COMPUTATIONAL INTELLIGENCE TENSORFLOW NEURAL NETWORK ASSIGNMENT

By

Aaron Ward

Supervisor(s):

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
B.SC IN COMPUTING AND INFORMATION TECHNOLOGY
AT
INSTITUTE OF TECHNOLOGY BLANCHARDSTOWN
DUBLIN, IRELAND
2017

Declaration

I herby certify that this material, which I now submit for assessment on the programme of study leading to the award of **B.Sc in Computing and Information Technology** in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfillment of the requirements of that stated above.

Dated: 2017

Author: Aaron Ward

Introduction

The goal of this assignment is to design a multiple layer neural network using the Tensorflow Library for a predictive model. Many factors will be taken into consideration for this project, such as the number of nodes needed in each layer, the number of layers and different learning rates. This report will document the two phases in implementing a neural network: Network Training and Network Evaluation. During training the model shall be trained by experimenting with different hyper parameters and the evaluation phase will load the trained model and test the metrics of the model in regards to data that it has not seen. The data being used consists of 500 rows of example data and 100 rows for evaluation.
The structure of this report will go by a design section with illustration, the training process and the evaluation process, then followed by a conclusion.

Assignment

Design of Network

Initially, the design of the network will consist of two input nodes for the two values contained in the training dataset. The learning rate will be set to 0.01 because it said that is a good rate to begin training at (https://www.youtube.com/watch?v=jWT-AX9677k). This will rate will be reduced after each number of runs. On the output layer, only one output node will be made to output the predicted value. The network shall also contain one hidden layer. This will consist of two nodes but will be increased incrementally to evaluate the optimal number. The goal of this network is see what is the optimum network topology to reduce the error in the shortest amount of time.

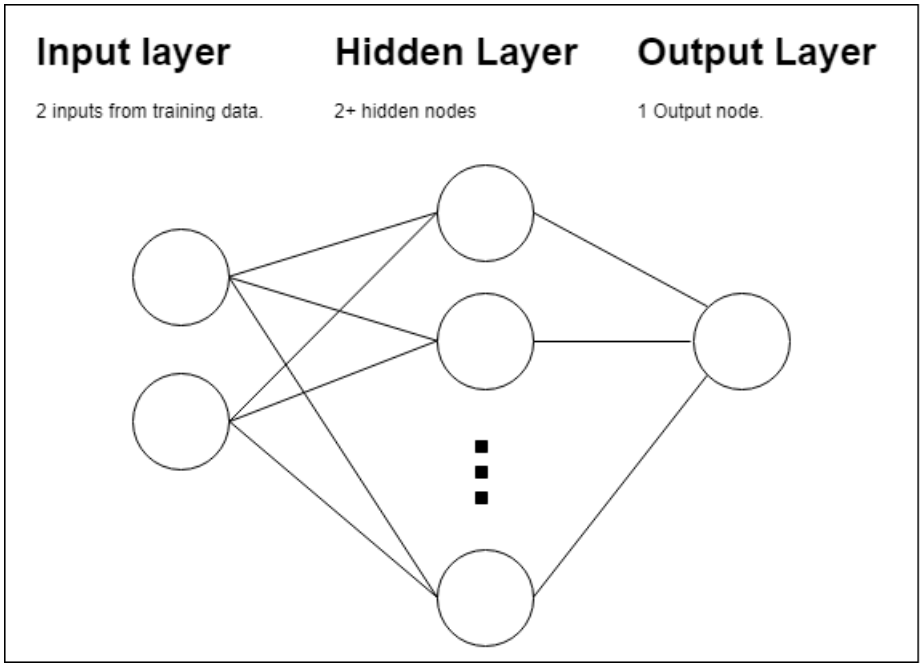

Figure 1: Architecture of neural network

Network Training

In this section we shall train the network. Each score declared in the tables are an average score over ten runs for each parameter. This process was done incrementally by first deciding a number of epochs, this is followed by a number of hidden layer nodes. And lastly, decreasing the learning rate from 0.01 to 0.0001.

1000 Epochs

A good starting off point to train the network began with a thousand epochs. Initially the network was trained with 2 hidden layer nodes for 1000 epochs. The cost, error and time are measured and the average over ten runs is calculated.

2 Hidden Layer Nodes

In the table 1 it can be seen that although the training time is low, the error rate never met the satisfactory rate 0.01, and the cost is relatively high. Therefore, it can be noted that is not a desirable model.

Learning rate	Cost	Error	Time
0.01	51.992	0.14	0.932
0.001	58.2	0.1916	1.006
0.0001	119.40	0.504	0.846

Table 1: Two Hidden Layer Nodes for 1000 Epochs

3 Hidden Layer Nodes

The three node model performed significantly better than the previous, with a low error score of 0.0098 on the learning rate 0.01, so this model can be taken into consideration for the evaluation model. However, the other scores did not meet the tolerable error score as the learning rate was decreased, as seen in Table 2.

Learning rate	Cost	Error	Time
0.01	1.34	0.0098	0.72
0.001	17.83	0.116	0.89
0.0001	119.47	0.495	0.874

Table 2: Three Hidden Layer Nodes for 1000 Epochs

2000 Epochs

Following a successful attempt with the previous model, further experimentation was done by increasing the number of epochs the network will be trained with.

2 Hidden Layer Nodes

Firstly, the network is reduced back to 2 hidden layer nodes and the average scores of the cost, error and elapse time are calculated. As seen in Table 3, no improvement was made to the model as the cost, error and training time were all increased.

Learning rate	Cost	Error	Time
0.01	50.77	0.147	1.606
0.001	51.985	0.123	1.606
0.0001	115.24	0.49	1.62

Table 3: Two Hidden Layer Nodes for 2000 Epochs

3 Hidden Layer Nodes

Secondly, an extra node was added to the model. It can be noted from Table 4 that a significant improvement was made to the model by adding an extra node to the hidden layer. The cost was reduced to 0.658 and error to 0.0053 when the learning rate was set to 0.01.

Learning rate	Cost	Error	Time
0.01	0.658	0.0053	1.94
0.001	8.72	0.054	1.81
0.0001	117.19	0.489	1.66

Table 4: Three Hidden Layer Nodes for 2000 Epochs

From these figures it is evident that the 2 node model trained for 1000 epochs and the 3 node model trained for 2000 epochs performed the best. Although the training time is shorter with 2 node network that was trained for 1000 epochs, the latter has a better accuracy. Also to take into consideration, there is only an average of 1.22 second training difference, which is not a high cost to pay. Therefore, the model with 3 hidden layer nodes that, trained for 2000 epochs, and the learning rate set to 0.01 shall be used for evaluation.

Network Evaluation

Upon inspection of the trained models results, the model was again trained with the optimal parameters specified and saved to disk, but upon evaluating the model with new data, the model did not perform as well as it did when training. An average error of 0.02775318, over 10 runs, was the result of using this model. Therefore, further training was implemented. In order reduce the error, the number of nodes and epochs were increased incrementally. These numbers are average results over 10 runs from the evaluation scripts in respect to the newly trained models. All models were trained with the 0.01 learning rate. See Table 5 for results.

Number of Epochs	Nodes	Average Error
3000	3	0.02332
3000	4	0.0224
3000	5	0.0184
3000	6	0.0128
3000	7	0.0175
3000	8	0.0125
4000	3	0.0256
4000	4	0.0139
4000	5	0.0143
4000	6	0.0118
4000	7	0.0121
4000	8	0.0118
5000	3	0.0184
5000	4	0.0146
5000	5	0.0145
5000	6	0.0115
5000	7	0.0105
5000	**8**	**0.0093**

Table 5: Incremental Experimentation with Parameters

As seen from this iterative process in the table above, the most optimal model for evaluation required 8 hidden layer nodes and to be trained for 5000 epochs in order to result in a error below 0.01.

Conclusion

In conclusion, the model received two inputs, one hidden layer and one output. The network was trained using 500 examples in the training dataset. The model performed well during training with 3 hidden layer nodes, but failed to reach the minimum tolerable error when fed new data. Therefore, further experimentation was conducted in order to find the optimal number of training epochs, learning rate and nodes that were required in order to reach an error below 0.01 in the evaluation phase. The project was successful in implementation, with an average evaluation error of 0.00093.