Fuzzy Based Modified SHL algorithm for Spiking Neural Networks

Haider Raza
Dept of Computer Science,
School of Mathematics and Computer Science,
Dilla University, Ethiopia.

Gashaw Tsegaye
Dept of Computer Science,
School of Mathematics and Computer Science,
Dilla University, Ethiopia.

Ranjit Biswas
Dept of Computer Science,
Jamia Hamdard University,
New Delhi, India.

ABSTRACT
Spiking neural network is the 3rd generation neural network. In this paper, we derive spiking neural network's topology and the fuzzy reasoning by restricting to the usage of biological components. Input encodes information in the timing of spike train. Fuzzy reasoning is used on biological components such as dynamic synapse, receptive field, inhibitory and excitatory neurons. The enrichment of the flow of information is done by dynamic synapse and the neuron selection by using receptive field. Modeling the dynamics of the limited synaptic resources makes neurons selective to particular spike frequencies. The receptive field behaves like fuzzy membership function which enables the individual neuron respond at certain spike train frequency. The network is supervised and learning occurs at the output layer of the network. Various issues arise while learning with supervised method takes place, namely convergence and continuous updating of weights after the goal is achieved. These issues are discussed in detail and are resolved. The modified SHL algorithm is used for learning. The classification problem of XOR is solved. The implementation is done on MATLAB.

General Terms
Spiking Neural Networks

Keywords
Spiking Neural Networks, Fuzzy, Dynamic Synapse, Receptive field, Classification.

1. INTRODUCTION
The recent development in spiking neural networks (SNNs) and computational neuroscience resembles the current understanding of neural mechanism within the human brain. From the history of neural network a common belief was that essential information in neurons is encoded by firing rate or rate codes. The new era of computational neuroscience shows spatio-temporal distribution of spikes in biological neurons. The third generation of neuron modeling (spiking neurons) is based on realization that the precise mechanism by which biological neuron encodes information is poorly understood. The models of spiking neurons are both computationally efficient and biologically accurate [1]. The two important roles of biological neurons in the flow of information within the neural circuit are excitatory and inhibitory. The excitatory neurons are responsible for routing information through the network and inhibitory neurons are for regulating the activity of excitatory neurons. There are more excitatory neurons than inhibitory neurons [2]. The synaptic transmission is unreliable and involves lot of uncertainty [3]. Due to this uncertainty it is difficult to say which biological feature improves the

computational capability in neural dynamics. The learning in neuro-computing draws its inspiration from behavior of human learning. Where the human expertise is implicit Fuzzy IF-THEN reasoning provides a language to deal [4] modeling biologically plausible SNNs presents a significant challenge given a vast scale of real networks. It is generally recognized that SNNs are capable of exploiting time as a resource for coding and computation in a more sophisticated manner than a typical neural computational model [5] [6] [7]. In Section 2 unsupervised and supervised learning methods, dynamic synapses and receptive fields are reviewed. Section 3 shows fuzzy reasoning and provides a basis for structuring the network topology.

2. REVIEW
Activity-dependent modification of synapses is a powerful mechanism for shaping and modifying the response properties of neurons, but it is also dangerous. Hebbian plasticity, in the form of long-term potentiation (LTP) and depression (LTD), provides the basis for most models of learning and memory, as well as the development of response selectivity and cortical maps. Hebbian learning is local modification of synaptic modification but it suffers from global stability. [8]

2.1 Forms of Learning
There are two types of learning algorithm that can be used for LTP or LTD for synaptic weights [9] which compare the interconnected pre and post-synaptic firing rates to a threshold to decide whether to stimulate LTD or LTP. The two types of learning are supervised learning and unsupervised learning. Bienenstock, Cooper and Munro's Model (BCM) also shows the lack of biological basis. Spike-Time-Dependent-Plasticity (STDP) is a non-Hebbian form of plasticity because it acts across many synapses and seems to depend primarily on the post synaptic firing rate rather than on correlation between pre and post-synaptic activity [8]. Hebbian plasticity can also be used to regulate the

overall synaptic activity but the strong balance is required between LTP and LDT. STDP is the solution to this problem of pure Hebbian form. Presynaptic firing that proceeds postsynaptic or depolarization can induce LTP, whereas reversing this temporal order causes LTD. [10] [11] [12] STDP and BCM are under the category of unsupervised learning algorithm and they do not lead to required specific goal. Under supervised learning schemes there are various categories of coding schemes for different algorithms such as Time-to-First Spike, Precise spike-Timing and Relative Spike Time. Time-to-First spike suffers from a problem that they cannot be used to learn the sequence of multiple spikes. There are many algorithms for supervised learning such as SpikeProp, Statistical Approach, Linear algebra method, Evolutionary strategies, Supervised Hebbian Learning (SHL) and Remote Supervision Method (ReSuMe). Each of the algorithms has its own advantages and disadvantages.

2.1.1 *Supervised Learning*

There are several methodologies for implementing supervised learning in SNNs. The following subsections illustrate the diversity of the different approaches.

2.1.1.1 *SpikeProp*

SpikeProp is a gradient descent training algorithm for SNNs that is based on backpropagation. The discontinuous nature of spiking neurons causes problems with gradient descent algorithms, but SpikeProp overcomes this issue by only allowing each neuron to fire once and by training the neurons to fire at a desired time. The algorithm can only be used in a time-to-first-spike coding scheme which means that it cannot learn patterns consisting of multiple spikes. [13]. The target of SpikeProp is to learn a set of the desired firing times, denoted t_j^d, at a neuron j for a given set of input patterns $S^{in}(t)$. The SpikeProp algorithm has been derived for the neurons modeled by the Spike Response Model. In this model the membrane potential of neuron j can be defined as:

$$U_{mj}(t) = \sum_{i \epsilon N_j^{pre}} \sum_k w_{ij}^k \, \varepsilon(t - t_i^o - d_{ij}^k)$$

The set N_j^{pre} represents all pre-synaptic neurons of the neuron j. The term w_{ij}^k is the weight of a synaptic terminal k of the connection between neuron i and j. It is assumed that ε(t) =t/**τ**exp (1-t/**τ**), with some time constant. The parameter t_i^o is the firing time of the neuron i, and d_{ij}^k is the delay of synaptic terminal.

2.1.1.2 *Evolutionary Learning*

Evolutionary strategies (ES) have been applied as a form of supervision for SNNs. ES differs from GAs in that they rely solely on the mutation operator. The accuracy of the resulting SNN provides the basis for determining the fitness function and the ES population was shown to produce convergence to an optimal solution. A limitation of this approach is that only the Time-to-First spike is considered by the ES, this learning process is very time-consuming and this renders them unsuitable for online learning [14].

2.1.1.3 *Supervised Hebbian Learning*

Supervised Hebbian Learning (SHL) [15] is arguably the most biologically plausible supervised SNN learning algorithm [16] .SHL simply seeks to ensure that an output neuron fires at the desired time, with the inclusion of a 'teaching' signal. Since the teaching signal comprises of intracellular synaptic currents, supervision may be envisioned as supervision by other neurons. SHL does suffer from the limitation that even after the goal firing pattern has been reached; SHL continues to change the weights. Thus, constraints must be added to the learning rule to ensure stability. However, the problem with setting constraints is that it is not easy to know at which point in the training they should be applied. The weights will continue to increase after each training epoch and eventually could cause the network to be unstable, or at least to generalize poorly in the testing phase of learning [16]. Ruf and Schmitt proposed one of the first spike-based methods similar to SHL approach. In the first attempt, they defined the learning rule for the monosynaptic excitation. The learning process was based on three spikes (two pre-synaptic and one post-synaptic) generated during each learning cycle. The first Presynaptic spike at the time t_1^{in} was considered as in input signal, whereas the second Presynaptic spike at $t_2^{in} = t^d$ pointed to the target firing time for the postsynaptic neuron. The learning rule is

$$\Delta w = \eta(t^{out} - t^d)$$

Where η>0 is the learning rate and t^{out} is the actual time of the postsynaptic spike.

2.1.1.4 *Remote Supervision Method (ReSuMe)*

The Remote Supervision Method (ReSuMe) is closely related to SHL but manages to avoid its drawbacks [16]. The 'remote' aspect comes from the fact that teaching signals are not delivered as currents to the learning neuron (as with SHL). Instead a teaching signal and STDP-like Hebbian correlation are employed to co-determine the changes in synaptic efficacy. In ReSuMe, the synaptic weights are modified according to the following equation:

$$\frac{d}{dt} w(t) = [S^d(t) - S^l(t)]\left[a + \int_0^\infty W(s) S^{in}(t - s) ds\right],$$

Where $S^t(t)$,$S^{in}(t)$ and S^l are the target, pre and post synaptic trains respectively. The parameter a expresses the amplitude of the non-correlation contribution to the total weight change while the convolution function represents the Hebbian-like modification of w. The high learning ability of ReSuMe has been confirmed through extensive simulation experiments **[16].**

2.1.2 *Unsupervised Learning*

STDP and BCM are under the category of unsupervised learning algorithm and they do not lead to required specific goal

2.1.2.1 *Spike Time Dependent Plasticity (STDP)*

The STDP learning rule dictates that long-term strengthening of the synaptic efficacy occurs when a pre-synaptic spike action potential precedes a post-synaptic one and this is called potentiation. Synaptic weakening occurs with the reverse temporal order of pre and postsynaptic spikes and this is called depression. The stability of STDP can be ensured by placing limits in the strengths of individual synapses and a multiplicative form of the rule introduces an adaptive aspect to learning, resulting in progressively smaller weight updates as learning progresses.

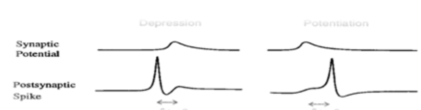

Fig: 1

2.2 Dynamic Synapse and Receptive Field

Synaptic efficacy changes on very short-time scales as well as over the longer time scale of training. The rate at which the synaptic efficacy changes, is determined by the supply of neurotransmitter and the number of receptor sites. Modeling the dynamics of the limited synaptic resources makes neurons selective to particular spike frequencies. The filtering effects of the dynamic synapses occur because there is a frequency of pre-synaptic spike train that optimizes the post-synaptic output [17]. It is particularly difficult to refrain dynamic synapses models to operate at specific frequency bands by changing the various model parameters. One way to guarantee that synapses are reactive to certain frequencies is with the use of Receptive Fields (RF). Experiments with retinal ganglion cells in the frog showed that the cell's response to a spot of light grew as the spot grew until some threshold had been reached. The part of the visual world that can influence the firing of a neuron is referred to as the RF of the neuron [18]. The implications for SNNs are that RFs can be used in conjunction with neuron models to promote feature selectivity and hence enhance the 'richness' of information flow.

3. FSNNs TOPOLOGY

Biological neuron dynamics are determined by the relationships between a sequence of action potential, synaptic resources, post-synaptic currents and membrane potentials. Neuron selectivity and information flow can be further strengthened using RFs [21]. Biological neurons show a form of human reasoning. Human reasoning is fuzzy in nature and involves much of uncertainty which involves a much higher level of knowledge representation [4]. Fuzzy rules may be expressed in terms such as

$$``IF\ ((A_1\ is\ \boldsymbol{X}_1) \wedge (A_2\ is\ \boldsymbol{X}_2) \wedge \ldots \wedge (A_N\ is\ \boldsymbol{X}_N), THEN(B\ is\ \boldsymbol{Z})''$$

Where "A" and "B" are both imprecisely (fuzzily) defined quantities, and "$\boldsymbol{X}$" and "$\boldsymbol{Z}$" are both fuzzy terms. Fuzzy logic, with fuzzy rules, has the potential to add human-like subjective reasoning capabilities to machine intelligences, which are usually based on bivalent Boolean logic. In this paper we use Fuzzy Logic to dictate the distribution of various biological plausible computational elements in SNNs. The Fuzzy IF-THEN rules are of the form:

$$``IF\ ((x_1\ is\ A_1) \wedge (x_2\ is\ A_2) \wedge \ldots \wedge (x_N\ is\ A_N), THEN(y\ is\ \boldsymbol{B})''$$

Where x_1 to x_N represent the network inputs, A_1 to A_N represent hidden layer RFs aSnd y is the network output.

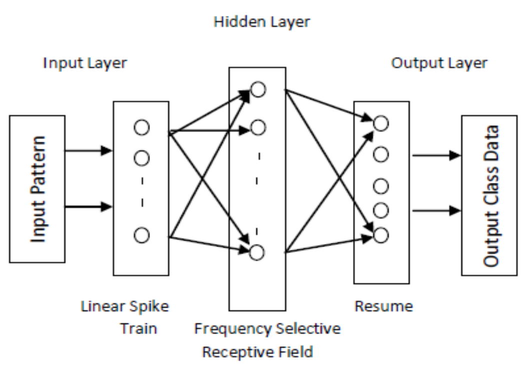

Fig: 2

Input Layer:
The input neuron is responsible for simply encoding the feature data into an appropriate frequency range. The spike trains are generated by linear encoding scheme. The encoding scheme takes the frequency data points and converts them into an inter-spike-interval (ISI) which is used to create input spike train. Each data point scaled into a particular frequency range in linearly scaled into an input spike train of a particular sample length.

Hidden Layer:
Gaussian RFs are placed at every synapse between the input and hidden neurons. The job of RFs is to determine the relation between the input frequencies f_i and the central operating frequency F_o of RF. The weight is then scaled by k_i. The process relates to the IF ($x_i\ is\ A_i$) part of fuzzy rule, where x_i is input and A_i represent the RF. By the use of RFs, proper tuning of dynamic synapse is not required. The function of each hidden layer neuron is to impose the remaining part of antecedent fuzzy IF-THEN rule, the conjunctive 'AND', now summing the PSP by performing the disjunctive 'OR'. The main function of RFs connecting to hidden layer neuron is to filter the spikes to the output layer.

Output Layer:
The action potential with the synapse is only significantly high in magnitude for a very short interval of time. This type of synapse has been considered as coincidence detector. It is then the task of supervised learning algorithm to associate the hidden layer neurons to the output layer neurons, thus performing fuzzy reasoning between the hidden layer (antecedents), and the output layer (consequents).

4. RELATED WORK

The XOR problem first has to be put into terms that it is understood by an SNN that transmits information in the form of spike trains. There are various ways in which the XOR problem may be encoded in terms of spikes. In a time-to-first-spike encoding scheme (Bohte et al., 2002), the XOR data can be translated into delays. However, although time-to-first-spike encoding schemes are successful at solving XOR but they lack biological plausibility. An alternative is to simply encode the 1s and -1s of the XOR truth table into two spike train frequencies. The frequencies used in this example are 60 and 100 (i.e. two numbers that will produce multiple spikes in a sample that are sufficiently different).

x_1	x_2	$Class$
60	60	-1
60	100	1
100	60	1
100	100	-1

x_1	x_2	d
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1

Fig.3

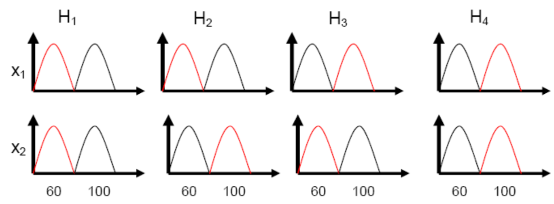

Fig. 4

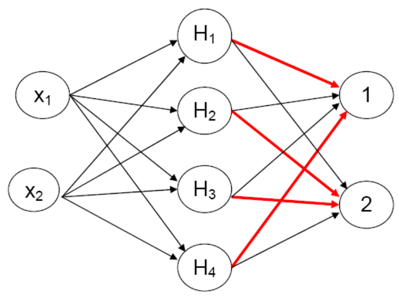

Fig. 5

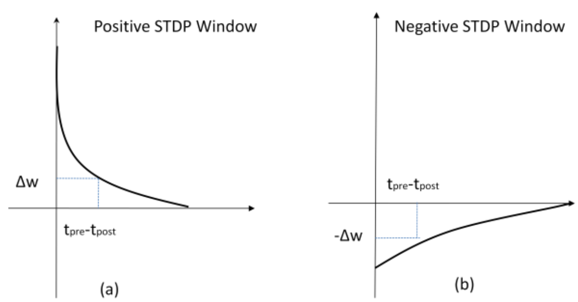

Fig.6

From Fig.4, it is an RBF like feed forward-network topology with two input neurons in input layer, four neurons are in hidden layer and two output neurons in output layer. From the example it is clear that the $H_1 and\ H_4$ (activated when both the input variable are same) are associated with the output neuron of class 1. The linear encoding presented here this is a trivial matter, since any two successive spikes can be used calculate the ISI and hence determine the frequency. The RFs are Gaussians, and whenever an input frequency falls within the non-zero part of the Gaussian, the weight between the input and the hidden layer node is scaled accordingly. In this way, the neurons in the hidden layer are frequency-selective. Fig.5 shows the Gaussians used in the hidden layer nodes. Without the Gaussian RFs the output from all the hidden layer neurons would be identical. With the RFs each hidden layer node responds to three of the four samples from the truth table. In fact, the hidden layer nodes only ignore an input data sample when both the $x_1 and\ x_2$ components of the sample lie outside the Gaussian RFs of the relevant hidden layer synapses. With standard RBF network the learning only occurs at the output layer of the network. The SHL algorithm is used, a supervisory spike train is delivered to the output neurons at the desired firing times. As with the SHL scheme STDP is used to modify the weights **[19]**. Unfortunately, with this approach, SHL has no way to 'prop up' the weights **[16]** once neurons have stopped firing.

STDP learning window **[8]** is used to modify the synaptic efficacy between the hidden and output weights. This modified form of SHL does not require an actual supervisory spike train as with the **[19]** approach. Fig. 5 illustrates the calculation of the weight updates for this modified form of SHL. By using this STDP window the weights are updated instead of supervisory spike. The modified SHL scheme controls the interaction of the STDP window, by allowing the positive weight updates when an output neuron is producing output spike at the desired output time Fig 6(a). Similarly when an output neuron is producing spikes at an undesired time, the negative of the STDP window is used to decrease the weight. In this way, the modified SHL implements STDP and anti STDP in order to relate the hidden nodes to the correct class output neurons. The output neuron are now trained since spike activity at the output only occur at the desired times. But the issue with SHL is that it continues to change the weight even after the goal has been reached. A more desirable function of the hidden layer would be to only process input data when both RFs are activated by the input data variable $x_1 and\ x_2$. Bu utilizing a product or conjunctive AND, a hidden layer neuron would only process a particular input data when both $x_1 and\ x_2$ components of the data point activate the positive part of their respective RFs. This is how the fuzzy rule base in SNN topology is implemented firstly. It is noticed by neurobiologist, the desired gating behavior in the experimental studies on RF dynamics [20]. In the presence of synchrony between inhibitory inputs, it is possible to implement an AND gate in hidden layer nodes. For the system containing more than two inputs the magnitude of the inhibitory part of the RF will need to be large since it will need to be large enough to cancel the sum of all other possible excitatory response. Implementing the inhibitory parts of the RFs results in a much clear cut classification of the input data by hidden layer neurons. This is due to the crisp classification of the hidden layer. Since each hidden node is crisply assigned to particular class, there is no negative weight updates for the output layer synapses. These weights receive far more positive weight increases than before and as such the output spike frequency arriving from these synapses is much higher than before.

5. CONCLUSION AND FUTURE WORK

We have solved the problem of dynamically optimizing network coverage and backbone connectivity in DWB-based wireless networks using genetic algorithm. This joint coverage coverage-connectivity optimization problem is a quadratic minimization problem which is a quadratic cost function for both coverage and connectivity in term of the square distance between neighbouring nodes. We developed a completely distributed mobility control algorithm based on genetic algorithm that computes the relocation direction of nodes on the basis of current control location. The simulation produces the encouraging results for the cost based control algorithm that optimizes both network coverage and backbone connectivity.

We proposed to use physical link cost models in coverage and connectivity optimization and perform quantitative analysis in terms of energy usage and bit-error rate. Further we would apply other methods of solution for optimization problem that minimize the cost of coverage and connectivity and improve energy consumption, throughput and end to end delay.

6. REFERENCES

[1] Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons. IEEE Trans. Neural Networks, 15(5), 1063-1070.

[2] Braitenberg, V; Schuz, A: .“Cortex-statistics and geometry of neuronal connectivity,.” Springer, 1998.

[3] Allen, C; Stevens, CF: .“An evaluation of causes for unreliability of synaptic transmission,.” Proceedings of the National Academy of Sciences, vol. 91, no. 22, pp. 10380-10383, 1994.

[4] Zadeh, L.A. (1965). Fuzzy sets. Information and Control, 8(3), 338—353.

[5] Gerstner W. and Kistler W. (2002a): Spiking Neuron Models. Single Neurons, Populations, Plasticity. — Cambridge: Cambridge University Press.

[6] Maass W. (1999): Paradigms for computing with spiking neurons, In: Models of Neural Networks, (L. van Hemmen, Ed.).—Berlin: Springer.

[7] Maass W. (2003): Computation with spiking neurons, In: The Handbook of Brain Theory and Neural Networks, 2nd Ed., (M. Arbib, Ed.). — MIT Press, Cambridge, pp. 1080–1083.

[8] Abbott, L. F., Nelson, S. B. (2000). Synaptic plasticity: taming the beast. Nature Neuroscience, 2, 1178-1183.

[9] Bienenstock, E. L., Cooper, L. N., Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. Journal Neuroscience, 2, 32-48.

[10] Levy, W. B. & Steward, D. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. Neuroscience 8, 791–797 (1983).

[11] Gustafsson, B., Wigstrom, H., Abraham, W. C. & Huang, Y.-Y. Long-term potentiation in the hippocampus using depolarizing current pulses as the conditioning stimulus to single volley synaptic potentials. J. Neurosci. 7, 774–780 (1987).

[12] Debanne, D., Gahwiler, B. H. & Thompson, S. M. Asynchronous pre- and postsynaptic activity induces associative long-term depression in area CA1 of the rat hippocampus in vitro. Proc. Natl. Acad. Sci. USA 91, 1148–1152 (1994).

[13] Bohte, S. M., Kok, J. N., La Poutré, H. (2002). Errorbackpropagation in Temporally Encoded Networks of Spiking Neurons. Neurocomputing, 48, 17-37.

[14] Belatreche, A., Maguire, L. P., McGinnity, T. M., Wu, Q. X. (2003). A Method for Supervised Training of Spiking Neural Networks. IEEE Cybernetics Intelligence – Challenges and Advances (CICA), 39-44.

[15] Ruf, B., Schmitt, M. (1997). Learning temporally encoded patterns in networks of spiking neurons. Neural Processing Letters, 5(1), 9-18.

[16] Kasiński, A., Ponulak, F. (2005). Comparison of Supervised Learning Methods for Spike Time Coding in Spiking Neural Networks.

[17] Natschläger, T., Maass, W., Zador, A. (2001). Efficient temporal processing with biologically realistic dynamic synapses. Network, 12, 75-87.

[18] Rieke, F., Warland, D., de Ruyter van Steveninck, R., Bialek, W. (1999). Spikes. MIT Press.

[19] Legenstein, R., Naeger, C., Maass, W., 2005. What can a neuron learn with spike-timing-dependent plasticity? Neural Computation 17 (11), 2337–2382.

[20] Tiesinga, P., Fellous, J., Salinas, E., Jos´e, J., Sejnowski, T., 2004. Inhibitory synchrony as a mechanism for attentional gain modulation. Journal of Physiology-Paris 98 (4-6), 296–314.

[21] Cornelius Glackin, Liam Maguire, Liam McDaid et al. (2011). "Receptive field optimisation and supervision of a fuzzy spiking neural network"., 247-56. In *Neural networks : the official journal of the International Neural Network Society* 24 (3).