LEARNING ARBITRARY FUNCTIONS WITH SPIKE-TIMING DEPENDENT PLASTICITY LEARNING RULE

Yefei Peng
Department of Information Science and
Telecommunications
University of Pittsburgh
Pittsburgh, PA 15260
ypeng@mail.sis.pitt.edu

Paul W. Munro
Department of Information Science and
Telecommunications
University of Pittsburgh
Pittsburgh, PA 15260
pmunro@mail.sis.pitt.edu

Abstract—**A neural network model based on spike-timing-dependent plasticity (STDP) learning rule, where afferent neurons will excite both the target neuron and interneurons that in turn project to the target neuron, is applied to the tasks of learning AND and XOR functions. Without inhibitory plasticity, the network can learn both AND and XOR functions. Introducing inhibitory plasticity can improve the performance of learning XOR function. Maintaining a training pattern set is a method to get feedback of network performance, and will always improve network performance.**

I. INTRODUCTION

The computational properties of a network of interconnected neurons can be investigated at either a neuronal level or at a system/network level. Barlow [1] made a compelling case for the former approach, citing the work of Hubel & Wiesel [2], in which the dependence of neuronal responses on features of environmental stimuli was demonstrated. Analyses of cortical neuronal receptive fields continue to dominate our understanding of cortical processing.

The response properties of a neuron are generally characterized in terms of some set of stimuli. Whether the stimuli in the set are patterns of afferent activity or are further removed from the neuron, such as visual patterns driving a cell in V1, the response of a neuron is a function of the stimuli. The computational nature of these functions remains an open question.

Theoretical accounts have assumed that cortical neurons develop response properties by virtue of a synaptic modification mechanism [3, 4]. Synaptic efficacies are generally assumed to change as a function of pre- and postsynaptic activities, as first hypothesized by Hebb [5]. Thus, neurons come to compute some function of their stimuli that are determined by some (unknown) statistical property of their patterned environment.

The discovery of long-term potentiation (LTP) by Bliss and Lømo [6] provided strong support for Hebb's hypothesis. Together with the functional inverse, long-term depression (LTD), these forms of longterm synaptic plasticity are regarded as endogenous mechanisms for changing synaptic weights within neural circuits [7, 8]. Typically, both LTP and LTD have been measured as a function of average firing rates of the pre- and/or postsynaptic neurons.

STDP is a phenomenon in which repetitive spike pairing on the order of milliseconds leads to LTP and LTD. STDP has been replicated in multiple laboratories working with neural tissues in culture [9], in slice [10-18], and in vivo [19-22]. Theoretical studies have indicated that STDP is a powerful learning rule supporting a range of computational functions [23-27].

In this paper, we build a feed forward neural network based on STDP learning rule, and train this network to compute arbitrary functions. In particular, we show that AND and XOR tasks can be learned by a network with inhibitory inter neurons.

II. NETWORK ARCHITECTURE AND TRAINING

The goal of the work presented in this paper is to study the rules that specify the activity-dependence of synaptic change in neural tissue, and how the computational properties of neurons depend on the statistics of the afferent activity patterns.

A. Network Architecture

The network architecture is shown in Figure 1. A postsynaptic output neuron R will be trained to respond to stimuli generated by a small number of extracellular stimulating electrodes, each driving multiple afferents to the target neuron, including interneurons within the cortical circuit. Activity and plasticity of these interneuron ‘hidden units’ is expected to play a major role in the proposed experiments [28-30]. In effect, the target neuron *R* computes a classification function; the stimulus is a member of the class if the neuron responds to it. Let *Si* be the activity level (injected current) of the *i*-th electrode, and let the binary vector **x**(*Si*) be the pattern of afferent spikes. Since increasing *Si* will increase the number of afferent spikes, the effective pattern to the neuron can be adjusted. Since the time interval of interest is very short, recurrent connections and time delays need not be included in the model. Afferents and interneurons are partially connected. Afferents and target neuron are partially connected. Interneurons and target neuron are fully connected. Only the weight connected to the target neuron could be changed. In our model, all neurons are integrate-and-fire neurons.

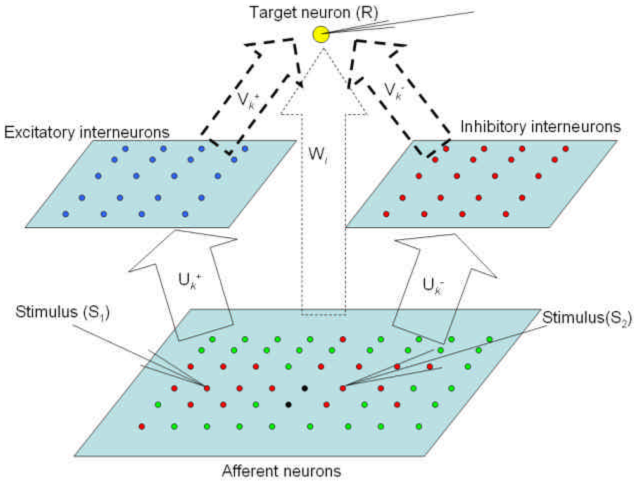

Figure 1. Network Architecture. W_j , U_k^+ and U_k^- are partial connections; V_k^+ and V_k^- are full connections. W_j , V_k^+ and V_k^- may be adjusted in training procedure.

B. Input Pattern

Input to the model is a vector corresponding to a set of afferents near the stimulating electrode(s). We assume an array of afferents, each generating a spike with a probability that is a function of distance from the electrode tip (Fig. 2). Each electrode induces action potentials with a probability that is near one within a small radius (shaded region). Outside this region, the probability decays to zero with distance.

Thus, the resulting input patterns tend to be noisy images of the pattern of electrode firing. The afferent stimuli will excite both the target neuron and interneurons that in turn project to the target neuron. The firing of the target neuron will depend upon the net input from the direct afferents as well as the interneurons. The parameters of the model will be fit according to experimental data. The simulations are thus an indirect approach to investigating the activity dependence of synapses throughout the network, excitatory as well as inhibitory.

In the simulations described in this paper, we also use a fixed coverage condition. That is, all afferents within some distance to the electrode tip will definitely fire; all other afferents will not fire. We call this fixed coverage condition. Similarly, we call the condition in which afferents fire with some probability unfixed coverage condition.

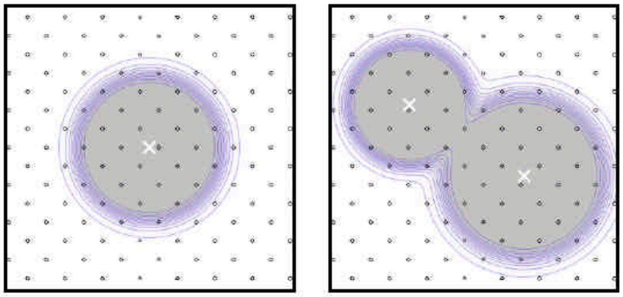

Figure 2. Stimulus patterns induced by electrodes. The simulated region is circular for a single electrode (left). Two or more electrodes may have overlapping regions of different sizes. Contours indicate probability of spike firing. Afferents (small dots) fire a spike with probability that is a function of distance from the electrode tip(s) (white Xs).

C. Training Method

In a manner derived from the conventional STDP induction procedure, we pair S_1 and S_2 with a postsynaptic spike evoked directly in cell *R* with depolarizing current through the whole-cell electrode (Fig. 3). Three sets of pairings will be performed and interleaved in a pseudo-random sequence. For AND function (study 1 of Figure 3), separate activation of S_1 and S_2 will be paired with a spike in *R* at a negative (post_pre) time interval of –5 ms (pairing one: R_S_1, pairing two: R_S_2), and S_1+S_2 will be paired with *R* at a positive (pre_post) interval of 5 ms (pairing three: S_1+S_2_R). For XOR function (study 2 of Figure 3), separate activation of S_1 and S_2 will be paired with a spike in *R* at a positive (pre_post) time interval of 5 ms (pairing one: S_1_R, pairing two: S_2_R), and S_1+S_2 will be paired with *R* at a negative (post_pre) interval of –5 ms (pairing three: R_S_1+S_2). The three training set is shown in Table 1.

The model is based on the assumption that plasticity will be induced on controlled pairs of individual pre and postsynaptic spikes operating at two fixed time delays, one for post following pre and one vice versa. The feed-forward processing of the model will be given by probabilistic feed-forward functions (*f*, *g*, and *h* in Table 2) and the plasticity will be computed according to functions corresponding to excitatory synapses (*F*) and inhibitory synapses (*G*). The parameters of the model will be tuned to test hypotheses about the form of the underlying plasticity functions, *F* and especially G (since little is known about the activity-dependence of inhibitory plasticity). If we look at the target neuron R, there are three inputs, weighted sum from all afferent neurons A_a, weighted sum from all excitatory interneurons A_e, and weighted sum from all inhibitory interneurons A_i. The output of target neuron R depends on these three inputs and the activity function h in Table 2. Generally, we use $A_{type}^{i,j}$ to represent all the inputs to the target neuron, where *type* can be *a* (afferent neurons), *e* (excitatory interneurons), or *i* (inhibitory interneurons), *i* and *j* can be 0 or 1, represent the electrode pattern. For example, $A_a^{0,1}$ represents the weighted sum from afferent neurons when input pattern is 0, 1 (only the 2nd electrode is active).

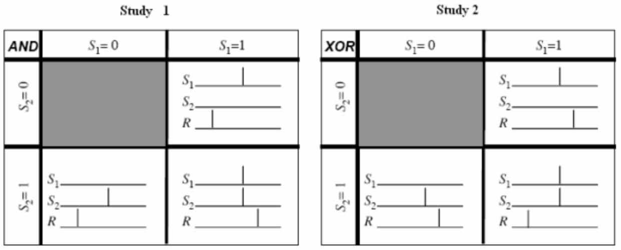

Figure 3. Training the network to learn the AND and XOR functions. Study 1, the AND function of two binary inputs S1 and S2. Study 2, the XOR function. Note that the training procedure for XOR is opposite that of AND, since the (0, 0) case is not included.

Table 1. Training set for AND and XOR functions

AND	Electrode 1	Electrode 2	Target
Pattern 1	0	1	0
Pattern 2	1	0	0
Pattern 3	1	1	1
XOR	Electrode 1	Electrode 2	Target
Pattern 1	0	1	1
Pattern 2	1	0	1
Pattern 3	1	1	0

In simulation, two types of learning rule are used. One is additive learning rule, the weight change is constant, and there is no limit on weight:

$$F(x_j, T) = \begin{cases} dw^+ \text{ if } T = +1 \\ dw^- \text{ if } T = -1 \end{cases} \tag{1.1}$$

where $dw^+ > 0$, $dw^- < 0$ .The other is multiplicative rule, the weight change is proportional to the difference between maximum weight and current weight if the weight will be increased, the weight change is proportional to the difference between minimum weight and current weight if the weight will be decreased:

$$F(x_j, T) = \begin{cases} k^+ * (W_{\max} - x) \text{ if } T = +1 \\ k^- * (W_{\min} - x) \text{ if } T = -1 \end{cases} \tag{1.2}$$

In both cases, the weight change is irrelevant to the network response r .

Since little is known about the activity-dependence of inhibitory plasticity, we propose a model for inhibitory plasticity. The weight change of inhibitory connection is just the opposite to that of excitatory connection. Corresponding to Eqn. (1.1), under additive learning rule, we have:

$$G(I_k, T) = \begin{cases} dw^- \text{ if } T = +1 \\ dw^+ \text{ if } T = -1 \end{cases} \tag{1.3}$$

Table 2. Model variables and dynamics

Connectivity(anatomy)
S_i = current strength of electrode i d_{ij} = distance from electrode i to afferent j U_{kj} = influence of afferent j on interneuron k W_j = influence of afferent j on target neuron V_k^+ = synaptic influence of excitatory interneuron k V_k^- = synaptic influence of inhibitory interneuron k
Processing and plasticity (physiology)
x_j = spiking of afferent j {0, 1} I_k = spiking of inhibitory interneuron k E_m = spiking of excitatory interneuron m r = "natural" spiking of target neuron T = forced spiking of target neuron
Processing
$x_j = \sum_i f(S_i, d_{ij})$ $I_k = g\left(\sum_j U_{kj} x_j\right); E_m = g\left(\sum_j U_{mj} x_j\right)$ $r = h\left(\sum_j W_j x_j, \sum_m V_m^+ E_m, \sum_k V_k^- I_k, \right)$ $T = \begin{cases} -1 \text{ if neuron is forced to spike early} \\ +1 \text{ if neuron is forced to spike late} \end{cases}$
Plasticity
$\Delta W_j = F(x_j, T)$ $\Delta V_k^+ = F(E_k, T)$ $\Delta V_k^- = G(I_k, T)$

Corresponding to Eqn. (1.2), under multiplicative learning rule, we have:

$$G(I_k, T) = \begin{cases} k^- * (W_{\min} - I_k) \text{ if } T = +1 \\ k^+ * (W_{\max} - I_k) \text{ if } T = -1 \end{cases} \tag{1.4}$$

There are two methods to choose the next pattern to feed to the network. The first one is randomly choosing a pattern from all three patterns. The second one is more complex. A training set is maintained, the next pattern is randomly chosen in the training set. At the beginning of the learning, it contains all patterns. After several epochs of training, there will be a testing period for all patterns. There will not be forced spikes in the target neuron in testing period; output of the target neuron will be recorded. If a pattern is computed correctly in ten trials (this number could be adjusted), then it will be removed from the training set. The training and testing will continue until maximum training epochs is reached or all patterns are learned.

The results presented in the remainder of the paper all pertain to networks with 625 afferents in 25× 25 grids, the positions of the two electrodes are (0.6, 0.6) and (0.4, 0.4), the stimulus radius is 0.2, the number of excitatory interneurons is 60, the threshold of the excitatory interneurons are randomly between 0 and 80, the number of inhibitory interneurons is 60, the threshold of the inhibitory interneurons are randomly between 0 and 80, the connection density between afferents and interneurons is 0.6, the connection density between afferents and target neuron is 0.2(there are cases that there is no connection between afferents and target neuron), the threshold of target neuron is 10.

III. RESULTS

Since little is known about the activity-dependence of inhibitory plasticity, at first, we suppose there is no inhibitory plasticity, which means, the weight between inhibitory interneurons and target neuron will not be changed during training period.

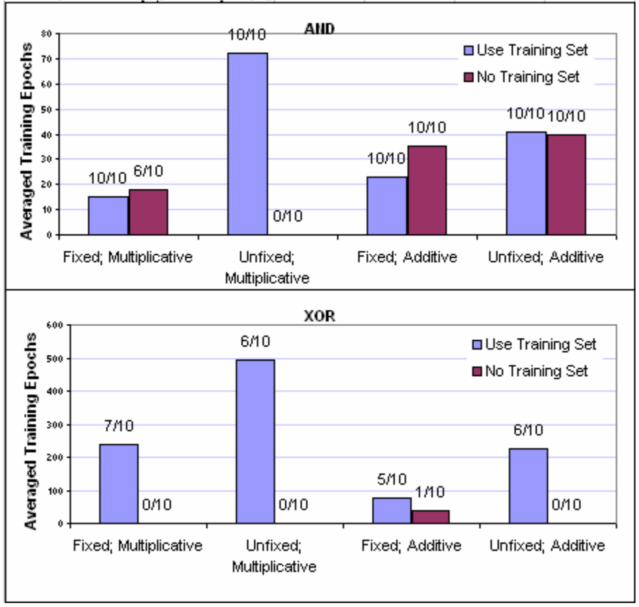

Figure 4. Training results in different conditions without inhibitory plasticity. Above: learning AND function. Below: learning XOR function. Meaning of the numbers: for example, " 6/10" means in 6 out of 10 trials the network learn the function successfully.

Figure 4 outlines the training results in different parameter combinations. The network can learn AND function in all conditions except the combination of multiplicative rule and no training set. The network can not learn XOR function under no training set condition, even using training set the network still can not learn XOR function very well.

Figure 5 shows how the average voltage of target neuron changes in training XOR function. The voltage is the weighted sum from afferents plus the weighted sum from excitatory interneurons minus weighted sum from inhibitory interneurons. If the voltage is larger then the threshold of target neuron, the output is 1; otherwise, the output is 0. Multiplicative learning rule is used, the electrode coverage is unfixed, and training set is used. At the beginning of training, the average voltage of pattern 3 is larger than that of pattern 1 and 2. Along with training, the average voltage difference between pattern 3 and pattern 1 or 2 is decreasing. At the final stage of the training, average voltage of pattern 3 is below the threshold, average voltage of pattern 1 and 2 are above the threshold.

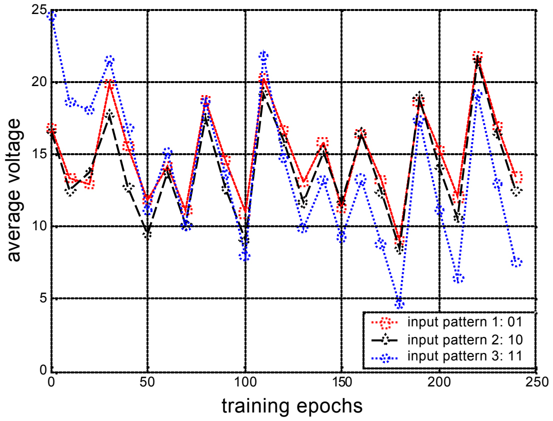

Figure 5. Average voltage of target neuron vs. training epochs in learning XOR function. Voltage of target neuron: $V^{i,j} = A_a^{i,j} + A_e^{i,j} - A_i^{i,j}$ (see Eqn.(1.5)). Threshold of target neuron is 10. The conditions are: use training set, multiplicative rule, and unfixed coverage.

Afferent neurons, excitatory interneurons, and inhibitory interneurons all have contributions to the voltage of target neuron. Their contributions before and after learning XOR function are shown in Figure 6 and 7 respectively. Before training, the voltage of target neuron is larger than the threshold for every pattern. After training, the voltage for pattern 1 and 2 are larger than threshold while that of pattern 3 is smaller than threshold. The network learned the XOR function. The contribution from afferent neurons doesn't change much after training. The contribution from inhibitory interneurons is almost unchanged after training since there is no inhibitory plasticity in this simulation; the only uncertainty is from randomly generated electrode coverage. But the contribution from excitatory interneurons changed a lot. Before training, $A_e^{1,1} \approx 2*A_e^{0,1}$, $A_e^{0,1} \approx A_e^{1,0}$, after training, $A_e^{1,1} \approx A_e^{0,1} \approx A_e^{1,0}$.

In the next simulation, we introduce inhibitory plasticity as Eqn. (1.3) and (1.4). Figure 8 outlines the training results in different parameter combinations. The network can learn AND function perfectly in all conditions except the combination of additive rule and no training set. The network can also learn XOR function when using training set.

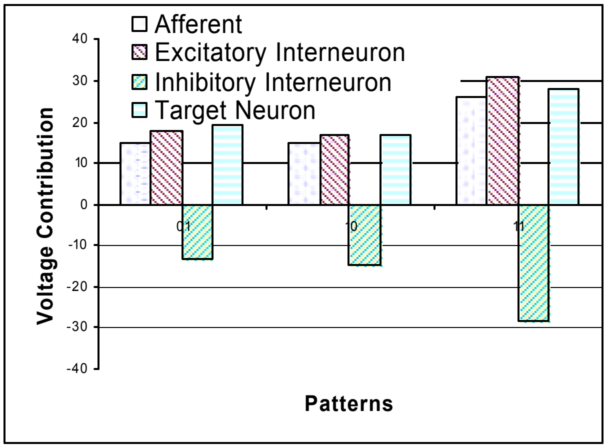

Figure 6. Contributions to target neuron voltage before learning XOR function for different input pattern. Threshold of target neuron is 10.

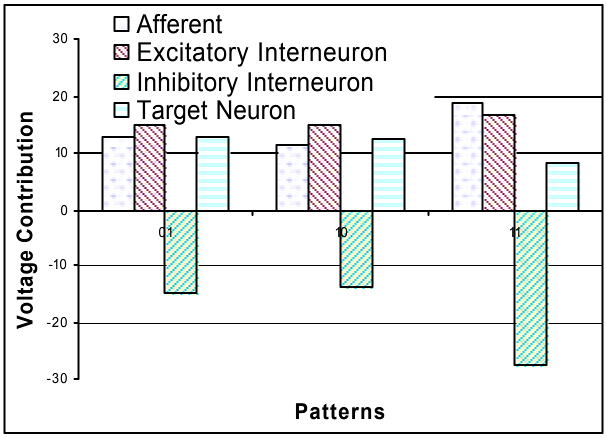

Figure 7. Contributions to target neuron voltage after learning XOR function for different input pattern. Threshold of target neuron is 10. The conditions are: using training set, multiplicative rule, and unfixed coverage.

IV. DISCUSSION

Since the target neuron is an integrate-and-fire neuron, we have:

$$r^{i,j} = \frac{sign(A_a^{i,j} + A_e^{i,j} - A_i^{i,j} - q_r) + 1}{2} \qquad (1.5)$$

To perform AND function, the following condition must be met:

$$\begin{aligned} A_a^{0,1} + A_e^{0,1} - A_i^{0,1} &< q_r \\ A_a^{1,0} + A_e^{1,0} - A_i^{1,0} &< q_r \qquad (1.6) \\ A_a^{1,1} + A_e^{1,1} - A_i^{1,1} &> q_r \end{aligned}$$

Eqn. (1.6) can be simplified to:

$$\Delta A_a^{j,k} + \Delta A_e^{j,k} > \Delta A_i^{j,k} \qquad (1.7)$$

$\Delta A_{type}^{j,k} = \Delta A_{type}^{1,1} - \Delta A_{type}^{j,k}$,$(type \in \{a, e, i\}; j, k \in \{0,1; 1,0\})$.

Similarly, to perform XOR function, the following condition must be met:

$$\Delta A_a^{j,k} + \Delta A_e^{j,k} < \Delta A_i^{j,k} \qquad (1.8)$$

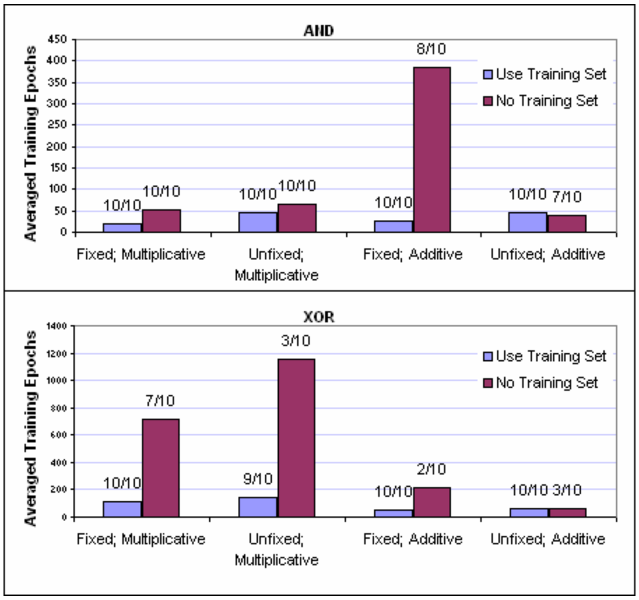

Figure 8. Training results in different conditions with inhibitory plasticity. Above: learning AND function. Below: learning XOR function. Meaning of the numbers: for example, " 8/10" means in 8 out of 10 trials the network learn the function successfully.

If there is no plasticity on the weights from inhibitory interneurons to target neuron, that means $\Delta A_i^{j,k}$ is fixed, no matter what value ΔA_i is, the network can still learn AND function if other weights to target neuron could be trained properly. Even if there is no inhibitory interneurons, AND function could still be learned. But for XOR function, if there is no plasticity on the weights from inhibitory interneurons to target neuron, $\Delta A_i^{j,k}$ must not be zero for the network to learn XOR function. If there is no inhibitory interneuron, that is $\Delta A_i^{j,k} = 0$, XOR function could not be learned.

As shown in Figure 5, before learning XOR function, $\Delta A_i^{0,1} = 15.07 < \Delta A_a^{0,1} + \Delta A_e^{0,1} = 10.98 + 13.07 = 24.05$, $\Delta A_i^{1,0} = 13.57 < \Delta A_a^{1,0} + \Delta A_e^{1,0} = 10.90 + 13.93 = 24.86$. As shown in Figure 6, after learning XOR function,

$$\Delta A_i^{0,1} = 12.55 > \Delta A_a^{0,1} + \Delta A_e^{0,1} = 6.31 + 1.71 = 8.02 ,$$

$$\Delta A_i^{1,0} = 13.55 > \Delta A_a^{1,0} + \Delta A_e^{1,0} = 7.52 + 2.02 = 9.54 .$$

Comparing Figure 4 and Figure 8 shows inhibitory plasticity improves the network performance of learning XOR functions.

Another conclusion we can make is using training set always improves network performance of learning AND and XOR functions. Using training set is a method to get feedback of network performance and refine training method by updating training set.

Our results show that a network based on STDP learning rule can learn AND and XOR functions even without inhibitory plasticity. In future, we expect some experiments on training cortical circuits could be done, so that the results from experiments and computer simulation could be combined.

REFERENCES

[1] H.B. Barlow, (1972). "Single units and sensation: A neuron doctrine for perceptual psychology," *Perception*, vol. 1, pp. 371-394, 1972.

[2] D.H. Hubel, and T.N. Wiesel, " Receptive fields, binocular interaction & functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106, 1962.

[3] E.L. Bienenstock, et al. "Theory for the Development of Neuron Selectivity: Orientation Specificity and Binocular Interaction in Visual Cortex," *J. Neurosci.*, vol. 2, pp. 32, 1982.

[4] S. Song, et al. " Competitive hebbian learning through spike-timing-dependent plasticity," *Nat. Neurosci.*, vol. 3, pp. 919, 2000.

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

[6] T.V.P. Bliss, and T. Lømo, " Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path," *J. Physiol.*, vol. **232**, pp. 331, 1973.

[7] W.B. Levy, and O. Steward, " Temporal contiguity requirements for long-term associative potentiation /depression in the hippocampus," *Neuroscience*, vol. **8**, pp.791, 1983.

[8] R.C. Malenka, and M.F. Bear, "LTP and LTD: An Embarrassment of Riches," *Neuron*, vol. 44, pp. 5, 2004.

[9] G.-Q. Bi., and M.-M. Poo, " Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, pp. 10464, 1998.

[10] J.C. Magee, and D. Johnston, " A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons," *Science*, vol. 275, pp. 209, 1997.

[11] D.E. Feldman, " Timing-based LTP and LTD at vertical input to layer II/III pyramidal cells in rat barrel cortex," *Neuron*, vol. **27,** pp. 45, 2000.

[12] V. Egger, et al., " Coincidence detection and changes of synaptic efficacy in spiny stellate neurons in rat barrel cortex," *Nat. Neurosci*, vol. 2, pp. 1098, 1999.

[13] D. Debanne, et al., "Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures," *J. Physiol*, vol. 507, pp. 237, 1998.

[14] P.J. Sjöström, et al., " Rate, timing and cooperativity jointly determine cortical synaptic plasticity," *Neuron*, vol. 32, pp. 1149, 2001.

[15] C.A. Boettiger, and A.J. Doupe, " Developmentally restricted synaptic plasticity in a songbird nucleus required for song learning," *Neuron*, vol. 31, pp. 809, 2001.

[16] R.C. Froemke, and Y. Dan, " Spike-timing-dependent synaptic modification induced by natural spike trains," *Nature*, vol. 416, pp. 433, 2002.

[17] M. Woodin, et al., " Coincident pre- and postsynaptic activity modifies GABAergic synapses by postsynaptic changes in CI transporter activity," *Neuron*, vol. 39, pp. 807, 2003.

[18] R.C. Froemke, et al., " Spike-timing-dependent plasticity depends on dendritic location," *Nature*, *in press*, 2005.

[19] L.I. Zhang, et al., " A critical window for cooperation and competition among developing retinotectal synapses," *Nature*, vol. 395, pp. 37, 1998.

[20] S. Schuett, et al., *Neuron*, vol. 32, pp. **3**25, 2001.

[21] H. Yao, and Y. Dan, " Stimulus timing-dependent plasticity in cortical processing of orientation," *Neuron*, vol. 32, pp. 315, 2001.

[22] A. Wolters, et al., " A temporally asymmetric Hebbian rule governing plasticity in the human motor cortex," *J. Neurophysiol*, vol. 89, pp. 2339, 2003.

[23] L.F. Abbott, and K.I. Blum, "Functional significance of long-term potentiation for sequence learning and prediction," *Cereb. Cortex*, vol. 6, pp. 406, 1996.

[24] P.D. Roberts, " Computational consequences of temporally asymmetric learning rules: I. Differential hebbian learning," *J. Comput. Neurosci.*, vol. **7**, pp. 235, 1999.

[25] R.P.N. Rao, and T.J. Sejnowski, 'Spike Timing Dependent Hebbian Plasticity as Temporal Difference Learning," *Neural Comput.*, vol. 10, pp. 2221, 2001.

[26] S. Song, et al., " Competitive hebbian learning through spike-timing-dependent plasticity," *Nat. Neurosci.***,** vol. 3, pp. 919, 2000.

[27] S. Song, and L.F. Abbott, " Cortical development and remapping through spike-timing dependent plasticity," *Neuron*, vol. 32, pp. 339, 2001.

[28] Y. Ikegaya, et al., " Synfire chains and cortical songs: temporal modules of cortical activity," *Science*, vol. 304, pp. 559, 2004.

[29] B.-Q. Mao, et al., " Dynamics of spontaneous activity in neocortical slices," *Neuron*, vol. 32, pp. 883, 2001.

[30] Y. Shu, et al., " Turning on and off recurrent balanced cortical activity," *Nature*, vol. 423, pp. 288, 2003.