Multilevel Neural Network on Student Performance Data

Shubham Barudwale
School of Computing Sciences and Engineering, VIT Chennai, Tamilnadu, India 600127
Email: barudwaleshubham.dinesh2015@vit.ac.in

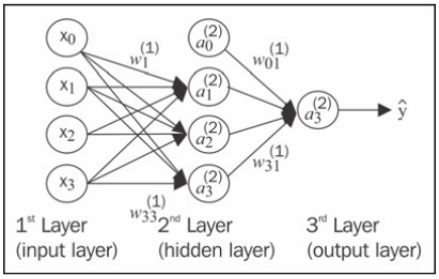
Abstract-

Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve performance) to do tasks by considering examples, generally without task-specific programming.

An ANN is based on a collection of connected units called artificial neurons, (analogous to axons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, and weight generally represented by real numbers, typically between 0 and 1. Using states and weights and adjusting them the ANN decides the final output.

1. Introduction

Multiple Layer Neural Networks is the unsuperwised learning algorithm. we will see how to connect multiple single neurons to a multi-layer feedforward neural network; this special type of network is also called a multi-layer perceptron (MLP). The following figure explains the concept of an MLP consisting of three layers: one input layer, one hidden layer, and one output layer. The units in the hidden layer are fully connected to the input layer, and the output layer is fully connected to the hidden layer, respectively. If such a network has more than one hidden layer, we also call it a deep artificial neural network.



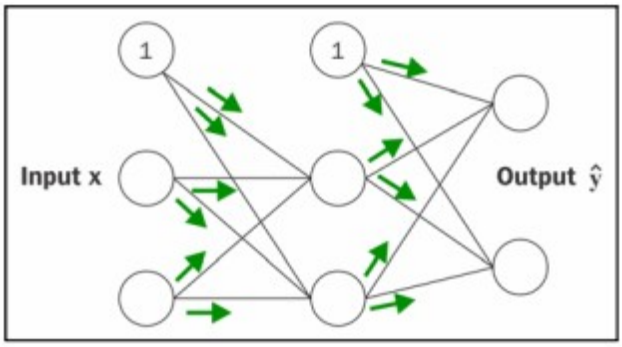
2. Methodology

In Multilayer Neural Networks generally used algoritm is backpropogation method. In this method first the values are feeded into the first layer of the network. And the next several layers are connected by the weigths. The values of the next layer's nodes are calculated by multiplying the input values and weights. And the same method is applied until the terminal output layer is reached.

Then if the output values are not distinct and close to the required values then again the values are backpropogated in oredr to change the weigts of the links and again the values are calculated for several times.

activation of the output layer, which we formulated as follows:

$Z^{(2)}=W^{(1)}[A^{(1)}]^T$ (net input of the hidden layer)
$A^{(2)}=\Phi(Z^{(2)})$ (activation of the hidden layer)
$Z^{(3)}=Z^{(2)}A^{(2)}$ (net input of the output layer)
$A^{(3)}=\Phi(Z^{(3)})$ (activation of the output layer)



the error vector of the output layer:

$$\delta^{(3)} = a^{(3)} - y$$

y is the vector of the true class labels.
error term of the hidden layer:

$$\boldsymbol{\delta}^{(2)} = \left(\boldsymbol{W}^{(2)}\right)^T \boldsymbol{\delta}^{(2)} * \frac{\partial \phi\left(z^{(2)}\right)}{\partial z^{(2)}}$$

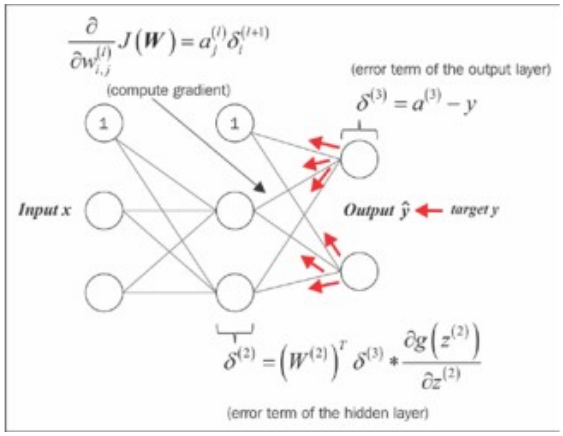
cost function as follows:

$$\frac{\partial}{\partial w_{i,j}^{(l)}} J(\boldsymbol{W}) = a_j^{(l)} \delta_i^{(l+1)}$$

Next, we need to accumulate the partial derivative of every j th node in layer l and

the i th error of the node in layer l +1 Then after derivation to reduce errors we get the following weight function to update weights:

$$W^{(l)} := W^{(l)} - \eta \Delta^{(l)}$$



3. Data

Student performance dataset is the dataset from which we are trying to gain some information from 650 student's academic performance data and various attributes which are affecting the performance. Attributes include sex, age, study time, free time, family information etc.

We will be using the Logistic regerssion method to learn and predict the performance. Out of 650 I have taken 455 data entries for lerning and rest for prediction i.e testing the performance of the algorithm.(70:30). The labels or target functions are mentioned in the 'new' field which is added manually. bad(0-7), medium(8-14), good(15-20) .

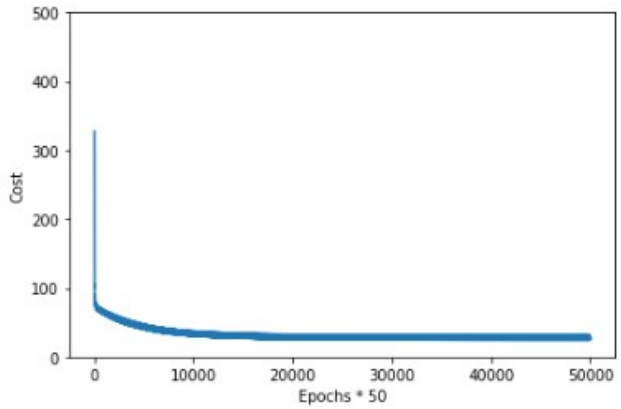
4. Algorithm

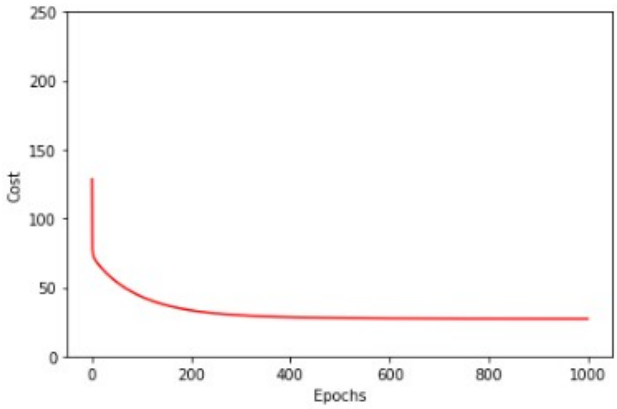
The threshold function of the units is modified to be a function that is continuous derivative, the Sigmoid function(Formula 4 The Sigmoid Function). The use of the Sigmoid function gives the extra information necessary for the network to implement the back-propagation training algorithm. Back-propagation works by finding the squared error (the Error function) of the entire network, and then calculating the error term for each of the output and hidden units by using the output from the previous neuron layer. The weights of the entire network are then adjusted with dependence on the error term and the given learning rate.

Training continues on the training set until the error function reaches a certain minimum. If the minimum is set too high, the network might not be able to correctly classify a pattern. But if the minimum is set too low, the network will have difficulties in classifying noisy patterns.

5. Experiments

We have used the algorithm for training the students data. We have given the 50 hidden nodes to train the data. But due to data is not distributed properly and not seperable properly hence the data was not properly trained. And the accuracy was very low as low as about 28%. Even after 1000 eopches the cost is not reduced significantly.





6. Conclusion

From the graph of Cost vs Epoch we can see that the cost is reduced to about 35 after 200 epochs from 135.

Since the data was not suitable for Neural network Algorithm he training was not proper and accuracy was about 28% while using both defined algorithm and Sklean Neural Network Algorithm. Which is the poorest among the others.

7. Referances

1) Python Machine Learning by Sebastian Raschka Chapter 12