Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Character Recognition using Spiking Neural Networks

Ankur Gupta* and Lyle N. Long†

***Abstract*— A spiking neural network model is used to identify characters in a character set. The network is a two layered structure consisting of integrate-and-fire and active dendrite neurons. There are both excitatory and inhibitory connections in the network. Spike time dependent plasticity (STDP) is used for training. The winner take all mechanism is enforced by the lateral inhibitory connections. It is found that most of the characters are recognized in a character set consisting of 48 characters. The network is trained successfully with increased resolution of the characters. Also, addition of uniform random noise does not decrease its recognition capability.**

I. Introduction

It is well known that biological neurons use pulses or spikes to encode information. Neurological research also shows that the biological neurons store information in the timing of spikes. Spiking neural networks belong to the third generation of neural networks and like their biological counterparts use spikes to represent information flow. They can use spatio-temporal information in communication and computation similar to biological neurons. As they use pulse coding for information processing, they are much faster than rate coding which implies some averaging mechanism, and is typically slower [1] [2]. This doesn't mean though that the rate coding scheme is never used but it means that pulse coding is used whenever faster speed is desired [2].

There have been many studies in the past using spiking neuron models to solve different problems. For example [3] used spiking neurons for spatial and temporal pattern analysis. They provided a biologically plausible learning algorithm for realizing RBFs (Radial Basis Functions), which themselves are quite powerful in function approximation, pattern classification etc. [4]. In this study, spiking neurons were used to compute RBFs by storing information in their delays. The time difference between the pre and the post synaptic spikes was used to learn these delays.

Panchev et al. [5] [6] [7] proposed a spiking neuron model, called the ADDS (Active Dendrite and Dynamic Synapse) model and used it for instructing a robot for navigation and grasping tasks. They used a mobile Khepera robot [8] equipped with a camera, gripper, and differential wheels for the real world tests. The task is to navigate the robot in a restricted environment with objects of different shape and color such that it executes specific instructions such as "go right" or "find blue ball" without colliding with other objects in the way.

Baig [9] developed a spatial temporal artificial neuron model for online cursive handwritten character recognition. Their model had the capability to handle spatial-temporal data sequences continuously. Buonomano et al. [10] proposed a model for position invariant character recognition by coding input intensity by relative firing time. They used traditional backpropagation to learn to discriminate histograms. Jaramillo et al. [11] used a network of spiking neurons to recognize short (~ 1 sec) spatial-temporal patterns.

The idea of using one spike per neuron to process information was explored by [12]. The order in which the neurons fired in a particular layer was used as a code. They justified their assumption by observing the electrophysical data of the monkey temporal lobe [13] [14] [15]. It was observed that some neurons of the monkey temporal lobe responded to a face stimuli with a latency of $80-100$ ms. After taking into account that the information has to pass through 10 different stages and the conduction velocities of the neocortical fibre, it was found that each stage had less than 10ms for computation. Such rapid processing presents problem for conventional rate coding if more than one spike is to be emitted by a cell [12]. Later [16], also using one spike per neuron, were able to train the network using spike time dependent plasticity (STDP) to recognize natural photograph of faces. They used spikeNet [17] for their simulations.

The present paper attempts to build and train a network using STDP such that after training it is able to identify different characters in a character set. The idea is that when the stimulus (in form of a constant current) is presented as input to the neurons, the output should be in the form of a series of spikes i.e. a neuron should respond to a familiar stimulus by spiking continuously unless another stimulus is presented.

The current paper is organized as follows: The next section presents the neuron model and the plasticity rule used. The neural network structure and the learning rule are presented in the third section. The last two sections present results and conclusions, respectively.

II. Neuron Model and Plasticity

The neuronal model used is the active dendrite and dynamic synapse model [6] [18] except that the dynamic synapses have not been used. We provide a brief introduction to the model in this section. In this model, a neuron receives input via spike(s) through a set of synapses and dendrites. The total post-synaptic

*Graduate student, Dept. of Computer Science and Engineering, Email: azg139@psu.edu

†Distinguished Professor of Aerospace Engineering and Mathematics, Email: lnl@psu.edu

The Pennsylvania State University, University Park, PA 16802

1-4244-1380-X/07/$25.00 ©2007 IEEE

current for the synapse i, with weight w^i attached to a dendrite is given by:

$$\tau_d^i \frac{dI_d^i(t)}{dt} = -I_d^i(t) + R_d^i w^i \delta(t - t_f^i) \quad (1)$$

Here t_f^i is the set of pre-synaptic spike times filtered as Dirac δ pulses. The time constant τ_d^i and resistance R_d^i defines the active property of the dendrites as the function of synaptic weights and are defined as:

$$\tau_d^i = \tau_{max} - |w_i|(\tau_{max} - \tau_{min}), |w_i| \leq 1 \quad (2)$$

From the above equation we can see that for high weights, τ_d^i is closer to τ_{min}, whereas for low weights, τ_d^i is closer to τ_{max}. Thus, as the time constant is low for stronger synapses, we have a earlier and steeper increase of the soma potential as compared to weaker synapses. The resistance R_d^i is given by:

$$R_d^i = \frac{\tau_d^i \theta}{R_m} \left(\frac{\tau_m}{\tau_d^i} \right)^{\frac{\tau_m}{\tau_m - \tau_d^i}} \quad (3)$$

Here θ is the neuron's firing threshold, R_m is the soma resistance, and τ_m is the soma time constant. The above equation for R_d^i ensures that the maximum value of the membrane potential change is proportional to the neuron's firing threshold θ.

The other influence to an output neuron comes from the somatic synapses feeding directly or close to the soma. These lateral or inhibitory connections enforce the winner-take-all mechanism as the synaptic activity of the neurons which are not the first to spike is inhibited. The equation governing the post-synaptic current for these synapses is give by:

$$\tau_s \frac{dI_s(t)}{dt} = -I_s(t) + \sum_i w^i \delta(t - t_f^i) \quad (4)$$

Finally, combining the contributions from the dendritic connections and the synapses feeding directly to the soma, we get the following governing equation for the total soma membrane potential, u_m:

$$\tau_m \frac{du_m(t)}{dt} = -u_m(t) + R_m \left(I_d(t) + I_s(t) \right) \quad (5)$$

where, $I_d(t) = \sum_i I_d^i(t)$ is the total dendritic current, τ_m is the soma time constant, and R_m is the soma resistance. When the membrane potential reaches the threshold value θ, it produces a spike, and immediately after that the membrane potential is reset to a value u_{reset} = $-1mV$. After this event, the membrane potential recovers to the resting potential value.

The learning for both, the synapses attached to active dendrites and the somatic synapses takes place using spike time dependent plasticity (STDP) [19] [20] [21] [22] [23] rules. STDP emphasizes that synaptic plasticity depends on the precise timing and temporal order between the pre and post synaptic spikes. It serves to address some of the major shortcomings of the Hebbian based learning such as competition between synapses and accidental pre/post synaptic correlation activity [22].

The general STDP learning rule for the synapses is given by:

$$\Delta w = \begin{cases} A^+ e^{\frac{\Delta t}{\tau^+}} & \text{if } \Delta t < 0 \\ A^- e^{\frac{-\Delta t}{\tau^-}} & \text{if } \Delta t > 0 \end{cases} \quad (6)$$

Where, $\Delta t = t_{pre} - t_{post}$. Here the goal is that weights are changed such that if the pre-synaptic spike occurs before the post-synaptic spike, the synapse is strengthened else it is weakened. This ensures that the next (after this training iteration) post-synaptic spike occurs closer to the pre-synaptic spike. The weights are then changed according to the relaxation rule:

$$w_{new} = \begin{cases} w_{old} + \eta \Delta w (w_{max} - w_{old}) & \text{if } \Delta w \geq 0 \\ w_{old} + \eta \Delta w (w_{old} - w_{min}) & \text{if } \Delta w < 0 \end{cases} \quad (7)$$

Here, η is the learning rate. For excitatory synapses $w_{min} = 0$ and $w_{max} = 1$, whereas for inhibitory synapses $w_{min} = -1$ and $w_{max} = 0$. If there is no pre-synaptic spike, the weight decays with a rate η_{decay}. This is similar to to the decay of synaptic strength in real neurons.

III. Test Problem and Network Structure

In order to test the network, a character set, shown in figure 1 consisting of 48 characters is used. Each character is represented by an array of $3X5$ pixels. The size of the input is increased by increasing the resolution of the characters. Thus, for a resolution of R, each pixel in the original character set is replaced by R X R identical pixels. The character set and the test problem is the same as used in [24], which used traditional neural networks and back propagation on massively parallel computers. Integrate and fire neurons [25] with constant input current are used as input neurons. If the pixel is "on", a constant current is supplied to the corresponding neuron, whereas if the pixel is "off", no current is supplied to that particular neuron. Figure 2 shows a typical representation of the character "A" by the input neurons.

The various parameters used in the network are defined in the appendix. The number of input neurons is equal to the number of pixels in the image. Thus there are 15 input neurons in the present case. The number of output neurons is the number of characters to be trained. There are two layers in the network. The first layer consists of simple leaky integrate and fire neurons which receive constant or zero input current, corresponding to 'on' or 'off' states of the input pixels. The next layer is the layer of active dendrite neurons, each of which is connected to all of the neurons in the previous layer. Finally, each of the output layer neuron is connected to every other output neuron via inhibitory lateral connections. These lateral connections reflect the competition among the output neurons.

IV. Results

For initial testing, the network was trained using only four characters ('A', 'B', 'C', and 'D'). There were 15 input neurons and 4 output neurons for this case. The training parameters used are described in the appendix. The weights

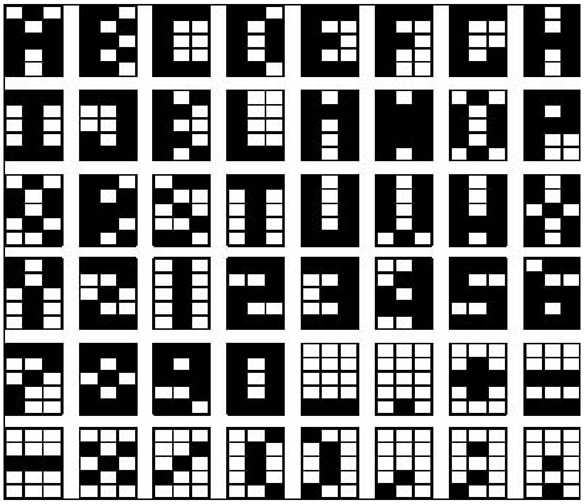


Fig. 1. Character set used

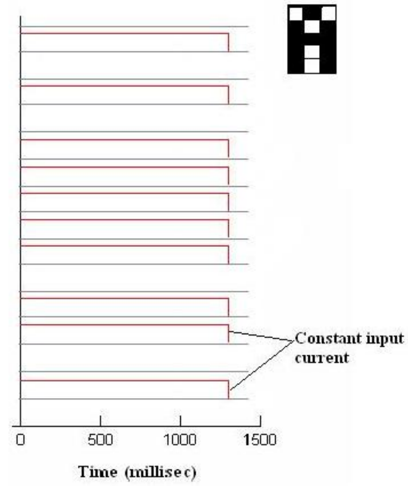
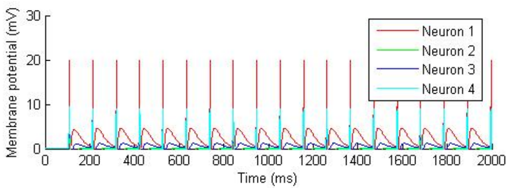
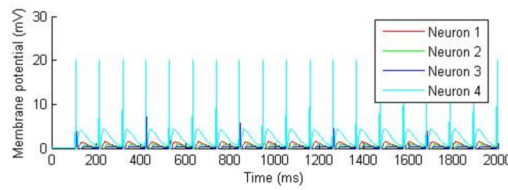


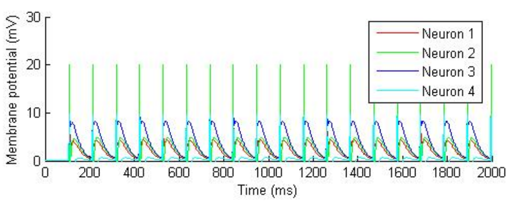
Fig. 2. Representation of 'A'



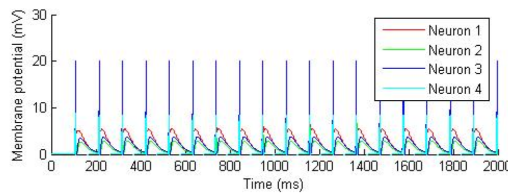
(a) Character 'A'



(b) Character 'B'



(c) Character 'C'



(d) Character 'D'

Fig. 3. Output when each character is presented individually

were initialized to random values between 0.5 and 1.0, so that all the output neurons spike at the first training iteration. Each character was presented one at a time sequentially during the training process. When the Frobenius norm of the weight change matrix was very small, it was assumed that the network is fully trained and no further significant learning is possible. For this simple case, the Frobenius norm of the weight change matrix was 10^{-3} after 100 epochs (an epoch means one full presentation of all the four characters), and thus the training was stopped at this point. The simulation time step was $0.2ms$.

Figures 3(a), 3(b), 3(c), 3(d) show the output of all the four neurons when the characters 'A', 'B', 'C', and 'D' are presented as inputs respectively after training. We can see that only one neuron responds to a particular character, by spiking continuously with spike-time interval of $100ms$, whereas the other neurons are below the threshold value of $10mV$. Particularly, neurons $1, 2, 3$, and 4 respond to characters 'A', 'C', 'D', and 'B' respectively. Note that here the current of the input neurons is set to a constant value such that the neurons spike after every $100ms$ interval. It was observed that the minimum spike-time interval allowed is roughly $80ms$, for the network to respond correctly. This is because this is roughly the time required for the neurons to reach the resting membrane potential.

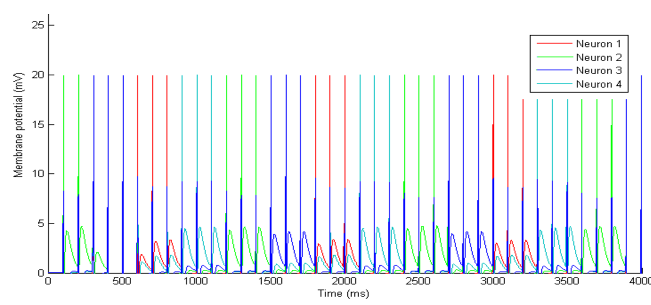
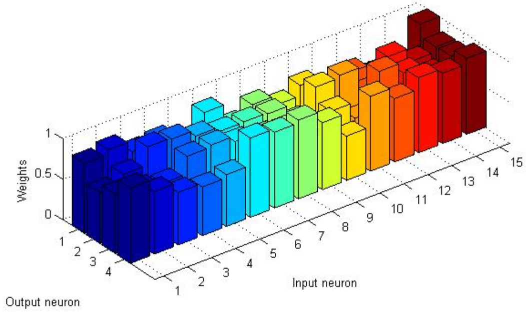
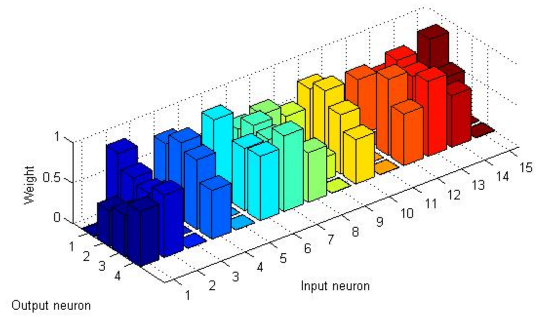


Fig. 4. Output when characters are presented in following order: 'C', 'D', 'A', 'B', 'C', 'D', ...

Figure 4 shows the output of the network when the

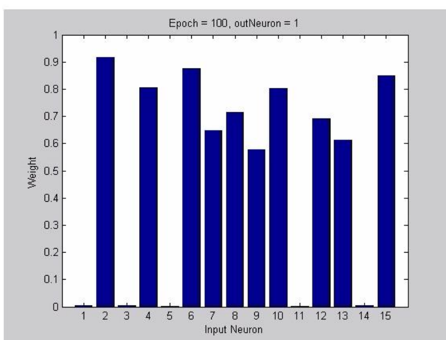


(a) Before training

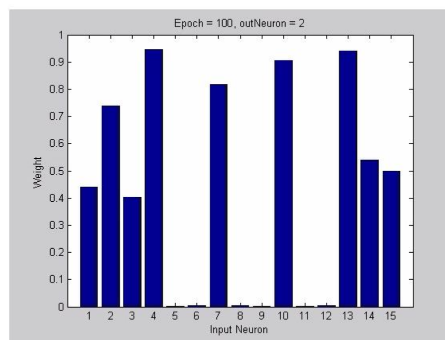


(b) After training

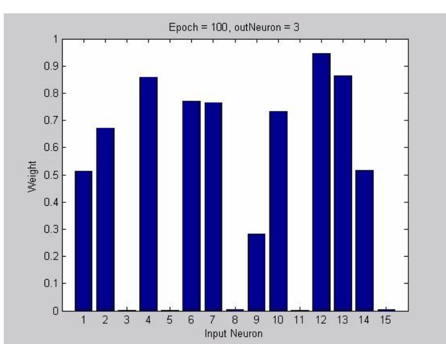
Fig. 5. Weight distribution



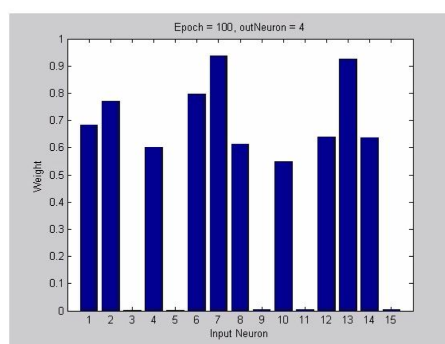
(a) Neuron 1



(b) Neuron 2



(c) Neuron 3



(d) Neuron 4

Fig. 6. Weight distribution for output neurons after training

characters are presented one by one in the order 'C', 'D', 'A', 'B', 'C', 'D', ... and so on. A new character was presented $1ms$ before every $300ms$. We can again see that only a particular neuron responds to a particular character by spiking continuously unless the next character is presented, when another neuron starts spiking, and so on. Figures 5(a) and 5(b) show the weights of each of the connections before and after training respectively.

Figures 6(a), 6(b), 6(c) , 6(d) show the weights of incoming connections for the output neurons after training. It can be seen that apart from other changes, each output neuron after learning a particular character, adjusts the corresponding weights for the pixels which were off to zero. For example, neuron 1 learned character 'A', and after training the weights of the incoming connections for which the corresponding pixels were off are adjusted to zero. This is expected as the synaptic strength decreases in the absence of any presynaptic spike. We would like to emphasize that it is not known in advance that which neuron learns which character, thus it is not possible to set the appropriate weights to zero initially in a hope to speed up the training.

Some trials with more number of inputs were also conducted. As mentioned in the previous section the size of the input was increased by increasing the resolution of the characters. In the present case, a resolution of 2 for the same problem as above was tried. Figure 7 shows the representation of characters with a resolution of 2. Thus, the number of inputs(and the no. of pixels) were now 60. The membrane resistance was scaled down by a factor of 4 as there were 4 times more input neurons as compared to the previous case.

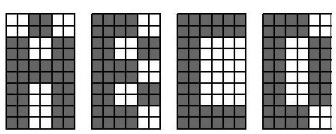


Fig. 7. Characters with resolution 2

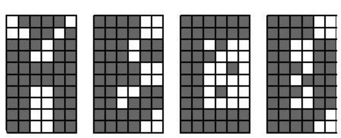


Fig. 8. Characters with noise added

After training again each output neuron "learns" to respond to a particular character. In particular, neurons $1, 2, 3$, and 4 respond to characters 'B', 'C', 'D', and 'A'. Random noise was also added to the characters as shown in figure 8 by switching 4 random "off" pixels in each character to "on". The network was still able to correctly recognize all the characters. Again, it was seen that each output neuron after learning a particular character, adjusts the corresponding weights for the pixels which were off to zero.

The next test case was trying to train the full 48 character set in figure 1. The network here consisted of 15 neurons in the input layer, and 48 in the output layer. The network structure was the same as previous except that there were more connections as the number of output neurons was 48. The network was trained for 100 epochs. During this training process each of the 48 characters was presented sequentially until the Frobenius norm of the weight change matrix was very small. Figure 9 shows the variation of the Frobenius norm with the number of training epochs. It decreases roughly linearly (log scale) with number of epochs.

Most of the characters (43) were recognized uniquely in the sense that either a unique neuron fired or the firing times of the same neuron were different. A single neuron responded with same firing times for the characters M, U, and W. Strikingly, all have the same number of pixels (11) in there character representations and look very similar. Similarly, characters J and 3, each having 10 pixels, also had non-unique representations. Rest of the characters had a unique representation.

Figure 10 shows the soma potentials of all the 48 neurons after training on presentation of a single character. Each symbol along with the color represents a unique neuron. We can see that only one neuron ($\diamond$ symbol with cyan color), which is the winner, spikes, and the rest remain below the threshold value of $10mV$. Similar observations were made with rest of the characters.

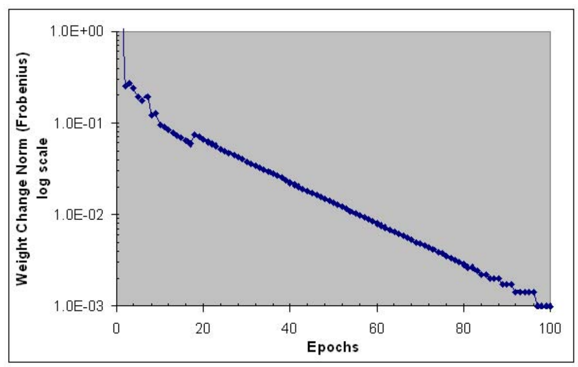


Fig. 9. Variation of Frobenius norm of the weight change matrix with no. of epochs

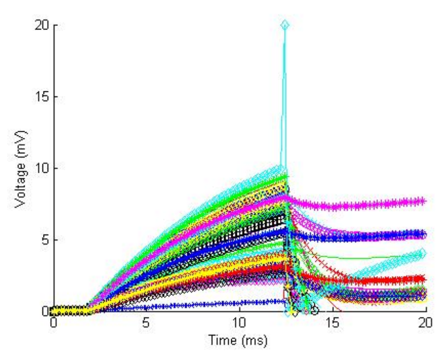


Fig. 10. Soma potentials for the case of 48 characters: Only one is the winner

V. Conclusion

A two layered spiking neural network was used to identify characters in a character set. STDP was used to train the network. The network was trained until no significant weight change was observed. Most of the characters were recognized when the network was trained using a character set of 48 characters. The network was successfully trained with characters of increased resolutions. The network was able to recognize characters after addition of random noise pixels. Evolution of the weights during training shows that each output neuron adjusts the weights corresponding to the pixels of the "learned" character which were off to zero. Preliminary results look encouraging and more complex problems such as image recognition will be tried in the future.

Acknowledgment

This research was supported by ONR Grant No. N00014-05-1-0844.

Appendix I
Running parameters

$R_m = 80$, $\theta = 10mV$, $\tau_s = 2ms$, $\tau_m = 30ms$, $\tau_{min} = 2ms$, $\tau_{max} = 30ms$, $\eta = 0.1$, $\eta_{decay} = 0.05$, $A^+ = 0.1$, $A^- = -0.105$, $\tau^+ = 1ms$, $\tau^- = 1ms$, $time\ step = 0.2ms$

References

[1] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, *Hebbian Learning of Pulse Timing in the Barn Owl Auditory System*, ser. Pulsed Neural Networks, W. Maass and C. M. Bishop, Eds. Berlin, Germany: MIT-press, 1999.

[2] S. Thorpe, A. Delorme, and R. van Rullen, "Spike based strategies for rapid processing," *Neural Networks*, vol. 14, pp. 715–726, 2001.

[3] T. Natschlager and B. Ruf, "Spatial and temporal pattern analysis via spiking neurons," *Network: Comput. Neural Syst.*, vol. 9, pp. 319–332, 1998.

[4] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice-Hall, 1999.

[5] C. Panchev and S. Wermter, "Spike timing dependent synaptic plasticity: from single spikes to spike trains," *Neurocomputing*, vol. 58-60, pp. 365–371, 2004.

[6] C. Panchev, "Spatio-temporal and multimodal processing in a spiking neural mind of a robot," Ph.D. dissertation, Univeristy of Sunderland, United Kingdom, May 2005.

[7] C. Panchev and S. Wermter, "Temporal sequence detection with spiking neurons: Towards recognizing robot language instruction," *Connection Science*, vol. 18, pp. 1–22, 2006.

[8] [Online]. Available: http://www.k-team.com/

[9] A. R. Baig, "Spatial-temporal artificial neurons applied to online cursive handwritten character recognition," in *European Symposium on Artificial Neural Networks*, April 2004, paper Bruges (Belgium), pp. 561–566.

[10] D. V. Buonomano and M. M. Merzenich, "A neural network model of temporal code generation and position invariant pattern recognition," *Neural Computation*, vol. 11, pp. 103–116, 1999.

[11] S. Jaramillo and F. Pereira, "Recognition of dynamic patterns with a network of spiking neurons." [Online]. Available: http://www.cs.unm.edu/ sjara/docs/Jaramillo_and_Pereira_CSSS2002.pdf

[12] R. van Rullen, J. Gautrais, A. Delorme, and S. Thorpe, "Face processing using one spike per neurone," *Biosystems*, vol. 48, pp. 229–239, 1998.

[13] C. J. Bruce, R. Desimone, and C. G. Gross, "Visual properties of neurons in a polysensory area in superior temporal sulcus of the macaque," *Journal of Neurophysiology*, vol. 46, pp. 369–384, 1981.

[14] D. I. Perrett, E. T. Rolls, and W. Caan, "Visual neurons responsive to faces in the monkey temporal cortex," *Experimental Brain Research*, vol. 47, pp. 329–342, 1982.

[15] D. A. Jeffreys, "Evoked potential studies of face and object processing," *Visual Cognition*, vol. 3, pp. 1–38, 1996.

[16] A. Delorme and S. J. Thorpe, "Face identification using one spike per neuron: resistance to image degradations," *Neural Networks*, vol. 14, pp. 795–804, 2001.

[17] [Online]. Available: http://www.sccn.ucsd.edu/ arno/spikenet/

[18] C. Panchev, S. Wermter, and H. Chen, "Spike-timing dependent competitive learning of integrate-and-fire neurons with active dendrites," in *Proceedings of the International Conference on Artificial Neural Networks*, August 2002, pp. 896–901.

[19] H. Markram, J. Lubke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps," *Science*, vol. 275, pp. 213–215, 1997.

[20] 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. 10 464–10 472, 1998.

[21] Y. Dan and M. Poo, "Spike time dependent plasticity of neural circuits," *Neuron*, vol. 44, pp. 23–30, 2004.

[22] S. Song, K. D. Miller, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.

[23] B. Berninger and G. Q. Bi, "Synaptic modification in neural circuits: a timely action," *BioEssays*, vol. 24, pp. 212–222, 2002.

[24] L. Long and A. Gupta, "Scalable massively parallel artificial neural networks," in *AIAA InfoTech@Aeropace Conference*, September 2005, paper 2005-7168.

[25] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.