A Comparative Study of Training Algorithms of Artificial Neural Network using MATLAB

Saumya Verma
G.L.B.I.T.M, Gr. Noida, India

Vaishnavi Gupta
G.L.B.I.T.M, Gr. Noida, India

Devraj Kamboj
Assistant professor
G.L.B.I.T.M, Gr. Noida, India

ABSTRACT

Neural networks are simplified models of the biological neuron system. A neural network is a massively parallel distributed system made up of highly interconnected neural computing elements that have the ability to learn and thereby acquire knowledge and make it available for use. ANN's, like people, learn by example. This paper gives overview of artificial neural networks, their working, architecture, learning methods, how one can create and train their own neural network using MATLAB, and their applications.

Keywords

ANN, MATLAB, Activation function, Training function

1. INTRODUCTION

An Artificial neural network is a computational model that replicates the working of the central nervous system of human beings. It is a web of interconnected computing units called artificial neurons that work in a parallel and distributed manner (Pai & Rajasekaran, 2003). It was first introduced in 1943, when Warren McCulloch, an American neurophysiologist, and a mathematician Walter Pitts wrote a paper on how neurons work and they made a simple model of an ANN using electrical circuits. They are very good learning devices that can be used for speech recognition and object recognition (Allen & J.B., 1994). Artificial neural networks are quite different from conventional computers in a way that conventional computers need a proper set of rules to follow in order to arrive at a solution. They provide us with solutions to the problems we already know how to solve whereas ANNs learn by examples (Minin, 2006). They can recognize complete patterns from incomplete, ambiguous and noisy ones. They cannot be programmed to perform a specific task. One major disadvantage is that they give solutions based on previous trends and experience as a result their solutions can be very unpredictable. Since traditional computers need step by step algorithm to perform a task, ANNs function via images and concepts. Hence, these two are not in competition but complement each other. There are various tasks which require a combination of both for effective output.

1.1 Biological Neuron And Artificial Neuron

As mentioned before that ANNs are a web of highly interconnected units (Kumar & Sharma, 2014). These units are called artificial neuron and work in the same way as a biological neuron does. A human brain is made up of about millions and millions of interconnected units called neurons. A neuron processes and transmits information through electrical and chemical signals.

A biological neuron consists of a cell body called soma. There are two types of links connected to the soma- dendrites and axon. Dendrites are used to get input. Axon is electrically active and transmits the information. Axon ends in a specialized link called synapse or synaptic junction which transfers the signal from axon to the dendrites of the other neuron. These synaptic links have neuron transmitter fluid which can accelerate or retard the signals to the soma (Eluyode & Akomolafe, 2013).

In case of an artificial neuron, this acceleration and retardation is modelled by weights. If the signal from one of the neuron is strong the synaptic weight will increase and if the signal is weak then the synaptic weight will decrease. Weights are the multiplicative factors of the input received.

The total input I received will be:

$$I= \sum XiWi$$

To get the final output y, we pass the weighted sum of all the inputs on to an activation function or transfer function,

$$y=\phi\ (I)$$

A very commonly used activation function is the Thresholding function. The y we receive is binary, depending on if the sum is greater than or less than the threshold value, θ.

$$\phi(I) = \begin{matrix} 1, & I>\theta \\ 0, & I\leq\theta \end{matrix}$$

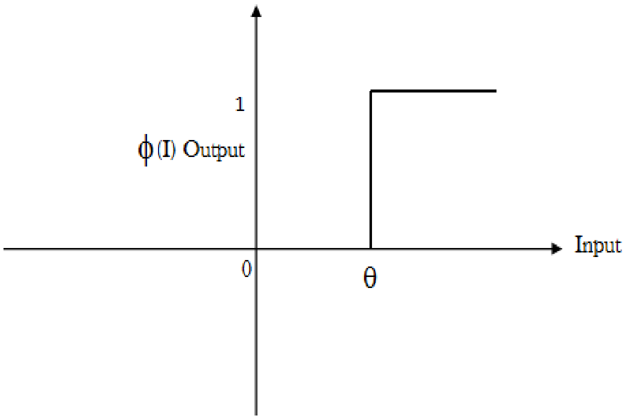


Fig 1: Thresholding Function

SIGNUM FUNCTION

Signum function, also known as sign function or quantizer function is defined as:

$$\phi(I)= \begin{matrix} +1, & I>\theta \\ -1, & I\leq\theta \end{matrix}$$

SIGMOIDAL FUNCTION

It's a continuous function that varies between asymptotic values 0 and 1 or -1 and +1. It is given by:

$$\phi(I)= \frac{1}{1+e^{-\alpha I}}$$

Where, α is the slope parameter that adjusts the abruptness of the function as it changes between two asymptotic values.

Sigmoidal functions are often used in artificial neural networks to introduce nonlinearity in the model.

2. NEURAL NETWORK ARCHITECTURE

ANN architecture can be represented in the form of directed graphs or digraphs. In these graphs the nodes represent the artificial neurons and the edges represent the synaptic links.

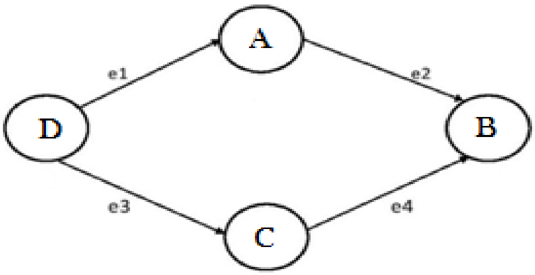


Fig 2: Example of a diagraph

In the above diagram A, B, C and D are artificial neurons while e1, e2, e3 and e4 the synaptic links.

There are 3 types of neural network architectures:

2.1 Single Layer Feedforward Network

Single layer feedforward network has the simplest of architectures. It comprises of two layers-the input layer and the output layer. The neurons in the input layer are termed as input neurons whereas the neurons in the output layer are termed as output neurons. The links between the input and the output layers carry weights. These two layers can only be connected unidirectionally, in this case, input to output hence the name feedforward. It is called single layer because the input layer transmits the signals to the output layer and the output layer performs the main computations (Maind & Wankar, 2014).

Fig.3 shows a single layer feedforward network. The input layer consists of different symptoms and the output layer consists of various diseases. When a set of medical symptoms are given as an input to the network, the network gives the corresponding output i.e. the disease. Thus, feedforward networks are widely used in medical diagnosis.

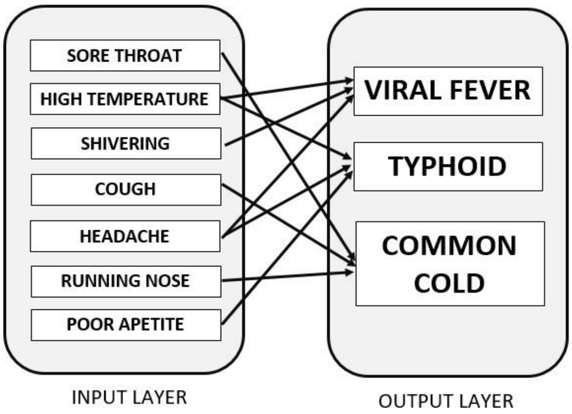


Fig 3: Single layer feedforward network

2.2 Multilayer Feedforward Network

They have a similar structure to single layer feedforward network in the sense that the layers are connected only in forward direction. But in multilayer feedforward networks there are one or more than one hidden layers and the computational units of these layers are called hidden neurons. Typically, the layers are fully connected meaning, each neuron of a layer is directed to each neuron of the subsequent layer. It is called multilayer because the processing and computations are performed by the hidden as well as output layers. Each connection between the nodes has a weight associated with it. Also, there is a special weight connected to the hidden and output layer separately called the bias which sets the thresholding value for the nodes.

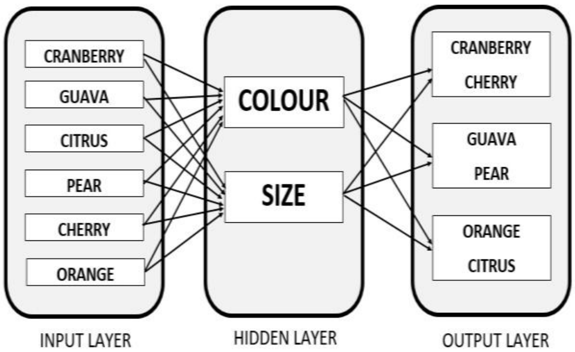


Fig 4: Multilayer feedforward network

Fig.4 shows a multilayer feedforward neural network. Inputs, in this case, fruits are given to the input layer. The hidden layer performs some computations and classifies the fruits according to their colour and size. The output obtained is classified fruits. For example cranberry and cherry are grouped together because their colour and size are similar- red and small. Multilayer feedforward networks are useful for classification and pattern recognition problems.

2.3 Recurrent Network

These networks differ with feedforward networks in the sense that there is at least one feedback loop. They are called recurrent network because they are a network of neurons with feedback connections. There could also be neurons with self-feedback connections like, the output of a neuron can be fed into itself as input. Recurrent networks are used for controlling a robot manipulator. Fig. 5 shows a recurrent network.

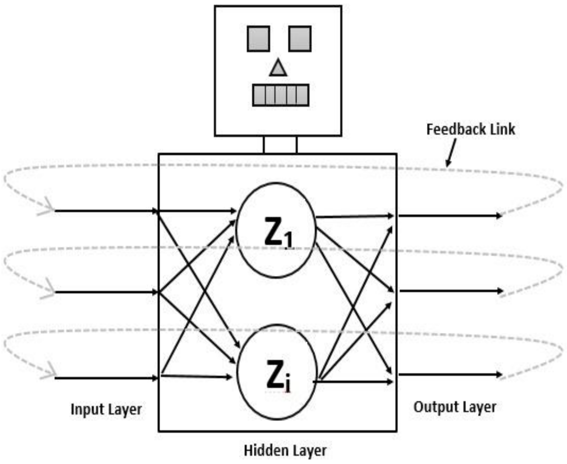


Fig 5: Recurrent Network

3. LEARNING METHODS

There are three major learning methods- supervised learning, unsupervised learning and reinforced learning.

3.1 Supervised Learning

In supervised learning, each example is a pair consisting of an input pattern and a desired output or target output pattern. A comparison is made between the network's computed output and the target output to determine the error. Using this error the network parameters are changed resulting in the improvement in performance.

Example:

1. Suppose there is a basket with 4 different fruits – apple, cherry, watermelon and grape.
2. Physical characters of the fruits are already known.
3. Now, arranging the same type of fruits at one place would be easy.
4. A new fruit is taken from the basket. Size, colour and shape of that particular fruit are observed.
5. If the size is small, colour is green and shape is round, therefore, the fruit is grape.
6. Similarly, other fruits are also observed.
7. Thus, supervised learning is very useful for classification.

3.2 Unsupervised Learning

Unsupervised learning is a type of machine learning algorithm used to draw inferences from input pattern without target output pattern. In this learning the network learns by itself by discovering and adapting to the structural features in the input pattern.

Example:

1. Suppose there is a basket and it is full of different types of fruits that are to be arranged into different groups.
2. This time there is no previous information about the fruits.
3. Now, a fruit is chosen and is arranged considering the physical characteristics of that particular fruit.
4. Suppose, colour of the fruit is considered first. Then the groups will be:

   a) RED COLOR GROUP: Apples and cherry.

   b) GREEN COLOR GROUP: Watermelons and grapes.

i. Now, another physical character is taken, such as size. The groups will be

   a) RED COLOR AND BIG SIZE: Apple.

   b) RED COLOR AND SMALL SIZE: Cherry.

   c) GREEN COLOR AND BIG SIZE: Watermelon.

   d) GREEN COLOR AND SMALL SIZE: Grape.

ii. Here, no information was provided beforehand i.e., target output was not presented. The system learnt by itself.

3.3 Reinforced Learning

In reinforced learning the target output is not presented to the network. It is only indicated if the computed output is correct or incorrect. The feedback helps the network in the learning process. A reward is given for a correct answer and a penalty for wrong answer. Reinforced learning is not one of the popular forms of learning. Reinforced learning is majorly used in the field of robotics.

4. MATLAB: THE TOOL

MATLAB is a software that creates an interactive environment for the users, letting them visualize their ideas and collaborating across multiple fields like signal and image processing, communications, control systems, and computational finance.

MATLAB student use software provides the same tools that professional engineers and scientists use every day (Mathworks).

The Neural Network Toolbox in MATLAB consists of various functions and apps for creating, training, and simulating neural networks. The apps make it easy to develop neural networks for tasks such as data-fitting (including time-series data), pattern recognition, and clustering.

4.1 Transfer Functions Available in MATLAB

Log-sigmoid transfer function

The log-sigmoid transfer function takes input that ranges from negative infinity to positive infinity and gives an output value that lies between 0 and 1. This function is commonly used for pattern recognition problems.

Syntax:

Y=logsig(N)

Where, Y is the output and N is the input.

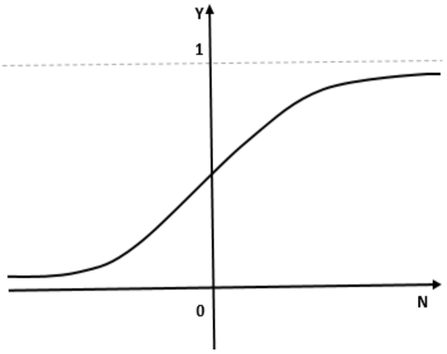


Fig 6: Log-sigmoid transfer function

Tan-sigmoid transfer function

The tan-sigmoid transfer function takes input that ranges from negative infinity to positive infinity and gives an output value that lies between -1 and 1. This function too is commonly used for pattern recognition problems.

Syntax:

Y=tansig(N)

Where, Y is the output and N is the input.

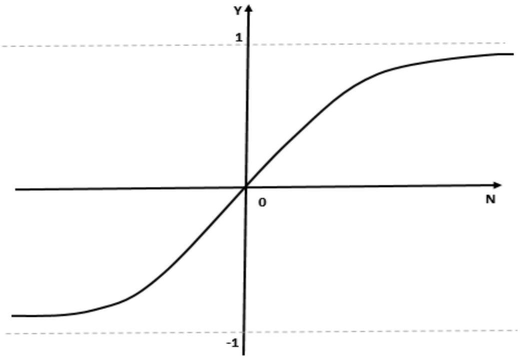


Fig 7: Tan-sigmoid transfer function

Linear Transfer function
The linear transfer function takes the net input of a layer and calculates its output.

This function is commonly used for fitting problems.

Syntax:

Y=purelin(N)

Where, Y is the output and N is the net input.

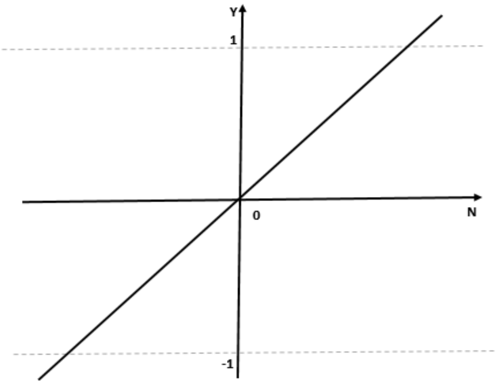


Fig 8: Linear transfer function

4.2 Neural network architectures available in MATLAB

4.2.1 Supervised Networks

Supervised networks are networks which work on supervised learning. There are 4 types of supervised networks in neural network toolbox:

i. **Feedforward networks**: These are the networks with one way connections i.e. each neuron of input layer is connected to each neuron of the output layer but not vice versa. This network is used for input to output mapping. Networks with one hidden layer with enough neurons can fit multidimensional mapping problems. They are mostly used in pattern recognition, prediction.

ii. **Radial basis networks**: These networks use radial basis functions as activation functions. They provide an alternative method for designing non-linear feed forward networks (Kon & Plaskota). It is used in function approximation, classification and time series prediction.

iii. **Dynamic networks:** In these networks the output not only depends on the current input but also on the current and previous inputs, outputs and states of networks. They are of two types- with feedforward connections and with recurrent connections. Prebuilt dynamic networks in the toolbox include focused and distributed time-delay, nonlinear autoregressive (NARX), layer-recurrent, Elman, and Hopfield networks. The toolbox also supports dynamic training of custom networks with arbitrary connections.

iv. **LVQ networks**: It stands for Learning Vector Quantization. It helps in classification of nonlinearly separable patterns.

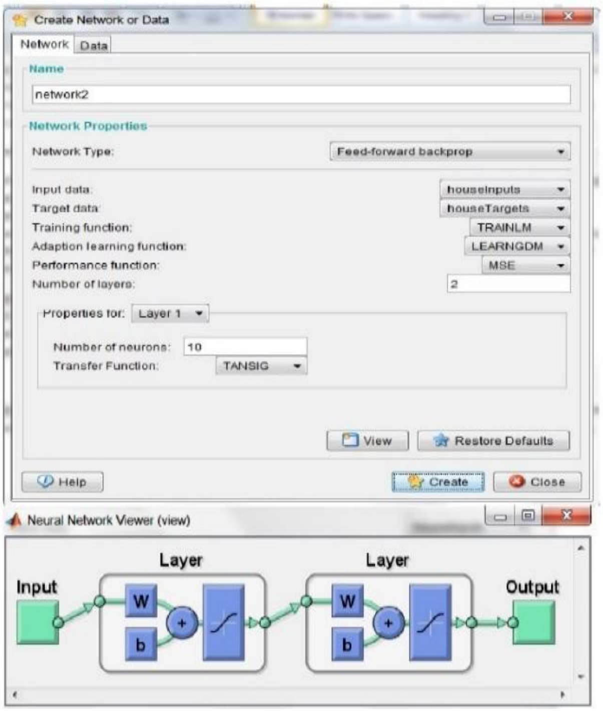
4.2.2 Unsupervised Networks

These are networks that run on unsupervised learning. Neural Network Toolbox includes two types of self-organizing, unsupervised networks:

i. **Competitive layers:** Neurons in this layer distribute themselves to recognize frequently presented input vectors. Competitive learning neural networks are a combination of both feedback and forward connection layers which results in the competition of the activation of the neurons in the output layer. Competitive layers are commonly used for classification and pattern recognition.

ii. **Self-organising maps**: They do classification of input vectors as they are grouped in input space. They are different from competitive learning as they learn both the distribution and to preserve the topology of input vectors (Vesanto & Alhoniemi, 2000). Like competitive layers, they are used for classification and pattern recognition tasks.

Using neural network toolbox, we can create various neural networks with different architectures. Through this we can select the number of hidden layers, import inputs, target data and train the network created. Errors can also be determined by drawing comparison between targets and the computed output. Finally, we can view the network architecture created. An example of feedforward network is shown in the figure given below:



Name
network2

Network Properties
Network Type: Feed-forward backprop

Input data: houseInputs
Target data: houseTargets
Training function: TRAINLM
Adaption learning function: LEARNGDM
Performance function: MSE
Number of layers: 2

Properties for: Layer 1
Number of neurons: 10
Transfer Function: TANSIG

Fig 9: Feedforward network created in MATLAB

4.3 Training algorithms available in MATLAB

The neural network toolbar has a number of training algorithms some of which are listed below:

1. **Gradient descent**: Gradient descent is an algorithm that is used to find local minima of a function. Given a function defined by a set of parameters, gradient descent starts with an initial set of parameter values and iteratively moves towards a set of parameter values that minimize the function.

2. C**onjugated gradient method:** The conjugated gradient method is an iterative technique used to solve sparse system of linear equations. The matrix of these equations must be symmetric and positive definite (Refsnæs, 2009).

3. **Levenberg-Marquardt (LM) algorithm:** The Levenberg-Marquardt (LM) algorithm is an iterative technique that locates the minimum of a function with several variables that is expressed as the sum of squares of non-linear real-valued functions. LM is a combination of steepest descent algorithm and the Gauss-Newton method.

4. **Resilient back propagation (Rprop):** Resilient backpropagation is an algorithm that can be used to train a neural network. It is similar to the regular back-propagation. Rprop has two main advantages over back propagation:

a) Training with Rprop is generally faster than training with back propagation.

b) Rprop doesn't require you to specify any free parameter values, as opposed to back propagation which needs values for the learning rate.

5. **Hebbian Learning:** Hebbian Learning Rule is a learning rule that specifies how much the weight of the connection between two units should be increased or decreased in proportion to the product of their activation (Munakata & Pfaffly, 2004).

6. **Learning vector quantization:** Learning vector quantization neural networks consist of two layers. The first layer maps input vectors into clusters that are found by the network during training. The second layer merges groups of first layer clusters into the classes defined by the target data.

7. **Widrow-Hoff learning:** Widrow-Hoff learning is similar to steepest descent learning. Widrow-Hoff learning is also known as least mean square learning. The LMS algorithm minimizes mean square error and therefore tries to move the decision boundaries as far from the training patterns as possible (Sivanandam, Sumathi, & Deepa, 2005).

8. **Kohonen neural network:** The Kohonen neural network does not use any sort of activation function. Further, the Kohonen neural network does not use any sort of a bias weight. Output from the Kohonen neural network does not consist of the output of several neurons. When a pattern is presented to a Kohonen network one of the output neurons is selected as a "winner". This "winning" neuron is the output from the Kohonen network. Often these "winning" neurons represent groups in the data that is presented to the Kohonen network. (Heaton, 2008)

In MATLAB these training algorithms can be applied on different input data sets and the neural network can be trained.

Procedure:

- First a neural network is created using NNTOOL. The input and the target data can be set according to the requirement. Any of the example datasets available in MATLAB can be imported. Here, the house pricing dataset is chosen as an example.

- The network type, training function, adaption learning function, performance function, number of layers, number of neurons, and transfer function are then set. The network type, in this case, is set as feedforward backpropagation adaption learning function as gradient descent with momentum weight and bias learning function, performance function as mean squared normalized error, number of layers 2, number of neurons 10, and transfer function tan sigmoidal.

- Different networks with different training functions are then created and trained.

- Hence, a regression graph for training, testing, validation, and one for all of the 3 is obtained.

- The network is trained 10 times each with different training algorithms and an average value for training, testing, validation and all of the three for the respective training algorithms is calculated. The values received are shown in Table 1.

- Now, the different values received, are plot on the graph and the training algorithm which gives the minimum house price is chosen.

- As it is clear from the graph (Fig. 10) variable learning rate backpropagation gives the best value of house price.

Table 1. Training of house pricing dataset using different Training algorithms

S NO.	TRAINING ALGORITHM	TRAINING: Regression	VALIDATION: Regression	TEST: Regression	ALL: Regression
1	Levenberg-Marquardt	0.98062	0.91136	0.9420034	0.958788
2	BFGS Quasi-Newton	0.849305	0.907561	0.919339	0.877887
3	Resilient Backpropagation	0.912349	0.895948	0.923835	0.906393
4	Scale Conjugate Gradient	0.919239	0.892487	0.904625	0.909732
5	Conjugate Gradient with Powell/Bealle Restarts	0.868886	0.879313	0.905837	0.870265
6	Fletcher-Powell Conjugate Gradient	0.899073	0.921032	0.930152	0.897146
7	One step secant	0.817638	0.884656	0.907148	0.829199
8	Polak-Ribiére Conjugate Gradient	0.885801	0.937925	0.929661	0.965748
9	Variable Learning Rate Backpropagation	0.804313	0.77619	0.773769	0.787731

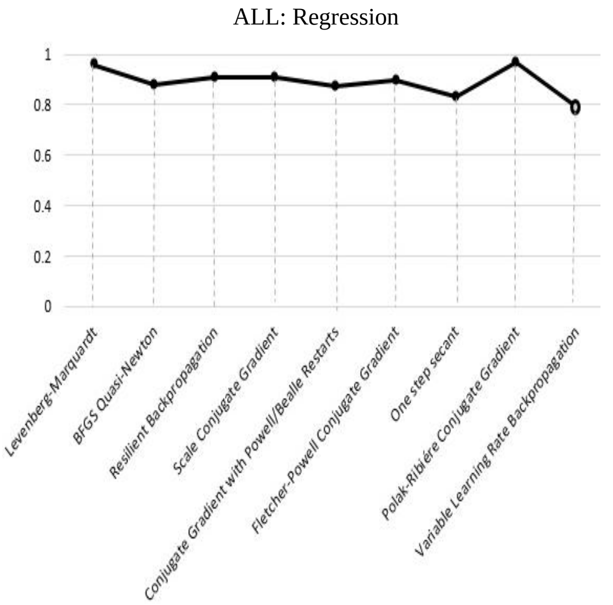


Fig 10: Graph of different values of regression for different training algorithms

5. APPLICATIONS

Neural networks are applied in miscellaneous fields today. Some of which are listed below:

1. Medical sciences: Artificial neural networks provide a powerful tool to help doctors to analyse, model and make sense of complex clinical data across a broad range of medical applications. They are used in cancer diagnosis, cardio vascular system modelling. (Ohno-Machado, 1996)
2. Business:
    a) Security and trading companies are improving their forecasting techniques with ANNs.
    b) Customer research.
    c) Target marketing.
3. Pattern recognition: Image analysis through the implementation of artificial neural network techniques has helped in the field of robotics and intelligent software processes (Basu, Bhattacharyya, & Kim, 2010).
4. Forecasting and risk management: ANNs have the capability to predict situations from past trends and hence have been used in stock market, metallurgy.
5. Example: NASA launched intelligent flight control system (IFCS) research with the goal to develop adaptive and fault tolerant flight control systems and improve aircraft performance in critical conditions. To achieve this goal it integrates self-learning neural network concepts into flight control software (NASA, 2014).
6. Self-diagnosis of medical problems using neural networks. (P.Gaikwad, 2012)

6. CONCLUSION AND FUTURE SCOPE

In this paper we discussed about the artificial neural network, creation of ANN and its training using MATLAB. Upon application of 9 different training algorithms on the house pricing dataset available in MATLAB and training the network 10 times each for each training algorithm it was found that variable learning rate backpropagation gives the best result.

NNs might, in the future, allow:

1. Hybrid neuro-fuzzy systems: A system that combines a neural network with a fuzzy system resulting in neuro-fuzzy systems. These systems can provide human-like expertise such as domain knowledge, uncertain reasoning, adaption to a noisy and time-varying environment, and are important in tackling practical computing problems.
2. Robots that can see, feel, and predict the world around them.
3. Improved stock prediction.
4. Self-driving cars.
5. Handwritten documents can be automatically transformed into formatted word processing documents.

7. REFERENES

[1] G. A. V. Pai and S. Rajasekaran, Neural Networks, Fuzzy Logic, And Genetic Algorithms: Synthesis And Applications, PHI Learning Pvt. Ltd., 2003.

[2] Graves, Alex, Mohamed, Abdel-rahman and Hinton, Geoffrey. Speech Recognition with Deep Recurrent Neural Networks. 2013.

[3] A.Minin, "The Neural Network Analysis," 2006.

[4] E. Kumar and E. Sharma, "Artificial Neural Networks-A Study," International Journal of Emerging Engineering Research and Technology, vol. 2, no. 2, pp. 143-148, 2014.

[5] O. Eluyode and D. T. Akomolafe, "Comparative study of biological and artificial neural networks," European Journal of Applied Engineering and Scientific Research, vol. 2, no. 1, pp. 36-46, 2013.

[6] M. S. B. Maind and M. P. Wankar, "Research Paper on Basic of Artificial Neural Network," International Journal on Recent and Innovation Trends in Computing and Communication , vol. 2, no. 1, pp. 96-100, 2014.

[7] "Mathworks," [Online]. Available: http://in.mathworks.com/products/matlab/.

[8] M. A. Kon and L. Plaskota, "Neural Networks, Radial Basis Functions, and Complexity".

[9] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 11, no. 3, pp. 586-600, 2000.

[10] R. H. Refsnæs, "A BRIEF INTRODUCTION TO THE CONJUGATE GRADIENT METHOD," 2009.

[11] Y. Munakata and J. Pfaffly, "Hebbian learning and development," Developmental Science , vol. 7, no. 2, pp. 141-148, 2004.

[12] S. N. Sivanandam, S. Sumathi and S. N. Deepa, Introduction to Neural Networks Using MATLAB 6.0, Mcgraw Hill Education, 2005.

[13] J. Heaton, Introduction to Neural Networks with Java, Heaton Research, Inc.; 1 edition, 2008.

[14] Importance of Artificial Neural Network in Medical Diagnosis disease like acute nephritis disease and heart disease. Khan, Irfan Y., Zope, P.H. and Suralkar, S.R. 2, Jalgaon : International Journal of Engineering Science and Innovative Technology (IJESIT), 2013, Vol. 2. ISSN: 2319-5967.

[15] J. K. Basu, D. Bhattacharyya and T.-h. Kim, "Use of Artificial Neural Network in Pattern Recognition," International Journal of Software Engineering and Its Applications, vol. 4, no. 2, pp. 24-34, 2010.

[16] "NASA," 1 March 2014. [Online]. Available: http://www.nasa.gov/centers/armstrong/news/FactSheets/FS-055-DFRC.html.

[17] P. P.Gaikwad, "SELF MEDICAL DIAGNOSIS USING ARTIFICIAL NEURAL NETWORK," Int.J.Computer Technology & Applications, vol. 3, no. 6, pp. 2006-2013, 2012.