# Fault Detection of Neural Network Using Machine Learning

**Farhana Kausar[1],Reddem Pradeep Kumar Reddy[2],Ravichandra[3],Shamanth S[4]**
[1, 2, 3, 4] Dept of CSE
[1, 2, 3, 4] Atria Institute of Technology, Bangalore

***Abstract-*** *An artificial neural network (**ANN**) is a computational model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the **ANN** because a neural network changes - or learns, in a sense - based on that input and output. It consists of Input, Hidden and Output Layers. Transfer Function (or) Activation Function is used in the hidden layers. Supervised, Unsupervised , Reinforced are learning methods in ANN. ANN architecture consists of Feed Back Networks and Feed Forward Networks.Various predictions are going on now a days. Artificial Neural Network is the trending technology for the predictions. By ANN we can perform various predictions like climate prediction , Travelling sales person problem , rainfall prediction and so on. Back propagation algorithm is the important concept involved in the ANN. We can perform Rainfall prediction by using Back Propagation Algorithm. Highly complex pattern recognitions can be achieved by using neural networks. In ANN hidden nodes and selection of hidden nodes concept plays a important role. In this paper we are dealing with how the fault is detected in neural network by using machine learning.*

***Keywords***- Artificial Neural Network(ANN), Back propagation, XOR, sigmoid function

## I. INTRODUCTION

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements(neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

The study of the human brain is thousands of years old. With the advent of modern electronics, it was only natural to try to harness this thinking process. The first step toward artificial neural networks came in 1943 when Warren McCulloch, a neuro physiologist, and a young mathematician, Walter Pitts, wrote a paper on how neurons might work. They modeled a simple neural network with electrical circuits. Neural networks, with their remarkable ability to derive meaning from complicated orim precise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyses.

In this paper we are considering XOR as a bench mark for the detection of fault.

The entire architecture is dived into three parts

a. Input layer : this layer is responsible receiving information , signals from external environments. By using activation function the output of the input layer is feed as input to the hidden layer.
b. Hidden layer: these layers perform most of the internal processing.
c. Output layer: this layer is responsible for producing the final outputs.

## II. ALGORITHMS

Back Propagation Algorithm

The most popular method for supervised learning is Back Propagation [BP] Algorithm. BP is based on gradient descent, which generally uses least square optimal criterion, defining a method for calculating the gradient of error with respect to input where the network is trained by a initial random weights and assigning these initial weights will make the network learn without any previous knowledge, by propagating error backward through network [21]. The basic procedure for the back propagation algorithm is: Initialize network weights randomly while not termination condition do Assign as net input to each unit in the input layer its

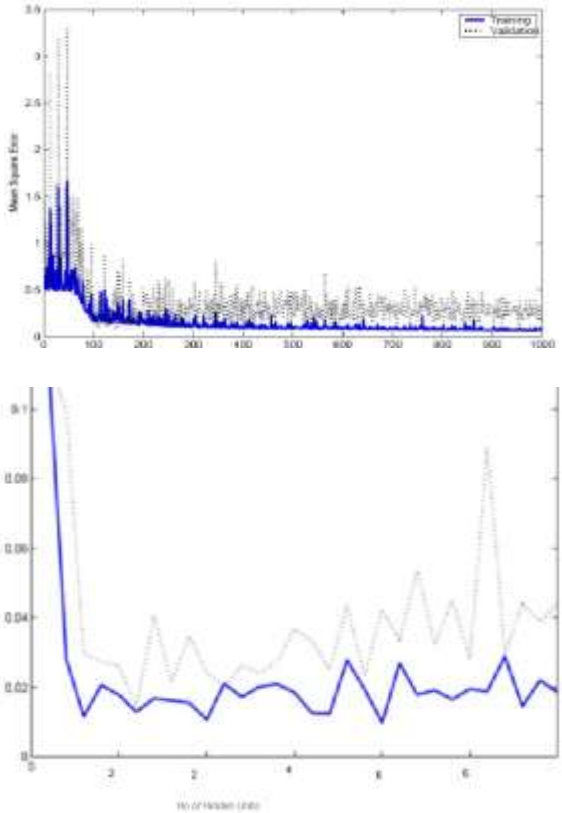
corresponding element in the input vector. The output for each unit is its net input Calculate network output by forwarding input signals in the network Calculate the error of each output neuron for all hidden neurons do Calculate weights updates Calculate where tj is target and oj is the actual jth output of the network Propagate the error back through the network. end for Update weights of the network. Check for the error rate.end while The training is supervised by having a target pattern associated with an input pattern. The pattern is given to the network and error is calculated to determine how weights should change. This process is repeated for each pattern.

## III. EXISTING SOLUTION

The detection of fault is already implemented by using MATLAB but as we know machine learning is an high level language to increase the performance.

Consider the behavior of a network , with inputs *X a*nd *Y* are two-valued. The two values are of course representing *True* and *False*, and we shall follow convention in giving these the values 0 and 1respectively. To construct a neural network that solves the classical (binary) XOR problem. The XOR problem behaves in a similar way for all 0's & all 1's.

These are the existing experimental results



## IV. PRESENT SITUATION

In this paper we are dealing with how the detection of fault will happen in neural network by using python machine learning. Normally we are using Anaconda Navigator 3 by taking the XOR as inputs.

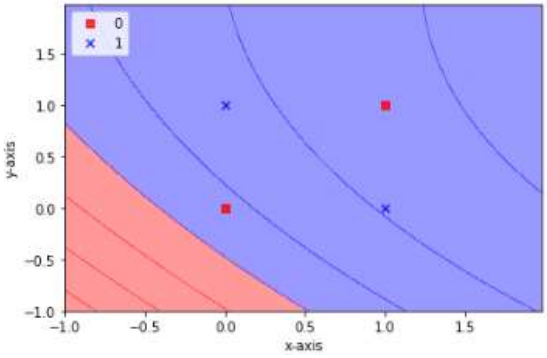
Change the input file used for the Binary XOR to a set of input/outputs that describe the OR on the value range between 0 and 1 with steps of 0.1. The transition with the true and false output can be placed somewhere in the middle. The value pairs should be randomly ordered before fed to the network. Complete the table below with training error and network behavior when tested. **Remember to reset the weights of the network before eachtraining.**

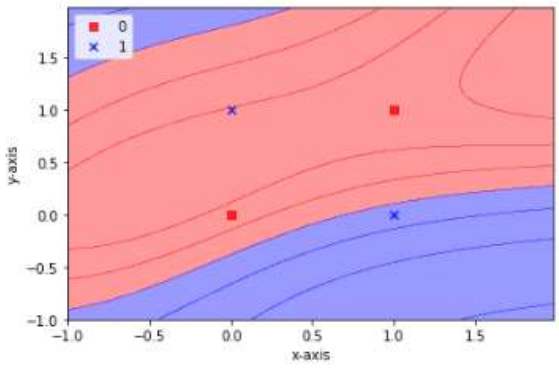
| *Epochs* | *Learning Rate* | *Momentum* | *RMSE* | *Behaviour?* |
|---|---|---|---|---|
| *1000* | *0.8* | *0.3* | | |
| *2000* | *0.8* | *0.3* | | |
| *3000* | *0.8* | *0.3* | | |
| *5000* | *0.8* | *0.3* | | |

**Sigmoid Function:**it's actually used for an activation it is an S shaped curve. Its refers to an logistic function .

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

Final prediction
[0 0] 0.5725987343720323
[0 1] 0.6634730888958555
[1 0] 0.7074681779844519
[1 1] 0.7651812610190181






## V. CONCLUSION AND FUTURE WORK

The final prediction output which we got has an minimized error. When fault is detected the machine undergoes back propagation algorithm and it will change the weights from output layer to input layer. This process is repeated until the minimzed error has been occurred.

By using machine learning we can use the inbuilt libraries such as MATPLOTLIB, NUMPY etc.

## REFERENCES

[1] R. P. Lippmann, "An Introduction to Computing withNeural Nets," *IEEE ASSP Magazine*, pp.4-22, April1987.

[2] F. M. Dias and A. Antunes, "Fault Tolerance of Artificial Neural Networks: an Open Discussionmfora Global Model," *International Journal of Circuits, Systems and Signal Processing*, Naun,July, 2008.

[3] C. Alippi, V. Piuri, and M. Sami, "Sensitivity to Errors in Artificial Neural Networks: ABehavioral Approach," RSME ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India.

[4] L.A.Belfore, Fault Tolerance of NeuralNetworks,Training Correctness(%) 99.00 99.00 Ph.D. Dissertation, Univ. of Virginia, 1989.

[5] Davis 19881 L. Davis, "Mapping Classifier Systemsinto Neural Networks," to appear in *Proceedings of the1988Conference on Neural Information ProcessingSystems*, Morgan Kaufimann..

[6] E.B. Tchernev, R. G. Mulvaney, and D.S. Phatak, "Investigating the Fault Tolerance of NeuralNetworks,"*Neural Computation*, vol. 17, no. 7, pp. 1646-1664,July 2005.

[7] P. Chandra and Y. Singh, "Fault Tolerance ofFeedforward Artificial Neural Networks – A Framework of Study," *Proceedings of the InternationalJoint Conference on Neural Networks*, vol. 1, pp. 489-494, July 2003.

[8] V. Piuri, "Analysis of Fault Tolerance in Artificial Neural Networks," Journal *of Parallel and Distributed Computing*, pp. 18-48, 2001.