Investigation of Unsupervised Learning in Spiking Neural Network using STDP Rules

Yanmin Ji
Department of Bioengineering
University of California, San Diego
La Jolla, CA 92093
y7ji@eng.ucsd.edu

Tianyi Lu
Department of Bioengineering
University of California, San Diego
La Jolla, CA, 92093
til061@eng.ucsd.edu

Abstract

Because of its energy efficiency and biological plausibility, spiking neural network is getting more and more attention recently, especially in the fields of neuromorphic engineering. Although several researchers have proved the feasibility of spiking neural networks in machine learning fields, we present a more biologically plausible network. In this project, we demonstrate the viability of Spiking-Time Dependent Plasticity of Spiking Neural Networks in unsupervised machine learning fields by digit recognition. With 100 excitatory neurons, we obtained an accuracy of 80%.

1 Introduction

It has long been a task facing researchers to construct computational programs that could achieve learning and pattern recognition like humans. Pattern recognition can be as simple as distinguishing handwritten digits and letters, or as difficult as distinguishing human voices. These topics are highlighted in machine learning. Researchers attempt to solve these problems with artificial neural networks (ANN). An artificial neural network can be simply described as a structure that resembles human brains with different layers of processing units that process input information and output towards the next layer. Previously these artificial neural networks are only similar to biological neural networks in structure but not in function because biological neural networks process information as spiking events in current, which is difficult to simulate without powerful computers. In recent years, along with fast development of supercomputers, researchers are able to construct artificial neural networks that are similar to biological neural networks not only in structure but also in function. Among these approaches, spiking neural network (SNN) is an increasing popular field. [1] SNN is different from older ANNs in that it takes time into account by processing inputs as spike trains, which provides SNN temporal resolution; SNN is also energy-conservative in that the neurons in SNN only fire when spike trains input into them and these units require no computational resources while there are no spiking events. [2]

One of the basic learning functions in human brains is spiking-timing-dependent plasticity (STDP). In STDP, a general rule is that the weight variables describing how information should be processed towards the next layer can be modified by the difference in time between the spiking events of the pre- and post- layers. Whether the time difference is positive or negative determines which function the weight change should follow. If input spiking event occurs immediately before the output spiking event, the weight of this processing unit will be made stronger; vice versa. Combining STDP and other network behaviors such as lateral inhibition turns out to be effective in digit recognition. [3]

A commonly used database for training artificial neural networks to recognize handwritten digits is MNIST. [4] This database consists of 60000 pictures for training and 10000 pictures for testing; each picture has 28 by 28 pixels with pixel intensities ranging from 0 to 255. In this project, the goal is to construct a program that can train under the training pictures and recognize digits from testing data. Since supervised learning is well investigated in SNNs, unsupervised learning is used in this project, which doesn't specify the categories of the patterns so that the patterns learned by the network are not necessarily ordered up with digits 0 to 9.

In the next section, the models used for neurons and synapses, as well as the network structure are introduced. Moreover, the detailed procedures of different processes are introduced in the next section. In section 3, we show our training results and some discussions on our network.

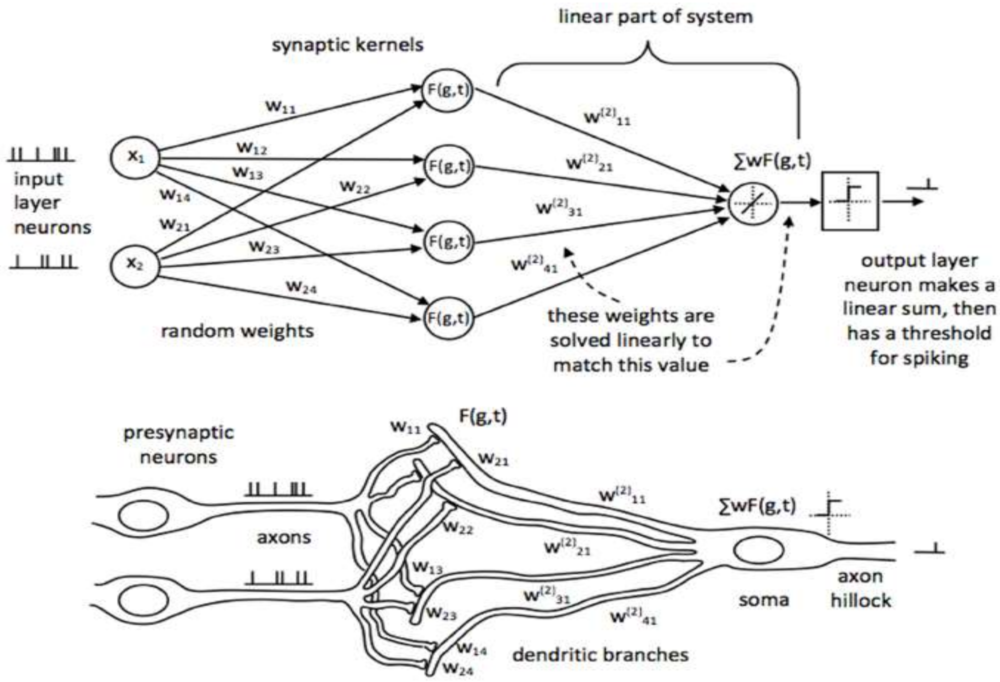

Figure 1: Data flow in SNN as spike trains. x1 and x2 are presynaptic neurons that output spike trains towards the synapses, whose functions are indicated by F (g, t). After processing the spike trains, sum of different synaptic outputs is inputted into soma, towards postsynaptic neurons.

2 Method

To simulate SNNs, Python-based simulator BRIAN2 is used to create the network.

2.1 Neuron and synapse model

To model individual neurons in the real world, we chose the Leaky Integrate-and-Fire (IAF) model. The reason why Hodgkin-Huxley is not chosen mainly because it is more complicated than the IAF model, and IAF model can already well describe the membrane potential of individual neurons. Specifically, the membrane voltage V is described as [5]

$$\frac{dV}{dt} = (-g_{leaky} * (V - E_{rest}) - (g_{excitatory} * V + g_{inhibitory} * (V - E_{reverse}) + I_{external})/C_{membrane} \quad (1)$$

where g_{leaky} is the leak conductance, E_{rest} is the resting potential, $g_{excitatory}$ is the excitatory conductance, $g_{inhibitory}$ is the inhibitory conductance, $E_{reverse}$ is the inhibitory reversal potential, $I_{external}$ is the external current, $C_{membrane}$ is the membrane capacitance. Also, the inhibitory and excitatory conductance are calculated as:

$$dg_{excitatory}/dt = -g_{excitatory}/\tau_{excitatory} \tag{2}$$

$$dg_{inhibitory}\ /dt = -g_{inhibitory}\ /\tau_{inhibitory} \tag{3}$$

where $\tau_{excitatory}$ is the excitatory synaptic time constant, and $\tau_{inhibitory}$ is the inhibitory synaptic time constant.

2.2 STDP model

In neuroscience, STDP is characterized as

$$W(\Delta t)\begin{cases} A_{pre}e^{-\Delta t/\tau_{pre}} \\ -A_{post}e^{-\Delta t/\tau_{post}} \end{cases} \tag{4}$$

However, in Brian2, a different function is employed to increase the efficiency of the program. a_{pre} and a_{post} are used as "traces" of pre- and post-synaptic activity, and can be described as:

$$\tau_{pre}\frac{d}{dt}a_{pre} = -a_{pre} \tag{5}$$

$$\tau_{post}\frac{d}{dt}a_{post} = -a_{post} \tag{6}$$

Therefore, when a presynaptic spike occurs:

$$a_{pre} = a_{pre} + A_{pre} \tag{7}$$

$$w = w + a_{post} * \gamma_{learning} \tag{8}$$

When a postsynaptic spike occurs:

$$a_{post} = a_{post} + A_{post} \tag{9}$$

$$w = w + a_{pre} * \gamma_{learning} \tag{10}$$

where $\gamma_{learning}$ is the learning rate, and w is the synaptic weight. The synaptic weight is also limited within the range between 0 and w_{max}, which is the maximum synaptic weight defined by users. By implementing and comparing with original STDP function in BRIAN2, this function is proved to have a similar effect but faster speed than the original STDP function.

2.3 Network structure

A three-layer spiking neural network as shown in Figure 2 is created for digit recognition task. The first layer is the photoreceptor layer, which has one-to-one connection with pixels and takes inputs from the intensity of each pixel from figures. Then, neurons from this layer fire with the Poisson pattern, the frequency of which is related to the intensity of the corresponding pixel. Since BRIAN2 keeps the neuron away from high-intensity firing, the highest frequency of Poisson stimulation is limited to be 64Hz, which is 1/4 of 255, the saturation intensity in MNIST datasets.

The second layer is a layer of excitatory neurons. The number of neurons is defined by users. However, the more neurons in this layer, the better prediction the final network will have. Each excitatory neuron receives spikes from all neurons from the first layer. The connections between the first two layers are determined by STDP rules. The initial synaptic weights are randomly assigned from 0 to 1 to ensure the difference between different neurons, otherwise all synapses will eventually have same weights.

The third layer is a layer of inhibitory neurons. The number of neurons in this layer is equal to the number of excitatory neurons in the second layer. The second layer sends excitatory signals to the third layer in a one-to-one connection. Then, each inhibitory neuron in the third layer inhibits all neurons in the second layer except the one it receives the signal from. This lateral inhibit can introduce competence between excitatory neurons in the second layer, and can, thus, increase the difference in synaptic weights.

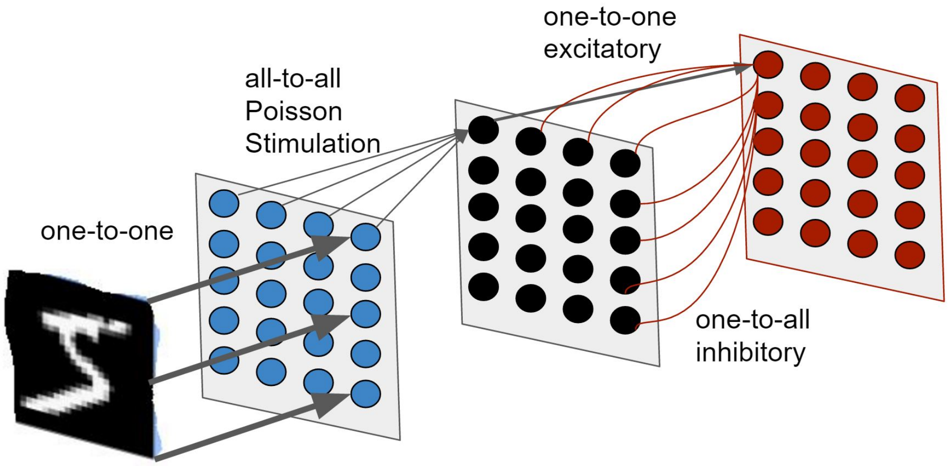

Figure 2: Network structure. The pixel intensity vector is used as the frequency for the first layer to generate Poisson spike trains. Each excitatory neurons in the second layer receives excitatory signals from all of neurons in the first layer, and the synaptic weights vary based on STDP rules. The second layer then send excitatory signals to the third layer of inhibitory neurons based on a one-to-one connection. The inhibitory neuron then send inhibitory signals in a one-to-all connection to the second layer.

2.4 Learning process

The learning process is an unsupervised learning process, in which the training group MNIST dataset is presented to the neural network in training process. After training process, neurons are labeled with different numbers. During test progress, test group of MNIST dataset and its label group are used to verify the prediction of the neural group.

2.4.1 Training process

During training process, the training group of MNIST dataset is shown to the neural network without labels. Each figure is presented to the network for 500ms. Prior to the whole process, the intensity of all pixels in each figure, which is a 28*28 matrix, is converted to a 1-D vector. Then, the pixel intensity of all figures is combined to a single vector with the length of 28*28*60000. The reason why intensity is converted in such way is because BRAIN2 does not allow the mixture of new network and the networks that are already simulated, meaning that between different figures, it is impossible to alter the previous Poisson stimulation with a new Poisson stimulation. By combining the intensity of all figures in a single vector, the program can pick intensity values of one figure every 500ms.

2.4.2 Labeling process

Since the training process is unsupervised, the cluster obtained from the training process needs to be labeled in order for human to understand. After training processes, a predefined vector of pixel intensities of number 0 to 9 in order is input to the network with zero learning rate. During this process, the firing activity of the network is recorded. Based on to which number it has the highest firing frequency in response, each neuron is assigned a number. The labeling results is saved for the future testing process.

2.4.3 Testing process

After training and labeling process, now the neural network is ready to be tested by the testing group of the MNIST dataset. All figures in testing group are presented to the neural network for 500ms with zero learning rate. Suring testing process, a similar intensity vector in training process is constructed for the testing group. For each figure, each neuron fires and vote once for its label. For example, if a neuron labeled “1” fires once, then label “1” is voted once for this figure. The total vote number is counted after each figure in order to determine the predicted label for the figure. The result is saved to determine the performance of the neural network.

2.5 Model parameters

To have a biologically plausible system, all parameters used in the system are as close to the real world as possible, as shown in Table 1. The external current is chosen to have neurons right below the threshold when they are not triggered by the Poisson stimulation.

Table 1. Parameters used for the model

Parameters	**Value**
g_{leaky}	10.0 nS
E_{rest}	-60 mV
$E_{reverse}$	-80 mV
$I_{external}$	200 pA
$C_{membrane}$	200 pF
$\tau_{excitatory}$	5 ms
$\tau_{inhibitory}$	10 ms

3 Results and Discussions

Due to the limit of PC resources, we only have the chance to test the training of 100 excitatory neurons with the training group of the MNIST datasets. As a demonstration, the synaptic weights of neuron 9 before and after the training process are shown in Figure 3. The weight vector with the length of 784 is converted back to 28*28 matrix for graphical presentation. From the comparison, it is clear that after the training process, each neuron can distinguish a certain type of patterns from the training figures. Neuron 9, for example, shows that the regions that is similar to number “3” have higher weights than other regions. Therefore, it is very possible that every time number 3 is presented to the network, neuron 9 will fire actively. In fact, neuron 9 is labeled as “3” in our labeling process. In terms of accuracy, with 100 excitatory neurons, the network can perform digit recognition with an accuracy of about 80%.

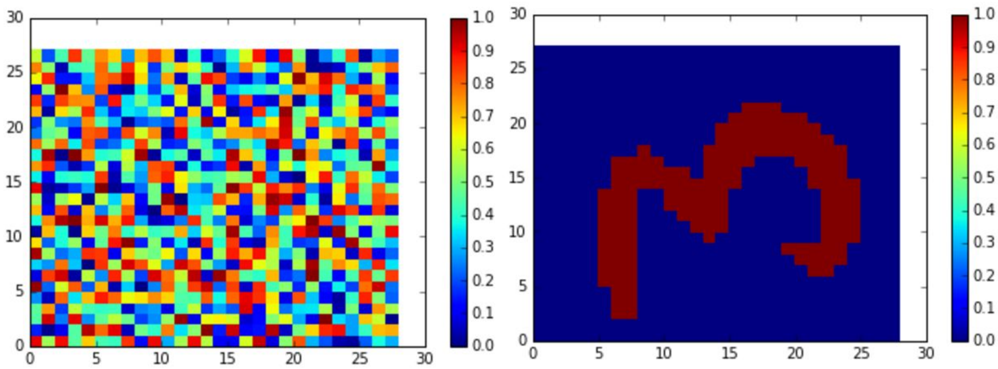

Figure 2: Left: synaptic weights between the photoreceptor layer and the neuron 9 before training process are assigned as random numbers. Right: synaptic weights between the photoreceptor layer and neuron 9 after training with the training group once for 500ms each figure. Values are reconstructed from a 784-length vector to a 28*28 matrix.

This value falls in the expected range since according to [3], the accuracy of neural networks with 100 excitatory neurons is approximated as 83%. One possible reason why the accuracy we obtained is lower is because the parameters we use in the model is more biologically plausible than others. Although the values we use can well simulate the real-world circumstances, these parameters are not optimized for machine learning, and, thus, is possible to be harder to converge than the parameters modified for machine learning purpose.

Several researchers [3, 6, 7] have used a similar network structure as ours. The main difference between their networks and ours is that our parameters are more biologically plausible, so ours is more meaningful for clinical or research uses. Especially, [3] designed a similar spiking neural network and performed digit recognition using STDP rules too. Although they used a biological neural model, our model further extends their neuron model to include leaky terms and membrane conductance. Therefore, our design is one of the models that can well simulate biological neural systems among others' work. Moreover, another advantage of our network is that the number of inhibitory neurons is same as the number of excitatory neurons. Therefore, when one neuron is actively firing, it not only can increase its weights through STPD rules, but also can inhibit other excitatory neurons through inhibitory neurons. This mechanism introduces the competition between excitatory neurons in the second layer.

In the future, it would be interesting to further investigate the scalability of our neural network by employing more excitatory neurons in the second layer. From [3], the accuracy is positively correlated to the number of excitatory neurons. Therefore, we can expect to have a much more accurate prediction with more excitatory neurons in our network. In addition, an increase in the number of layers may also hugely increase the accuracy. Also, a mask layer may also be useful to further increase the efficiency or the convergence of the network. Another meaningful improvement would be introducing the stochastic voting mechanism. In our current network, when the neuron fires, it will vote for its labelled number once. However, it is possible that a neuron actively responds to multiple numbers. Therefore, a stochastic mechanism can be used to further increase the accuracy.

In conclusion, while well simulate the biological neural network, our artificial spiking neural network can perform digit recognition using STDP rules with a good accuracy. With the help of super computer, it is expected that such a neural network can have state-of-the-art performance.

Acknowledgments

We would like to thank our instructor Dr. Gert Cauwenberghs, our TAs, Aashish Patel and Jun Wang, for their help throughout the course and the project.

References

[1] Lee, K. & Kwon, D.C. (2008) Synaptic plasticity model of a spiking neural network for reinforcement learning. *Neurocomputing* 71.13: 3037-3043.

[2] Maass, W. (1997) Networks of spiking neurons: the third generation of neural network models. *Neural networks* 10.9: 1659-1671.

[3] Diehl, P. U. & Cook M. (2015) Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience* 9.

[4] LeCun, Y., Cortes, C. & Burges, C.J.C. THE MNIST DATABASE of handwritten digits. http://yann.lecun.com/exdb/mnist/

[5] Gerstner, W. & Kistler, W.M. (2002) *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press.

[6] Querlioz, D., Bichler, O. & Gamrat, C. (2011) Simulation of a memristor-based spiking neural network immune to device variations. *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE.

[7] Querlioz, D., Bichler, O., Dollfus, P. & Gamrat, C. (2013) Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Transactions on Nanotechnology* 12, no. 3: 288-295.