Learning Categories with Spiking Nets and Spike Timing Dependent Plasticity

Christian Huyck[1]

Middlesex University, London NW4 4BT UK
c.huyck@mdx.ac.uk

Abstract. An exploratory study of learning a neural network for categorisation shows that commonly used leaky integrate and fire neurons and Hebbian learning can be effective. The system learns with a standard spike timing dependent plasticity Hebbian learning rule. A two layer feed forward topology is used with a presentation mechanism of inputs followed by outputs a simulated ms. later to learn Iris flower and Breast Cancer Tumour Malignancy categorisers. An exploration of parameters indicates how this may be applied to other tasks.

Keywords: Spiking Neural Network · STDP · Categorisation

1 Introduction

AI is a critical technology with immense interest from governments, companies and society at large. Recent developments in machine learning and deep networks in particular have achieved success in a wide range of areas, such as face recognition [15] and games [13].

Deep Nets [8] are a diverse group of systems typically with large numbers of units between layers, and many layers that are well connected. These connectionist systems are typically inspired by the brain, and called neural networks.

Simulated biological neural networks, on the other hand, attempt to reproduce the behaviour of brains, or parts of brains [12]. These are based on models of biological neurons, models of biological learning, and biological topologies. The neural models are typically spiking neurons.

This paper describes a system that categorises data based on a neural network. The network has aspects of biological plausibility combined with a biologically unrealistic topology. The plausible aspects include learning via spike timing dependent plasticity, a Hebbian learning rule, and a widely used, though simple, biological neuron model. It is not clear that the presentation or testing mechanism is psychologically realistic.

2 Literature Review

There are many neural models including relatively simple point models that represent neurons by simple equations and elaborate compartmental models [9] that break neurons into compartments and evaluate the conductance of these compartments. Leaky integrate and fire (LIF) neurons integrate activation from other neurons. The activation leaks away, but if enough accumulates, the neuron fires emitting a spike. The activation resets after it fires. In this paper, the LIF neural model is from Brette and Gerstner [4]. The model includes exponential current transmission, so that the current is transferred across the synapse (after the pre synaptic neuron fires) at an exponentially decaying rate.

In the brain, most if not all learning is Hebbian [7]. If the presynaptic neuron tends to cause the postsynaptic neuron to fire, the weight will tend to increase. There are many rules, but a great deal of biological evidence supports Spike Timing Dependent Plasticity (STDP) [3]. Bi and Poo [3] have perhaps the first published example that shows the performance of biological neurons. Song et al. [14] have developed an idealised curve that fits the biological data, though it is a curve fitting exercise. If the pre-synaptic neuron fires before the post, the weight is increased; if the post-synaptic neuron fires first, the weight is decreased. Note that the closer to precisely co-firing, the more the weight change. The simulations in the remainder of the paper use an STDP learning rule.

LIF neurons were used as the neural model. The system was developed using PyNN middleware [5], a python package to specify the topology and manage inputs. The backend was the NEST neuron simulation platform [6].[1]

3 Methods

Data was taken from the widely used University of California at Irvine (UCI) benchmark [1]. A commonly used task, categorisation of Iris flowers, is used initially. The data was split into two equal sized groups. The Iris data has 150 instances, 50 of each of three categories, so the data was split into two 75 item data sets with 25 of each category in each.

First the data is preprocessed by scaling the range of features to 0 to 100 with two digits of precision. Now all features are represented by an integer between 0 and 100 inclusive. The input to the system is represented by a neuron for each number. So, for the Iris data, there are four features, and thus 404 input neurons.

There is a neuron for each output category. For the Iris data, there are three categories, and thus three neurons. The input neurons are well connected to the output neurons using plastic synapses. The plasticity rule is a variant Hebbian STDP (consistent with Song et al. [14]).

During training, the input neurons are sent a spike, and the output neurons are sent a spike one ms. later. This uses the PyNN spike source, an impossible biological mechanism for learning. The input neurons consist of those with the input feature, and in a window of three. So, when the first training feature is 19, the neurons numbered 15 to 21 are stimulated as numbering is zero based. The neurons are stimulated so that they fire once.

One of the parameters that was explored in development was the number of training epochs. An epoch is the presentation of all the training examples; in

[1] The code can be found on http://www.cwa.mdx.ac.uk/NEAL/NEAL.html.

the case of irises, all 75 training items. There may be several epochs of training with all of the training items presented in sequence in each epoch.

The time between each example was another parameter that was explored. This time can affect the system for two main reasons. First, if the time is too small, an input example can continue to spike into the next example. Second, the STDP synaptic reduction window is effected by the prior example; if the prior example fires nearer to the time of the current example, the synaptic weight from the prior input neurons to the current category neurons may be reduced. During testing, the input neurons are stimulated, and the spikes of the output neurons are counted. The input is categorised based on the spiking behaviour with the neuron that spiked most winning.

It is possible to set the neural parameters for the model, but the default parameters were used, and varying them was not explored. Some important parameters are: the firing threshold, the higher the threshold, the more activation is required for a spike; the refractory period, after a neuron spikes, all input activation is ignored during the refractory period; and the leak rate, the higher the leak rate the faster activation leaks away making the neuron more difficult to fire over short periods of time. Another simulation parameter is the time step that sets how often are the neural and synaptic variables are updated. These simulations used a 1ms time step.

The STDP learning rule is described by seven parameters. The first is the initial synaptic weight i, the second is the maximum weight m. The third is the minimum weight that has always been 0 in the simulations described in this paper; input features may have no influence on output categories, so the neurons that represent these values should have a 0 connection. There are many STDP rules, and a spike pair rule is used in the simulations in this paper. The weight is modified based on spike pairs alone. There are four parameters associated with this, two for increasing the weight and two for decreasing the weight. The increasing parameters are $A+$, for scaling how much the weight increases, and $\tau+$ for stretching the window of that the weight increases. The parallel decreasing parameters are $A-$ and $\tau-$. $\tau+$ and $tau-$ are in ms.

Parameters are explored to develop a system that categorises reasonably well. The data has been broken into a training and a test set. Parameters are explored using the training set, and the test set is used, largely, for reporting.

In the first example there are 5 training epochs; the time between examples is 30; the initial synaptic weight was 0; the maximum synaptic weight was 0.05; the synaptic increase parameters were $\tau+ = 20.0ms.$ and $A+ = 0.004$, and the decrease parameters were $\tau- = 20.0ms.$ and $A- = 0.003$. This is represented by the first line in table 1.

4 Results

Exploration of parameters can include a change of topology, but some simple things to explore are the learning parameters, and presentation mechanism. In particular, the five learning parameters, the number of training epochs, and the length of a presentation are explored.

Exploration is done by training the system on the training set, and testing on the testing set. It is not a two fold cross validation. The system uses no randomness. Repeating a run will have the same results.

One large piece of information is how many test examples actually have neurons firing. The default categorisation works with no neurons firing but it shows that the system is not categorising. The parameters affect how many categorisation neurons fire in the test. Unsurprisingly, typically, the more training epochs, the more test firing; synaptic weights are initially zero and each presentation provides the opportunity to increase, and increased synaptic weight leads to further firing in future epochs, further increasing weights. This may not be the case if the initial weight is larger than zero.

Too much firing also has a problem in that the output becomes random. The neurons integrate input from a set of pre-synaptic neurons, all firing once at the same time. The neuron can get enough activation so that it fires multiple times. However, if two or three of the categorisation neurons get a great deal of input, they saturate and fire the same number of times. So, there is an ideal window of incoming synaptic strength to differentiate between the categories.

Perhaps the most powerful mechanism for increasing firing is to increase the maximum synaptic weight m. So, if the system with a particular parameter set had many tests with output neurons firing, m was increased. Similarly, the total output spikes can also be tabulated, and if this is very high, m can be reduced.

The synaptic weight increase and decrease constants $A+$ and $A-$ also influence output neuron firing. Increasing $A+$ or decreasing $A-$ leads to increased firing; decreasing $A+$ or increasing $A-$ leads to decreased firing.

The goal is to have a system that categorises well. So, the categorisation results also matter. By following the gradient so that all tests have categorisation neurons firing, but many with only one spike, parameters can be set to find a good result on the training set (e.g. the second row of table 1).

Two other systems are shown for comparison. The first [10], in the third row is a spiking net using a compensatory Hebbian rule; the neurons have adaptation. As it uses randomness, the average results of a two fold test are shown. The second [16] uses a specialised feed forward neural topology and a variant of STDP that incorporates a learning signal; training and testing are separated in their evaluation, and in table 1.

Wisconsin Breast Cancer Categorisation

A second task, the Wisconsin Breast Cancer Categorisation task, again from the UCI benchmark [1] is reported. An item refers to a patient and is represented by 9 relevant features, and the output is a binary value referring to whether the tumour was benign or malignant. There were 699 items, with 241 malignant category items. The data set was split into two with the training set having an extra item and an extra malignant item.

Each feature had a range from 1 to 10, one feature had missing values, and one feature had one value that was not represented. So, each input feature was

Table 1. Categorisation Results: WBC refers to the Wisconsin Breast Cancer task.

Task	Epochs	Example Time	Maximum Weight m	$A+$	$\tau+$	$A-$	$\tau-$	Train Result	Test Result
Iris	5	30	0.005	0.004	20.0	0.003	20.0	86.67%	90.6%
Iris	6	50	0.003	0.005	20.0	0.002	20	92%	90.6%
Iris [10]									93.53%
Iris [16]								95.5%	95.3%
WBC	6	50	0.002	0.006	20.0	0.009	20	95.14%	95.7%
WBC [16]								96.2%	96.7%

represented by 10 neurons and the two output categories by one neuron each. When an item was presented, only one neuron was stimulated for each feature, and the missing feature was simply ignored; one of the benefits of this approach is that missing features are readily ignored.

A simple exploration of the parameter space began with the parameters from the Iris data set (line 2 of table 1). This exhibited a great deal of firing during testing, so the Maximum Weight w was reduced. This left little firing, so $A+$ was increased. Somewhat surprisingly, increasing $A-$ also improved results leading to several training parameter sets that got 95.14%. One was chosen, and the results are displayed in the first WBC line of table 1. Below that the results reported from another spiking system [16] are shown.

5 Discussion

This paper has shown an exploration of simple feed forward topology and a standard learning rule based on a standard biological LIF neuron, and a standard STDP rule. The presentation mechanism of turning on the input neurons one step before the output neurons is clearly biologically impossible. Similarly, the uniformity of the initial feed forward topology is also biologically implausible. While the results are below the state of the art, and the tasks are simple, the results are quite near the state of the art. This merely shows how powerful the strictly Hebbian STDP learning mechanism is.

STDP, with the topology and presentation mechanism used above, has a result that is a type of covariance rule. The synaptic weight from a neuron representing an input feature will increase if it is used as a member of the category. If it is also used for another category, it will decrease, so the weight roughly reflects the likelihood the feature discriminates between the categories. If it is involved in two categories, the weight will be lower, and if in three lower still. The feature breadth mechanism used in the Iris task supports learning from fewer examples, and generalisation to unpresented data.

STDP is strictly Hebbian, so is an entirely unsupervised mechanism. Reinforcement can be included by adding extra topology to encourage neurons to fire at appropriate times; this is a solution that has been included in spiking networks [2, 16]. Adjusting synaptic weights to reflect desired outputs, as is done in supervised rules such as back propagation [11] does not seem to have a biological basis. This supervised learning is a powerful mechanism, particularly for feed forward networks. However, the brain is not feed forward but highly recurrent.

This paper has used a simple two layer feed forward approach. Learning here is based on particular inputs. Another approach would be to extend across layers with different times so that input could cascade through layers. Other precise timing mechanisms can be developed, but in the brain, most neurons fire more or less continuously at a low rate. Closely timed mechanisms will not work as models of actual biological processing.

References

1. Bache, K., Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
2. Belavkin, R., Huyck, C.: Conflict resolution and learning probability matching in a neural cell-assembly architecture. Cognitive Systems Research **12**, 93–101 (2010)
3. Bi, G., Poo, M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. Journal of neuroscience **18**(24), 10464–10472 (1998)
4. Brette, R., Gerstner, W.: Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. J. Neurophysiol. **94**, 3637–3642 (2005)
5. Davison, A., Yger, P., Kremkow, J., Perrinet, L., Muller, E.: PyNN: towards a universal neural simulator API in python. BMC neuroscience **8**(S2), P2 (2007)
6. Gewaltig, M., Diesmann, M.: NEST (NEural Simulation Tool). Scholarpedia **2**(4), 1430 (2007)
7. Hebb, D.: The organization of behavior: A neuropsychological theory. New York: Wiley (1949)
8. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation **18:7**, 1527–1554 (2006)
9. Hodgkin, A., Huxley, A.: A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology **117**, 500–544 (1952)
10. Huyck, C., Mitchell, I.: Post and pre-compensatory Hebbian learning for categorisation. Computational Neurodynamics **8:4**, 299–311 (2014)
11. Rumelhart, D., McClelland, J.: Parallel Distributed Processing. MIT Press (1986)
12. Sejnowski, T., Koch, C., Churchland, P.: Computataional neuroscience. Science **241:4871**, 1299–1306 (1988)
13. Silver, D., Schrittwieser, J., et al, Hassabis, D.: Mastering the game of go without human knowledge. Nature **550**, 354–59 (2017)
14. Song, S., Miller, K., Abbott, L.: Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nature neuroscience **3:9**, 919–926 (2000)
15. Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: Face recognition with very deep neural networks. CoRR **abs/1502.00873** (2015), http://arxiv.org/abs/1502.00873
16. Wade, J., McDaid, L., Santos, J., Sayers, H.: Swat: a spiking neural network training algorithm for classification problems. IEEE Transactions on Neural Networks **21:11**, 1817–1830 (2010)