Self Organising Maps with a Point Neuron Model

Ian Mitchell and Christian Huyck
Middlesex University, London, UK
i.mitchell@mdx.ac.uk and c.huyck@mdx.ac.uk

This abstract describes simulations using a reasonably biological accurate point neuron model, a fatiguing leaky integrate and fire model. These model neurons use a novel compensatory Hebbian learning rule to categorise data items, a standard machine learning task. The resulting system is a kind of self organising map, which compares favourably with a Kohonen map on one machine learning task.

The simulation is discrete. The fatiguing leaky integrate and fire neural model integrates activity from connected firing neurons, retaining some from prior cycles (equation 1).

$$a_i^t = \frac{a_i^{t-1}}{d} + \sum_j w_{ji}, d > 1 \quad (1)$$

The activation at time t, a_i^t is the sum of the new activation the neuron receives, and the activation from the prior time step divided by a decay factor d. The decay factor models the leak. If a neuron spikes, it loses all activation, though incoming activation may cause it to spike in the next cycle.

The neuron accumulates fatigue when it fires, increasing by a constant F_c and decreasing by a constant F_r when it does not fire when fatigue is non-negative. It fires when $a_i^t - F_i^t > \theta$, where θ is a constant firing threshold.

Parameters were tuned to recorded rat somatosensory cortical neurons with $\theta = 2.2$, $d = 1.12$, $F_c = 0.045$, and $F_r = 0.01$. With these parameters, assuming each cycle is 10 ms., over 90% of the spikes match within 2 cycles on a widely varying input regime. The model was extended so that neurons that were hypofatigued, $F_i^t > \theta$, fired with no input. In this case the fatigue was halved when the neuron fired and $F_i^t < -.25$. Negative fatigue reduction was more complex. This led to an improved fit to biological data even though there was no spontaneous firing in that data.

Weights were learned following a Hebbian learning rule that increased the strength when the pre and postsynaptic neurons co-fired in a cycle (equation 2), and decreased the weight when the presynaptic neuron fired and the postsynaptic neuron did not fire (equation 3). This rule was compensatory. The presynaptic compensatory rule used the total weight leaving the neuron and the postsynaptic compensatory rule used the total weight entering the postsynaptic neuron W_j.

$$\Delta_+ w_{ij} = (1 - w_{ij}) * R * 10^{(W_B - W_j)} \quad (2)$$

$$\Delta_- w_{ij} = w_{ij} * -R * 10^{(W_j - W_B)} \quad (3)$$

The weight w_{ij} is modified using the learning rate $R = 0.01$ and the constant desired synaptic strength W_B. This rule forces the incoming or outgoing strength toward W_B.

Recently, spike timing dependent plasticity (STDP) has been typically used in similar circumstances, but this rule is not available to a model using a 10 ms. cycle because it depends on spike latencies on the order of 1 ms. The authors have used presynaptic compensatory learning to form hierarchical categories with a network in earlier work. Note that STDP can make use of a compensatory rule.

The topology used was a set of neurons for input, with connections leading to a second set of 1000 neurons. The second set also had connections between them. The learning rule for neurons leaving the first set was presynaptic compensatory with $W_B = 5$, and the rule for those between the second set was postsynaptic with $W_B = 1$. All connections were sparse and random, and all neurons are excitatory.

Introducing the firing of hypofatigued neurons led to a principled spontaneous firing. Via Hebbian learning, a connection to an unfiring neuron cannot gain strength, so neurons not externally stimulated could never be involved. Combining spontaneous firing and compensatory learning provides a principled way of moving beyond externally stimulated neurons, involving the second set of neurons.

The system was used to categorise the car data set from the University of California at Irvine's Machine Learning repository. This consisted of data items described by 6 input fields (with 3 or 4 values) and a category (4 values). Each feature-value was represented by 10 neurons and all were externally stimulated for 40 cycles followed by no input for 35 for each training epoch. During this phase of training, the neurons are not reset so that the target neurons can spontaneously fire, and become involved in the system.

R is multiplied by .7 every 5,000 cycles, and after 20,000 cycles, the training set was presented again. The firing behaviour of the second set of neurons was recorded over an epoch of 75 cycles yielding a vector of integer values between 0 and 75. The system was reset after each of these with each neuron having activation and fatigue set to 0. This reset prevented unstimulated neurons firing during the test. After the second phase of training, the test set was presented. Its firing behaviour was recorded and a Pearson test was done between the test set and all the training items. The category of the closest training item was taken as the category of the test item.

A four-fold test was performed with the system trained on 432 items and tested on 1296. This was done on 100 nets for each fold. The result was 78.92%. The test was duplicated on a Kohonen net yielding a score of 75.75%. Finally, the original system was tested on the training set with a score of 99.99% showing that the net does memorise the input.

The second set of neurons acts as a distributed self-organising map, responding to the input in the first set of neurons. The combination of pre and post-synaptic compensatory learning with the topology allow all of the neurons to be productively used in the map.

This shows that a biologically plausible neural model and learning rule have comparable performance to a standard machine learning algorithms. Moreover, the introduction of spontaneous firing neurons and postsynaptic compensatory learning provide a mechanism to involve neurons not directly stimulated by the environment.