

Mathematical models of neuron cell and their applications.

Pongsak Suvanpong

Department of Biological Sciences, Macquarie University, NSW 2109, Australia

Abstract

This paper presents mathematical models of neuron and neural networks and their application in machine learning. Starting with the simple mathematical model of a neural cell introduced by McCulloch and Pitts. The idea of perceptron as a neural network and back-propagation learning algorithm. A successful used of back-propagation and multilayer perceptron to drive a vehicle ALVINN.

Introduction

A neuron or nerve cell is a type of cell found in nervous system of most animals. It processes inputs and provides appropriate output. A neuron cell consists mainly of soma, axon, dendrites and synapses. Soma is cell body part of a neuron cell where the nucleus lives. Axon, dendrites and synapse are extended part from the cell body. In a neuron cell there is only one axon, this is where the output of the neuron occurs. The dendrites, on the other hand, is where the inputs to the neuron cell occurs and there are many dendrites in a neuron cell. There are synapses at the end of axon and dendrites. Synapse is where the transmission of signal between neighboring neurons occurs. This connection between two neighboring neurons is called synaptic junction. Synapses releases substances called neurotransmitter when the neuron fires its action potential.

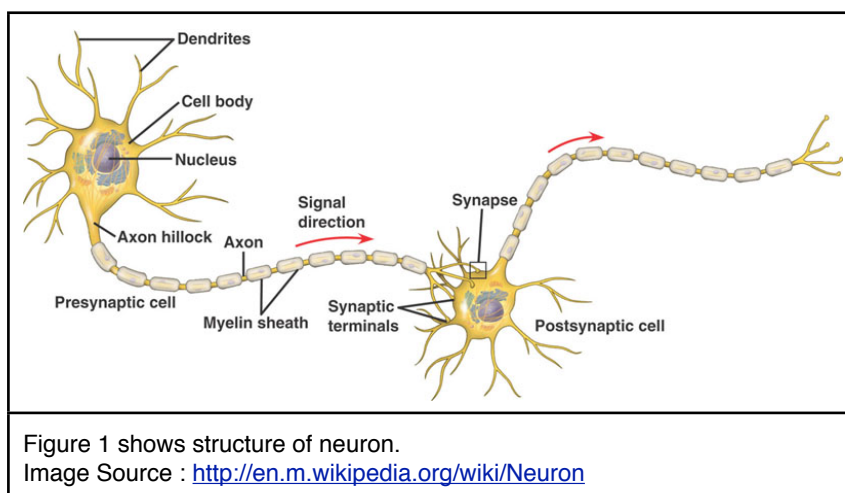


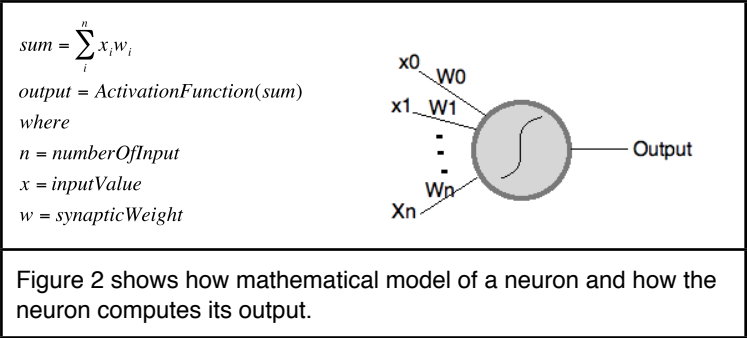
Figure 1 shows structure of neuron.

Image Source : <http://en.m.wikipedia.org/wiki/Neuron>

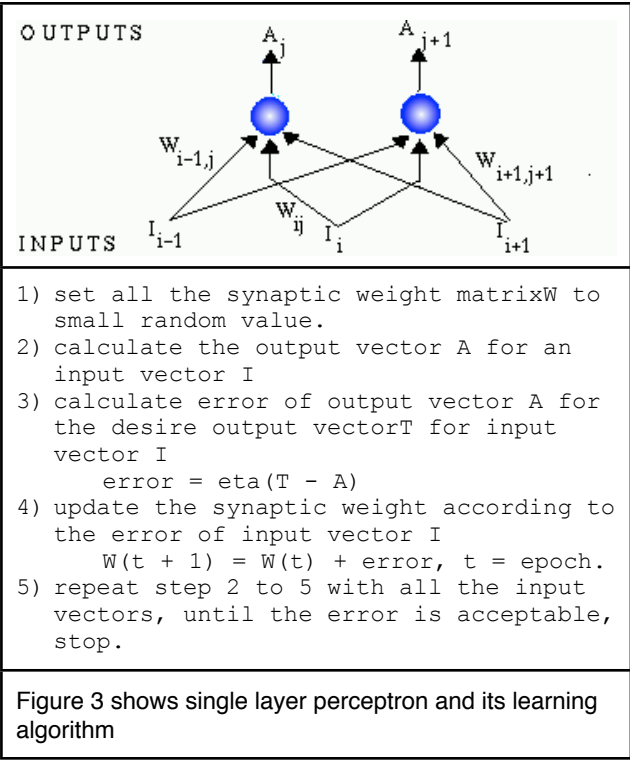
A neuron cell can be stimulated by electrical inputs and may or may not fire its action potential (Wickens, Wilson, 1988). Many neurons can form connections to do specific work, the formation is called a neural network. A neural networks, once formed, can encode informations, this process is called neural coding. The informations, once encoded, can also be decoded. The decoding process is sometimes called recalling and the encoding process is called training or learning.

Mathematical Models

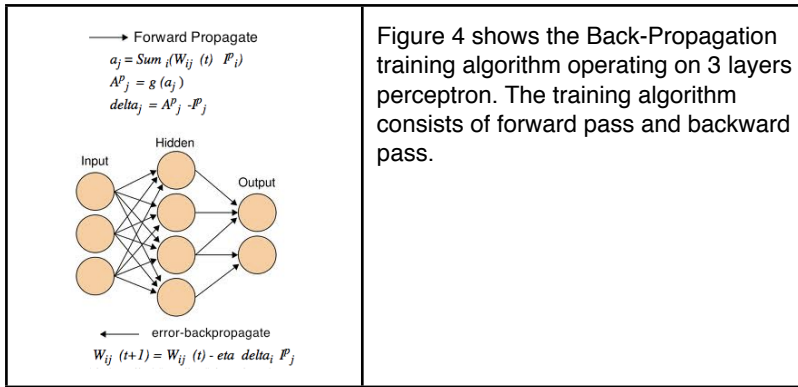
To understand the encoding and decoding process of biological neural networks, McCulloch and Pitts (1949) had developed a mathematical model of a neuron. The model describes a neuron as binary processing unit. The neuron is either fired and not fired its output signal and is controlled by a threshold logic or activation function. The model neuron, like its biological counter part, has number of inputs and an output. At each input to the neuron, there is a weight. The weight acts like an input synaptic resister. To calculate if a neuron will fire output signal, first the sum of multiplication between each input and the input's synaptic weight, is calculated. The activation function, then, calculated using the sum value from the first step. This process is illustrated in the figure 2.



A more complex model called perceptron (Minsky, Marvin, and Seymour, 1988) was later introduced to extend the simple one neuron model. The model introduced learning and uses group of neurons, and connects them together like layer. A single layer perceptron neural network can be trained to solve simple problem. The problem, however, must be linearly separable. A linear separable problem is a kind of problem where a straight line can be drawn to separate each classes of the problem space. For example logical AND, OR are linearly separable, but XOR is not. The Training algorithm is described in the figure 3.

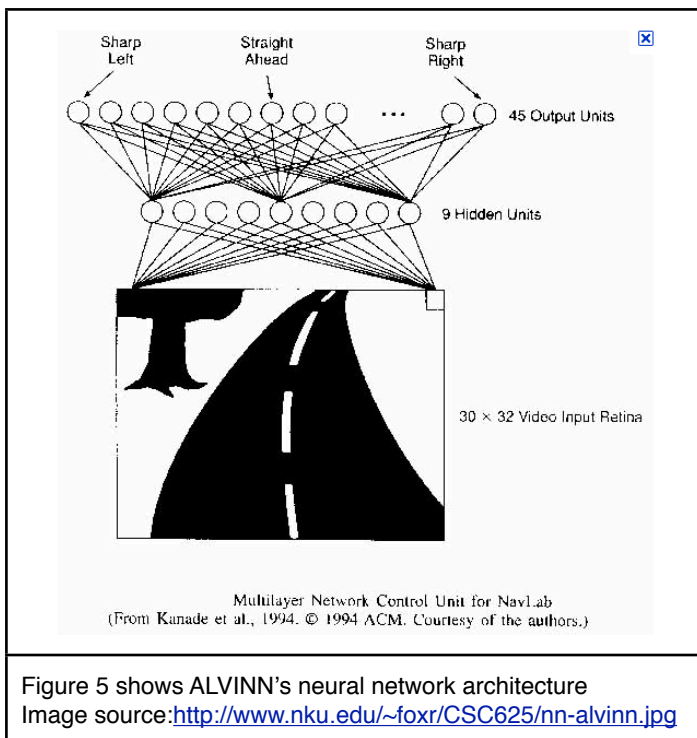


Back-propagation(Williams and Zipser, 1989) is a kind of supervised learning algorithm used in multilayer perceptron. To train a multilayer perceptron with Back-propagation algorithm, the perceptron must have at least 3 layers: input, hidden and output layer. There is only one input and output layer, but there can be more than one hidden layer. Back-propagation has 2 phases: feed-forward and error-backpropagate. The feed-forward propagates a input vector through the layers to produce the output vector. The root mean square error is, then, calculated between the perceptron output and the desire output for the input vector. The error-backpropagate propagates error back through the layers from output to hidden and input layer. In each layers and each neurons in the layer, the synaptic weights are updated. The process is repeated with all input vectors over and over until the perceptron is converged to the solution. This can be determined from the sum of the root mean square error of each input vector. Figure 4 illustrated the back-propagation training process.



ALVINN

ALVINN(Pomerleau, 1989 & Sukthankar, Pomerleau, Thorpe C, 1993) stands for an autonomous land vehicle in a neural network. The purpose of the project was to develop neural network based real-time controller to autonomously driving a vehicle. The neural network used in this research had 960 neurons at input layer. The input layer was fed with video image from a video camera attached at the front of the test vehicle. There were 45 neurons at the output layer. The output layer gave steering wheel position to the test vehicle. Figure 5 shows more details of the output layers and neural network's architecture.



The sample data that was used to train ALVINN, were generated using real human driver with simulated and real road images. A sensor at the test vehicle's steering wheel recorded the steering wheel position while a video camera recorded the video frames that associated with the turn angles. The back-propagation algorithm was used to train the neural network.

The results from this research were very encouraging. After training the neural net controller for a short amount of time, it could drive the car in variety of road conditions, both paved and unpaved, roads that have median dash line, solid dash line and no median line at all. The controller could drive the vehicle at maximum speed of 32 kilometers per hour.

Summary

There are about 100 billion neuron cells in human brain. Each neuron's processing unit, is very simple, but when group of neurons form a path way as neural network, the network can perform very complex tasks in lightning fast speed. Just thinking about your ability to distinguish most of your friends by their voices, or ability to understand senses and emotions from music. In a biological organism, when born, the brain has no preprogrammed knowledge, but as the organism starts learning process when it is interacting with surrounding environment. The neurons, then, start forming network to correctly responding to environment. The exact details of how this process is done, is still not known, but in spite of that, many have successfully used neural network to do complex task mimicking human.

References

1. Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford University Press, Oxford.
2. Bem T, Meyrand P, Branchereau P, Hallam J (2008). 'Multi-Stability and Pattern-Selection in Oscillatory Networks with Fast Inhibition and Electrical Synapses'. *PLoS ONE*, Vol.3, no 11, pp 30-383.
3. Freeman J. A. and Skapura D. M. (1992). *Neural Networks Algorithms, Applications, and Programming Techniques* Addison Wesley Press, Boston, MA
4. Gallant S. I. (1994). *Neural network learning and expert system*, MIT Press, Cambridge MA.
5. McCulloch W. S. and Pitts, W. H. (1943). 'A logical calculus of the ideas immanent in nervous activity.' *Bulletin of Mathematical Biophysics*, Vol. 5, pp.115-133.
6. Minsky H., Marvin K., and Seymour P. (1988). *Perceptrons*, 2nd edition, MIT Press, Cambridge MA.
7. Orchard G. A. and Phillips W. A. (1991). *Neural computation: A beginner's guide*, LEA publishers, Hillsdale USA.
8. Pomerleau, D.A. (1989). 'ALVINN: An autonomous land vehicle in neural network'. *Advance in Neural information processing system 1*, Vol. 12, no. 1, pp. 230-237.
9. Pomerleau, D.A. (1991). 'Efficient Training of Artificial Neural Networks for Autonomous Navigation.' *Neural Computation*. Vol. 3, no. 1, pp. 88-97.
10. Sukthankar R., Pomerleau D. A. and Thorpe C. (1993). 'Panacea: An Active Sensor Controller for the ALVINN Autonomous Driving System'. *Robotics*, Vol. 8, no.1, pp. 120-127.
11. Wickens J. R. and Wilson C. J. (1988). 'Regulation of Action-Potential Firing in Spiny Neurons of the Rat Neostriatum In Vivo'. *The Journal of Neurophysiology* Vol. 79, no. 5, pp. 2358-2364
12. Williams, R. J, and Zipser D. (1989). 'A Learning algorithm for continually running fully recurrent neural networks'. *Neural Computation*, Vol. 1, no. 3, pp. 270-280.