Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 6, June 2015, pg.76 – 85

SURVEY ARTICLE

Back-Propagation Neural Network for Speech Recognition- A Survey

Shashi[1], Sonia[2]
[1]M.Tech student & DPGITM College (MDU Rohtak), India
[2]Assistant professor & DPGITM College (MDU Rohtak), India
[1] shashiahlawat9@yahoo.in; [2] sonya_leo2007@yahoo.co.in

Abstract— ***This research present a review of BACK PROPAGATION NEURAL NETWORKS (BPNN) on speech recognition system. Speech user interface to computer is further an important step that technology needs to take for people. BPNN is principle procedure for training feed-forward neural networks. It is a systematic method of training multilayer artificial neural network. It is built on high mathematical foundation and has very good application potential. All important variables of Back Propagation Algorithm (BPA) that influence BPA are discussed. This paper is outline the basic back-propagation and periodically improvements over back-propagation technique.***

Keywords — ***ANN, BPNN, ASR, Recognition system, Adalines.***

I. INTRODUCTION

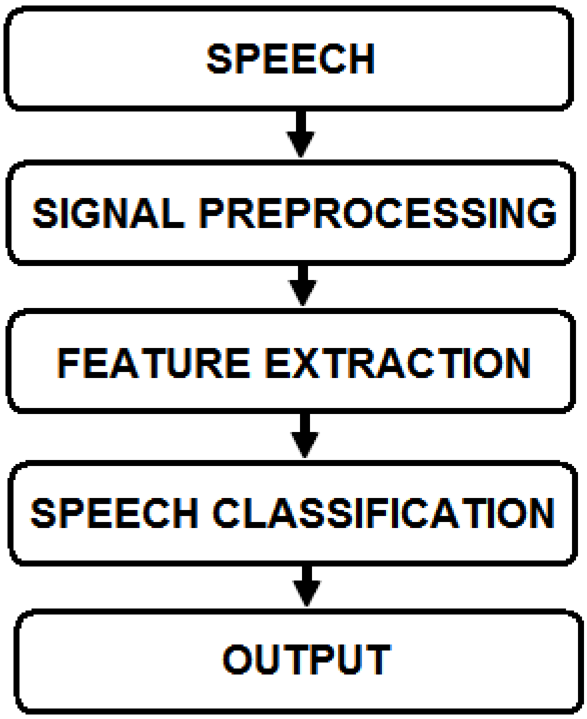
An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the biological nervous systems, such as the brain. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. A Neural Network is configured for pattern recognition or data classification, through a learning process. In biological systems, Learning involves adjustments to the synaptic connections that exist between the neurons. Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements working in parallel to solve a specific problem. Neural networks learn by example. A neuron has many inputs and one output. The neuron has two modes of operation (i) the training mode and (ii) the using mode. In the training mode, the neuron can be trained 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 training rule is used. Neural network has many applications. The most likely applications for the neural networks are (1) Classification (2) Association and (3) Reasoning. An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward neural network that has been trained accordingly. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern.

II. AUTOMATIC SPEECH RECOGNITION

Acoustic pattern recognition determines a reference model which best matches the input speech, as an output. Acoustic modelling, naturally posed as a static pattern matching problem is amenable to neural networks. Many ASR systems in existence employ DTW or HMM for feature recognition. DTW method measures the distance between each reference frame and each input frame using the dynamic algorithm to obtain the best warping of the pattern. HMMs characterize speech signals using a pre-trained Markov chain. But, some difficulties still exist in such ASR systems, since speech recognition is a complex phenomenon due to the asymmetries involved in speech production and speech interpretation. For effective results, ASR can employ an approach that is closer to human perception. Neural networks are modelled after the human brain. Hence, we use neural network for feature recognition in our ASR system [1], [2].

A. RECOGNITION SYSTEM

Fig. 1 shows the speech recognition process can generally be divided in many different components illustrated in Fig. 1



- The first block, which consists of the acoustic environment plus the transduction equipment (microphone, preamplifier and AD-converter) can have a strong effect on the generated speech representations. For instance we can have additional impact generated from additive noise or room reverberation.
- The second block is intended to deal with these problems, as well as deriving acoustic representations that are both good at separating classes of speech sounds and effective at suppressing irrelevant sources of variation.

- The third block must be capable of extracting speech specific features of the pre-processed signal. This can be done with a variety of techniques like cepstrum analysis and the spectrogram.
- The fourth block tries to classify the extracted features and relates the input sound to the best fitting sound in a known 'vocabulary set' and represents this as an output.

III. ARTIFICIAL NEURAL NETWORKS

Many tasks involving intelligence or pattern recognition are extremely difficult to automate, but appear to be performed very easily by human beings. Human beings recognize various objects, apparently with very little effort. The neural network of human beings contains a large number of interconnected neurons. Artificial neural networks are the computing systems whose theme is borrowed from the analogy of biological neural networks [2], [4].

Neural network is a useful tool for various applications which require extensive classification. The advantage of parallel processing in neural networks and their ability to classify the data based on features provides a promising platform for pattern recognition. Traditional sequential processing techniques have limitations for implementing pattern recognition problems in terms of flexibility and cost whereas neural networks perform the processing task by training instead of programming in a manner analogous to the way human brain learns. Unlike the traditional sequential machines where rules and formula need to be specified explicitly, a neural network learns its functionality by learning from the samples presented [3], [7].

A. Characteristics of artificial neural networks

Artificial neural networks have labelled directed graph structure where nodes perform some computations. They consist of a set of nodes and a set of connections connecting pair of nodes. Each connection carries a signal from one node to another. Label represents the connection strength or weight indicating the extent to which signal is amplified or diminished by a connection. Different choices for the weights result in different functions being evaluated by the network. Weights of the network are initially random and a learning algorithm is used to obtain the values of the weights to achieve the desired task. A graph structure, with connection weights modifiable using a learning algorithm, results in a network called artificial neural network. Neural network stores the knowledge specific to a problem in the weights of connections using learning algorithm [3], [7].

B. Classification

Classification means assignment of each object to a specific class or group. It is of fundamental importance in a number of areas ranging from image and speech recognition to the social sciences. We use a training set consisting of sample patterns representing all classes, along with class membership information for each pattern. Using the training set, rules for membership in each class are deduced to create a classifier, which later assigns other patterns to their respective classes according to these rules. We use neural networks to classify samples, i.e., map input patterns to different classes. Each output node can stand for one class. An input pattern is determined to belong to class *i* if the *i*th output node computes a higher value than all other output nodes when that input pattern is fed into the network [3], [4].

C. Perceptrons and Linear separability

Perceptron is a machine that learns using examples i.e. training samples to assign input vectors to different classes. Perceptron uses a linear function of inputs. Perceptron has a single output whose value determines to which class each input pattern belongs and it is represented by a single node that applies a step function to the net weighted sum of its inputs.

If there exists a line, whose equation is w0+w1x1+w2x2 = 0, that separates all samples of one class from the other perceptron, with weights w0, w1, w2 for the connections from inputs 1, x1, x2, respectively, can be derived from the equation of class, then a that line. Such classification problems are said to be linearly separable, since they are separable by a linear combination of inputs. The inter-relationship between perceptron weights and the coefficients of terms in the equations of lines holds true for the converse as well [3], [7]..

D. Limitations of using perceptrons

If there are three input dimensions, a two class problem can be solved using a perceptron only if there is a plane that separates samples to different classes. For simple examples and two dimensional spaces it is relatively easy to determine by geometric construction whether two classes are linearly separable. But it becomes very difficult for higher dimensional spaces. If no line can separate samples belonging to two different classes i.e., the samples are not linearly separable, then a simple perceptron cannot classify the samples. It is the fundamental limitation of simple perceptron. Real life classification problems are linearly non-separable and hence perceptron training algorithm cannot achieve accurate results for such classification problems [3]. A robust algorithm would achieve a reasonable separation between most of the samples of the two classes. Two algorithms achieve robust classification for linearly non-separable classes - pocket algorithm and least mean square algorithm. The LMS algorithm minimizes the mean square error instead of the number of misclassified samples, while the pocket algorithm stores information about the better weight vectors observed in the process of modifying weights [3], [7].

E. Adalines

Robust recognition may also be achieved by minimizing the mean square error (MSE) instead of the number of misclassifies samples. An adaptive linear element or adaline accomplishes classification by modifying weights in such a way as to minimize the MSE at every iteration training. This can be achieved using gradient descent, since MSE is a quadratic function whose derivative exists everywhere. When the sample input is presented during training the network, the linear weighted net input is computed and compared with the desired output for that sample, generating an error signal. This error signal is used to modify each weight in the adaline. Unlike the perceptron training algorithm, weight changes are made to reduce MSE even when a sample is correctly classified by the network [3], [7].

F. Supervised learning using multi-layer networks

Perceptron approach can be extended to solve linearly non-separable classification problems, using layered structure of nodes. Such networks contain one or more layers of hidden nodes that isolate useful features of the input data. However it is not easy to train these networks. Given that the network makes an error on some sample inputs, identifying which weights in the network must be modified, and to what extent is a tough task. Hence, perceptron and other one layer networks are seriously limited in their capabilities. Feed-forward multilayer networks with non-linear node functions can overcome these limitations, and can be used for many applications. Hence a more powerful supervised learning mechanism called back-propagation is used for multi-class, multi-level discrimination [3], [5].

IV. BACK-PROPAGATION NETWORKS

Back-Propagation Neural Network (BPNN) algorithm is the most popular and the oldest supervised learning multilayer feed-forward neural network algorithm proposed by Rumelhart, Hinton and Williams [2]. The back propagation algorithm is the modification of least mean square algorithm. It modifies network weights to minimize the mean squared error between the actual and desired outputs of the network. Back propagation algorithm makes use of supervised learning in which the network is trained using training samples for which inputs as well as desired outputs are known. The weights are frozen once the network is trained and it can be used to compute output values for new input samples. The feed forward process involves presenting an input pattern to input layer nodes that pass the input values onto the first hidden layer. Each of the hidden layer nodes computes a weighted sum of its inputs and passes the sum through its activation function before presenting the result to the output layer. An error at a higher layer of multi-layer network is propagated backwards to nodes at lower layers of the network. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function. The Basic Equation of a Back Propagation Algorithm are:- Wt+1 = wt +-For weight *A. Components of BPA* Back Propagation algorithm uses Gradient descent learning rule which requires careful selection of parameters such as initial weights and biases, learning rate value, activation function should be selected carefully. An improper choice of these parameters can lead to slow network convergence, network error or failure.

a. Activation Function

It is a function used to transform the activation level of a unit (neuron) into an output signal. Activation functions are also called as transfer functions. It is the function that is applied to the net output of any node before it is fed to the next layer.

The activation function is used for 2 things. To make the unit to be active (near +1) when the right inputs are given and to make inactive (near 0) when the wrong inputs are given. Second to make the activation nonlinear, otherwise the entire neural network collapses into a simple linear function. Some basic types of Activation Functions are Identity Function, Step function, Sigmoidal function. The output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units.

b. Learning rate coefficient (η) Learning rate coefficient determines the size of the weight adjustments made at each iteration and hence influences the rate of convergence. Poor choice of the coefficient can result in failure in convergence. Too fast or too slow of a learning rate is detrimental to network convergence.

c. Momentum Term (α) There is another way possible to improve the rate of convergence by adding some inertial or momentum to the gradient expression. This can be accomplished by adding a fraction of the previous weight change to the current weight change. A commonly used update rule introduced by Rumelhart et al. includes such a momentum term. The updating equations used by Rumelhart are defined as: $[\Delta W]t+1= -\eta(dE/dW) + \alpha\ [\Delta W]t$ It is added to smooth out oscillation, increase convergence speed.

d. Proportional Factor(β) Standard Back-propagation Algorithm (BP) usually utilizes two term parameters; Learning Rate α and Momentum Factor β, but sometimes a 3rd term called proportional factor is also added to increase convergence speed, escape from local minima.

e. Cost Functions Back-propagation algorithm uses Mean Squared Error (MSE) cost function. But it has been observed that Mean squared error (MSE) cost function has some drawbacks such as incorrect saturation and tend to trap in local minima, resulting in slow convergence and poor performance. MSE gives more emphasis on reducing the larger errors as compared to smaller errors due to the squaring that takes place. Hence research has been done to find a better cost function and new cost functions such as Bernoulli error measure (Chow et al.), New Modified cost function (Samsuddin et al.), Classification-Based (CB) cost functions (Rimer Martinez) etc. have been proposed.

V. RESEARCH SO FAR

The backpropagation (BP) algorithm is widely recognized as a powerful tool for training feed-forward neural networks. But since it applies the steepest descent method in updating the weights, it suffers from a slow convergence. Several iterations are required to train a small network, even for a simple problem. Much work, therefore has been done in search of faster methods.

Table1: MAJOR CONTRIBUTION AND STUDY IN AREA OF BPNN

YEAR	RESEARCHER	CONCLUSION OF WORK
1958	Rosenblatt,Frank	Introduced the PERCEPTRON Model
1969	Minsky,M.L ad Papert, S[1].	Perceptrons:An introduction to computational geometry
1986	Rumelhart et al.	Introduce the error back propagation method to train multilayer feed forward networks
1987	Richard P. Lippmann	Given the introduction to computing with neural nets and its different algorithms and classification problems including single layer perceptron and kohonan and multilayer perceptron and introduction of BACK PROPAGATION algorithm.
1988	Robert A. Jacobs[4].	To achieve faster rates of convergence than steepest descent algorithms he has examined two implementations namely the momentum and delta-bar-delta and the hybrid of them also.
1988	Scott E. Falhman[5].	Compared the Quick propagation against standard back propagation algorithm using test over benchmark problems like Exclusive-Or problem, Encoder problem and found positive results.
1993	Yam and Chow[6].	According to the coefficient of correlation between the prior weight change and the downhill gradient the momentum factor and the learning rate are modified
1994	Verma and Mulawka[7].	By solving , weight matrix for the output layer of the network using least squares method and theory of equations, adjustment is made .
1995	Drago et al[8].	An Adaptive Momentum Back Propagation (AMBP,ABP, Accelerated Learning) is said to attain very satisfying performance for achieving fast minimum search. The network weight update rule is chosen such that the error function is forced to behave in a certain manner that accelerates convergence. Besides the good convergence speed, a high generalization capability has been achieved,
1995	Bossan et al[9].	This technique try to decrease the time which has been spent rigorously for attaining low MSE in those dense regions of the pattern space while

		ignoring patterns in sparse regions of it until large number of training epochs occur.
1997	Chen et al[10].	A randomized BP algorithm in which a series of weight vectors are chosen over the learning phase is proposed.
1998	Fukuoka et al[11].	Each connecting weight in a network is multiplied by a factor in the range of (0,1] at a constant interval during a learning process. The basic idea of the method is to keep sigmoid derivative relatively large while some of the error signals are large.
1999	Ng and Leung[12].	The new proposed back-propagation algorithm is to change the derivative of the activation function so as to magnify the backward propagated error signal, thus the convergence rate can be accelerated and the local minimum can be escaped.
2000	Wen et al.	An adaptive backpropagation algorithm which can update learning rate and inertia factor automatically based on dynamical training error rate of change.
2001	Abid et al.	This approach minimizes a modified form of the criterion used in the standard backpropagation algorithm. This criterion is based on the sum of the linear and the nonlinear quadratic errors of the output neuron.
2002	Yu and Liu[13].	BPALM (Backpropagation with adaptive learning rate and momentum term) - adaptive learning rate and momentum term where the learning rate and momentum factor are adjusted at each iteration to reduce the training time.
2003	Zweiri et al[14].	Besides learning rate and momentum factor of backpropagation algorithm a new third term called proportional factor is proposed to fasten the weight adjustment process.
2004	Wang et al.	Improved BP where each training pattern has its own activation function of neurons in hidden layer to avoid local minima.
2005	Pernia-Espinoza et al.	The benfit of the non-linear regression model - estimates (introduced by Tabatabai and Argyros, 1993) is combined with the backpropagation algorithm to produce the TAO-robust learning algorithm.
2006	Kathivalavakumar and Thangavel.	A new technique and optimization criterion is proposed to train single hidden layer FFNN where it trains the hidden layer and output layer independently.
2007	Wang et al.	An Individual Inference Adjusting Learning

		Rate technique (IIALR) is proposed to enhance the learning performance of the BPNN.
2007	Sammy Siu et al[15].	By using Evolutionary technique Back propagation algorithm is improved.
2007	Guijarro and Fontenla[16].	An algorithm which applies linearleast-squares is proposed. It combines linear-least-squares with gradient descent. It improves the learning rate of the basic backpropagation algorithm in several orders of magnitude, while maintaining good optimization accuracy.
2008	Bumghi, Ju, Deok[18].	A novel idea is proposed to solve the LOCAL MINIMA problem faced in FFNN.
2011	Kavita Burse,M.manoria,vishnu[19].	Implemented Zweiri's three -term BPLA over XOR problem and found it is very easy to solve the local minima problem in Multiplicative Neuron Model.
2011	Zhen G che,Zhen H che,Tzu[20].	Compared the BPLA with the Genetic algorithm and found in some cases it is faster than even Genetic and not much complex.
2011	Chukwuchekwa Ulumma Joy[21].	By using pattern recognition problems, comparisons are made based on the effectiveness and efficiency of both backpropagation and genetic algorithm training algorithms on the networks. The backpropagation algorithm is found to outperform the genetic algorithm in this instance.
2013	Yeremia, Hendy, et al[22].	In this study, backpropagation network algorithm is combined with genetic algorithm to achieve both accuracy and training swiftness for recognizing alphabets. The training time needed for backpropagation learning phase improved significantly from 03 h, 14 min and 40 sec, a standard backpropagation training time, to 02 h 18 min and 1 sec for the optimized backpropagation network. A hybrid approach proves to be improvising the network performance.
2014	Geraldo Miguez, AdilsonElias Xavier and NelsonMaclan	It is suggest that by replacing bihyperbolic functions the backpropagation networks performs better than traditional Sigmoid functions.

VI. CONCLUSIONS

Back-propagation neural networks employ one of the most popular neural network learning algorithms, the Back-propagation (BP) algorithm. It has been used successfully for wide variety of applications, such as speech or voice recognition, image pattern recognition, medical diagnosis, and automatic controls. Algorithm is known for its mathematical simplicity and accuracy. It is the simplicity that attracts researchers and so that, many improvements and variations of the BP learning algorithm have been reported to be at its limitations such as slow convergence rate and convergence to the local minima. Significant research has been done to overcome these problems and different variations of BPA have been proposed. From this paper we can conclude that even though several variations and different techniques have been suggested to improve the performance of BPA there is still room for further research.

REFERENCES

[1] Minsky, M. L., and S. Papert. "Perceptrons: an introduction to computational geometry. Expanded." (1988).

[2] Rumelhart, D.E., Hinton, G. E., and Williams, R, J. Learning Internal Representations by Error Propagation.1986

[3] Lippmann, Richard. "An introduction to computing with neural nets." ASSP Magazine, IEEE 4.2 (1987): 4-22.

[4] Jacobs, Robert A. "Increased rates of convergence through learning rate adaptation." Neural networks 1.4 (1988): 295-307.

[5] Fahlman, S.E. (1988a), An Empirical Study of Learning Speed in Backpropagation Networks, Carnegie Mellon Report, No CMU- Cs, pp. 88-162.

[6] Yam, Y. F., and T. W. S. Chow. "Extended backpropagation algorithm." Electronics Letters 29.19 (1993): 1701-1702.

[7] Verma, B. K., and J. J. Mulawka. "A modified backpropagation algorithm." Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on. Vol. 2. IEEE, 1994.

[8] Drago, Gian Paolo, Mauro Morando, and Sandro Ridella. "An adaptive momentum back propagation (AMBP)." Neural Computing & Applications 3.4 (1995): 213-221.

[9] Bossan, M. C., et al. "A modified backpropagation algorithm for neural classifiers." Circuits and Systems, 1995., Proceedings., Proceedings of the 38th Midwest Symposium on. Vol. 1. IEEE, 1995.

[10] Xiao-hu yu ,Guo-an Chen, Efficient Backpropagation learning using optimal learning rate and momentum. Neural networks Vol. 10. No. 3. pp 517-527,1997.

[11] Fukuoka, Yutaka, et al. "A modified back-propagation method to avoid false local minima." Neural networks 11.6 (1998): 1059-1072.

[12] Ng, Sin Chun, Shu Hung Leung, and Andrew Luk. "Fast convergent generalized back-propagation algorithm with constant learning rate." Neural Processing Letters 9.1 (1999): 13-23.

[13] Yu, Chien-Cheng, and Bin-Da Liu. "A backpropagation algorithm with adaptive learning rate and momentum coefficient." Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on. Vol. 2. IEEE, 2002.

[14] Zweiri, Yahya H., James F. Whidborne, and Lakmal D. Seneviratne. "A three-term backpropagation algorithm." Neurocomputing 50 (2003): 305-318.

[15] Siu, Sammy, et al. "Improving the back-propagation algorithm using evolutionary strategy." Circuits and Systems II: Express Briefs, IEEE Transactions on 54.2 (2007): 171-175.

[16] Guijarro-Berdiñas, Bertha, et al. "A linear learning method for multilayer perceptrons using least-squares." Intelligent Data Engineering and Automated Learning-IDEAL 2007. Springer Berlin Heidelberg, 2007. 365-374.

[17] Shamsuddin, Siti Mariyam, Maslina Darus, and Fadhlina Izzah Saman. "Three term backpropagation algorithm for classification problem." Neural Network World 17.4 (2007): 363.

[18] Bumghi Choi,Ju-Hong Lee,Deok-Hwan Kim, Solving local minima problem with large number of hidden nodes on two-layer feed forward artificial neural networks Neurocomputing 71 (2008).

[19] Burse, Kavita, Manish Manoria, and Vishnu Pratap Singh Kirar. "Improved back propagation algorithm to avoid local minima in multiplicative neuron model."Information Technology and Mobile Communication. Springer Berlin Heidelberg, 2011. 67-73.

[20] Che, Zhen-Guo, Tzu-An Chiang, and Zhen-Hua Che. "Feed-forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm." Int J Innov Comput Inf 7 (2011): 5839-5850.

[21] Joy, Chukwuchekwa Ulumma. "Comparing the Performance of Backpropagation Algorithm and Genetic Algorithms in Pattern Recognition Problems."International Journal of Computer Information Systems 2.5 (2011): 7-12.

[22] Yeremia, Hendy, et al. "GENETIC ALGORITHM AND NEURAL NETWORK FOR OPTICAL CHARACTER RECOGNITION." Journal of Computer Science 9.11 (2013): 1435.

[23] Rajasekaran, Sundaramoorthy, and GA Vijayalakshmi Pai. *NEURAL NETWORKS, FUZZY LOGIC AND GENETIC ALGORITHM: SYNTHESIS AND APPLICATIONS (WITH CD)*. PHI Learning Pvt. Ltd., 2003.

[24] Falas, Tasos, and A-G. Stafylopatis. "The impact of the error function selection in neural network-based classifiers." *Neural Networks, 1999. IJCNN'99. International Joint Conference on*. Vol. 3. IEEE, 1999.

[25] Rimer, Michael, and Tony Martinez. "CB3: an adaptive error function for backpropagation training." *Neural processing letters* 24.1 (2006): 81-92.