experimental studies suggest that the cortex can adapt its functions to optimize performance during learning [8]–[10]. In this process, several studies have shown that populations of neurons can simulate animals behaviors quite well. For example, a delayed matching-to-sample task drawn by [8] has demonstrated that populations of neurons in the prefrontal cortex continually changed their response properties during the visual learning process. Adaptive and extended training of working memory tasks can improve the performance of cognitive tasks, which is associated with changes of population of neurons activities in frontal cortex and basal ganglia [9]. In the work of [10],

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

functional adaptation-related changes in the neurons of motor cortex have been demonstrated. These neurons are found to change their tuning properties with only feedback of movement results.

Inspired by these neurobiological mechanisms, the REINFORCE class of algorithms have been applied to train a neural network, which enable it to fulfill the delayed-reinforcement tasks [11]. The recurrent neural network is applied in the reservoir computing framework, which takes the advantages that this recurrent neural network in near-chaotic regime exhibits complex dynamics, highly reminiscent of neural activity, and it can successfully implement flexible decision making, motor control, associations and memory maintenances [12].

In this paper, a novel reward-modulated Hebbian learning rule is introduced, which is able to adjust not only the internal recurrent weights but also the input connected and readout weights by a delayed reward signal. Based on the REINFORCE class of algorithms, this method combines liquid computing theory and node-perturbation Hebbian learning to train the weights of each layer in the network, which enables further parameter tuning to promote performance. After training, the model is able to control a two-link robotic manipulator arm to perform a center-out task with eight target points successfully, and eventually reproduce the corresponding trajectories with great accuracy in response to different inputs. In addition, the trained network can also accomplish the delayed nonmatch-to-sample tasks successfully, which indicates that this model also has the ability to fulfill working-memory related tasks.

II. Related Work

Recently, cortical computation in cognitive tasks has been seen as a highly dynamic activity of populations of neurons [13]–[16]. In some researches [17], [18], it is shown that models based on recurrent neural networks may capture similar dynamics in higher cortical areas such as motor cortex. As a result, such network models have been applied to perform many cortical computation tasks, such as motor control [19]–[21] and working memory [22].

A range of supervised RNN training methods have been proposed from the classical BP to reservoir methods: (1) Backpropagation Through Time [23], [24]; (2) Atiya-Parlos recurrent learning [25]; (3) BackPropagation-DeCorrelation [26]; (4) Echo State Networks (ESNs) [27]. In this list, the focus of training gradually moves from the entire network towards the output for faster convergence, and weights of the networks are usually optimized by regression or gradient descent method. ESN is a famous architecture and supervised learning principle for RNNs, pertaining to liquid computing theory. When liquid computing theory is implemented into these recurrent network models, specific computational functions are acquired through modification of the weights from a population of neurons to readout neurons [4]. Hence, only weights of readout neurons are adjusted during learning process. However, these methods based on supervised learning are not enough biologically plausible, and generally require a derivable cost function and a constant supervisory signal.

Besides supervised learning principle, reward-modulated Hebbian learning [4], [12] is another effective rule for training RNNs. This method is inspired by the modulation effects of dopamine on synaptic plasticity in neuroscience, where the changes of synaptic weights depend not only on the activities of pre-synaptic and post-synaptic neurons, but also on the reward or punishment signals [28]. In contrast with supervised learning rules, this is more biologically plausible. For training process, employing the node-perturbation method [29], [30] to estimate gradient has been demonstrated effective to train the RNNs in reward-modulated Hebbian learning [4], [31]. From reinforcement learning perspective in this paper, a novel reward-modulated Hebbian learning rule is proposed to train the RNN with sparse and delayed rewards for flexible decision tasks.

III. Method

A. Network Architecture

How to train a biological recurrent neural network to generate different patterns just by one-time reinforcement of reward after each trial is an interesting question. To fulfil a wide range of temporal and spatiotemporal tasks usually requires encoding time in the dynamic changes in the pattern of activity of neurons. Population of neurons could be considered as an important way to encode information for neural computations, where the activities of neurons at any given time can be projected to a point in a high-dimensional space. These points could form a network trajectory over time, which can be visualized by principal component analysis of the activities of neurons. Clearly, the advantage of such computing is to encode time into the trajectory so that it can deal with temporal and spatiotemporal process. In this paper, a fully connected recurrent neural network of neurons in applied, and each of them is a leaky integrator. In this model, the membrane potential of each neuron follows first order differential equations.

$$\tau \dot{x}_i = -x_i + \sum_{j=1}^{N} W_{ij}^{rec} r_j + \sum_{k=1}^{N_{in}} W_{ik}^{in} u_k + \sqrt{2\tau\sigma_{rec}^2}\xi_i \quad (1)$$

$$r_i = \tanh(x_i) \quad (2)$$

$$z_l = \sum_{i=1}^{N} W_{li}^{out} r_i \quad (3)$$

where τ is the time constant of neuron activation, x_i is the neurons membrane potential, W_{ij}^{rec} is the synaptic weight from neuron j to neuron i, W_{ik}^{in} is the weight from input current u_k to neuron i, W_{li}^{out} is the readout weight, r_j is the firing rate of the neuron j, z_l is the output of the readout unit l. The recurrent noise has been supposed to yield to normal distribution with zero mean and $\sqrt{2\tau\sigma_{rec}^2}$ variance [23]. These noises are actually used to estimate the policy gradient [26] in order to produce exploratory variation in the output of network. If noises are set to zero, the network wont have the learning ability.

B. Reward-modulated Hebbian learning Rule

A neural network of N neurons is trained from episode to episode, where each episode consists of T time steps. At the end of each episode, a reinforcement reward based on the performance is delivered to the network.

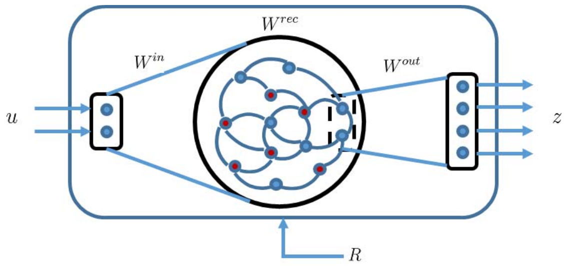

Fig. 1. The structure of the neural network.

According to episodic REINFORCE theory [11], each synaptic weight W_{ij}^{rec} is modified as

$$\triangle W_{ij}^{rec} = \eta(R - R_b)\sum_{t=1}^{T} e_{ij}^{rec}(t) \tag{4}$$

where η is a learning rate, R_b is the baseline of reinforcement, $e_{ij}(t)$ represents the characteristic eligibility for weight W_{ij}^{rec} at time step t. The baseline R_b is an adaptive estimate of upcoming reward based on past experience, so that $R - R_b$ represents the reward prediction error signal. A simple approach to compute the baseline is to maintain a running average of actual rewards [32]. So at the n^{th} episode, the reinforcement baseline is updated by

$$R_b(n) = \alpha_r R_b(n-1) + (1 - \alpha_r)R(n) \tag{5}$$

where $0 < \alpha_r \leq 1$. The characteristic eligibility $e_{ij}(t)$ represents a potential synaptic weight change accumulated by synapse from neuron j to neuron i. Define $G(\rho) = Pr\{z = \rho | W^{rec}\}$, then the characteristic eligibility is $e_{ij} = \partial \ln G_i / \partial W_{ij}^{rec}$ which is used to estimate the decent direction. However, in the first order differential equation above, it is difficult to get the derivative of weight W_{ij}^{rec} from readout units. To solve this problem, one method is to implement derivation based on the machine learning library, such as theano [23]. Another method is to estimate the eligibility trace [12] as

$$e_{ij}^{rec}(t) = S(x_i(t) - \bar{x}_i(t))r_j(t-1) \tag{6}$$

where S is a supralinear function, for example $S(x) = x^3$, for amplifying the large deviations and suppressing small ones to learn from delayed, time-sparse rewards.

In this paper, we propose that not only recurrent weights W^{rec} are modified during learning, but also input weights W^{in} and output weights W^{out} should be adjusted to maximize the rewards. Similar to the description above, input weight W^{in} is changed according to the following equation:

$$\triangle W_{ik}^{in} = \eta(R - R_b)\sum_{t=1}^{T} e_{ik}^{in}(t) \tag{7}$$

Algorithm 1 Reward-modulated Hebbian Learning Rule.

```
Initialize:
  Initialize a recurrent neural network.
Training:
for $i = 1$ to $N_{trial}$ do
    for $t = 1$ to $T$ do
        Compute the output of network $z_t = net.step(u_t)$;
    end for
    return  $z = [z_1, z_2, ..., z_T]$;
    Compute the response of network $y = Sys(z)$;
    Compute the reward of this trial $R = Rew(y, y^{target})$
    based on an objective function;
    Update the weight of network $net.update(R)$.
end for
```

where the characteristic eligibility for input weight W_{ij}^{in} at time step t is:

$$e_{ik}^{in}(t) = S(x_i(t) - \bar{x}_i(t))u_k(t-1) \tag{8}$$

The update of the output weigh W^{out} is achieved following a process similar to the methods above, except for the computation of the eligibility trace. The reason is that there is a linear relationship between z and r.

$$\triangle W_{li}^{out} = \eta(R - R_b)\sum_{t=1}^{T} e_{li}^{out}(t) \tag{9}$$

where the eligibility trace for input weight W_{li}^{out} is inspired by the work of [23]:

$$e_{li}^{out}(t) = (z_l(t) - \bar{z}_l(t))r_i(t-1) \tag{10}$$

To train the RNN, an objective function that includes not only the error but also some regulation terms is defined. Generally, the error can be measured by computing the error sum of squares of the differences between the target and actual results. The regulation terms are designed to encourage sparse weights or minimize the activation energy. The objective function is further described in the experiment section. To learn cognitive and motor control tasks, the reward-modulated hebbian learning rule includes several important steps. The algorithm is shown as Algorithm 1.

IV. Experiment

To evaluate the effectiveness and efficiency of the new proposed algorithm, the new learning rule is applied to train a recurrent neural network in two tasks: (1) the center-out reaching task and (2) delayed nonmatch-to-sample task, respectively. The first task is designed to test the ability of motor pattern generation and control; while the second task is to test working memory and flexible decision making.

A. Center-Out Reaching Task

The classic task for studying voluntary decision making and motor control is the ”center-out reaching task”, where a monkey moves its hand from a central location to target

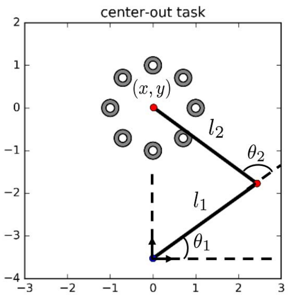

Fig. 2. The center-out reaching task.

points on a circle around the starting position. An experiment is designed where a two-link robotic manipulator arm can change its pose in order to make the end point of link reach the indicated target point from central starting point. In this experiment, there are eight possible target points surrounding the starting point as shown in Fig. 2. Let $p = (x, y)$ denote the position of the end of link. Letting l_1 and l_2 be the length of the two links, we have

$$\begin{cases} \dot{x} = -(l_1 s_1 + l_2 s_{12})\dot{\theta_1} - l_2 s_{12}\dot{\theta_2} \\ \dot{y} = -(l_1 c_1 + l_2 c_{12})\dot{\theta_1} - l_2 c_{12}\dot{\theta_2} \end{cases} \tag{11}$$

where $s_i = \sin\theta_i$, $s_{ij} = \sin(\theta_1 + \theta_2)$, and similarly for c_i and c_{ij}. Then we let the output of network be the input of system as $z_i = \triangle\theta_i$.

After maintaining fixation on the central location for 50 ms, the robotic manipulator arm executes a movement from current position to target point within 100 ms. For this task, a recurrent neural network with 200 units is trained, and the weights of network are updated by reward-modulated Hebbian learning rule at the end of each trial. During training, every trial consists of eight sub-trials, in which the network randomly receives an input to indicate one of eight possible target positions. Moreover, the robotic arm move to point y under the control of a sequence of readout signals at each trial is assumed, and the target point is defined as y^{target}. Then the objective function is defined as

$$L = \frac{1}{d}\sum_{i}^{d} |y_i - y_i^{target}| + \frac{\lambda}{TN}\sum_{j=1}^{T}\sum_{i=1}^{N}|r_{ij}| \tag{12}$$

$$R = -L \tag{13}$$

where $d = 2$ is the dimension of the coordinates of the point, λ is the parameter of regularization term, r_{ij} represents the fire rate of neuron i time step j.

As we know, a good choice of initialization parameters can decrease the learning time. In the experiments, each element in the recurrent weight W^{rec} is set to zero with probability $1 - p$, and the remaining fraction is initialized to non-zero values drawn from a Gaussian distribution with zero mean and variance $g/(pN)$ [17]. For the input weight W^{in} and output weight W^{out}, they are initialized with a uniform distribution over a small range. In addition, the membrane potential x_i of each neuron i is initialized with uniform noise in a small range at the start of every episode. Parameters of the recurrent neural networks are shown in TABLE I in detail.

TABLE I
PARAMETERS OF THE RECURRENT NEURAL NETWORK

Parameter	Symbol	Default Value
Learning rate of input and recurrent weight	η_{rec}, η_{in}	0.5
Learning rate of output weight	η_{out}	0.01
Spectral radius of recurrent weight	g	1.1
Probability of recurrent weight	p	0.5
Number of neurons	N	200
Number of input units	N_{in}	9
Number of output units	N_{out}	2
Unit time constant	τ	30ms
Time step	$\triangle t$	1ms
Standard deviation for recurrent noise	σ_{rec}	0.15
Filter factor of reward	α_r	0.33
Parameter of regularization term	λ	0.1

Results are shown in Fig.4. As we can see, performance improves with increasing trials, and error reaches a low value after about 1000 trials. The proposed method can obtain a higher accuracy quickly in contrast with the original one. After training, one hundred test trials are performed to investigate the trajectories of motion and errors between end point and target position (Fig.3). In summary, the training is sufficient and the network is able to reproduce the corresponding trajectories with great accuracy in response to different inputs. The robotic arm is able to correctly reach the corresponding target according to the input signal.

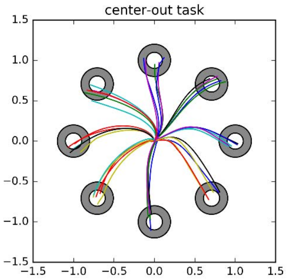

Fig. 3. Trajectory of motion after training.

To visualize the influence of the training process in the recurrent neural network, first three principal components of the firing rates of neuron population are extracted, based on the five trials for the eight targets both before (Fig.5) and after training (Fig.6). It is clear that training can change the trajectories in the state space for each sequence, and similar characteristics between different movements can be well captured.

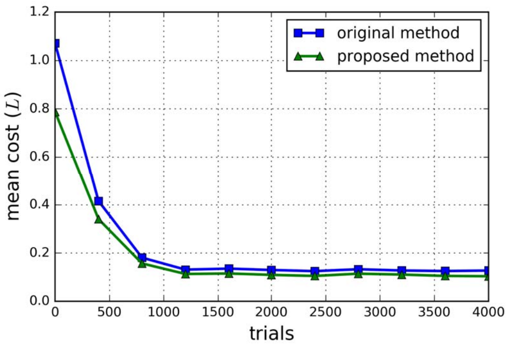

Fig. 4. The change of cost during training.

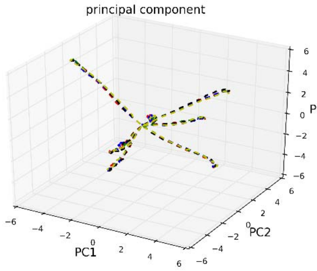

Fig. 5. Principal components of neurons activities before training.

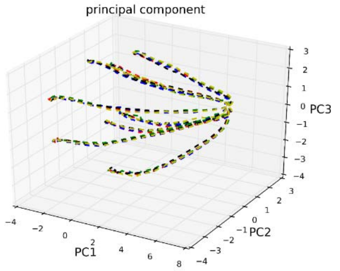

Fig. 6. Principal components of neurons activities after training.

After 4000 trials training, 100 test trials are carried out to evaluate the performance. The mean of total errors is 0.106 ± 0.045. The errors of reaching the eight target points are shown in TABLE II.

TABLE II
THE ERRORS OF REACHING EIGHT TARGET POINTS

Point	Original Method	Proposed Method
1	0.126 ± 0.054	0.117 ± 0.025
2	0.123 ± 0.054	0.133 ± 0.078
3	0.137 ± 0.076	0.097 ± 0.059
4	0.121 ± 0.040	0.094 ± 0.027
5	0.112 ± 0.048	0.097 ± 0.036
6	0.124 ± 0.059	0.093 ± 0.047
7	0.137 ± 0.051	0.082 ± 0.031
8	0.125 ± 0.048	0.138 ± 0.053

B. *Delayed Nonmatch-to-Sample Task*

The delayed nonmatch-to-sample task is widely used to test working memory for learned associations in animals. Typically, the animal (such as rats or monkeys) is presented with a sample stimulus and a comparison stimulus, but there is a short delay between them. In the nonmatching paradigm, the animal gets reward if it selects the nonmatch stimulus. Moreover, by changing the length of the delay, people can study how long the animal can retain information in its working memory. Here, the reward-modulated hebbian learning rule is employed to train the population of neurons to finish this task with great accuracy.

In the experiment, a simple delay nonmatch-to sample task is designed following the work [12]. It can be considered a timed XOR problem, in which if the two successive stimuli are identical, the output of network should be -1, and otherwise the network should output 1. Each trial is set to 1 second long, and the first input is lasting for 200 ms, then second input is also lasting for 200 ms after a 200 ms delay. Response of the network is determined by the activities of neurons over last 200 ms. Moreover, the objective function is defined as

$$L = \frac{1}{T_{eval}} \sum_{t=1}^{T_{eval}} |y_t - y_t^{target}| \tag{14}$$

$$R = -L \tag{15}$$

The cost change curve is shown in Fig.7. It is clear that the network is able to respond to different inputs correctly with higher accuracy when the number of training trials is increasing. After 4000 training trials, 100 test trials are carried out to evaluate the performance, and the precision can reach $99.5 \pm 0.4\%$. In Fig.7, the blue line represents the model where only internal recurrent weights are adjusted, and the green line is the newly proposed model. Clearly, changing all the weights with newly proposed learning rule can increase the speed of learning.

In this task, there are four cases in total. In each case, the activities of output neuron and two other neurons are drawn in Fig.8. Clearly, the activities of the output neuron is able to make correct decision based on past delayed stimulus during last 200ms. At the same time, the traces of other neurons still remain highly dynamical.

V. CONCLUSION

In this paper, inspired by the modulation of human emotions on the learning process of cognition, motion control and decision making, a novel reward-modulated Hebbian learning rule is developed to train a recurrent neural network. Based on the REINFORCE class of algorithms, the new algorithm changes not only the internal recurrent weights but also the input connected weights and readout weights by a delayed reinforcement signal. This method combines liquid computing theory and node-perturbation Hebbian learning to train the weights of each layer, which allows for further parameters tuning to improve performance. To evaluate the effectiveness and efficiency, this new learning rule is applied to train a recurrent

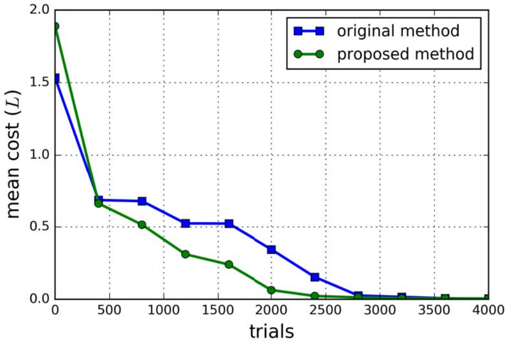

Fig. 7. The change of cost in original and proposed method.

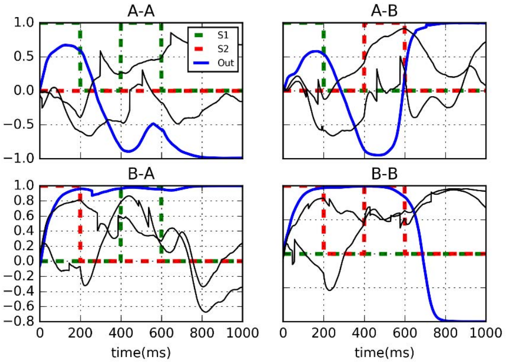

Fig. 8. The activities of output neuron and two other neurons.

neural network in (1) the center-out reaching and (2) delayed nonmatch-to-sample task, respectively. In the first task, the model is able to control a two-link robotic manipulator arm to finish a center-out task with eight target points successfully, and eventually reproduces the corresponding trajectory in response to different inputs with a high accuracy. In the second task, the network is trained to do the delayed nonmatch-to-sample task. After training, the activities of output neuron are able to make correct decision based on past delayed stimulus, which indicates that this model has the ability to fulfill working-memory tasks. Comparing with the model where only internal recurrent weights are adjusted, the new proposed algorithm can increase the speed of learning. In the future, it is interesting to analyze the relationship between the new proposed method and traditional reinforcement learning.

Acknowledgment

This work is supported by the Strategic Priority Research Program of the CAS (Grant XDB02080003), Beijing Municipal Science and Technology (Grant D16110400140000 and D161100001416001), National Natural Science Foundation of China (Grant 61210009), National Natural Science Foundation of China (Grant 61627808) and National Natural Science Foundation of China (Grant 91648205).

References

[1] H. Qiao, Y. Li, F. Li, X. Xi and W. Wu, ”Biologically Inspired Model for Visual Cognition Achieving Unsupervised Episodic and Semantic Feature Learning”, IEEE Transactions on Cybernetics, vol. 46, no. 10, pp. 2335-2347, 2016.

[2] H. Qiao, X. Xi, Y. Li, W. Wu and F. Li, ”Biologically Inspired Visual Model With Preliminary Cognition and Active Attention Adjustment”, IEEE Transactions on Cybernetics, vol. 45, no. 11, pp. 2612-2624, 2015.

[3] Wu W, Qiao H, Chen J, et al. Biologically inspired model simulating visual pathways and cerebellum function in human-Achieving visuomotor coordination and high precision movement with learning ability[J]. arXiv preprint arXiv:1603.02351, 2016.

[4] G. Hoerzer, R. Legenstein and W. Maass, ”Emergence of Complex Computational Structures From Chaotic Neural Networks Through Reward-Modulated Hebbian Learning”, Cerebral Cortex, vol. 24, no. 3, pp. 677-690, 2012.

[5] T. Brosch and D. Sander, Comment: The Appraising Brain: Towards a Neuro-Cognitive Model of Appraisal Processes in Emotion, Emotion Review, vol. 5, no. 2, pp. 163C168, Apr. 2013.

[6] M. L. Kringelbach, A. Stein, and T. J. van Hartevelt, The functional human neuroanatomy of food pleasure cycles, Physiology & Behavior, vol. 106, no. 3, pp. 307C316, Jun. 2012.

[7] E. A. Phelps, Emotion and Cognition: Insights from Studies of the Human Amygdala, Annual Review of Psychology, vol. 57, no. 1, pp. 27C53, Jan. 2006.

[8] G. Rainer and E. Miller, ”Effects of Visual Experience on the Representation of Objects in the Prefrontal Cortex”, Neuron, vol. 27, no. 1, pp. 179-189, 2000.

[9] T. Klingberg, ”Training and plasticity of working memory”, Trends in Cognitive Sciences, vol. 14, no. 7, pp. 317-324, 2010.

[10] B. Jarosiewicz, S. Chase, G. Fraser, M. Velliste, R. Kass and A. Schwartz, ”Functional network reorganization during learning in a brain-computer interface paradigm”, Proceedings of the National Academy of Sciences, vol. 105, no. 49, pp. 19486-19491, 2008.

[11] R. Williams, ”Simple statistical gradient-following algorithms for connectionist reinforcement learning”, Machine Learning, vol. 8, no. 3-4, pp. 229-256, 1992.

[12] T. Miconi, Biologically plausible learning in recurrent neural networks for flexible decision tasks[J]. bioRxiv, 2016.

[13] E. Meyers, D. Freedman, G. Kreiman, E. Miller and T. Poggio, ”Dynamic Population Coding of Category Information in Inferior Temporal and Prefrontal Cortex”, Journal of Neurophysiology, vol. 100, no. 3, pp. 1407-1419, 2008.

[14] O. Barak, M. Tsodyks and R. Romo, ”Neuronal Population Coding of Parametric Working Memory”, Journal of Neuroscience, vol. 30, no. 28, pp. 9424-9430, 2010.

[15] D. Raposo, M. Kaufman and A. Churchland, ”A category-free neural population supports evolving demands during decision-making”, Nature Neuroscience, vol. 17, no. 12, pp. 1784-1792, 2014.

[16] M. Churchland, J. Cunningham, M. Kaufman, J. Foster, P. Nuyujukian, S. Ryu and K. Shenoy, ”Neural population dynamics during reaching”, Nature, 2012.

[17] D. Sussillo and L. Abbott, ”Generating Coherent Patterns of Activity from Chaotic Neural Networks”, Neuron, vol. 63, no. 4, pp. 544-557, 2009.

[18] D. Buonomano and W. Maass, ”State-dependent computations: spatiotemporal processing in cortical networks”, Nature Reviews Neuroscience, vol. 10, no. 2, pp. 113-125, 2009.

[19] D. Sussillo, M. Churchland, M. Kaufman and K. Shenoy, ”A neural network that finds a naturalistic solution for the production of muscle activity”, Nature Neuroscience, vol. 18, no. 7, pp. 1025-1033, 2015.

[20] R. Laje and D. Buonomano, ”Robust timing and motor patterns by taming chaos in recurrent neural networks”, Nature Neuroscience, vol. 16, no. 7, pp. 925-933, 2013.

[21] G. Hennequin, T. Vogels and W. Gerstner, ”Optimal Control of Transient Dynamics in Balanced Networks Supports Generation of Complex Movements”, Neuron, vol. 82, no. 6, pp. 1394-1406, 2014.

[22] K. Rajan, C. Harvey and D. Tank, ”Recurrent Network Models of Sequence Generation and Memory”, Neuron, vol. 90, no. 1, pp. 128-142, 2016.

[23] H. Song, G. Yang and X. Wang, ”Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks: A Simple and Flexible

Framework”, PLOS Computational Biology, vol. 12, no. 2, p. e1004792, 2016.

[24] P. J. Werbos, Backpropagation through time: what it does and how to do it, Proceedings of the IEEE, vol. 78, no. 10, pp. 1550C1560, 1990.

[25] A. F. Atiya and A. G. Parlos, New results on recurrent network training: unifying the algorithms and accelerating convergence, IEEE Transactions on Neural Networks, vol. 11, no. 3, pp. 697C709, May 2000.

[26] J. J. Steil, Backpropagation-decorrelation: online recurrent learning with O(N) complexity, 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541).

[27] H. Jaeger, Echo state network, Scholarpedia, vol. 2, no. 9, p. 2330, 2007.

[28] E. Izhikevich, ”Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling”, Cerebral Cortex, vol. 17, no. 10, pp. 2443-2452, 2007.

[29] I. Fiete and H. Seung, ”Gradient Learning in Spiking Neural Networks by Dynamic Perturbation of Conductances”, Physical Review Letters, vol. 97, no. 4, 2006.

[30] I. Fiete, M. Fee and H. Seung, ”Model of Birdsong Learning Based on Gradient Estimation by Dynamic Perturbation of Neural Conductances”, Journal of Neurophysiology, vol. 98, no. 4, pp. 2038-2057, 2007.

[31] R. Legenstein, S. Chase, A. Schwartz and W. Maass, ”A Reward-Modulated Hebbian Learning Rule Can Explain Experimentally Observed Network Reorganization in a Brain Control Task”, Journal of Neuroscience, vol. 30, no. 25, pp. 8400-8410, 2010.

[32] N. Fremaux, H. Sprekeler and W. Gerstner, ”Functional Requirements for Reward-Modulated Spike-Timing-Dependent Plasticity”, Journal of Neuroscience, vol. 30, no. 40, pp. 13326-13337, 2010.