Asia-Pacific Journal of Neural Networks and Its Applications
Vol.4, No.1 (2020), pp.21-28
http://dx.doi.org/10.21742/ajnnia.2020.4.1.03

A Study on Backpropagation in Artificial Neural Networks

Ch Sekhar[1] and P Sai Meghana[2]

Department of CSE, VIIT(A), AP, India University, India
[1]sekhar1203@gmail.com, [2]palavalasasaimeghana@gmail.com

Abstract

Innovation assumes essential job nowadays in human life to limit the manual work. Execution and exactness with innovation will be high. The Backpropagation neural framework is multilayered, feedforward neural framework and is by a full edge the most extensively utilized. It is moreover seen as one of the least demanding and most wide systems used for managed planning of multilayered neural systems. Backpropagation works by approximating the non-direct association between the data and the yield by changing the weight regards inside. It can furthermore be summarized for the data that is rejected from the planning structures (perceptive limits).

Keywords: *Backpropagation, ANN, Neuron, Nervous system, MLP, Feedforward networks*

1. Introduction

A neural system is a gathering of associated I/O units where every association has a weight-related with its PC programs. It encourages you to develop prescient models from enormous databases. This model expands upon the human sensory system. It encourages you to lead picture understanding, human learning, speech recognition, and so on.

Backpropagation is the embodiment of neural net preparing. It is the technique for tweaking loads of a neural net dependent on the error value got in the past epoch (i.e., emphasis). Legitimate tuning of the loads permits you to lessen error value and to make the model dependable by expanding its speculation. Backpropagation is a compact structure for "backward propagation of errors." It is a standard technique for preparing artificial neural systems [1]. This technique assists with computing the inclination of a misfortune work regarding all the loads in the system. Backpropagation recipes from essential standards and genuine qualities. The neural system uses three information neurons, one shrouded layer with two neurons, and a yield layer with two neurons.

2. The literature on back propagation

During the feed forward computation neural networks, the result output value is not near to target or teacher output value. There is a difference between target, and actual feed forward values lead error value. The neural network model tries to give the best prediction output will good tolerance for that we have to minimize the error rate. This can be done with Backpropagation

Milestone of Backpropagation:

Article history:
Received (April 26, 2020), Review Result (May 29, 2020), Accepted (July 10, 2020)

Print ISSN: 2207-8738, eISSN: 2207-8746 AJNNIA

- In 1961, the essentials idea of ceaseless Backpropagation was inferred with regards to control hypothesis by J. Kelly, Henry Arthur, and E. Bryson.
- In 1969, Bryson and Ho gave a multi-orchestrate dynamic system improvement method.
- In 1982, Hopfield brought his idea of a neural framework.
- In 1986, by the effort of David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams, Backpropagation got affirmation.
- In 1993, Wan was the primary individual to win a stellar example acknowledgement challenge with the assistance of the backpropagation strategy.

3. Backpropagation algorithm and computational process

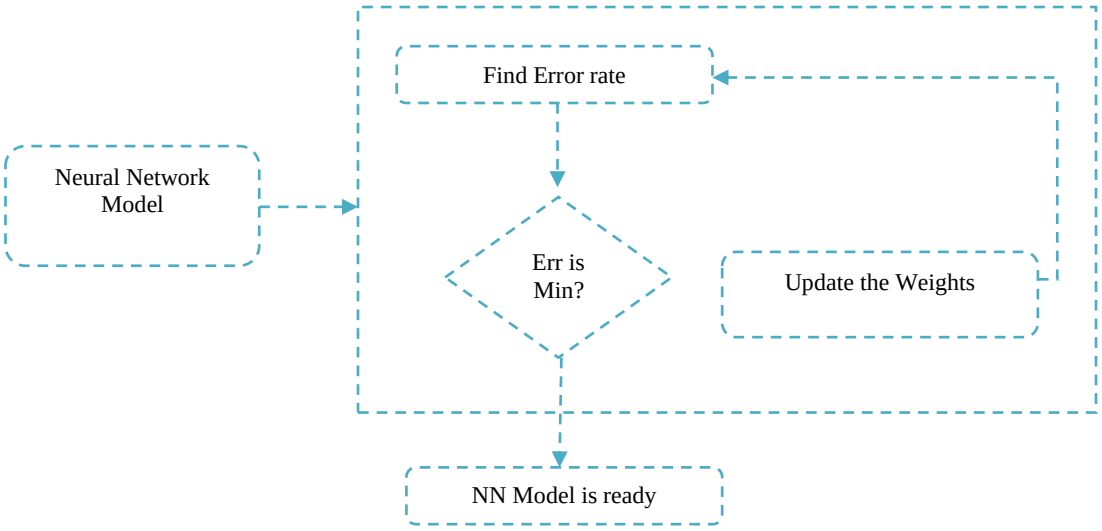


Figure 1. Flow diagram of the working mechanism of backpropagation

The above [Figure 1] shows that the everyday workflows of the backpropagation process mechanism. During the backwards propagation, these computation activities will happen as mentioned below.

- Find Error rate: Here we need to calculate the model output with actual output
- Minimum Error: Cross verifying whether the error is minimized or not.
- Update the Weights: The error is more than the acceptable range then, update the weights and biases. After that, again check the error. Repeat the process until the error becomes low.
- Neural Network Model: Once the error rate was acceptable range then the model is ready to use for forecasting the data

The generalized workflow and stepwise computation in Backpropagation given as pseudo-code as follows:

```
1. Assign all network inputs and output
2. Initialize all weights with small random numbers, typically between -1 and 1
3. repeat
       for every pattern in the training set
       Present the pattern to the network
4. //     propagated the input forward through the network:
         for each layer in the network
           for every node in the layer
              1. Calculate the weight sum of the inputs to the node
              2. Add the threshold to the sum
              3. Calculate the activation for the node
           end
         end
5. //     Propagate the errors backward through the network
         for every node in the output layer
           calculate the error signal
         end
6. for all hidden layers
              for every node in the layer
                 1. Calculate the node's signal error
                 2. Update each node's weight in the network
              end
         end
7. //     Calculate Global Error
          Calculate the Error Function
8. end
9. while ((maximum number of iterations < than specified) AND
          (Error Function is > than specified))
```

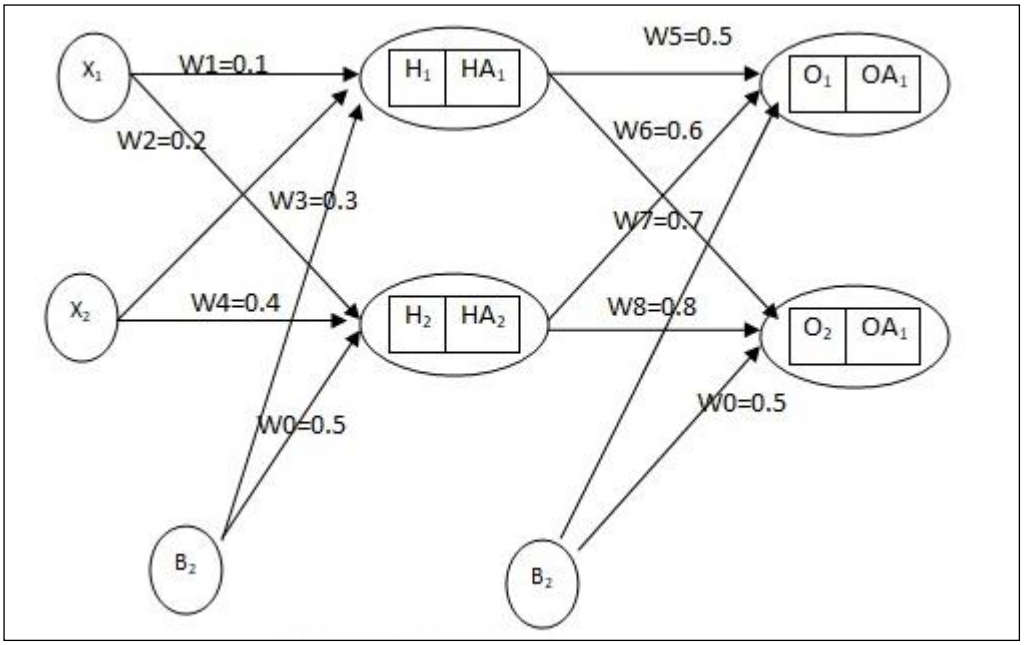


Figure 2. Feed forward ANN

The above [Figure 2] shows the feedforward artificial neural network contains Input, Hidden and Output layers. Each layer contains two nodes with respective weights. All nodes are fully connected model, including the bias node.

Notions of the above network as follows:

- *X1, X2: Input Nodes*
- *H1, H2: Hidden Layer Nodes with net out from respective inputs*
- *HA1, HA2: Hidden Layer Nodes with activation output*
- *O1, O2: Output Layer Nodes with net out from respective inputs*
- *OA1, OA2: Output Layer Nodes with activation output*
- *W1 to W8: Weights of respective layers from input to output*
- *B1, B2: Bias Nodes for Hidden and Output layers respectively*

4. Working with backpropagation

The Following steps are followed involved during Backpropagation of above feedforward network.

Step 1

The input and target values for this problem are X1=1, X2=2, I and target values t1 =0.5 and t2=0.05. The weights of the network need to be randomly chosen within the range of 0 to 1. Here we initialize weights as shown in the figure above for understanding the process.

Step 2

From the Step1, we got the inputs and respective weights, as it was a feed-forward network.

The neuron will send to next neuron, i.e. hidden layer neuron. As it was fully connected network, each node/neuron will receive inputs from all the nodes/neurons of the input layer.

Now here we are going to calculate the summation output of at each node of the hidden layer as follows,

$$H_1 = (W_1 \times X_1) + (W_3 \times X_2) + (B_1 \times W_0) \quad (1)$$

$$H_2 = (W_2 \times X_1) + (W_4 \times X_2) + (B_1 \times W_0) \quad (2)$$

From the above equations, we are going to calculate feed-forward computation from input to hidden and hidden to output layers respectively

Input to the hidden layer

$$H_1 = (1 \times 0.1) + (2 \times 0.3) + (1 \times 0.5) = 1.2 \quad (3)$$

$$H_2 = (1 \times 0.2) + (2 \times 0.4) + (1 \times 0.5) = 1.5 \quad (4)$$

Applying activation function for both hidden nodes, here we are using sigmoid activation functions.

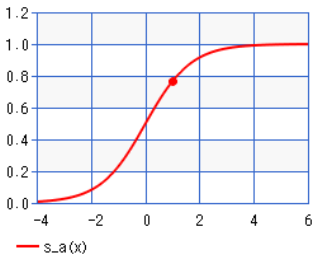
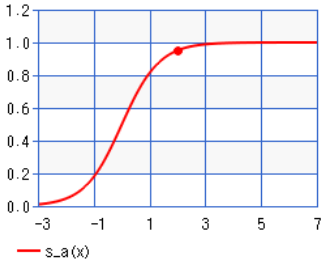
$S= 1/ (1+e^{-x})$		

Figure 3. (a) Sigmoid activation function equation (b) With HA1 (c) With HA2

$$HA_1 = \frac{1}{(1+e^{-H_1})} = \frac{1}{(1+e^{-1.2})} = 0.6456 \quad (5)$$

$$HA_2 = \frac{1}{(1+e^{-H_2})} = \frac{1}{(1+e^{-1.5})} = 0.9525 \quad (6)$$

Hidden layer to Output layer

$$O_1 = (HA_1 \times W_5) + (HA_2 \times W_7) + (B_2 \times W_0) \quad (7)$$

$$O_2 = (HA_1 \times W_6) + (HA_2 \times W_8) + (B_2 \times W_0) \quad (8)$$

$$O_1 = (0.6456 \times 0.5) + (0.9525 \times 0.6) + (1 \times 0.5) = 1.3943 \quad (9)$$

$$O_1 = (0.6456 \times 0.7) + (0.9525 \times 0.8) + (1 \times 0.5) = 1.71392 \quad (10)$$

Applying activation function for both hidden nodes, here we are using sigmoid activation functions.

$$OA_1 = \frac{1}{(1+e^{-O_1})} = \frac{1}{(1+e^{-1.3943})} = 0.8012 \quad (11)$$

$$OA_2 = \frac{1}{(1+e^{-O_2})} = \frac{1}{(1+e^{-1.7139})} = 0.9685 \quad (12)$$

Step 3

In this step, we need to compute the error value occurred w.r.t. to target output and feed-forward computation values

$$Error = Actual\ Output - Target\ Output$$

$$E_1 = OA_1 - T_1$$

$$E_2 = OA_2 - T_2$$

$$Etotal = E_1 + E_2 \quad (13)$$

Step 4

After the above operation, need to start backwards step. To update the weights based on the error value.

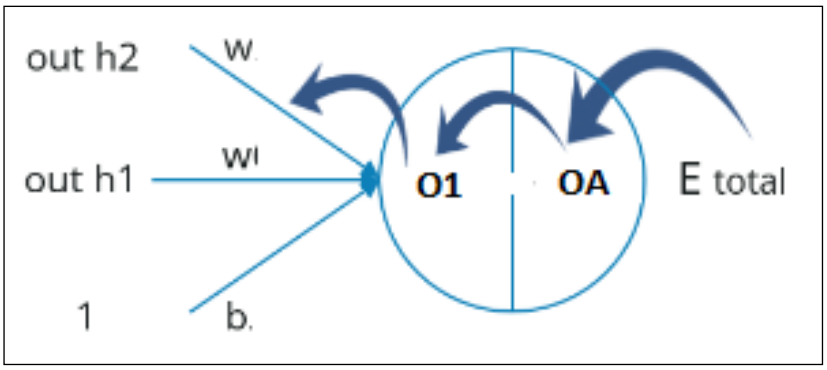


Figure 4. Error backpropagated from output to hidden, Hidden to the input layer

Here will compute the what change the error concerning the weight w5

$$\frac{\delta Etotal}{\delta w5} = \frac{\delta Etotal}{\delta outO1} \times \frac{\delta outO1}{\delta net01} \times \frac{\delta net01}{\delta w5} \tag{14}$$

We are spreading in reverse; the first thing we have to do is, compute the adjustment in simple mistakes w.r.t the yield O1 and O2. New weight is calculated based

$$W_{new} = W_{old} + learning\ rate\left(\frac{\delta Etotal}{W_{old}}\right) \tag{15}$$

The above process explained for one node or perceptron, and it needs to be repeated for all the nodes and update the weights. With new weights need to calculate the new error again until getting the error with minimal.

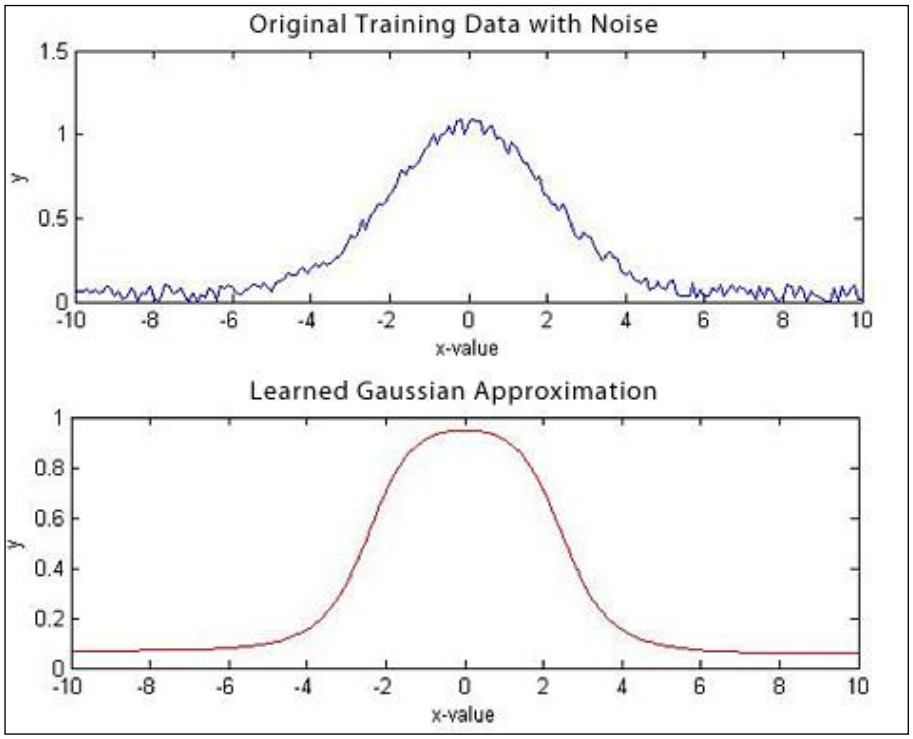


Figure 5. Gaussian function estimation

5. Applications of back propagation

Classification Problems: Right now, the objective is to distinguish whether a specific "information point" has a place with Class 1, 2, or 3. Irregular focuses are allocated to a specific class, and the neural system is prepared to discover the example. When preparing is finished, it will utilize what it has figured out how to group new focuses precisely.

Functional Approximation: Right now, organize attempts to inexact the estimation of a specific capacity. It is encouraged with full information, and the objective is to locate the actual example. In the wake of preparing, the system effectively gauges the estimation of the gaussian capacity (underneath).

Time-Series Forecasting: Right now, the objective is to structure a neural system to foresee a worth dependent on a piece of given time-arrangement information (for example, financial exchange forecast dependent on given patterns). To move toward this issue, the contributions to the neural system must be refactored in lumps, and the subsequent yield will be the following information thing straightforwardly following that piece (see beneath).

6. Conclusions

In this paper, we have shown that the process of a backpropagation neural network performs well on large sets of data. The performance can be improved by changing the number of hidden neurons and the learning rate. Because of its iterative training and gradient-based training, the general speed is far slower than required, so it takes a significant amount of time to train on an extensive set of data. We cannot say that there is a whole network for every kind of database out there. So keep testing your data on multiple neural networks and see what fits the best.

References

[1] Budiharjo S. Triyuni W. Agus Perdana, and H. Tutut, "Predicting tuition fee payment problem using backpropagation neural network model," **(2018)**

[2] M. Huan, C. Ming, and Z. Jianwei, "Study on the prediction of real estate price index based on hhga-rbf neural network algorithm," International Journal of u - and e-Service, Science and Technology, SERSC Australia, ISSN: 2005-4246 (Print); pp.2207-9718 (Online), vol.8, no.7, July, **(2015)** DOI: 10.142 57/ijunnes st.2015.8.7.11.

[3] A. Muhammad, A. Khubaib Amjad, and H. Mehdi, "Application of data mining using artificial neural network: survey," International Journal of Database Theory and Application, vol.8, no.1, **(2015)** DOI: 10.14257/ijdta.2015.8.1.25.

[4] P. Jong, "The characteristic function of CoreNet (Multi-level single-layer artificial neural networks)," Asia-Pacific Journal of Neural Networks and Its Applications, vol.1, no.1, **(2017)** DOI: 10.21742/AJNNIA.201 7.1.1.02

[5] L. Wei, "Neural network model for distortion buckling behaviour of cold-formed steel compression members," Helsinki University of Technology Laboratory of Steel Structures Publications 16, **(2000)**

[6] The concept of Back-Propagation Learning by examples from the http://hebb.cis.uoguelph.ca/~skremer /Teachin g/27642/BP/node3.html

