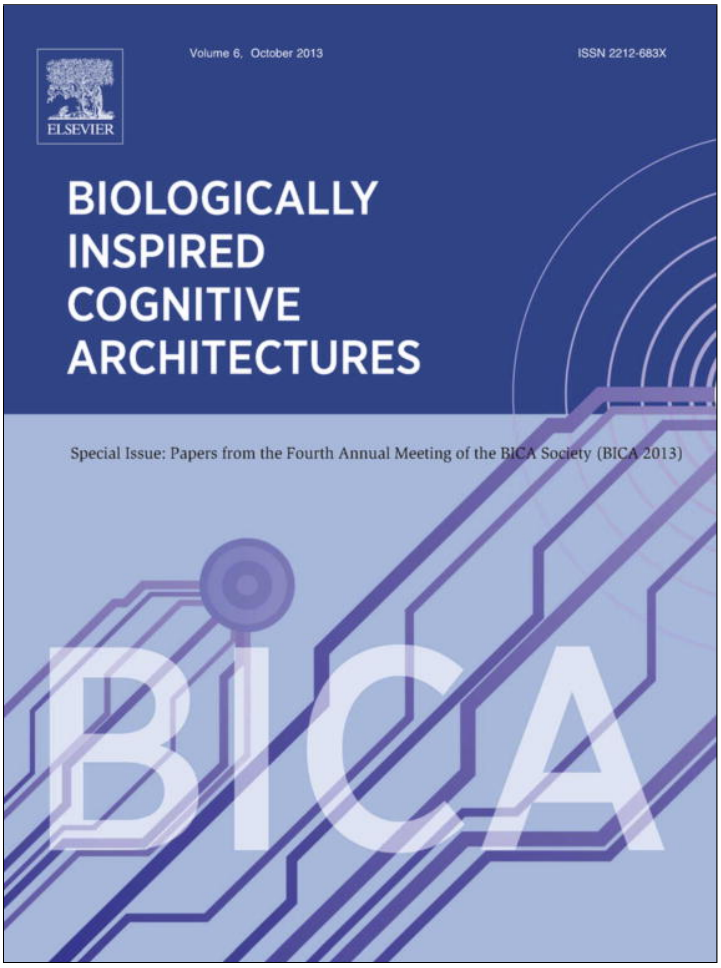

Volume 6, October 2013
ISSN 2212-683X

BIOLOGICALLY
INSPIRED
COGNITIVE
ARCHITECTURES

Special Issue: Papers from the Fourth Annual Meeting of the BICA Society (BICA 2013)

Biologically Inspired Cognitive Architectures (2013) **6**, 3–7

Available at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

Compensatory Hebbian learning for categorisation in simulated biological neural nets

Christian R. Huyck *, Ian G. Mitchell

Middlesex University, Dept. of Computer Science, London, UK

Received 15 March 2013; received in revised form 18 June 2013; accepted 22 June 2013

KEYWORDS
Fatiguing leaky integrate and fire neuron;
Compensatory learning;
Biological cognitive architecture;
Categorisation

Abstract
Using a reasonably accurate fatiguing leaky integrate and fire (FLIF) neural model, and biologically plausible compensatory Hebbian learning rules, simulations categorise benchmark machine learning data. The FLIF model is a simple, efficient point model with discrete cycles roughly corresponding to 10 ms. of biological time. The model is applied to the yeast categorisation task and the results are compared with those of other mature machine learning algorithms, including a new Kohonen net. Synaptic weights are changed following a compensatory Hebbian rule that includes the total synaptic weight of a neuron. The neural model leads to spontaneous neural firing that enables neurons not directly stimulated by the environment to be included in the neural categorisation circuit. The network is sparsely connected, and broken into two subnets, with the first subnet directly stimulated by the environment, and using compensatory learning based on the strength leaving the neuron. The second subnet initially fires only spontaneously, and uses compensatory learning based on the weight entering the neuron. After learning, new items are categorised based on a Pearson measurement comparing the firing behaviour of the second subnet on trained items, and the test item. The simulation is self-organising using only unsupervised learning. This ‘‘biologically'' plausible learning mechanism and network is close to the machine learning algorithms' performance; the biological network categorises 53% correctly, while the Kohonen net categorises 56% correctly. This neural simulation is incomplete, but supports further developments in biological neural cognitive architectures.

---

* Corresponding author. Tel.: +44 208 411 5412.
E-mail address: c.huyck@mdx.ac.uk (C.R. Huyck).

2212-683X/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bica.2013.06.003

Introduction

One long-term research goal is to create a simulated mind; indeed there is a research effort with substantial support to build a computer capable of simulating the brain at a relatively fine-grained level (Markram, 2006). To create a simulated mind from a simulated brain, it is essential to understand the basic functions of the brain. This goal relates to work in cognitive architectures leading to the goal of a biological neural cognitive architecture.

One problem that a biological cognitive architecture must address is how one set of neurons autonomously learns from the behaviour of another set of neurons. Assuming learning is done by a Hebbian learning rule and the first set is directly stimulated by the environment, the second set of neurons, not being directly stimulated by the environment, must fire spontaneously to allow synaptic strength to them to grow.

Relatively coarse point models can be used to simulate this behaviour, exploring spontaneous activation and model Hebbian learning rules. As a bonus, these models can be applied to standard machine learning problems. A simulated neuro-biological model that categorises yeast data is described in the remainder of the paper.

Background

Modelling the brain has a long history, and both neurons and learning have been modelled. Computational models of neurons can be broadly grouped into compartmental models and point models. Compartmental models break the neuron into 3D parts, and then use models of electrical conductance to dynamically determine the neuron's electrical properties. An early example of these is the Hodgkin—Huxley model (Hodgkin & Huxley, 1952). Almost without exception, these models are a more accurate reflection of biology than are point models, but they are expensive to simulate.

Point models are widely used, with integrate and fire models (McCulloch & Pitts, 1943) being an early and simple model. These are also used in the Hopfield model (Hopfield, 1982). A widely used extension includes leak (Amit, 1989), leading onto the fatiguing leaky integrate and fire (FLIF) model described below.

Boltzmann machines are another type of point model (Ackley, Hinton, & Sejnowski, 1985). They fire stochastically, but without input they fire on a regular basis. Increasing input increases their firing rate. The FLIF model approximates this behaviour without the stochasticity.

Many learning rules are Hebbian, and in all cases they are local rules based on the behaviour of adjacent neurons. Biologically, learning appears to be Hebbian, with the spike timed dependent plasticity (STDP) having solid biological support (Bi & Poo, 1998). Hebbian learning and variants have proved to be quite powerful. For instance, it is possible to separate independent and principal components with local rules (Fyfe, 2005). Some form of decorrelation is necessary so that all of the neurons do not become highly connected, and Fyfe uses anti-Hebbian rules to decorrelate the behaviour of co-firing neurons.

The standard Kohonen learning rule (Kohonen, 1997), which is not Hebbian, uses a ''Mexican Hat'' function to move neurons away from each other, decorrelating them. STDP also decorrelates neurons when the pre-synaptic neuron fires after the post-synaptic neuron. This has been used in spiking models to duplicate the standard Kohonen map (Rumbell, 2012).

Model

Broadly speaking, the components of the model used in the simulation are the neural model, the learning algorithm, and the topology (the way the neurons are connected). The system can be found at http://www.cwa.mdx.ac.uk/chris/hebb/yeast/yeast.html.

FLIF model

The FLIF neural model is a point model, which is in the family of integrate and fire (McCulloch & Pitts, 1943) models, with two dynamic variables, activation A and fatigue F. An integrate and fire model is described by Eq. (1). The neuron integrates activity from other neurons that fired in the last cycle (V_i) weighted by the synaptic strength w_{ij}. It fires if activity surpasses a threshold θ.

$$\theta < A_j = \sum_{i \in V_i} w_{ij} \tag{1}$$

The model is discrete and runs in cycles that roughly correspond to 10 ms. of time. It is leaky as described by Eq. (2), so if it does not fire in one cycle, it retains some of the activation for the next cycle. Without input, activation decays from step $t-1$ to step t, being divided by a constant $D > 1$.

$$A_j^t = A_j^{t-1}/D \tag{2}$$

The neuron also fatigues each step it fires. Each neuron has a fatigue value that is increased by a constant F_c each step it fires. The neuron's fatigue is added to the threshold, producing a dynamic threshold where neurons that frequently fire require more activation to fire.

When a neuron does not fire, its fatigue is reduced. In earlier models this was reduced by a constant F_r in each step that the neuron did not fire, but the fatigue value, F, never went below zero. The new version allows fatigue to be negative. When a neuron is hypo-fatigued, it will fire when fatigue is negative enough ($-F > \theta$). In this model, if the neuron fired and fatigue was less than $-.25$, the fatigue was halved as described in Eq. (3). Otherwise, it was increased by F_c as usual.

$$F_i^{t+1} = F_i^t/2 \tag{3}$$

This leaves four parameters to describe the neural model. Threshold θ is 2.2; decay D is 1.12; fatigue increase F_c is 0.45; and fatigue recovery F_r is 0.01. In past simulations, these were free parameters for simulation, but these values have been selected to fit the firing behaviour to biological neurons. The particular neurons modelled were rat somatosensory neurons under a widely varying direct current injection regime. Similarly, the fatigue rule's, inclusion led to a closer fit to the biological firing behaviour (Huyck & Parvizi, 2012). Fit to neural spiking behaviour is over 90% with an average difference of less than two cycles (17 ms.).

In addition to an improved fit to neural data, this provides a mechanism for neurons not directly stimulated to fire. For neurons to be included in a Hebbian circuit, they need to fire. In earlier simulations, neurons were selected randomly to fire spontaneously (Huyck & Bowles, 2004) to enable them to be included in the circuit. In this neural model, spontaneous firing emerges from fatigue, and thus is theoretically simpler. This spontaneous firing is related to Boltzmann machines (Ackley et al., 1985).

Compensatory learning

The learning mechanism is another component of the model. While evidence points to biological modification of synaptic weights being Hebbian, this leaves a vast range of possible rules. The simulations below use a compensatory learning rule. In addition to the firing behaviour of the two neurons a synapse connects, a compensatory rule takes into account the total weight of the synapses in these neurons, forcing the total weight toward a target total in conjunction with the firing behaviour. The authors have used a compensatory rule based on the total weight from the pre-synaptic neuron's synapses in earlier work to learn hierarchical categories (Huyck, 2007). The simulations described below make use of two variations of the compensatory rule, one based on the total weight to the post-synaptic neuron, and other prior rule based on the total weight from the pre-synaptic neuron; these are termed post-compensatory and pre-compensatory.

Hebbian rules are typically a combination of a rule for neurons co-firing, and one for when they do not. Eq. (4) is used when the neurons co-fire, and Eq. (5) when the pre-synaptic neuron fires and the post-synaptic neuron does not. When the pre-synaptic neuron does not fire, the weights do not change.

$$\Delta_+ w_{ij} = R * T[(1 - w_{ij}) * 10^{(W_B - W_j)}] \tag{4}$$

$$\Delta_- w_{ij} = -R * T[w_{ij} * -10^{(W_j - W_B)}] \tag{5}$$

In Eqs. (4) and (5), R is the learning rate, which is 0.01 in the simulations below. W_B is the neuron's target synaptic weight and W_j is the neuron's total synaptic weight. T is a threshold function capping its output at 1. Thus a synaptic weight can change by at most the learning rate in a given cycle, and this is often the case in the beginning of a simulation as the initial synaptic weights are low. The synaptic weights are restricted to have values between 0 and 1.

Topology

The topology is divided into two subnets (see Fig. 1). The Input subnet acts as a proxy for environmental stimulus. The SOM subnet acts as a self-organising map, though it differs from a Kohonen net (Kohonen, 1997). Each neuron in both subnets is excitatory. The Input subnet has no internal connections; its neurons do not fatigue and thus fire each time they are externally stimulated, and only when they are externally stimulated.

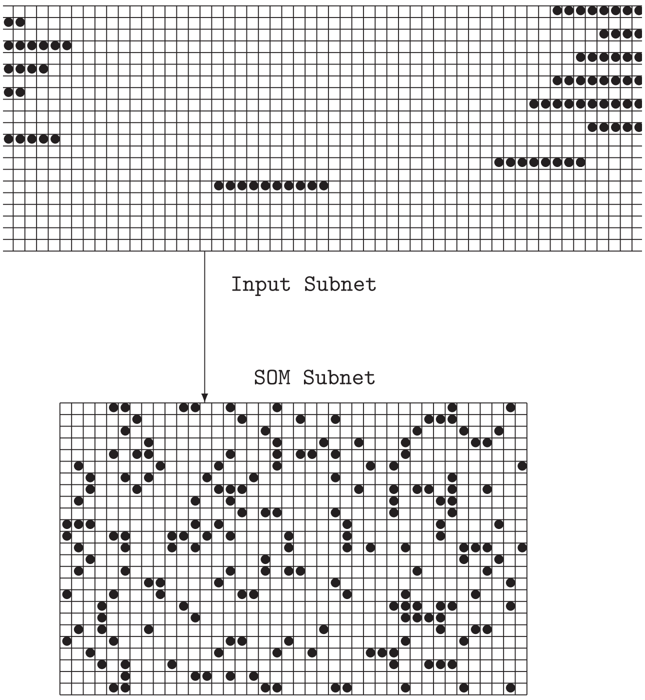

Fig. 1 Biological neural net: circles refer to firing neurons. The arrow represents all of the synapses from the Input to the SOM subnet. The Input subnet represents 8 numerical inputs. The SOM subnet represents one cycle of firing, with many neurons representing this particular category. Several of those neurons will also be part of the representation of other categories.

Each neuron in the Input subnet has 10 connections to the SOM subnet. All connections are randomly assigned without duplication or (in the case of the SOM subnet) self-connections. Within the SOM subnet, each neuron has 20 connections.

Pre-compensatory learning is applied to synapses leaving the Input subnet; W_B is 5. Synapses between neurons in the SOM subnet change by post-compensatory learning; W_B is 1.

The Input subnet has 1080 neurons. The SOM subnet has 1000 neurons. This topology (size of input aside), learning forms and target weights, duplicate an earlier simulation for car data (Mitchell & Huyck, 2013).

Experiment

The authors have been interested in applying these types of biological neural networks as a solution to machine learning problems for some time. Networks where all neurons were stimulated could effectively learn categories (Huyck & Orengo, 2005). However, these networks had problems with categories that were not linearly separable, and could not easily memorise large numbers of items.

In other earlier work (Cairns, Huyck, Mitchell, & Wu, 2001), the authors (and colleagues) applied a range of machine learning techniques to the yeast categorisation task from the University of California at Irvine's Machine Learning Repository (Asuncion & Newman, 2007). Recently, the authors applied this compensatory solution to a relatively large car categorisation task from the Repository. A straight forward translation was made to apply this type of network to the yeast problem.

The initial solution was directly derived from the earlier car categorisation problem. The car problem had discrete input features, while the yeast problem consisted of a set 8 input features with values between 0 and 1, with two digits of precision. There were 10 categories.

The input was encoded in a relatively obvious way. Each input feature was given 110 neurons. When an item was presented, the 10 adjacent neurons to the value were stimulated. For example, if the first feature had a value of .01, the 2nd to 11th neuron were stimulated, and if it had a value of .99 the 100th to 109th neuron were stimulated. During training, each category was also stimulated with each being represented by 20 neurons.

The SOM subnet contained 1000 neurons. Each training and test item was presented for 40 cycles, and the net allowed to run for a further 35, providing a 75 cycle epoch. The first phase of training was 20,000 cycles; this phase set the synaptic weights. During the second phase, all training items were presented; the firing behaviour was used as a basis for comparison for categorisation. This all followed the earlier car categorisation work (Mitchell & Huyck, 2013).

During the first phase of training, the model was allowed to run freely, with the appropriate change of training input for each new epoch. The idea was that 40 (10 ms) cycles of input was roughly equivalent to 400 ms of physical time, and thus was sufficient to activate a concept; however, the activity did not persist, and thus these networks do not contain Cell Assemblies (see concluding section). The 75 cycle epoch provides time for neurons to fire spontaneously enabling their synaptic strengths to increase via Hebbian learning, and become part of the classification mechanism. The Input subnet's learning rate is multiplied by 0.7 every 5000 cycles.

During the second phase of training, learning was turned off, the model was reset at the beginning of each epoch, and firing behaviour was recorded. In a given test of the entire network, 268 items were presented during the first phase and 1335 items were presented during the second phase. Learning during the second phase led to degraded (though reasonable) performance. All 8 input and 1 output feature were presented during both training phases. During the second training phase, reset zeroed each neuron's activation and fatigue at the beginning of each epoch; this made the initial state at each epoch in the second training phase roughly equivalent. The 1000 dimensional firing vector provided the bases for categorising.

The testing phase was similar to the second training phase. The test items were presented without the output feature, and the firing behaviour and the category were recorded. There was a reset at the beginning of each epoch.

Categorisation was done by comparing the firing pattern of the test item with the training items from phase 2 using a Pearson's Product Moment Correlation. The category of the training item with the closest Pearson's Correlation was selected.

Following the earlier machine learning work (Cairns et al., 2001), a 10-fold test was used. A particular net was trained on 90% of the data and tested on 10%. This was repeated 10 times, each with a new net, to get a complete 10-fold test.

Table 1 10-fold categorisation results.

Algorithm	Result (%)
Growing cell structure	55
MLP with Backprop	57
Genetic algorithms	55
Expanding range rules	56
FLIF neurons	52.81
SOM	56.00

Results from earlier systems, shown in Table 1, show that it is a difficult task. On 100 complete 10 fold tests, the average performance was 52.18% of correct categorisation. A slightly different topology led to a slightly better result with a high of 55.19% on one 10 fold test. This is below the supervised algorithms, but quite near them. It is also below the SOM performance, despite having the FLIF network perform better on the earlier car data.

In Cairns et al. (2001) Kohonen's SOM was not used and hence is included here. The parameters of the Kohonen's model used are as follows: $10 * 10$ hexagonal grid; neighbourhood set at 2/3 reducing to a single node; and the learning rate, α, initialised at 0.05 reducing to 0.01 over 1000 iterations. The Kohonen SOM was exposed to the same 10-fold test on yeast and the results are the mean of training and testing five times.

Discussion and conclusion

The FLIF neural model is a reasonably accurate biological model. It is efficient to simulate, mostly because it has a 10 ms step while most other models simulate either continuously or with a 1 or .1 ms step. The emergent spontaneous activity allows neurons that are not directly stimulated by the environment to be incorporated into neural circuits.

The compensatory learning mechanism is biologically motivated. It merely limits the strength of synapses from (in the pre-compensatory case) or to (in the post-compensatory case) a neuron. In this particular topology, it rapidly increases strength from the initial neurons. The SOM neurons also increase in strength even when they initially receive little activity because the post-compensatory rule spreads the synaptic strength to neurons that do not have much, thus increasing their participation in the circuits.

The resulting FLIF topology yields a self-organising system that has several neurons representing a state. In Kohonen's SOM, one unit represents a state. The multi-unit representation allows more states to be remembered. The Pearson measurement is one way to separate the states, but a neural mechanism will be needed for an eventual biological neural cognitive architecture.

This simulation has separated pre and post-compensatory learning, but there is no reason that a synapse cannot use both components, or that this could change over time in response either to age or some other feedback. Moreover, compensatory mechanisms can be combined with STDP.

One of the main goals of the authors' work in this broader area is that the neurons will form Hebbian Cell Assemblies, (Hebb, 1949) that persist after stimulation. These Assemblies provide a link between psychological short and long-term memory and neural behaviour, but this simulation does not exhibit Cell Assemblies.

Obviously, the work is far from complete. The topology is simple and the number of neurons is very small. However the compensatory mechanism could reasonably be a component of a complete simulated biological neural cognitive architecture.

References

Ackley, D., Hinton, G., & Sejnowski, T. (1985). A learning algorithm for boltzmann machines. *Cognitive Science, 9*, 147–169.

Amit, D. (1989). *Modelling brain function: The world of attractor neural networks*. Cambridge University Press.

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Bi, G., & Poo, M. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience, 18*(24), 10464–10472.

Cairns, P., Huyck, C., Mitchell, I., & Wu, W. (2001). Comparison of categorisation algorithms for predicting the cellular localization sites of proteins. *Knowledge and Information Systems: An International Journal, 12*, 296–300.

Fyfe, C. (2005). *Hebbian learning and negative feedback networks*. Springer.

Hebb, D. (1949). *The organization of behavior: A neuropsychological theory*. Wiley & Sons.

Hodgkin, A., & Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology, 117*, 500–544.

Hopfield, J. (1982). Neural nets and physical systems with emergent collective computational abilities. *PNAS, 79*, 2554–2558.

Huyck, C. (2007). Creating hierarchical categories using cell assemblies. *Connection Science, 19*(1), 1–24.

Huyck, C., & Bowles, R. (2004). Spontaneous neural firing in biological and artificial neural systems. *Journal of Cognitive Systems, 6*(1), 31–40.

Huyck, C., & Orengo, V. (2005). Information retrieval and categorisation using a cell assembly network. *Neural Computing and Applications, 14*, 282–289.

Huyck, C., & Parvizi, A. (2012). Parameter values and fatigue mechanisms for FLIF neurons. *Journal of Systemics, Cybernetics and Informatics, 10*(4), 80–86.

Kohonen, T. (1997). *Self-organizing maps*. Springers.

Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience, 7*, 153–160.

McCulloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115–133.

Mitchell, I., & Huyck, C. (2013). Self organising maps with a point neuron model. In *17th international conference on cognitive and neural systems*.

Rumbell, T. (2012). Self organisation and hierarchical concept representation in networks of spiking neurons. Ph.D. thesis, Plymouth University, UK.