ARTIFICIAL NEURAL NETWORK: A REVIEW

Jaswinder Kaur[1], Neha Gupta[2]
E-Mail Id: [1]jasukaur@rediffmail.com, [2]nehagupta@ansaluniversity.edu.in
School of Engineering & Technology, Ansal University, Gurgaon, India

Abstract- In this paper an introduction of Artificial Neural Network is presented. Learning Algorithms like Supervised Algorithms, Reinforcement Algorithms and Unsupervised Algorithms are discussed. Also, optimization methods like Gradient Descent, Newton Method, Conjugate Gradient Method, Quasi Newton and Levenberg Marquardt are presented.

Keywords: Artificial Neural Network Supervised Algorithms, Reinforcement Algorithms, Unsupervised Algorithms, Gradient Descent, Newton Method, Conjugate Gradient Method, Quasi Newton and Levenberg Marquardt.

1. INTRODUCTION

Artificial Neural Networks are almost used in most of the applications nowadays. It is an expanding field. Researchers from a number of disciplines are attracted to Artificial Neural Networks [1]. It includes fields like neuroscience, psychology, medicine, physics, mathematics etc. [2]. It is applied in wide variety of applications which includes heart attack diagnosis, character recognition, face recognition, speech recognition, cancer diagnosis, image classification, customer segmentation, gaming, market forecasting, classification of mails (spam and non spam) etc. [3].

It is a system which processes information same as in case of human beings. It has same structure and principles for operations as in human brain (interconnected cells of brain known as neurons process the information operating in parallel) [4]. The neuron was designed by Rosenblatt in 1958 and named as perceptron [5].

2. ARTIFICIAL NEURAL NETWORK

An artificial neural network consists of input nodes, hidden nodes and output nodes [6]. Usually there are three layers namely input layer, hidden layer and output layer.

The number of input nodes and output nodes are decided by the problem to be solved. There is a bias input (B) always set to 1. It provides flexibility in network learning. The input node is given input information like X_1, X_2……..X_n.

A connection weight is the only adjustable parameter in a neural network. During training the values of these weights are increased or decreased for emphasizing the connections between nodes [7]. Weights W_1, W_2……W_n are assigned to connections between nodes.

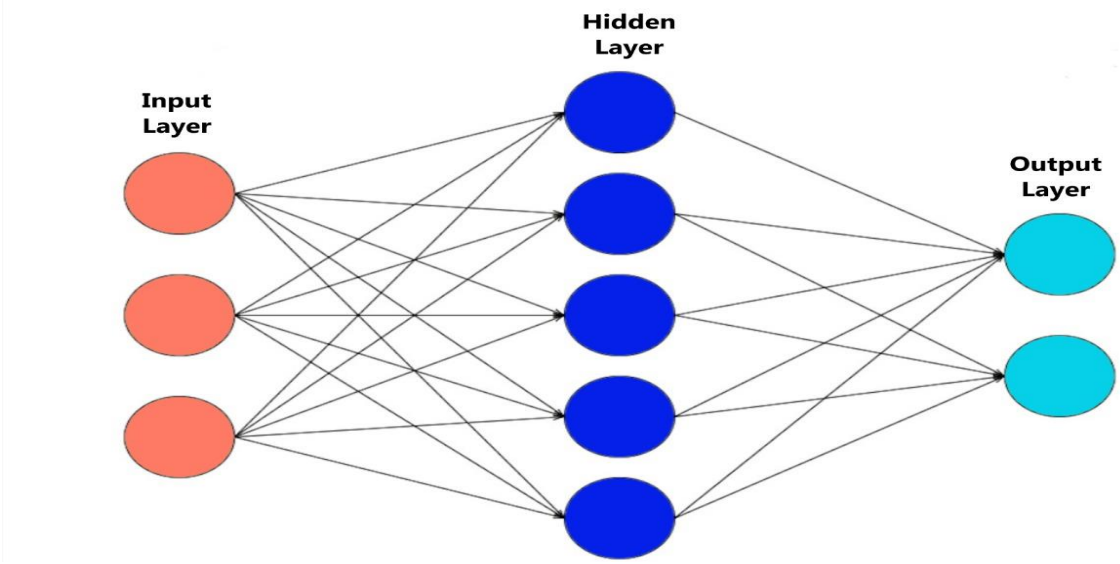

Fig. 2.1 An Artificial Neural Network

The output of hidden node is weighted sum of input values and connection weights given by equation below

$$H_i = \sum X_i W_i \quad (1)$$

The output Y_i at output layer is given by

$$Y_i = F(X_i) \quad (2)$$

Where $F(X_i) = F(H_i + B)$ and $F(X_i)$ is an activation function (i.e. sigmoid, linear, tanh, threshold etc.) to add non linearity [8].

3. LEARNING ALGORITHMS

The learning algorithms are used for training the neural network (NN). These algorithms are of three types.

3.1 Supervised Algorithms

It has labeled data set means the target value is known in advance. It uses training data set. It helps in predicting. Techniques like classification, ranking and regression come under supervised learning techniques [9].

It helps in weather forecasting, classification of images, detection of fraud, market analysis and forecasting, estimation of life expectancy, email spam detection, diagnostics, risk assessment and prediction of popularity of an advertisement.

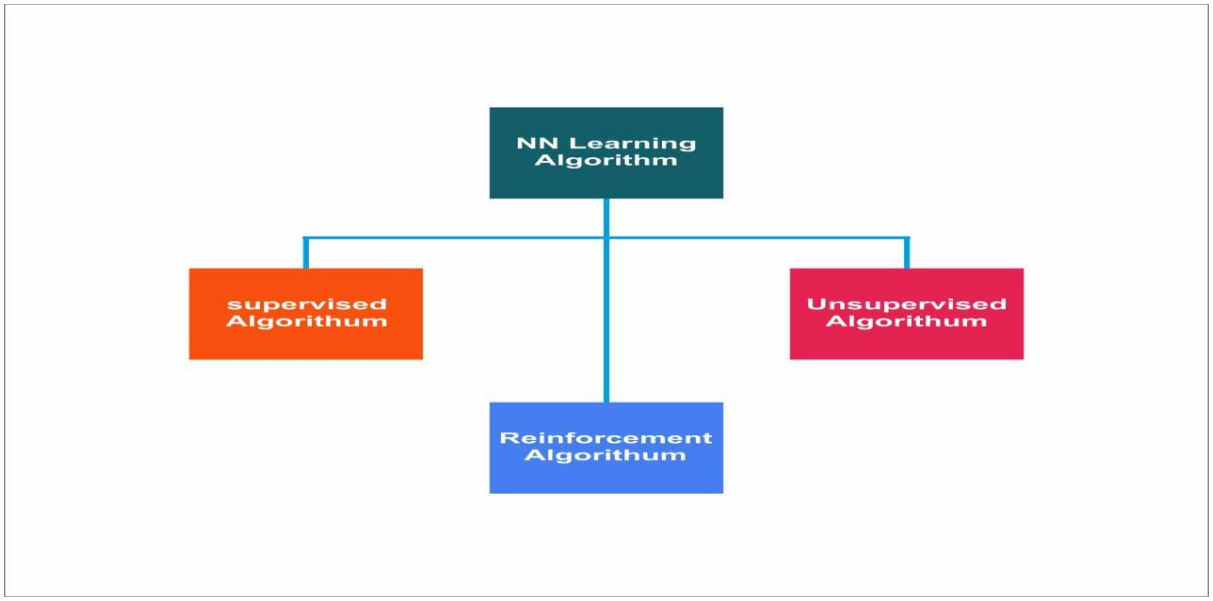

Fig. 3.1 Various NN Learning Algorithms

3.2 Unsupervised Algorithms

It has data set which is not labeled means the target value is not known in advance. It uses input data set only [10]. It helps in analysis of data. Clustering, association mining, segmentation and reduction in dimensionality come under unsupervised learning techniques.

It helps in customer segmentation, visualization of big data, targeted marketing, feature elicitation, text mining, face recognition and image recognition.

3.2 Reinforcement Algorithms

In these algorithms the network learns from its correct and wrong actions. It works better with data set which is unknown [11].

It helps in making real time decisions, robotic navigation, Gaming, acquisition of skill, learning different tasks, reward system, recommendation system and inventory manufacturing.

4. OPTIMIZATION METHODS

Optimization methods are used in NN for adapting to the connection weights in a network. In a NN the network is trained many times whenever there is modification in the network architecture. An important factor of network training is its computation efficiency. Many optimization methods can be used for connection weights optimization of nodes in hidden layer. These non linear methods are discussed below.

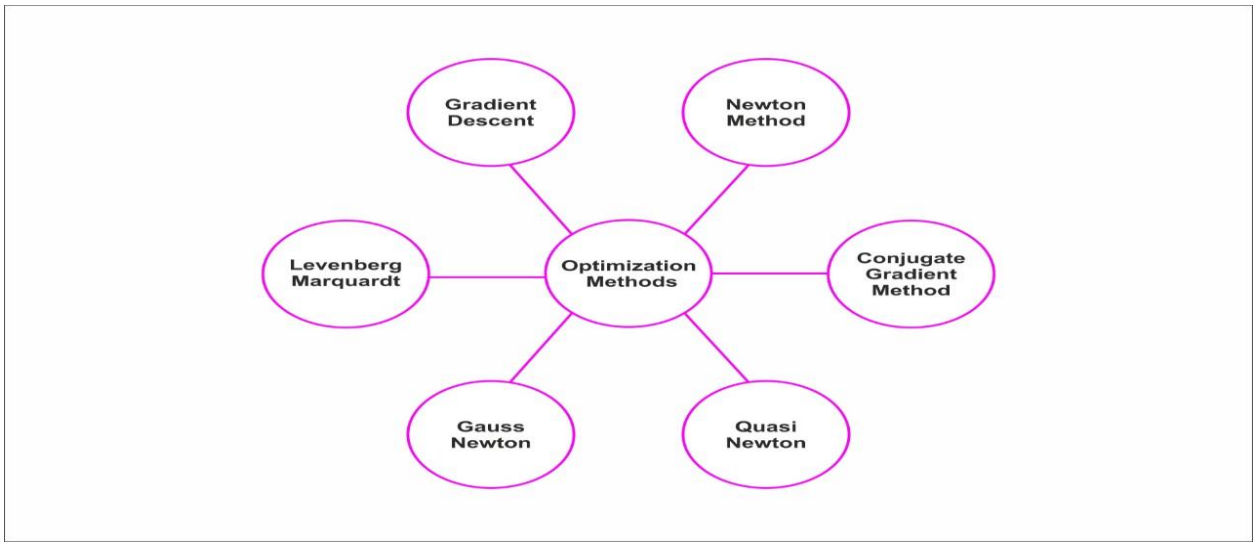

Fig. 4.1 Optimization Methods

4.1 Gradient Descent

Gradient Descent is the mostly used optimization method for training a neural network. It is a minimization algorithm. When input is changed my some amount then the gradient will change the output. It helps in reducing the cost function by adjusting the values. The input is known as slope [12]. The model learns fast when slope is high. But if the gradient is too high or too low, it may vanish. It is one of the slowest methods and requires less memory as compared to other optimization methods. The computational requirement is $Q(W_n)$ while training a NN having weight W_n.

4.2 Newton Method

Newton optimization method uses Hessian matrix and is a second order method. The calculation of hessian matrix and inverse of it is very expensive computationally [13]. It also takes fewer steps in minimizing the cost function. It is not widely used as it requires more computational power. The computational requirement is $Q(W_n^3)$ number of operations for one iteration while training a NN having weight W_n.

4.3 Conjugate Gradient Method

Conjugate Gradient method is an iterative method. It can be applied to linear and non linear system. It helps in the convergence of algorithm. It is faster than gradient descent method [14]. It can used for large NN.

4.4 Quasi Newton

Quasi Newton is one of the best methods for large NN. It is fast, takes less computation time and also less expensive computationally. There is no need for calculating hessian matrix and its inverse [15]. The computational requirement is $Q(W_n^2)$ number of operations for one iteration while training a NN having weight W_n.

4.5 Gauss Newton

Gauss Newton is a popular method for problems which are non linear least square and this algorithm is iterative [16]. It performs calculations in order to search for a solution. It keeps on guessing the values of X while making calculations. If the guess is not good enough, the algorithm may be slow and may not be able to find a good solution. It is based on Newton Method. In this firstly the values of X are guessed, then Jacobian Matrix is created and partial derivatives are calculated. The computational requirement is $Q(W_n^3)$ number of operations for one iteration while training a NN having weight W_n.

4.6 Levenberg Marquardt

It is an approximation of newton method. It solves minimization problems arising from curve fitting [17]. It is slower than Guass Newton in solving non linear least square problems. In was developed by Kenneth Levenberg in 1944 and again improved by Donald Marquardt in 1963. The computational requirement is $Q(W_n^3)$ number of operations for one iteration while training a NN having weight W_n.

CONCLUSION

Artificial Neural Networks are very powerful and nowadays used in almost all applications. Learning algorithms are used to solve problems using supervised, unsupervised or reinforcement learning techniques. Optimization methods play an important role in NN training. Six optimization methods explained above can be used for solving complex problems.

REFERENCES

[1] Hassoun, M. H. (1995). Fundamentals of artificial neural networks. MIT press.

[2] Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. Journal of biomedical informatics, 35(5-6), 352-359.

[3] Hopfield, J. J. (1988). Artificial neural networks. IEEE Circuits and Devices Magazine, 4(5), 3-10.

[4] Verma, A. K., R. Anil, and Om Prakash Jain. "Fuzzy Logic Based Revised Defect Rating for Software Lifecycle Performance Prediction Using GMR." Bharati Vidyapeeth's Institute of Computer Applications and Management (2009): 1.

[5] Kaur, J. and Gupta, N. (2019). Constructive Neural Network: A Framework. International Journal of Engineering and Advanced Technology (IJEAT), Volume-9, Issue-2, December 2019, Page No 5321-5324.

[6] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.Khashei, M., & Bijari, M. (2010). An artificial neural network (p, d, q) model for timeseries forecasting. Expert Systems with applications, 37(1), 479-489.

[7] Daniel, W. B., & Yeung, E. (2019). A constructive approach for one-shot training of neural networks using hypercube-based topological coverings. arXiv preprint arXiv:1901.02878.

[8] Kaur, J. and Gupta, N. (2020). Bipolar Sigmoid Algorithm for Designing Constructive Neural Network. International Journal on Emerging Technologies, 11(2): 991–996.

[9] Kaur, J. and Gupta, N. (2020). Extended Bipolar Sigmoid Algorithm for Enhancing Performance of Constructive Neural Network. International Journal on Emerging Technologies, 11(2): 1034–1038.

[10] Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168).

[11] Rodriguez-Nieva, J. F., & Scheurer, M. S. (2019). Identifying topological order through unsupervised machine learning. Nature Physics, 15(8), 790-795.

[12] Schwartz, A. (1993). A reinforcement learning method for maximizing undiscounted rewards. In Proceedings of the tenth international conference on machine learning (Vol. 298, pp. 298-305).

[13] Hochreiter, S., Younger, A. S., & Conwell, P. R. (2001, August). Learning to learn using gradient descent. In International Conference on Artificial Neural Networks (pp. 87-94). Springer, Berlin, Heidelberg.

[14] Mangasarian, O. L. (2004). A Newton method for linear programming. Journal of Optimization Theory and Applications, 121(1), 1-18.

[15] Johansson, E. M., Dowla, F. U., & Goodman, D. M. (1991). Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. International Journal of Neural Systems, 2(04), 291-301.

[16] Hennig, P., & Kiefel, M. (2013). Quasi-Newton method: A new direction. Journal of Machine Learning Research, 14(Mar), 843-865.

[17] Dora, L., Agrawal, S., Panda, R., & Abraham, A. (2017). Optimal breast cancer classification using Gauss–Newton representation based algorithm. Expert Systems with Applications, 85, 134-145.

[18] Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In Numerical analysis (pp. 105-116). Springer, Berlin, Heidelberg.