International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN (P): 2249–6831; ISSN (E): 2249–7943
Vol. 11, Issue 1, Jun 2021, 129-134

BACK PROPAGATION ALGORITHM - A REVIEW

MRINALINI SMITA & Dr. ANITA KUMARI

Assistant professor, St. Xavier's College, Intermediate Section, Ranchi, Jharkhand, India

Assistant professor, Dr. Shayama Prasad Mukherjee, University, Ranchi, Jharkhand, India

ABSTRACT

The application of deep learning models in financial forecasting comes in to overcome the difficulty in stock price prediction from traditional methods. The use of neural network architecture in deep learning models is called ANN. An ANN is a form of computation inspired by the structure and function of the brain.[** *Padhy, 2005]* ***One of the most important property of ANN is that NNs are capable of Learning by example means utilising examples from the data and organising them into a useful form constitutes a model representing the relationship between input and output variables. Back propagation algorithm is a most popular neural network training algorithm for financial forecasting as it is an advancement over the imitations of the one-and two-layer networks. Back propagation is the heart of deep learning process. This paper is a review on back propagation algorithm.

KEYWORDS: Deep Learning, Neural Network Architecture & Back Propagation Algorithm

Received: May 05, 2021; **Accepted**: May 25, 2021; **Published**: Jun 03, 2021; **Paper Id.**: IJCSEITRJUN202113

Original Article

INTRODUCTION

Every neuron consists of a functional unit. An ANN is interconnection of neurons. A number of neurons collect input information X. At every level, it passes through or multiplied by weight vector. This processed information from every neuron is passed to another neuron. They are also multiplied by another set of w and it continues. The mathematical function of ANN, a model of biological neurons, in which each input is separately weighted, the sum is then passed through a non-linear function f, known as activation function or transfer function. The output of every unit in the ANN is given by

$$y = f(w^T X)$$

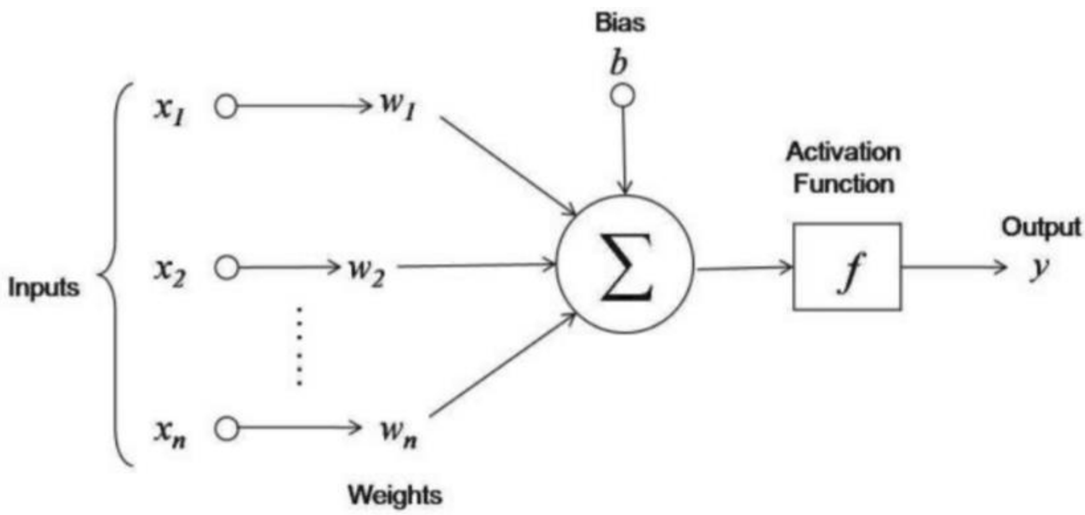

Figure 1

MATHEMATICAL MODEL FOR NEURAL NETWORK

Here X is the input, w is the weight vector and f is a non-linear function. In the figure, b is the bias, an error term.

Neural Network Architecture

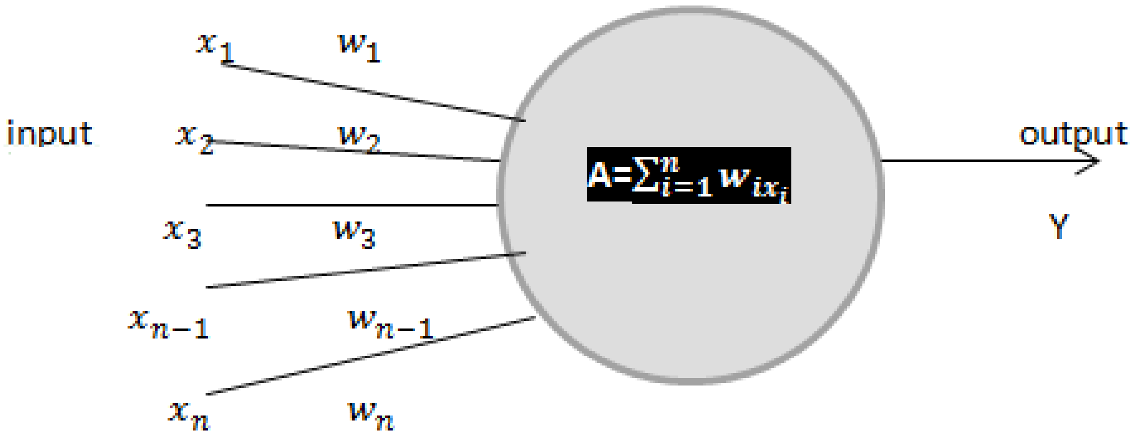

Figure 2

Architecture of Neural Network

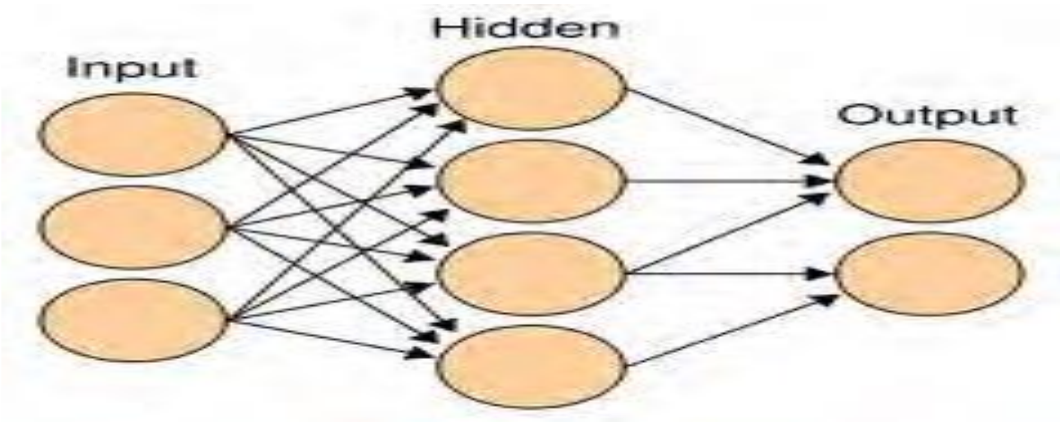

Figure 3

Layers in Neural Network

- Input layer node gives actually the '***identity functions'*** i.e. whatever be the input to a node in the input layer that node simply outputs that input(I/P) and feeds it to the nodes in the next layer.
- The other layer called ***intermediate or hidden layer*** because it usually has no connections with the outside world. The hidden layer actually improves the non- linearity's so as we increase the number of hidden layers, we can capture more and more complex form of non-linearity, then finally at the output layer node (classifying node), it implements actually the linear classifiers.
- ***Bias*** in artificial neural network (ANN) is nothing but the "***error***" term. The "***error***" basically signifies how well

our network is performing on a certain dataset.

- ***Activation function basically means "Input times weights, add bias and activate***". *A NN without an activation function is essentially just a linear regression model.* The activation function does the non-linear transformation to the inputs making it capable to learn and perform more complex tasks. Here the activation function is ***Sigmoid Function.***

- $g(z) = \frac{1}{1+e^{-z}}$

- This function is also referred to as a ***Squashing Function*** since it maps a large input domain onto the smaller range of 0 to 1.**[Kamber et al, 2001]**

OBJECTIVE OF THE STUDY

- To understand the mechanism of Back propagation algorithm.

- How to apply this algorithm to solve real-world problems such as classification, prediction etc.

Back Propagation Algorithm

Being the most popular algorithm of neural network training algorithm for financial forecasting, back propagation is used to train the Neural Network by chain rule method. In this chain rule, after each feed-forward pass through a network, this algorithm does the backward pass to adjust the model parameters based on weights and biases.

Training of Back Propagation

- For training the NN, error is propagated backward by adjusting the weights in between the layers but not through the Feed forward Network as we input the feature vectors X at the input layer(I/L), they are being processed in every layer and get output finally and nowhere propagated in the backward direction. ***This is the reason a NN is Feed Forward but the learning is Back Propagation Algorithm.***

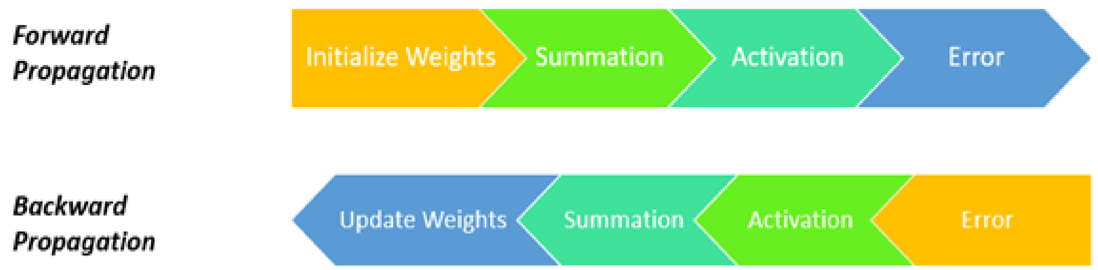

Figure 4

- ***Partition*** is very important in machine learning. It is generally done in two parts **-** ***training dataset*** **and** ***testing dataset***. The reason behind partition is that most of the ANN techniques are ***black box techniques,*** we do not have minute details of what is exactly going inside the network and as the process is so complex that we cannot control the entire process. Generally, what happens, we give it some data, NN tries to train it again and again (being an iterative process) and after some iterations, we come up with exact kind of prediction. But when we deployed it for new dataset, it does not work well as it has over trained itself on the training dataset. So whenever we have a dataset, we try to divide it randomly into two parts- training and testing dataset. The model is trained on the training dataset and then it is deployed for testing to find that the model is improved or not.

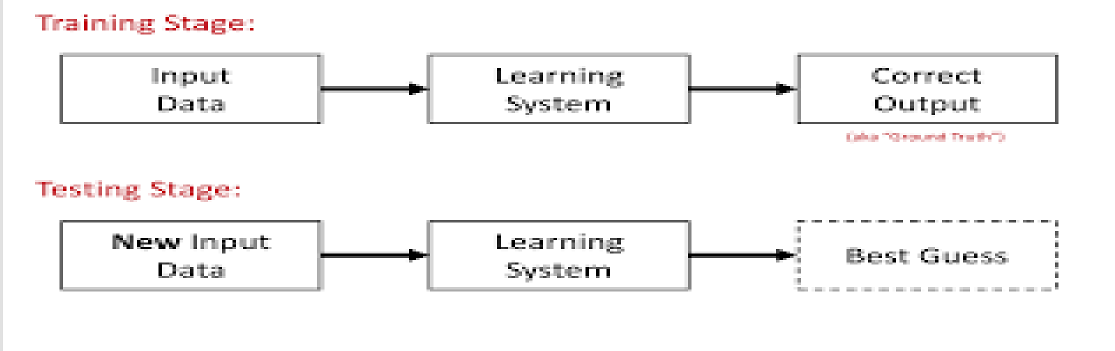

Figure 5

- During the training of multilayer neural network. the updating of weights of Neural network is done in such a way that the error observed can be reduced. This error is directly observed at the output(O/P) layer, which is back propagated to the previous layer and with that notional error which has been back propagated, the weight updating in previous layers are done.
- The error can only be observed at the output layer(O/L). For a training example, we do not know what should be the error at hidden layers. So the error that we find at the output layer(O/L), is back propagated and we estimate the error at the inside hidden layers. So we say that the error at the output layer(O/L) is because of the error which was computed at the hidden layers(H/L). Thus the error at the output layer(O/L) is partially dependent on the error at these hidden layers**.** In fact**, the *Error is proportional to the weight.***
- Back propagation works in this way that when we apply the NN on a particular input, the input signal propagate forward but the error at this layer is back propagated to the previous layer. Based on that error we do weight updating of errors in previous layers.
- When we have a sequence of functions from input(I/P) to output (O/P), we cannot directly compute the partial derivative of output, $\frac{\partial O}{\partial X}$ where

$$O = f^{(k)}(f^{(k-1)} \dots\dots \left(f^{(i)} \dots\dots \left(f^{(2)}\left(f^{(1)}(X)\right)\right)\right)$$

as output O is far away from the input X. rather we have to make use of chain rule of differentiation to find out the partial derivatives of output (O/P) with respect to input(I/P). ***The partial derivatives are the sensitivity of output on the input X*** as we have sequence of neural network functions from input to output which cannot be directly computed so in back propagation we make use of these properties of chain rule of partial derivatives.

- While applying the back propagation algorithm, any non-linear function can be used if it is differentiable everywhere and monotonically increasing with the inputs. Sigmoidal functions, including logistic, hyperbolic tangent and arctangent functions meet these requirements.

(**Tsoukalas & Uhrig 1996**)

- Since ***the error sent back for training in back propagation is proportional to the derivative of the sigmoid function,***

  $g'(z) = g(z)(1 - g(z))$

  Very little training take place. This ***derivative*** of Sigmoid function is the most attractive feature of Sigmoid function which is extremely simple to compute and making use of it for classification problem. This network paralysis can sometimes be avoided by reducing the value of the training co-efficient, which unfortunately results in extending the training time.

[**Padhy, 2005**]

- Applying activation function, the output we get (predicted value) should be compared using loss function. Our aim is to reduce this loss function using an optimizer, Gradient Descent here. The gradient descent will try to find out the most, weights that have to be updated in a very efficient way which is the best way to calculate estimated (predicted) variable. That basically means, we update all the weights(i.e.whole number of records) in each and every layer by gradient descent (an optimizer) using back propagation.

WEIGHT UPDATION FORMULA

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W}$$

Here the derivative $\frac{\partial L}{\partial W}$is taken which help us to find whether this derivative or slope we are finding is positive or negative. This loss value L should be reduced to the minimum value. If this loss function given by,

L = (actual value – predicted value)^2

is not decreasing, that means the whole function is not converging somewhere. Until or unless we reach the global minimum, updating weights are done.

- During weight updation, we come up with a constant, called ***Learning Rate***, which controls the rate of convergence. This learning rate should be small to reach the global minimum function. Of course, ifη is equal to 0, then no learning will take place. Therefore, η must always be positive.[**Tsoukalas & Uhrig 1996]** But it should not be very large as we will never come towards the goal of the minimal point then. The default value for the learning rate is 0.1or 0.01.

Merits of Back Propagation

- The linear models generally fail to understand the data pattern and analyze when the underlying system is a non-linear one. But ANNs are non-linear in nature hence preferred over the traditional linear models.
- Stock markets are chaotic. The use of ANNs in a chaotic market does not require an understanding of the market dynamics. This is why***it is practically feasible and profitable to use Machine learning systems like NN to predict the behaviour of financial instrument such as stocks.***

Limitation of Back Propagation

Although back propagation is viewed as a powerful model; it has some of its drawbacks in financial applications which include

- ***Overfitting*** is one of the limitations of back propagation, which occurs when a model can predict data well when trained but unable to generalise accurately in the testing phase .i.e. the model has high variance but low bias.
- ***Underfitting*** is another limitation. A model is said to be underfitting when it is unable to classify/ predict means the training accuracy of the model is very low i.e.model has low variance and high bias.
- Another drawback is that ***deactivating***, or dropping neurons will reduce the performance of the neural network. To further limit is model, it is important to note that it is difficult to predictable which nodes will be disabled.

CONCLUSIONS

As researchers and investors strive to out-perform the market, the use of neural networks to forecast stock market prices will be a continuing area of research. The ultimate goal is to increase the yield from the investment. It has been proven already through research that the evaluation of the return on investment in share markets through any of the traditional techniques is tedious, expensive and a time consuming process. Thus the use of back propagation is just a stepping stone in future prediction technologies. Moreover, back propagation of ANN can be used for other applications and comparative study with other models as well, in future.

REFERENCES

1. *Himali Jani, Chirag Jain, Shreyansh Jain & Kiran Bhowmick, "Neural Network Applications in Stock Market Predictions",International Journal of Computer Science and Engineering (IJCSE),Vol. 2, Issue 5,pp, 135-146*
2. *Padhy, N.P., "Artificial Intelligence and Intelligent Systems", ISBN-13: 978-0-19-567154-4,400-437,2005*
3. *Adnan A. Ugla*, Dhuha J. Kamil & Hassan J. Khaudair, "Quality Improvement of Shaped Metal Depositedcomponents Using Non-Traditional Techniques: A Review",International Journal of Mechanical and Production Engineering Research and Development (IJMPERD),Vol. 10, Issue 3,pp, 2411–2430*
4. *Han, Jiawei, Kamber, Micheline, "Data Mining: Concepts and Techniques" ISBN:1-55860-489-8, 301-310, 2001*
5. *Amitvikram Nawalagatti & Prakash R. Kolhe, "A Comprehensive Review on Artificial Intelligence Based Machine 3 Learning Techniques for Designing Interactive Characters",International Journal of Mathematics and Computer Applications Research (IJMCAR),Vol. 8, Issue 3,pp, 1-10*
6. *.Tsoukalas, L.H. and Uhring, R.E., " Fuzzy and Neural Approaches in Engineering", JohnWiley, New York1996.*
7. *Azriyenni, M. W. Mustafa, D. Y. Sukma & M. E. Dame, "Application of Backpropagation Neural Network for Fault Location in Transmission Line 150 kV ",International Journal of Electrical and Electronics Engineering (IJEEE),Vol. 2, Issue 4,pp, 21-30*