ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Review Paper on Artificial Neural Networks

Rani Pagariya
Computer science and engineering
G.H.R.C.E & M, Amravati
Amravati ,india
ranipagariya19@gmail.com

Mahip Bartere
Computer science and engineering
G.H.R.C.E & M, Amravati
Amravati ,india
mahip_media@yahoo.co.in

Abstract: Since the digital computers are being invented, the human being has been attempting to create machines which directly interact with the real world without his intervention. In this sense Artificial Neural Networks comes with an alternative for endowing to the computers one of the characteristic that is intelligence. This report is an introduction to Artificial Neural Networks The various types of neural networks are explained and demonstrated, benefits of artificial neural network are described, and a detailed historical background is provided. The connection between the artificial and the real neurons is also investigated and explained. Finally, the mathematical models involved are presented and demonstrated.

Keywords: component Feed Forward, ANN, sigmoid, distributed nature, firing rules.

I. INTRODUCTION

An Artificial Neural Network (ANN) as its name suggests it is a network, network of neurons. It is non-linear mapping structure, an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. An artificial neural network may be called as neural network or neural net (NN). They are powerful tools for modeling , especially when the underlying data relationship is unknown. The key element of this paradigm is the structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) as in human brain, working in unison to solve specific problems. ANNs to learn by example just like human being. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. As learning in biological systems involve adjustments to the synaptic connections that exist between the neurons. This is true of ANNs to.

II. LITERATURE SURVEY

Work on artificial neural networks, commonly referred to as *neural networks*, has been motivated right from its inception by the recognition that the brain computes in an entirely different way from the conventional digital computer[1]. An Artificial Neural Network (ANN) is an information processing paradigm that is inspired 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[2].An ANN is a computational structure that is inspired by observed process in natural networks of biological neurons in the brain. It consists of simple computational units,neurons, which are highly interconnected. ANNs have become the focus of much attention, largely because of their wide range of applicability and the ease with which they treat complicated problems.

ANNs are parallel computational models comprised of densely interconnected adaptive processing units. These networks are fine-grained parallel implementations of nonlinear static or dynamic systems. An important feature of these networks is their adaptive nature, where “learning by example” replaces “programming” in solving problems. This makes computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved but where training data is readily available. ANNs are recognized in the area of classification and prediction, where regression model and other related statistical techniques have traditionally been employed[3].

III. HISTORICAL BACKGROUND

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.. The struggle to understand the brain owes much to the pioneering work of Ramón y Cajál (1911), who introduced the idea of neurons as structural constituents of the brain [1]. The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much [2]. They combined many simple processing units together that could lead to an overall increase in computational power. They suggested many ideas like: a neuron has a threshold level and once that level is reached the neuron fires. It is still the fundamental way in which ANNs operate. The McCulloch and Pitts's network had a fixed set of weights. Hebb (1949) developed the first learning rule, that is if two neurons are active at the same time then the strength between them should be increased. In the 1950 and 60's, many researchers (Block, Minsky, Papert, and Rosenblatt worked on perceptron. The neural network model could be proved to converge to the correct weights, that will solve the problem. The weight adjustment (learning algorithm) used in the perceptron was found more powerful than the learning rules used by Hebb. The perceptron caused great excitement. It was thought to produce programs that could think. Minsky & Papert (1969) showed that perceptron could not learn those functions which are not linearly separable .The neural networks research declined throughout the 1970 and until mid 80's because the perceptron could not learn certain important functions. Neural network regained importance in 1985-86. The researchers, Parker and LeCun discovered a learning algorithm for multi-layer networks called back propagation that could solve problems that were not linearly separable[4].

IV. HUMAN AND ARTIFICIAL NEURONES

Typically, neurons are five to six orders of magnitude slower than silicon logic gates; events in a silicon chip happen in the nanosecond (10-9 s) range, whereas neural events happen in the millisecond (10-3 s) range. However, the brain makes up for the relatively slow rate of operation of a neuron by having a truly staggering number of neurons (nerve cells) with massive interconnections between them. It is estimated that there must be on the order of 10 billion neurons in the human cortex, and 60 trillion synapses or connections. The net result is that the brain is an enormously efficient structure. Specifically, the energetic efficiency of the brain is approximately 10-16 joules (J) per operation, whereas the corresponding value for the best computers (in 1994) is about 10-6 joules per operation. The brain is a highly complex, nonlinear, and parallel information-processing system. It has the capability of organizing neurons so as to perform certain computations (e.g. pattern recognition, perception, and motorcontrol) many times faster than the fastest digital computers.

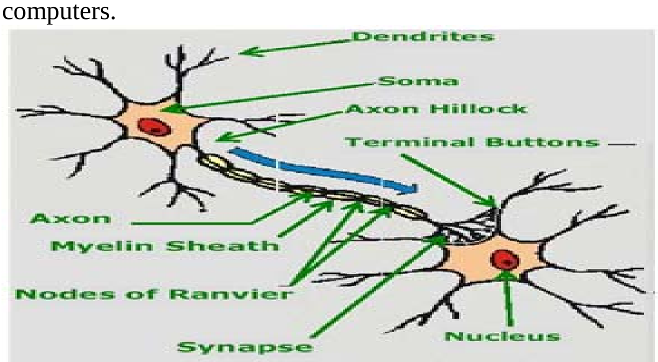

Figure 1. Structure of Nneuron [1]

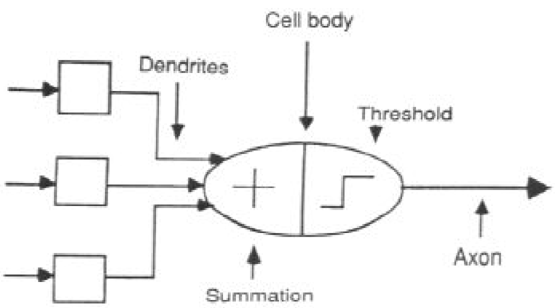

Figure 2: Artificial neuron model[2]

Consider, for example, human vision, which is an information-processing task. It is the function of the visual system to provide a representation of the environment around us and, more important, to supply the information we need to interact with the environment. To be specific, the brain routinely accomplishes perceptual recognition tasks (e.g. recognizing a familiar face embedded in an unfamiliar scene) in something of the order of 100-200 ms[1].

V. AN ENGINEERING APPROACH

a. A simple neuron An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

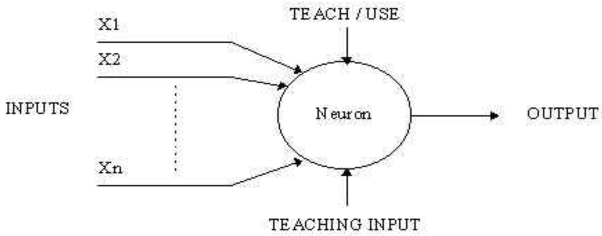

Figure 3:A simple neuron[2]

b. **Firing rules** The firing rule is an important concept in neural networks and accounts for their high flexibility. A firing rule determines how one calculates whether a neuron should fire for any input pattern. It relates to all the input patterns, not only the ones on which the node was trained. A simple firing rule can be implemented by using Hamming distance technique. The rule goes as : Take a collection of training patterns for a node, some of which cause it to fire (the 1-taught set of patterns) and others which prevent it from doing so (the 0-taught set). Then the patterns not in the collection cause the node to fire if, on comparison , they have more input elements in common with the 'nearest' pattern in the 1-taught set than with the 'nearest' pattern in the 0-taught set. If there is a tie, then the pattern remains in the undefined state.For example, a 3-input neuron is taught to output 1 when the input (X1,X2 and X3) is 111 or 101 and to output 0 when the input is 000 or 001[2].

Table 1: truth table before applying the firing rule

X1:		0	0	0	0	1	1	1	1
X2:		0	0	1	1	0	0	1	1
X3:		0	1	0	1	0	1	0	1
OUT:		0	0	0/1	0/1	0/1	1	0/1	1

VI. ARCHITECTURE OF NEURAL NETWORKS

There are several types of architecture of ANN. However, the two most widely used ANN are discussed below:

a. ***Feed-forward ANNs*** allow signals to travel one way only; from input to output.

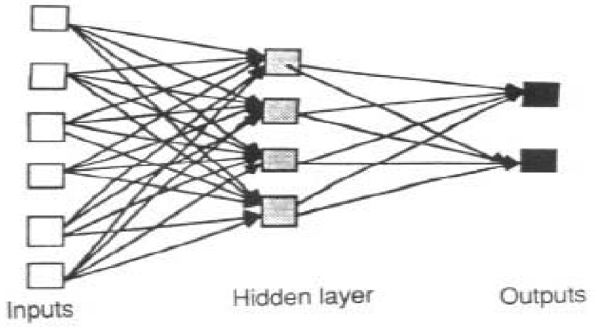

Figure 4: feed forward networks[2]

There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down.

b. ***Feedback/Recurrent Networks*** can have signals traveling in both directions by introducing loops in the network. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.[2]

VII. MATHEMATICAL MODEL INVOLVED:

The back propagation algorithm (Rumelhart and McClelland, 1986) is used in layered feed-forward ANNs. This means that the artificial neurons are organized in layers, and send their signals “forward”, and then the errors are propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an *output layer*. There may be one or more intermediate hidden layers. The back propagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated.

The idea of the back propagation algorithm is to reduce this error, until the ANN *learns* the training data. The training begins with random weights, and the goal is to adjust them so that the error will be minimal. The activation function of the artificial neurons in ANNs implementing the back propagation algorithm is a weighted sum

$$A_j\left(\overline{x},\overline{w}\right)=\sum_{i=0}^{n}x_i w_{ji} \quad (1)$$

We can see that the activation depends only on the inputs and the weights. If the output function would be the identity (output=activation), then the neuron would be called linear. But these have severe limitations. The most common output function is the sigmoidal function:

$$O_j\left(\overline{x},\overline{w}\right)=\frac{1}{1+e^{A_j\left(\overline{x},\overline{w}\right)}} \quad (2)$$

The sigmoidal function is very close to one for large positive numbers, 0.5 at zero, and very close to zero for large negative numbers. This allows a smooth transition between the low and high output of the neuron.

The goal of the training process is to obtain a desired output when certain inputs are given. Since the error is the difference between the actual and the desired output, the error depends on the weights, and we need to adjust the weights in order to minimize the error. We can define the error function for the output of each neuron:

$$E_j\left(\overline{x},\overline{w},d\right)=\left(O_j\left(\overline{x},\overline{w}\right)-d_j\right)^2 \quad (3)$$

We take the square of the difference between the output and the desired target because it will be always positive, and because it will be greater if the difference is big, and lesser if the difference is small. The error of the network will simply be the sum of the errors of all the neurons in the output layer:

$$E\left(\overline{x},\overline{w},\overline{d}\right)=\sum_j\left(O_j\left(\overline{x},\overline{w}\right)-d_j\right)^2 \quad (4)$$

The back propagation algorithm now calculates how the error depends on the output, inputs, and weights. After we find this, we can adjust the weights using the method of *gradient descendent*:

$$\Delta w_{ji}=-\eta\frac{\partial E}{\partial w_{ji}} \quad (5)$$

This formula can be interpreted in the following way: the adjustment of each weight wji will be the negative of a constant eta (0) multiplied by the dependence of the previous weight on the error of the network, which is the derivative of E in respect to Wi .The size of the adjustment will depend on 0, and on the contribution of the weight to the error of the function. This is, if the weight contributes a lot to the error, the adjustment will be greater than if it contributes in a smaller amount. (5) is used until we find appropriate weights (the error is minimal).This is the goal of the backpropagation algorithm, since we need to achieve this backwards. First, we need to calculate how much the error depends on the output, which is the derivative of E in respect j to O (from (3)).

$$\frac{\partial E}{\partial O_j}=2\left(O_j-d_j\right) \quad (6)$$

And then, how much the output depends on the activation, which in turn depends on the weights (from (1) and (2)):

$$\frac{\partial O_j}{\partial w_{ji}}=\frac{\partial O_j}{\partial A_j}\frac{\partial A_j}{\partial w_{ji}}=O_j\left(1-O_j\right)x_i \quad (7)$$

And we can see that (from (6) and (7)):

$$\frac{\partial E}{\partial w_{ji}}=\frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial w_{ji}}=2\left(O_j-d_j\right)O_j\left(1-O_j\right)x_i$$

(8)

And so, the adjustment to each weight will be (from (5) and (8)):

$$\Delta w_{ji}=-2\eta\left(O_j-d_j\right)O_j\left(1-O_j\right)x_i \quad (9)$$

We can use (9) as it is for training an ANN with two layers. Now, for training the network with one more layer we need to make some considerations. If we want to adjust the weights (let's call them Vik) of a previous layer, we need first to calculate how the error depends not on the weight, but in the input from the previous layer.

$$\Delta v_{ik} = -\eta \frac{\partial E}{\partial v_{ik}} = -\eta \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial v_{ik}} \quad (10)$$

Where:

$$\frac{\partial E}{\partial w_{ji}} = 2(O_j - d_j)O_j(1 - O_j)w_{ji} \quad (11)$$

And, assuming that there are inputs Uk into the neuron with Vik (from (7)):

$$\frac{\partial x_i}{\partial v_{ik}} = x_i(1 - x_i)v_{ik} \quad (12)$$

If we want to add yet another layer, we can do the same, calculating how the error depends on the inputs and weights of the first layer.

For practical reasons, ANNs implementing the backpropagation algorithm do not have too many layers, since the time for training the networks grows exponentially. Also, there are refinements to the backpropagation algorithm which allow a faster learning[5].

VIII. BENEFITS OF NEURAL NETWORKS

Its massively parallel distributed structure and its ability to learn and, therefore, *generalize*. Generalization refers to the neural network producing reasonable outputs for inputs not encountered during training (learning). These two information processing capabilities make it possible for neural networks to solve complex (large-scale) problems that are currently intractable. In practice, however, neural networks cannot provide the solution working by themselves alone. Rather, they need to be integrated into a consistent system engineering approach. Specifically, a complex problem of interest is decomposed into a number of relatively simple tasks, and neural networks are assigned a subset of the tasks that match their inherent capabilities. The following useful properties and capabilities:

a. Nonlinearity. A neuron is basically a nonlinear device. Consequently, a neural network, made up of an interconnection of neurons, is itself nonlinear.
b. Input-output mapping. A popular paradigm of learning called *supervised learning* involves the modification of the synaptic weights of a neural network by applying a set of training samples. Each sample consists of a unique input signal and the corresponding desired response. The network is presented a sample picked at random from the set, and the synaptic weights (free parameters) of the network are modified so as to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with an appropriate criterion. The training of the network is repeated for many samples in the set until the network reaches a steady state, where there are no further significant changes in the synaptic weights.
c. Adaptivity. Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions. Moreover, when it is operating in a nonstationary environment a neural network can be designed to change its synaptic weights in real time. The natural architecture of a neural network for pattern classification, signal processing, and control applications, coupled with the adaptive capability of the network, makes it an ideal tool for use in adaptive pattern classification, adaptive signal processing, and adaptive control.
d. Contextual information. Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt with neural network.
e. Fault tolerance. A neural network, implemented in hardware form, has the potential to be inherently fault tolerant in the sense that its performance is degraded gracefully under adverse operating. Due to the distributed nature of information in the network, the damage has to be extensive before the overall response of the network is degraded seriously. Thus, a neural network exhibits a graceful degradation in performance rather than catastrophic failure.
f. VLSI implementability. The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks. This same feature makes a neural network ideally suited for implementation using very-large-scale-integrated (VLS1) technology.
g. Uniformity of analysis and design. Basically, neural networks enjoy universality as information processors. This feature manifests itself in different ways:
a) Neurons, in one form or another, represent an ingredient common to all neural networks.
b) This commonality makes it possible to share theories and learning algorithms in different applications of neural networks.
c) Modular networks can be built through a seamless integration of modules.
h. Neurobiological analogy. The design of a neural network is motivated by analogy with the brain, which is a living proof that fault-tolerant parallel processing is not only physically possible but also fast and powerful. [1]

IX. CONCLUSION

This is a review paper which focusing on history, architecture, as it can be helpful in developing better approaches in developing neural network. Basic methods how neuron learn, so it get easy for beginner to learn about ANNs. It compares human neurons with ANNs giving idea about how concept of ANN evolved and how it works. Finally it is focusing on benefits ,the reason why to use ANNS. It is providing mathematical model-'backpropogation algorithm which is one of the best model to start with. Various other mathematical models can be studied and complete artificial human being can be developed using ANNs.

X. REFERENCES

[1]. M. Hajek (2005)" NEURAL NETWORKS".

[2]. Christos Stergiou and Dimitrios Siganos "NEURAL NETWORK"http://www.doc.ic.ac.uk/~nd/surprise_96 /journal/vol4/cs11/report.html

[3]. Girish Kumar Jha" ARTIFICIAL NEURAL NETWORKS" ,Indian Agricultural Research Institute,PUSA.

[4]. R. C. Chakraborty"Fundamentals of Neural Networks": AI Course lecture 37 – 38, notes, slides www.myreaders.info/,e-mail :rcchak@gmail.com June 01, 2010

[5]. Carlos Gershenson "Artificial Neural Networks for Beginners" C.Gershenson@sussex.ac.uk

[6]. Eric Davalo and Patrick Naim " Neural Networks".

[7]. Rumelhart, D. and J. McClelland (1986)." Parallel Distributed Processing." MIT Press, Cambridge, Mass.

[8]. Cheng, B. and Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. Statistical Science.

[9]. Anderson, J. A. (2003). An Introduction to neural networks. Prentice Hall.

[10]. Hassoun, M. H. (1995). Fundamentals of Artificial Neural Networks. Cambridge: MIT Press.

[11]. Daniel Graupe "Princilpe Of Artificial Neural Networks":2nd edition.

[12]. Rojas, R. (1996). Neural Networks: A Systematic Introduction. Springer, Berlin.