On modelling biological neurons with artificial neural networks

C. Coomber, P. Horan, R. Dew

email: coomber@deakin.edu.au

School of Computing & Mathematics

Deakin University

Victoria

Australia, 3217

Abstract

The first few pages of any good introductory book on neurocomputing contain a cursory description of neurophysiology and how it has been abstracted to form the basis of artificial neural networks as we know them today. In particular, artificial neurons simplify considerably the behavior of their biological counterparts. It is our view that in order to gain a better understanding of how biological systems learn and remember it is necessary to have accurate models on which to base computerized experimentation. In this paper we describe an artificial neuron that is more realistic than most other models used currently. The model is based on conventional artificial neural networks (and is easily computerized) and is currently being used in our investigations into learning and memory.

keywords: artificial neural networks, brain modelling, artificial neurons

Introduction - neurophysiology to neurocomputing

The human brain is the focus of much inter-disciplinary scientific research. In particular, researchers from the artificial intelligence (AI) community are actively involved in trying to automate the remarkable abilities of the brain, such as memory, learning, and intelligence. AI has many streams, one of which is neurocomputing. A distinctive feature of neurocomputing is that it approaches the questions we have about the brain by emulating its neurophysiology. That is, neurocomputing is concerned with automating models of brain structure and function in an effort to replicate the abilities of the brain.

Neurocomputing is built around several key abstractions of neurophysiology, in particular the abstractions of the neuron and the synapse. The neuron, which is the basic cellular component of the brain, is unlike other cells in the body because it can integrate electrical potentials across its cellular membrane and can emit regenerative impulses termed action potentials (of constant amplitude) when a critical membrane potential (the threshold) is exceeded. When a neuron is stimulated by other neurons, it is either moved closer to its threshold (excitation) or moved away (inhibition). The information transmitted by a neuron is frequency-rate encoded.

The neuron itself has three main parts: the soma, the dendrites, and the axon. The soma, or cell body, contains all the biological machinery necessary for the functioning of the neuron. The dendrites are fine, branch-like ramifications that are the principal regions for interneuronal contact. They integrate afferent, graded electrical potentials and carry the resulting electrical potential to the cell membrane. The axon carries electrical impulses away from the cell body and generally branches into many fibres that terminate at the dendrites, axons or cell bodies of other neurons. The points at which the axonal fibres of one neuron meet parts of another neuron are called synapses, which may be electrical or chemical. An electrical synapse is formed when a neuron is in electrical contact with another neuron. A chemical synapse is formed when a neuron transmits a chemical substance (a neurotransmitter) to another neuron. When an electrical impulse reaches the presynaptic terminal of an axon, a measure of neurotransmitter is released into the cleft between the presynaptic terminal and the postsynaptic neuron. The neurotransmitter diffuses across the cleft to the postsynaptic neuron, affecting the ionic permeability of the cell membrane and produces an excitory or inhibitory response.

This view of neurophysiology is the inspiration for the ‘artificial neural networks’ used in neurocomputing. A typical artificial neural network consists of a number of simple, interconnected artificial neurons (or units) that function in parallel and are structured in layers. The connections between these units are weighted to represent synaptic strength. A unit produces output by subtracting a threshold value from the weighted algebraic (linear) sum of its inputs and then passes this result through a transfer function to yield the output. The artificial neural network learns by making appropriate adjustments to the connection weights [Hinton, 1992].

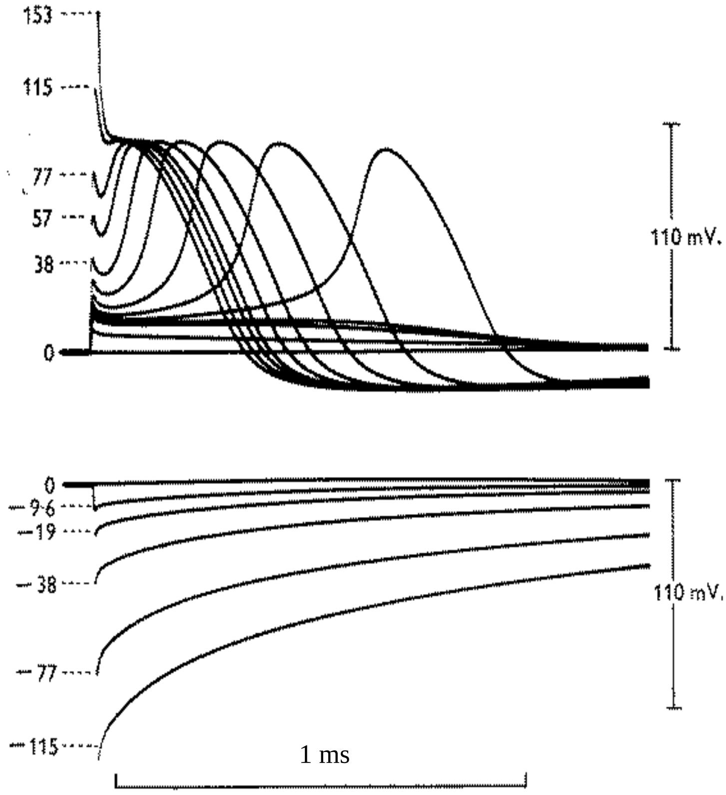

Figure 1. Time course of membrane potential for a particular axon following a short shock at 23˚ C. Adapted from [Hodgkin, Huxley and Katz, 1952]

Real and artificial neurons

Hodgkin, Huxley and Katz (1952) conducted a range of experiments with the squid giant axon in an effort to determine the laws that govern movements of ions through nerve cells during electrical activity. The experimental method they devised employs an internal electrode made up of two fine wires, a guard system for measuring membrane current, and a feed-back amplifier for clamping the membrane potential at a desired level.

In the early stages of the work it was important to prove that the membrane under investigation was capable of generating a normal action potential. This was achieved by applying a brief shock (in mμcoulomb/cm^2) to one internal electrode and recording the change in membrane potential with the other electrode. Typical results are shown in Fig. 1. The numbers attached to the records indicate the shock applied. Observe that if the membrane potential is displaced above the threshold level (about 15 mV above the resting potential of -70 mV) an action potential is produced, the main part of which takes about 1 ms. During this time the membrane enters an absolute refractory state during which no stimulus, however strong, can evoke a second action potential. All the action potentials have the same characteristic, non-linear shape and have a maximum amplitude of about 100 mV. If the membrane potential is displaced to the threshold level it might remain in a state of unstable equilibrium for some time. If the depolarization is less than the threshold level it is followed by a subthreshold response similar to that seen in most excitable tissues. If no further shock is applied, the membrane potential will settle at the resting potential V_r.

From experimental records like that in Fig. 1 it is possible to determine the relation between shock (presynaptic stimulus) and membrane potential (postsynaptic response). For instance, if we just consider depolarizing shocks, the relation has an “S” or sigmoidal shape [Cooke and Lipkin, 1972]. In neurocomputing, transfer functions are often given this sort of shape, which lends a degree of biological plausibility to artificial neurons (units) but still understates greatly the behavioral complexities of real neurons. In particular, the time course of neural behavior is overlooked. By oversimplifying things we are likely to discard important mechanisms and structures that are vital for learning and memory.

Experimenting with computerized models of biological neural systems can help us gain a better understanding of how learning and memory operates. The more realistic the models are (within reason since overly complex models present performance problems) the better [Clery, 1992; Sejnowski and Churchland, 1992]. Our research on neural networks [Lim, Horan, and Jarvis, 1992] and simulation models [Coomber, 1993] is directed toward this end.

A neural model based on traditional artificial neural networks

Overview

We have designed an artificial neuron that embodies the chief characteristics of biological neurons. It responds to stimuli by changing its membrane potential over time in the same manner as a real neuron. The neural model is unique in that it is built from traditional artificial neural networks that have been trained on data obtained from experiments with squid giant axons kept at around 20° C.

The experimental temperature is important because the rate at which membrane potential changes with time is increased about threefold for a rise of 10° C [Hodgkin and Katz, 1949]. In other words, an experimental record such as that in Fig. 1 would basically be stretched left or right depending on the temperature at which the experiment was conducted. By choosing data from a particular temperature range we simplify the architecture of the artificial neural networks because they do not have to deal with translational effects.

The artificial neuron has five input parameters and one output parameter (Fig. 2). The input parameters are V_s (electrical stimulus or synaptic shock), V_r (resting potential), V_m (current membrane potential), threshold (the potential at which the neuron gives an action potential), and t (time). The output parameter V is the new membrane potential at time t. The time t may be varied in arbitrary fractions of a millisecond. All the other parameters are measured in millivolts. In addition, the artificial neuron has an absolute refractory period of 1 millisecond. However, this may be varied as required.

Architecture of the artificial neuron

The artificial neuron is made up of four conventional artificial neural networks and a control routine that governs which networks are to be activated at a given time. These components are illustrated in Fig. 3. Two of the neural networks, the *translation net* and *action_potential net*, operate together to determine the changes in membrane potential as an action potential evolves over time. The other two neural networks, the *Vr_threshold net* and the *sub_Vr net*, are used to determine membrane potential for below threshold but above resting potential stimuli, and below resting potential stimuli, respectively.

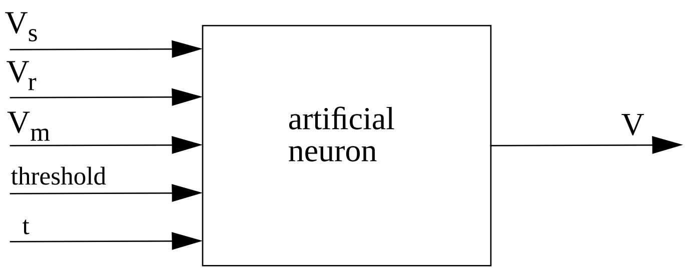

Figure 2. Inputs and outputs of the artificial neuron. V_s=synaptic shock (mV); V_r=resting potential (mV); V_m=current membrane potential; t=time (ms); threshold (mV); V=membrane potential at time t.

V_s	synaptic shock (mV)
V_r	actual resting potential (mV) of neuron;
V_m	current membrane potential (mV) relative to V_r.
t	time in milliseconds
threshold	millivolts above V_r at which neuron produces an action potential
refractory_t	absolute refractory time (usually 1ms)
state	1 = action potential; 2 = resting potential to threshold; 3 = below resting potential

```
main
        S := 0
        refractory_t := 0
        state := 3
        t := 0
        V_r := -70 “resting potential”
        V_m := V_r
        threshold := 15 “assume threshold is 15 mV”
        while(true)
          V_s := input() “see if synaptic shock applied”
          if (refractory_t > 0)
            refractory_t := refractory_t - Δt
            call stabilize(S, t, state)
          else
              if (V_s <> 0) “process stimuli”
                S := V_s + V_m
                call synaptic_shock(S, t, state, threshold, refractory_t, V)
              else
                call stabilize(S, t, state, threshold)
          t := t + Δt “next time unit”
          V_m := V + V_r

synaptic_shock(S, t, state, threshold, refractory_t, V)
        If (S > threshold)
            state := 1
            S := translation.net(S) “get relative shift”
            V := action_potential.net(S + t)
            refractory_t := 1 “absolute refractory period of 1 ms”
        else
            If (S >= 0)
                state := 2
                V := V_r_threshold.net(S, t, threshold)
            else
                state := 3
                V := sub_V_r.net(S, t)

stabilize(S, t, state, V, threshold)
        If (state = 1) “evolve action potential”
            V :=  action_potential.net(S + t)
        else
            If (state = 2)
                V := V_r_threshold.net(S, t, threshold)
            else
                V := sub_V_r(S, t)
```

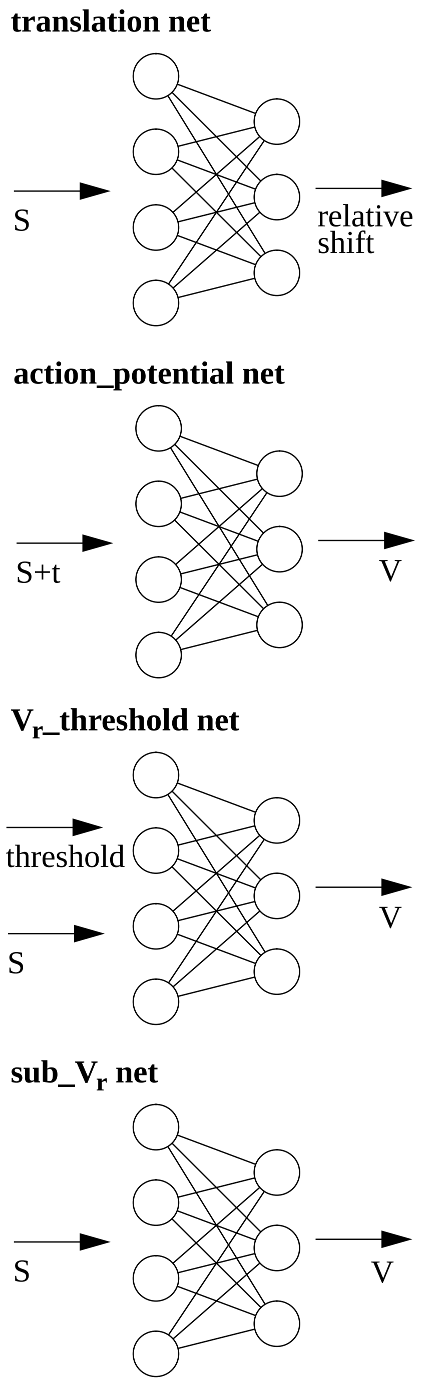

Figure 3. Components of the artificial neuron. The *main* routine calls either *synaptic_-shock* or *stabilize* to change membrane potential over time. These routines execute appropriate artificial neural nets (right).

In the early stages of our work we concentrated on designing artificial neural networks to reproduce the time courses of membrane potential produced by suprathreshold stimuli. The neural networks we experimented with had at most two hidden layers and were either feedforward or recurrent nets. Also, we used either vanilla backpropagation or backpropagation with momentum as the learning algorithm. The first layer of any network had no more than 20 units and, if it had a second layer, this had up to 8 units. The training set had approximately 10000 examples in it. After experimenting with various ways of scaling the training data and also trying different network configurations we could only manage to reduce the average error between expected and desired outputs to about 0.02, which was unsatisfactory for our purpose. Training times were also exorbitant and reached up to one week (on a SPARC SLC station). With more hidden units it would have been possible to train the network to yield an acceptable error value, but the resulting network would have been too large and inefficient. The trouble with training can be explained by the fact that the data set the network was trying to learn is nonmonotonic - a characteristic with which backpropagation networks have difficulty [Caudill, 1993]. To overcome this problem we used a different approach that took more advantage of the symmetric nature of the training data. The *translation net* and the *action_potential net* were devised subsequently.

The translation net - using the symmetric nature of the training data

As illustrated in Fig. 1, the rate at which an action potential rises is dependent on the strength of the shock applied to the membrane. The stronger the shock, the quicker the action potential reaches its peak amplitude. If we assume that depolarizing shocks are no greater than 100 mV relative to V_r then all action potentials produced by shocks in this range will have the same general shape. If we now plot the time course of the action potential produced by the shock just over the threshold value and treat it as a reference curve then all other action potentials can be shifted right and overlaid on it. There is a perfect fit, except for slight inaccuracies at the beginning of some action potentials.

The *translation net* accepts a suprathreshold stimulus up to 100 mV and produces the relative shift required to map to the reference curve to the time course of the action potential initiated by the stimulus. It is then a simple matter to determine the membrane potential at a given time t by adding the relative shift to t and reading off the potential from the reference curve. This is the job of the *action_potential net*.

The translation net has a feedforward architecture and has one input unit, three hidden units, and one output unit. All of the units have hyperbolic tangents as transfer functions. The training set consists of about 50 mappings of the form (shock $\longrightarrow$ relative shift). Shocks are scaled in the range [-1,+1] and relative shifts in the range [-0.7, +0.7]. The artificial neural network was trained until the average error between expected and desired outputs was less than 0.001 and took approximately one hour to do so.

The action_potential net - producing regenerative behavior

The *action_potential net* encodes the time course of an action potential that is used as a reference curve for determining the membrane potential evoked by a given suprathreshold stimulus (input to the *translation net*) at time t. This network takes as input the relative shift from the *translation net* added to the time t at which a membrane potential reading is required. The network outputs the membrane potential V for time t.

The *action_potential net* has a feedforward architecture with one hidden layer that has six units. The training set comprises 1000 associations of the form (time $\longrightarrow$ membrane potential), for which time, normally in the range 0 to 10 ms, is scaled to [-1, +1], and membrane potential, normally from -30 mV to 100 mV relative to V_r is scaled to the range [-0.7, +0.7]. The network was trained until the error between the expected and desired outputs was less than 0.005. This took about half a day. We used backpropagation with a momentum value of 0.1 and a learning rate of about 0.05. Fig. 4 shows the time course of the action potential (reference curve) produced by the network.

The sub_Vr net - sub-resting potential responses

If the artificial neuron receives a stimulus V_s that hyperpolarizes the membrane below the resting potential then the *sub_Vr net* comes into play. This net encodes the time courses of membrane potential (capacity discharge curves) taken by five different stimuli below resting potential ranging uniformly from 0 to -80 millivolts. The training set consists of 5000 mappings of the form (time, shock) $\longrightarrow$ membrane potential. The network has a feedforward architecture with two input

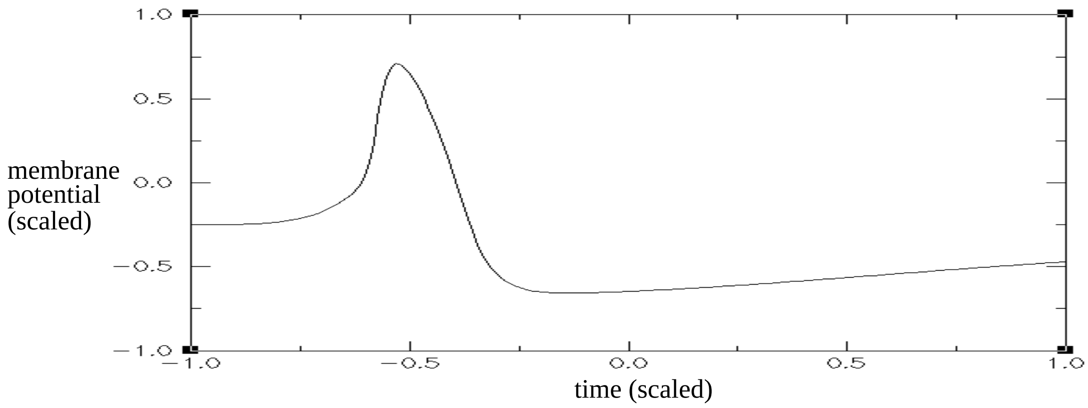

Figure 4. The action potential (reference curve) produced by the action_potential net.

units, two hidden layers of six and four units respectively, and one output unit. The components of the input vector are scaled to fit in the range [-1, +1] and the output is scaled in the range [-0.7, +0.7].

Unlike trying to train a network to learn the time courses of membrane potential followed by several suprathreshold stimuli, it was relatively easy to train a network to learn the time courses of several below resting potential stimuli. This was because the membrane potential of each time course increases monotonically (exponential in shape) and is approximately linear beyond the one millisecond mark. It was possible to train the *sub-Vr net* down to an error of 0.002 in about 24 hours. To test the generalization capabilities of the network we compared, for several stimuli not in the training set, a number of time courses of membrane potential produced by the network with actual time course records obtained experimentally from squid giant axons. The network generalized very well and had a nominal error.

The Vr_threshold net

To handle subthreshold but above resting potential stimuli we have designed the *Vr_threshold net*. This is a feedforward network with two hidden layers that have eight and four units, respectively. The network has one output unit (millivolts) and three input units (threshold, time, and stimulus). Stimuli around the threshold level produce time courses of membrane potential that indicate the artificial neuron is in a state of unstable equilibrium (on the edge of giving an action potential). Other stimuli have time courses that quickly taper off and reestablish the resting potential. Again, this behavior is shown clearly in Fig. 1. The training set for the *Vr_threshold net* has 8000 cases, scaled in the same manner as the training data for the other artificial neural networks. It took approximately 24 hours to train the network to an error of 0.002.

Coordinating the actions of the artificial neural networks

The coordinated action of the four artificial neural networks is controlled by the *main* routine (Fig 3). This either passes control to the *synaptic_shock* or the *stabilize* function depending on the presence of a synaptic stimulus V_s. Whenever V_s is not equal to zero, *synaptic_shock* is called, otherwise *stabilize* is called to bring the artificial neuron to resting potential. Notice that during the first millisecond of an action potential the artificial neuron is refractory and will not be affected by other stimuli during this period. At any point in time the artificial neuron will be in one of three states; either producing an action potential (state 1), undergoing a subthreshold but above resting potential response (state 2), or a below resting potential response (state 3).

Networks of artificial neurons

Using an object-oriented approach, we represent an artificial neuron by an instance of the class NEURON which defines the attributes (V_s, V_r, V_m, threshold, t, state, refractory) and implements the message NEXT_POTENTIAL. If this message is sent to an instance of NEURON, along with appropriate parameters, the receiving object will return its membrane potential for the next time unit. The message encodes the control routine shown in Fig. 3 and makes calls to the four conventional artificial neural networks described above.

Conclusion - What next?

We have described an artificial neuron that behaves essentially like a real neuron in so far as the time course of membrane potential is concerned. However, much more is to be done. In the next stage of our research we will investigate realistic synaptic models, and then (perhaps the most challenging part) we will focus our attention on ways of making networks of our artificial neuron learn and store information. Recent advances in neuroscience, especially the evidence that indicates synaptic transmission may be bidirectional (i.e., retrograde transmission) [Jessell and Kandel, 1993] will influence our learning frameworks.

References

Caudill, M. (1993): Avoiding the Great Backpropagation Trap, *Neural Network Special Report*, AI Expert, pp. 23-29.

Clery, D. (1993): Chips with a life of their own. *New Scientist*, pp. 29-33, January.

Cooke, I. M., Lipkin, M. (1972): Cellular Neurophysiology: A source book. Holt, Rinehart and Winston, New York.

Coomber, C. (1993): An Object-Oriented Simulation Procedure for Automating Transformation Schema Execution, *The International Journal in Computer Simulation: Special Issue on Object-Oriented Simulation*, Ablex Publishing Corporation, NJ, (to be published early 1993).

Hinton, G. (1992): How Neural Networks Learn from Experience, *Scientific American*, May 1992, pp.105-109.

Hodgkin, A. L. and Katz, B. (1949): The effect of temperature on the electrical activity of the giant axon of the squid. *Journal of Physiology*, 109, pp. 240-249.

Hodgkin, A. L., Huxley, A. F., and Katz, B. (1952): Measurement of Current-Voltage Relations in the Membrane of the Giant Axon of Loligo, *Journal of Physiology*, 116, pp. 424-448.

Jessell, T. M., and Kandel, E. R. (1993): Synaptic Transmission: A Bidirectional and Self-Modifiable Form of Cell-Cell Communication, *Cell Neuron*, 72, pp. 1-30, January.

Lim, R., Horan, P., and Jarvis, R. A. (1992): Implementation of a Neural network model for control in grasping a moving target, *IEEE Region 10 International Conference on Computers, Communications and Automation Towards the 21st Century*, Melbourne,11-13 November.

Sejnowski, T. and Churchland, P. (1992): Silicon Brains. *APC*, pp. 131-139, November.