Research Journal of Computer and Information Technology Sciences ______________________E-ISSN 2320 – 6527
Vol. **4(7),** 6-13, July **(2016)** *Res. J. Computer and IT Sci.*

Review Paper

Mathematical Analysis for Training ANNs Using Basic Learning Algorithms

Vaibhav Kant Singh
Dept. of Computer Science and Engineering, Institute of Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, CG, India
vibhu200427@gmail.com

Available online at: www.isca.in, www.isca.me
Received 24th Feburary 2016, revised 7th July 2016, accepted 13th July 2016

Abstract

Artificial Neural Network is the Stream of Computer Science that considers the construction of programs that are analogous to the working of human brain. There are three basic entities of Artificial Neural Network i.e. Artificial Neuron, Network Topology and Learning Rule. In this paper we will have an insider to some of the basic learning rules. The pattern in which the various units present in the ANN are structured is dependent on the training algorithm used to train the network. Thus, we can say that Learning rules used for the design of ANN are being structured. In this paper we have discussed two classifications of learning algorithms on the basis of the procedure exhibited while training. The two classifications are Supervised and Unsupervised Learning. Some of the Learning rules discussed in this paper are Error Correction Learning, Hebbian Learning and Competitive Learning. The basic structure incorporated by the Learning algorithm is discussed in the paper. Also various features exhibited are concluded on the basis of the mathematical explanation given specifying the working of Learning Algorithm. In the paper Supervised learning which employs the help of teacher while training generally produces accurate result. As the teacher in the supervised learning paradigm by virtue of his experience in the field for Artificial Neural Network is constructed produces desired response hence guides the process towards producing correct and consistent result. Unsupervised learning take help of the set of adaptation rules for making changes in the adaptable parameters of the network. As the system is unaware of the desired response and is guided by quasi biological process the Learning algorithm is less accurate.

Keywords: Artificial Neural Network, Neurons, Supervised Learning, Unsupervised Learning, Error Correction Learning, Hebbian Learning, Competitive Learning

Introduction

Artificial Neural Network is a system that operates like a parallel and distributed processor, the discussed processor is having a natural tendency of storing knowledge acquired during its practice in a particular environment and also having the capability of utilizing the knowledge gained. ANN is a machine that is modeled to perform the way in which brain performs a task of interest. It is simulated in Software in a Digital Computer or is implemented as Electronic components. ANN is having a degree of resembles human brain in two scenarios i.e. Knowledge is obtained from the domain through learning process and Inter-neuron connection strength known as Synaptic weights are used to store the obtained knowledge. The concept of ANN started when in 1943 the classical paper which was written by McCulloh and Pitts was published and presented in International Conference in University of Chicago[1]. Von-Neumann made EDVAC after getting influenced from the paper. EDVAC gave rise to the emergence of ENIAC the first digital computer. In 1949 D.O. Hebb proposed Hebb's learning law[2]. In 1954 Minsky proposed learning machines[3]. In 1958 Rosenblatt proposed perceptron learning and convergence[4]. In 1960 Widrow and Hoff proposed ADALINE[5]. In 1969 Minsky and Papert proposed Multilayer perceptron[6]. In 1974 Werbos gave Error backpropagation framework[7]. In 1985 Ackley, Hinton and Sejnowski proposed Boltzmann machine[8]. In 1986 Rumelhart, Hinton and Williams proposed generalized delta rule[9]. ANN is having analogy with human brain i.e. BNN. In Brain there are neurons, where every neuron contains cell body or soma where the cell nucleus is located. Tree like nerve fibers called dendrites are associated with the cell body. These dendrites receive signals from other neurons. In cell body there is an extension which is called axon. Which is a long fiber that eventually branches into strands and sub strands connecting to many other neurons at the synaptic junction, or synapses. There are around 10^{11}neurons, 10^{14} synapses and about 10^{4} interconnections. The nerve cell i.e. the neuron is at a resting potential of -70mV, when neuron is at this potential the cell membrane associated with the neuron is impermeable. The intracellular fluid is negatively charged whereas the extracellular fluid is containing Na+ ions.

Since the membrane is impermeable the Na+ ions are not allowed to get inside the cell. But when the dendrites present in the neuron receive signal than if the potential of the cell increases to an amount -60mV. The neuron is said to fire, which causes change in the permeability of the neuron. Now, because of firing the membrane becomes permeable and the Na+ ions present outside are allowed to enter the cell. This causes chemical reaction as a result of which certain chemical substances called neurotransmitters are produced which travel from pre-synaptic region to post-synaptic region causing change in polarization and result in the modification in the value of synapse. Synapses are responsible for storing the knowledge that we obtain from the environment. ANN is also having numeric values called synaptic weights to store the knowledge.

There are three basic techniques described by Simon to represent ANN i.e. Block Diagram Representation, Signal Flow Graph and Architectural Graph expressed in Figure-1 (b), (c) and (d)[10]. Artificial Neural Network exhibits various benefits including Nonlinearity, Input-Output Mapping, Adaptivity, Contextual Information, Fault Tolerance, Uniformity of Analysis and Design, Neurobiological analogy, Evidential Response, VLSI Implementability etc. The Artificial Neural Network systems include ADALINE (Adaptive Linear Neural Element), SOFM (Self Organizing Feature Map), AM (Associative Memory), Boltzmann Machine, BAM (Bidirectional Associative Memory), RNN (Recurrent Neural Network), Neoconition, BSB (Brain-State-in-a-Box), Cauchy machines, Hopfield Network, ART (Adaptive Resonance Theory), LVQ (Learning Vector Quantization), RBF (Radial Basis Function), Perceptron etc.

The three basic elements of Artificial Neural Network are Network Architecture specifying the way the neurons are present in the topology, Individual elements present in the topology and Training Algorithm that is used to train the network. There are three basic class of the first element i.e. Network Architecture namely Single Layer Feedforward, Multi Layer Feedforward and Recurrent Network. Different types of Artificial Neural Network System fall in one of these three categories. The second element i.e. the Neuron is the basic computation unit of the Artificial Neural Network. The third and a very important element of Artificial Neural Network is Learning. In this paper we will see the basic working of some of the Learning algorithms employed for training the Artificial Neural Network.

Literature Survey

In the year 2013 Sathya and Abraham considered two learning algorithms for ANN supervised and unsupervised for making an investigation on classification of post graduate students and found that error correction learning a type of supervised learning is less efficient than KSOM (Kohonen's Self Organizing Model) an unsupervised learning approach[11]. Hormozi et. al. in 2012 described various machine learning classification techniques used in robotic manipulators[12]. Aruna et. al. made empirical comparison on various supervised learning algorithms in the year 2011. UCI machine learning repository was used for carrying out experiment in WEKA[13]. In the year 1999 Altug et. al. presented two neural fuzzy inference systems, namely, FALCON and ANFIS, with applications to induction motor fault detection/diagnosis problems[14]. Shieh and Lin proposed VNN a supervised learning algorithm for classification in general and for emitter identification in particular in the year 2002[15]. In 1985 Ackley et. al. proposed a parallel search method, based on statistical mechanics and showed how the method leads to a general learning rule for modifying the connection strengths so as to incorporate knowledge about a task domain in an efficient way[16]. In the year 1949 Hebb gave the statement in his book which was later on utilized for implementing the Hebbian learning in ANN[2]. In 1988 Jacobs examined the steepest descent and analyzed why it is slow to converge. He also proposed four heuristic mechanisms for achieving faster rates of convergence while adhering to the locality constraint[17]. Rupp and Sayed in 1997 provided a time-domain feedback analysis of the perceptron learning algorithm and of training schemes for dynamic networks with output feedback[18]. In 2012 Praveen et. al. demonstrated the minimization of error using the gradient descent method implemented using Multilayer Neural Network[19]. Loganathan and Girija in 2013 determined appropriate neural network architecture for training the ANFIS structure inorder to adjust the free parameters[20]. Mohd in 2013 reviewed the backpropagation algorithm[21]. Budura et. al. introduced DPRCL in the year 2006. The algorithm is a competitive learning algorithm and is used for data clustering[22]. In the year 2004 Papageorgion et. al. proposed AHL Hebbian learning algorithm that alleviates the problem of the potential convergence to a steady state[23]. Sonali and Priyanka provided an overview of ANN in the year 2014[24]. In the year 2014 Arijit and Asoke gave a theoretical study on how ANN can be used for data encryption[25]. Foram and Mahesh in the year 2015 proposed a method for calculating the number of hidden layers for a given problem on the basis of input data[26]. In the year 2015 Abdullahi et. al. discussed different techniques for application of Neural Networks and discussed their present limitation and scope for future research in Artificial Neural Networks[27]. During the couple of years i.e. 2015 and 2016 Vaibhav Kant Singh the author of the current paper proposed solution to the XOR problem using MLP[28-31]. Vaibhav Kant Singh proposed a Solution for the Ex-NOR problem in the current year i.e. 2016[32].

Learning in Artificial Neural Network

The basic property present in the Artificial Neural Network that makes it extraordinary is the capability to learn from the environment for which the ANN is prepared and to improve its performance through the training process. The enhancement in the performance and structure takes place under some prescribed limitations and for having enhancement it takes time. As already discussed the way in which neurons of an Artificial Neural Network are connected is directly linked with the learning algorithm used to train the network. Therefore we say that learning algorithms which are used to train the network are structured. Generally an Artificial Neural Network adapts to its environment by iteratively making changes in its weights and bias level. The artificial neural network becomes more knowledgeable after each iteration of the training process. Mendel and Maclaren (1970) gave definition of learning in the context of Artificial Neural Network. The definition they gave is "Learning is a process by which the free parameters of neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place"[10]. Neural Network Learning algorithms are classified into three basic types namely Supervised learning (Error based), Reinforced learning (Output based) and Unsupervised learning. In this paper we made a mathematical description of working of Supervised and Unsupervised learning. Supervised learning is further classified into Stochastic and Error Correction (Gradient descent). Error Correction is further having Least Mean Square (LMS) algorithm and Back Propagation algorithm. Supervised learning algorithms are implemented in the Artificial Neural Network systems like perceptron, ADALINE/MADALINE and various Multilayer networks. Two phases are involved in Supervised learning namely Learning phase and Retrieving phase. Unsupervised learning is employed in systems like ART, Kohonen's Self Organizing Map etc.

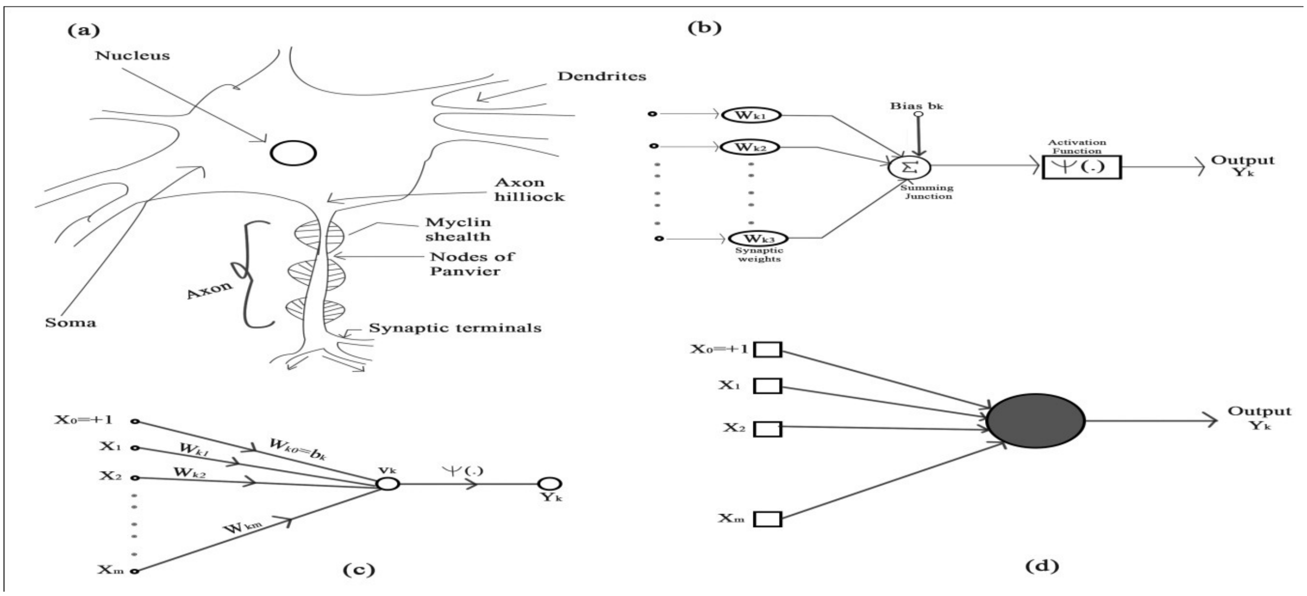

Figure-1
(a) Biological neuron; (b) Block Diagram of neuron in ANN; (c) Signal flow graph of neuron; (d) Architectural Graph of neuron

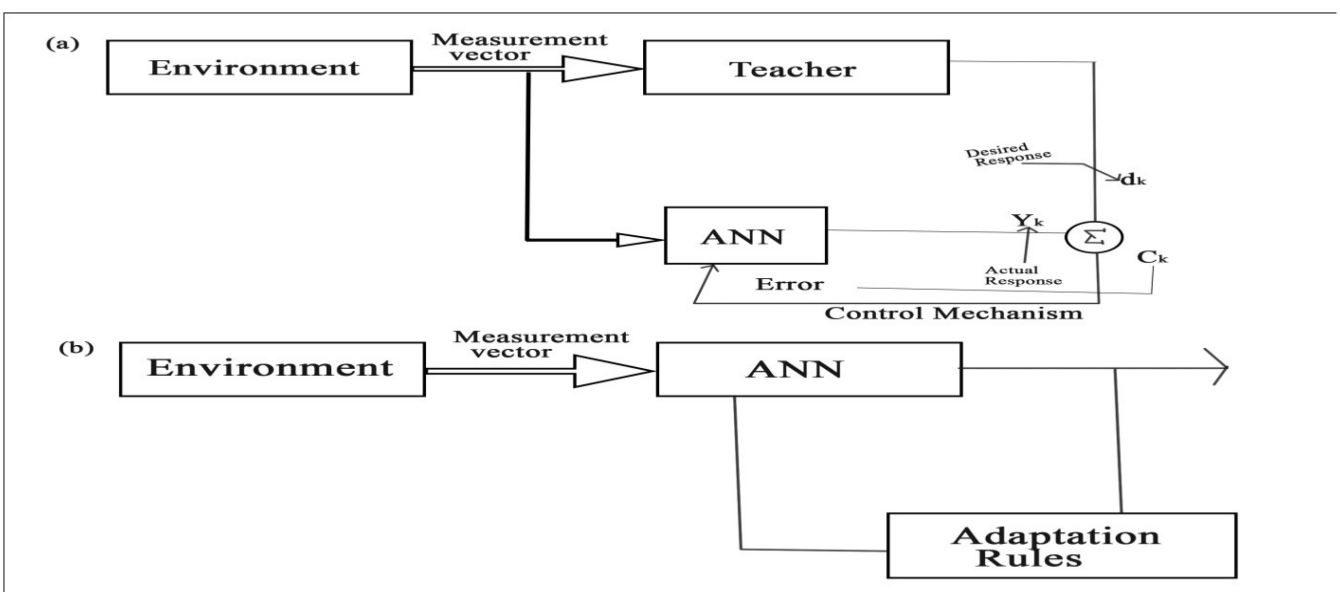

Figure-2
(a) Supervised Learning (b) Unsupervised Learning

Supervised Learning: Supervised learning as the name refers there is a teaching element who guides the Artificial Neural Network during the training process. The environment in which the Artificial Neural Network is supposed to operate is exposed to both the teaching component and the Artificial Neural Network system as expressed in Fig. 2(a). Every stimulus applicable for the Artificial Neural Network is provided to both the Teaching element specified by Teacher and the Artificial Neural Network system represented by ANN. The Artificial Neural Network by virtue of its structure produces response for the obtained stimulus. The response obtained for the provided stimulus will be called as actual output. The teacher is supposed to be an element that by experience of the environment produces the desired response for the obtained stimulus. Now the output obtained from the Artificial Neural Network system is compared with the output provided by the teacher. If an error is generated then the generated error invokes control mechanism which makes changes in the values of synaptic weights present in the Artificial Neural Network structure. The process continues until the error value is minimized to the maximum extent.

Basic types of Supervised Learning are: **Error Correction Learning:** In this section we will discuss the working of a Error Correction learning is perceptron as expressed in Figure-3. Mathematical modeling of the way how training is performed in perceptron is discussed in this section. For classification task generally the value of Desired response D= -1 or +1. To understand the problem consider a perceptron network with a set of input training vectors X1, X2, X3 and the initial weight vector is W1. The learning rate constant C is assumed to be 0.2. The teachers desired response for X1, X2 and X3 are D1=-1, D2=1 and D3=-1 respectively. Activation function taken is sgn i.e. Sign function.

$$sgn(NET(N)) = \begin{cases} 1 \; if \; NET(N) > 0 \\ -1 \; Otherwise \end{cases}$$

$$NET(N) = W(N)X(N)$$

$$sgn(NET(N)) = Y(N)$$

$$E(N) = D(N) - Y(N)$$

$$W(n+1) = W(N) + C \times E(N) \times X(N)$$

Let, $X1 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 2 \end{bmatrix} X2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} X3 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} W1 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

Training according to the perceptron algorithm progresses as follows:-
Step1: Using X1 and W1 Calculate W2

$$\boldsymbol{NET1 = W1 \times X1 = [-1 \quad 1 \quad 1 \quad 1] \times \begin{bmatrix} 1 \\ 1 \\ -1 \\ 2 \end{bmatrix} = 1}$$

$$Y1 = sgn(1) = 1$$

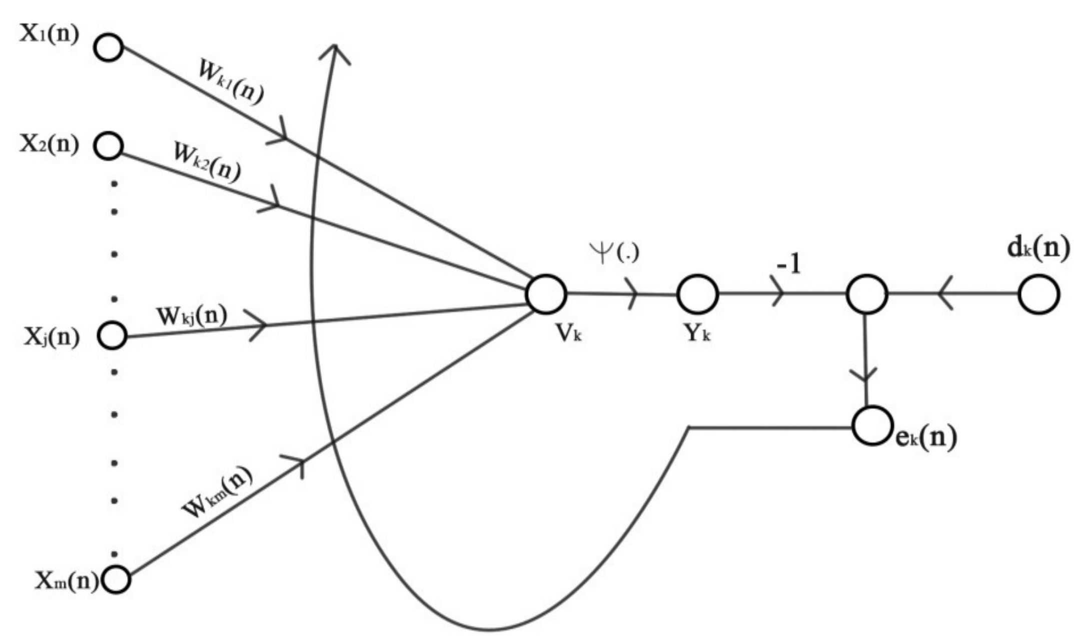

Figure-3
Output neuron in Error Correction Learning

Here, Since D1=-1 Therefore $D1 \neq Y1$ Therefore $E1 = D1 - Y1$ which means that E1=-2. Therefore there is going to be weight change i.e.

$$W2 = W1 + C \times E1 \times X1 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + 0.2 \times (-2) \times \begin{bmatrix} 1 \\ 1 \\ -1 \\ 2 \end{bmatrix}$$

Therefore,

$$W2 = \begin{bmatrix} -1.4 \\ 2 \\ 0 \\ 3 \end{bmatrix}$$

Step2: Using X2 and W2 Calculate W3

$$NET2 = W2 \times X2 = [-1.4 \quad 2 \quad 0 \quad 3] \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = -1$$

$$Y2 = sgn(-1) = -1$$

Here, D2=1 i.e. Therefore $D2 \neq Y2$ Therefore $E2 = D2 - Y2$ which means that E2=2. Therefore there is going to be weight change i.e.

$$W3 = W2 + C \times E2 \times X2 = \begin{bmatrix} -1.4 \\ 2 \\ 0 \\ 3 \end{bmatrix} + 0.2 \times 2 \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix}$$

$$W3 = \begin{bmatrix} -1 \\ 2.4 \\ 0.4 \\ 2.6 \end{bmatrix}$$

Step3: Using X3 and W3 Calculate W4

$$NET3 = W3 \times X3 = [-1 \quad 2.4 \quad 0.4 \quad 2.6] \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 4.4$$

$$Y3 = sgn(4.4) = 1$$

Here, D3=-1 i.e. Therefore $D3 \neq Y3$ Therefore $E3 = D3 - Y3$ which means that E3=-2. Therefore there is going to be weight change i.e.

$$W4 = W3 + C \times E3 \times X3 = \begin{bmatrix} -1 \\ 2.4 \\ 0.4 \\ 2.6 \end{bmatrix} + 0.2 \times (-2) \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$W4 = \begin{bmatrix} -1.4 \\ 2 \\ 0 \\ 2.2 \end{bmatrix}$$

Here, W4 represents the value of Weight Matrix after learning X1, X2 and X3. In each of the steps the value of the actual output i.e. Y was compared with Desired output D. In the above example in all the steps there was a mismatch and thus adjustment in weight took place.

Unsupervised Learning: Unsupervised learning as the name reveals learning is done in the absence of the Teacher as expressed in Figure-2(b). In other words there is no one to guide the learning process. In Unsupervised learning there are set of adaptation rules defined on the domain for which the Artificial Neural Network is constructed. The response produced by the Artificial Neural Network is guided by the hypothesis employed by the unsupervised learning algorithm used. Generally Hebbian Learning algorithms and Competitive Learning algorithms come under unsupervised learning. In this section we will see a mathematical formulation that is employed by the two approaches to implement adjustment in weight.

Hebbian Learning: Hebbian learning is one of the oldest form of learning. The Learning advocated is influenced from the statement made by Hebb in his book Organization of behavior[2]. Hebb's statement is "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, Some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased."The synapses that follow Hebb's statement are called Hebbian synapse. The four key mechanisms that are in Hebbian learning are namely Time-dependent, local, Interactive and Conjunctional. The Hebbian rule implies that if the cross product of output and input is positive there is an increase in weight. In case the cross product is negative the weight decreases. In this section we will see an example representing the working of Hebbian learning mathematically. Consider the same set of input values X1, X2 and X3 used for the case of perceptron. We Assumed learning rate constant C as 1. And W1 i.e. initial weight vector is assumed to be same as that for the perceptron case. Activation function used is sgn.

$$sgn(NET(N)) = \begin{cases} 1 \; if \; NET(N) > 0 \\ -1 \; Otherwise \end{cases}$$

$$NET(N) = W(N)X(N)$$

$$sgn(NET(N)) = Y(N)$$

$$W(N+1) = W(N) + C \times sgn(NET(N)) \times X(N)$$

$$Let, \quad X1 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 2 \end{bmatrix} X2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} X3 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} W1 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Step 1: Using X1 and W1 Calculate W2

$$NET1 = W1 \times X1 = [-1 \quad 1 \quad 1 \quad 1] \times \begin{bmatrix} 1 \\ 1 \\ -1 \\ 2 \end{bmatrix} = 1,$$

$$Therefore \; sgn(NET1) = 1$$

$$W2 = W1 + C \times sgn(NET1) \times X1$$

$$W2 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + 1 \times 1 \times \begin{bmatrix} 1 \\ 1 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 3 \end{bmatrix}$$

Step2: Using X2 and W2 Calculate W3.

$$NET2 = W2 \times X2 = \begin{bmatrix} 0 & 2 & 0 & 3 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = -1$$

$$Therefore\ sgn(NET2) = -1$$

$$W3 = W2 + C \times sgn(NET2) \times X2$$

$$W3 = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 3 \end{bmatrix} + 1 \times -1 \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 4 \end{bmatrix}$$

Step3: Using X3 and W3 Calculate W4.

$$NET3 = W3 \times X3 = \begin{bmatrix} 0 & 1 & -1 & 4 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 4$$

$$Therefore\ sgn(NET3) = 1$$

$$W4 = W3 + C \times sgn(NET3) \times X3$$

$$W4 = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 4 \end{bmatrix} + 1 \times 1 \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 5 \end{bmatrix}$$

In the case of Hebbian learning in the above example it can be seen that learning with discrete f(NET) (here f=sgn) and C=1 results in adding or subtracting the entire input pattern vectors to and from the weight vector respectively.

Competitive Learning: In Competitive learning as the name reveals the output unit compete to become active and only one output unit or one unit per group is active at a time. Rumelhart and Zipser proposed that competitive learning rule comprises of 3 basic formulations which are, first “A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input pattern”, second “A limit imposed on strength of each neuron” and third “A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron, or only one neuron per group, is active at a time”. The unit that wins the competition is called “Winner-takes-all-neuron”10.

$$Yk = \begin{cases} 1\ if\ NETk > NETj\ for\ all, j \neq k \\ 0\ Otherwise \end{cases}$$

Where k and j represents the index value of the neuron

$$\sum_j Wkj = 1$$

$$\Delta Wkj = \begin{cases} C(Xj - Wkj) if\ neuron\ k\ wins\ the\ competition \\ 0\ if\ the\ neuron\ k\ loses\ the\ competition \end{cases}$$

Let, {W11=0.25, W12=0.25, W13=0.25, W14=0.25} be the values of weights attached to neuron 1 producing output Y1, {W21=0.1, W22=0.4, W23=0.3, W24=0.2} be the values of weights attached to neuron 2 producing output Y2and {W31=0.2, W32=0.25, W33=0.25, W34=0.3} be the values of weights attached to neuron 3 producing output Y3. Also, let x1=2, x2=3, x3=4 and x4=1.

$$NET1 = 2 \times 0.25 + 3 \times 0.25 + 4 \times 0.25 + 1 \times 0.25 = 2.50$$

$$NET2 = 2 \times 0.1 + 3 \times 0.4 + 4 \times 0.3 + 1 \times 0.2 = 2.80$$

$$NET3 = 2 \times 0.2 + 3 \times 0.25 + 4 \times 0.25 + 1 \times 0.3 = 2.45$$

Here, since $NET2 > NET1 > NET3$ Therefore output signals of neuron 2 will be active and rest will be deactivated. According to the formulation made weight change is going to take place only in the weights of neuron 2 which is the winner neuron. The procedure continues until all the patterns applicable for the network are learned by the Artificial Neural Network.

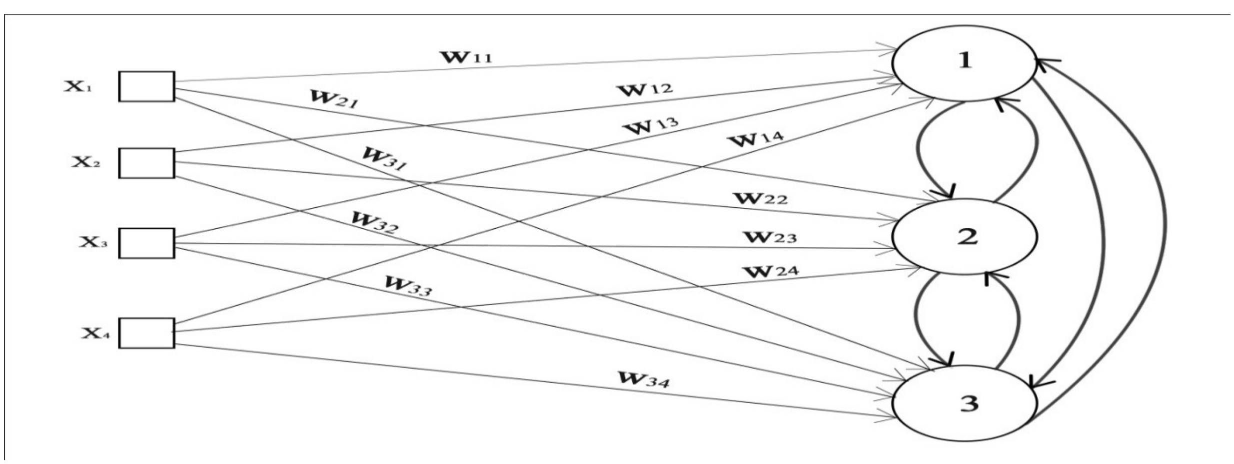

Figure-4
Architectural Graph representing three neurons, employing Competitive Learning

Conclusion

In the survey that we made we concluded that the approach Supervised learning is most widely used for the construction of the Artificial Neural Network the reason being the simplicity in the structure required for training. Also the approach is more accurate as the Artificial Neural Network is under supervision of a teacher who knows the desired response. Perceptron employs error correction learning a type of supervised learning as shown above. Supervised learning refers to search in weight space along error gradients. In supervised learning there is user class membership concept. Unsupervised learning as discussed above operate without a teacher, there are set of adaptation rules that are used for making improvement. Learning algorithm is comparatively less accurate. Unsupervised learning refers to the modification in the free-parameters using quasi biological method. In Hebbian learning which is basically used as Unsupervised learning algorithm in the case of a continuous f(NET), the weight incrementing and decrementing vector is scaled down to a fractional value of the input pattern. In competitive learning only in the neuron that is active weight change takes place.

Acknowledgment

I would like to place my thanks to my mother Smt Sushma Singh, my wife Smt Shubhra Singh, my brothers Abhinav Kant Singh and Abhuday Kant Singh, my sister Priyamvada Singh, my daughter Akshi and son Aishwaryat for the support that they give for the preparation of this work. Last but not the least I want to thank my Father Late Shri Triveni Kant Singh for his blessing, which enabled me to this work.

References

1. McCulloch W.S. and Pitts W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophy.,* 5, 115-133.
2. Hebb D. O. (1949). The Organization of Behaviour : A Neuropsychological Theory. Wiley, Newyork, United States of America.
3. Minsky M.L. (1954). Theory of neural-analog reinforcement systems and its application to the brain-model problem. Doctorate Thesis, Princeton University, Princeton, NJ.
4. Rosenblatt D.E. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
5. Widrow B. and Hoff M.E. (1960). Adaptive switching circuits. IRE WESCON Convention Record, 4, 96-104.
6. Minsky M.L. and Papert S.A. (1969). Perceptron : An introduction to computational geometry. Cambridge, MA : MIT Press.
7. Werbos P.J. (1974). Beyond regression : New tools for prediction and analysis in the behavioural sciences. Doctorate Thesis. Harward University, Cambridge, MA.
8. Ackley D.H., Hinton, G.E. and Sejnowski, T.J. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, 9, 147-169.
9. Rumelhart D.E., Hinton G.E. and Williams R.J. (1986). Learning internal representations by error propagation. Parallel Distributed Processing : Explorations in the Microstructure of Cognition. 1, Cambridge, MA : MIT Press, 318-362.
10. Haykin S. (1994). Neural Networks : A Comprhensive Foundation. New York : Macmillan College Publishing Company Inc.
11. Sathya R. and Abraham A. (2013). Comparison of Supervised and Unsupervised Learning Algorithm for Pattern Classification. *International Journal of Advance Research in Artificial Intelligence*, 2, 34-38.
12. Hormozi H., Hormozi E. and Nohooji H.R. (2012). The Classification of the Applicable Machine Learning Methods in Robot Manipulators. *International Journal of Machine Learning and Computing*, 2, 560-563.
13. Aruna S., Rajagopalan S.P. and Nandakishore, L.V. (2011). An Emperical Comparison of Supervised Learning Algorithms in Disease Detection. *International Journal of Information Technology Convergence and Services,* 1, 81-92.
14. Altug S., Chow M.Y. and Trussell H.J. (1999). Fuzzy Inference Systems Implemented on Neural Architectures for Motor Fault Detection and Diagnosis. *IEEE Transactions on Industrial Electronics*, 46, 1069-1079.
15. Shieh C.S. and Lin C.T. (2002). A Vector Neural Network for Emitter Identification. *IEEE Transactions on Antennas and Propagation*, 50, 1120-1127.
16. Jacobs R.A. (1988). Increased Rates of Convergence through Learning Rate Adaptation. *Neural Networks*, I, 295-307.
17. Rupp M. and Sayed A.H. (1997). Supervised Learning of Perceptron and Output Feedback Dynamic Networks : A Feedback Analysis Via the Small Gain Theorem. *IEEE Transactions of Neural Networks*, 8, 612-622.
18. Sehgal P., Gupta S. and Kumar D. (2012). Minimization of Error in Training a Neural Network using Gradient Descent method. *International Journal of Technical Research*, 1, 10-12.
19. Loganathan C. and Girija K.V. (2013). Hybrid Learning for Adaptive Neuro Fuzzy Inference Systems. *International Journal of Engineering and Science*, 2, 6-13.
20. Saduf M.A.W. (2013). Comparative Study of Backpropagation Learning Algorithms for Neural

Networks. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3, 1151-1156.

21. Budura G., Botoca C. and Miclau N. (2006). Competitive Learning Algorithms for Data Clustering. *Facta universitatis - series: Electronics and Energetics,* 19, 261-269.

22. Papageorgiou E.I., Stylios C.D. and Groumpos P.P. (2004). Active Hebbian Learning Algorithm to Train Fuzzy Cognitive Maps. *Elsevier International Journal of Approximate Reasoning*, 37, 219-249.

23. Maind S.B., and Wankar P. (2014). Research Paper on basic of Artificial Neural Network. *International Journal of Recent and Innovation Trends in Computing and Communication*, 2, 96-100.

24. Ghosh, A. and Nath A. (2014). Cryptography Algorithms using Artificial Neural Networks. *International Journal of Advance Research in Computer Science and Management Studies*, 2, 375-381.

25. Panchal F. and Panchal M. (2015). Optimizing Number of Hidden Nodes for Artificial Neural Network using Competitive Learning Approach. *International Journal of Computer Science and Mobile Computing*, 4, 358-364.

26. Muhammad A.U., Musa A.G. and Yarima K.I. (2015). Survey on Training Neural Networks. *International Journal of Adavnce Research in Computer Science and Software Engineering*, 5, 169-173.

27. Singh V.K. (2015). One Solution to XOR Problem using Multilayer Perceptron having Minimum Configuration. *International Journal of Science and Engineering*, 3(3), 32-41.

28. Singh V.K. (2015). Two Solution to the XOR Problem using Minimum Configuration MLP. *International Journal of Adavance Engineering Science and Technological Research*, 3(3), 16-20.

29. Singh V.K. (2016). Proposing Solution to XOR Problem using Minimum Configuration MLP. *Procedia Computer Science Elsevier*, ScienceDirect, International Conference on Computational Modeling and Security, CMS 2016, Bangalore, India, 11-13 Feb, 255-262.

30. Singh V.K. (2016). Minimum Configuration MLP for solving the XOR problem. Proceeding of IEEE International Conference on Computing for Sustainable Global Development and Accepted for Publishing in IEEE Explore, IEEE Conference ID:37465, INDIA Com-2016, Delhi, India, 16-18 March, 168-173.

31. Singh V.K. (2016). Mathematical Explanation to Solution for Ex-NOR problem using MLFFN. *International Journal of Information Sciences and Techniques*, 6(1/2), 105-122.