ISSN 2319 – 7757 EISSN 2319 –7765

Indian Journal of Engineering

Artificial neural network: all about artificial neural world

Vicky Ahuja, Vivek Anand, Varun Gandhi, Varsha Yadav

Department of Computer Science and Engineering, Dronacharya College of Engineering, Khentawas, Farukhnagar, Gurgaon, India

Received 28 August; accepted 20 October; published online 5 November; printed 30 November 2013

ABSTRACT

An Artificial Neural Network (ANN) is an information processing model that is encouraged by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Key Words: Hebbian Learning, Backpropagation Learning Rule, Perceptron learning rule

How to Cit this Article
Vicky Ahuja, Vivek Anand, Varun Gandhi, Varsha Yadav. Artificial neural network: all about artificial neural world. *Indian Journal of Engineering*, 2013, 6(15), 6-8

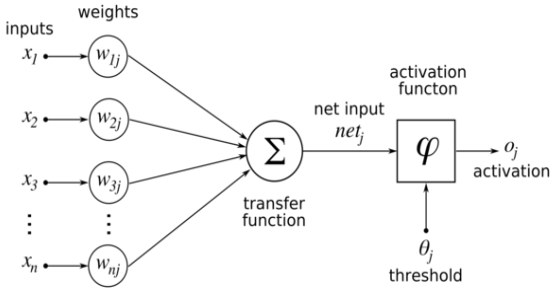
1. INTRODUCTION

Artificial neural networks are computational models enthused by animal central nervous systems (particularly the brain) that are capable of machine learning and pattern recognition. They are generally presented as systems of interconnected "neurons" that can compute outputs from inputs by feeding information through the network. For example, in a neural network for handwriting recognition, a set of input neurons may be activated by the pixels of an input image representing a letter or digit. The activations of these neurons are then passed on, weighted and transformed by some function determined by the network's designer, to other neurons, etc., until finally an output neuron is activated that determines which character was read. Like other machine learning methods, neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including computer vision and speech recognition.

2. LEARNING ALGORITHMS

Hebbian Learning Rule

In the Hebbian learning, weights between learning nodes are adjusted so that each weight better represents the relationship between the nodes. Nodes which tend to be positive or negative at the same time will have strong positive weights while those which tend to be opposite will have strong negative weights. Nodes that are uncorrelated will have weight near zero.



Vicky Ahuja,
Artificial neural network: all about artificial neural world,
Indian journal of engineering, 2013, 6(15), 6-8,

For Example: If two nodes x and y are often simultaneously active, Hebbian Learning will increase the connection strength between the two so that excitation of either one tends to cause excitation of the other. While, if nodes x and z were of opposite activations at all times, then Hebbian learning would gradually decrease the connection in between below zero so that an excited x and z would inhibit the other.

For this rule the learning signal is equal simple to the neuron's output.

We have

r ~ = f($W_i^t\ x$)

The increment ΔW_i = c f ($W_i^t\ x$)x

The single weight adjustment using the following increment:

ΔW_{ij} = c f ($W_i^t\ x$)x_j

This can be written as

ΔWi_j = c $o_i\ x_j$ for j= 1, 2, 3….n

Hebbian Learning Rule has four Features:

- It is unsupervised.
- It is a local learning rule which means that it can be applied to a network in parallel.
- It is simple and therefore requires very little computation.
- It is biologically plausible.

Perceptron learning rule

The Perceptron learning rule was originally developed by Frank Rosenblatt in the late1950s.Training patterns are presented to the network's inputs; the output is computed. Then the connection weights w_j are modified by an amount that is proportional to the product of

- The difference between the actual output, y, and the desired output, d, and
- The input pattern, x.

The algorithm is as follows:

- Initialize the weights and threshold to small random numbers.
- Present a vector x to the neuron inputs and calculate the output.
- Update the weights according to:

$$w_j(t+1) = w_j(t+1) + \eta\,(d-y)x$$

where

1. d is the desired output,
2. t is the iteration number, and
3. eta is the gain or step size, where 0.0 < n < 1.0

- Repeat steps 2 and 3 until:

1. the iteration error is less than a user-specified error threshold or
2. a predetermined number of iterations have been completed.

Notice that learning only occurs when an error is made, otherwise the weights are left unchanged.

This rule is thus a modified form of Hebb learning.

During training, it is often useful to measure the performance of the network as it attempts to find the optimal weight set. A common error measure or cost function used is sum-squared error. It is computed over all of the input vector/output vector pairs in the training set and is given by the equation below:

$$\mathbf{E} = \frac{1}{2} \sum_{i=1}^{n} \| y^i - d^i \| 2$$

Where p is the number of input/output vector pairs in the training set.

Widrow- Hoff Learning Rule

The Widrow-Hoff Learning Rule is applicable for

1. The supervised training of neural networks.
2. Weights are initialized at any value in this method.
3. This rule is also called LMS (least mean square).

Vicky Ahuja,
Artificial neural network: all about artificial neural world,
Indian journal of engineering, 2013, 6(15), 6-8,

4. It is independent of the activation function of neurons used since it minimizes the squared error between the desired output valued, and the neuron's activation value $net_j = W_i^t x$.

The learning signal r for this rule is

$\mathbf{r} = d_i - W_i^t x$

$\Delta W_i = \mathbf{c}\ (\mathbf{d} - W_i^t x) x$

For single weight adjustment

$\Delta W_{ij} = \mathbf{c}\ (d_i - W_i^t x) x_j$

Note: Widrow- Hoff learning rule is a special case of Delta learning rule.

If

$f(W_i^t x) = W_i^t x$

Then

$f(net) = net$ **and** $f'^{(net)} = 1$

where net = $W_i^t x$

Then from Delta learning Rule, the learning signal is

$r = [d_i - f(W_i^t x)]\, f'(W_i^t x)$

$r = [d_i - f(W_i^t x)] * 1$ **Or** $r = [d_i - (W_i^t x)]$

Therefore the learning signals of delta learning rule and Widrow Hoff learning rules are equal for this special case.

Correlation Learning Rule

1. Correlation learning rule is for supervised learning.
2. By substituting $\mathbf{r} = d_i$ in general learning rule :

$w_i(t) = cr\,[w_i(t), x(t)\, d_i(t) x(t)$

We obtain the correlation learning rule. The Weight adjustment are

$\Delta W_i = \mathbf{c}\ (d_i) x$

Or for single weight adjustment

$\Delta W_i = \mathbf{c}\ (d_i) x_j$

This rule states that if d_i is the desired response due to x_j the corresponding weight increase is proportional to their product..

1. The rule typically applies to recording data in memory networks with binary response neurons.

This rule is a special case of Hebbian Learning Rule with a binary activation and for $O_i = d_i$. But the main difference is that Hebbian Rule is performed in unsupervised mode and correlation learning rule in supervised learning mode. Similar to Hebbian Learning Rule, this learning rule also requires weight initialization $w = 0$.

REFERENCES

1. Artificial Neural Network: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
2. Introduction to Artificial Neural Networks by Gunjan Goswami

Vicky Ahuja,
Artificial neural network: all about artificial neural world,
Indian journal of engineering, 2013, 6(15), 6-8,