Note: I talked about the general concepts of the problem with Diandra Prioleau but all of the works in the homework are mine without any help.

Multilayer Perceptron

In this homework, we have implemented the error backpropagation to increase the accuracy of the neural network. The mlptrain function would do one forward pass and compute the error. Using the backpropagation signal from the error function, the weights in both the hidden layer and the output layer would be updated. The accuracy of a neural network could be impacted by many factors including but not limited to the learning rate, the number of iterations, and the number of hidden nodes.

Each of these factors would be investigated according to our code and discuss the corresponding plots to show how it had affected our results.

Learning rate:

Learning rate tuning in the neural network is a challenging task. It could affect the performance and the entire training process. As it was mentioned in the class, the learning rate would show the step size in the learning process. A large value of learning rate could lead to too much oscillating between the values to find the local optima.

In contrast, too small values of learning rate could lead to smaller step size and ultimately longer time to converge to the local optima. As it has been suggested in [1], most developers pick different learning rates and check if it is performing well on the data and model or not. Therefore, in this assignment, a range of learning parameters equal to [0.001,0.01,0.1,0.5,1] has been used to check if there is any difference within the error. The error curve based on different learning parameters is shown in Figure 1.

It has been shown that learning rate equal to one has the least error while performing on the model and use the backward and forward pass. However, we investigated 0.1,0.01 and 1 for the number of iterations to indeed find the best one.

```
Confusion matrix is:
[[ 60.   0.]
 [  0.  60.]]
Percentage Correct:  100.0
```

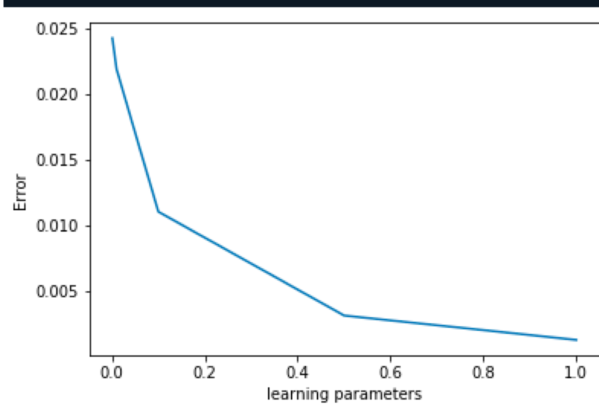

Figure 1: learning rate: [0.001,0.01,0.1,0.5,1] , hidden nodes: 9, niterations :100000

Number of iterations:

We investigate how the number of iterations while doing the backpropagation and forward pass could help to reach higher accuracy. Therefore, 1000000 was chosen as the number of iterations and backpropagation was conducted followed by a forward pass with the learning rate of 0.01 and hidden nodes of 9. Fixed number of iterations would do to test the performance and converge to the local optima. The error vs. the number of iterations plot has been shown in Figure 2. As it has been demonstrated after a couple of iterations, the model error would not change anymore which implies the convergence of the model to the local optima. Less number of iterations could lead to less accuracy as the whole training process including the backpropagation and updating the weights are happening less. Therefore, we need to make sure to pick a higher number of iterations to reach the point at which error is minimum. As we mentioned previously, the learning rate equal to one could cause more oscillating between the points. In Figure 2.b, when the learning rate is 1 with the same number of iterations, the error function curve is swinging more and finally end up with the less overall accuracy according to the confusion matrix in the third picture.
In Figure 2.a with more iterations and medium learning rate, %100 accuracy was achieved. As it is shown in Figure 2.a and 2.c, the same learning rate with a smaller number of iterations lead to lower accuracy.

```
Confusion matrix is:
[[ 60.   0.]
 [  0.  60.]]
Percentage Correct:  100.0
```

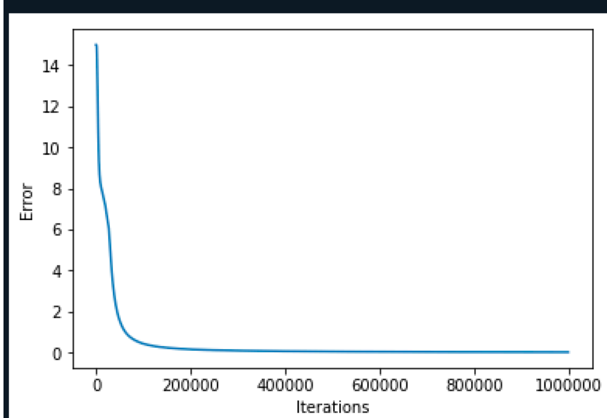

Figure 2.a: learning rate 0.01, niterations; 1000000, hidden nodes: 9

```
Confusion matrix is:
[[ 40.   0.]
 [ 20.  60.]]
Percentage Correct:  83.3333333333
```

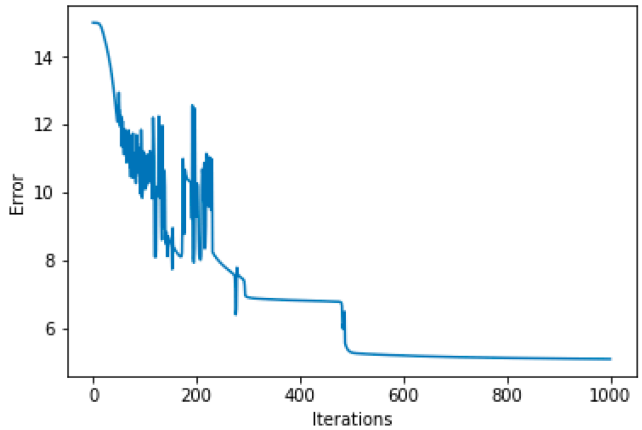

Figure 2.b: learning rate: 1, hidden nodes: 9, niterations :1000000

```
Confusion matrix is:
[[ 40.   0.]
 [ 20.  60.]]
Percentage Correct:  83.3333333333
```

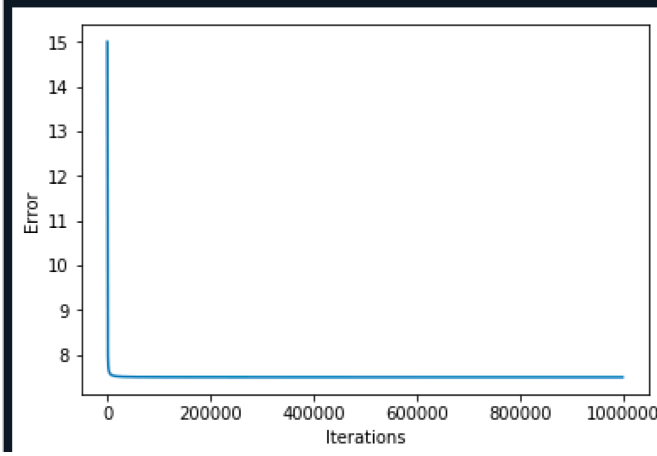

Figure 2.c: learning rate: 0.01, hidden nodes: 9, niterations :100000

Hidden nodes:

The two previous experiment using only 3 and nine hidden nodes to investigate the impact of learning rates and the number of iterations. However, in this section, a varying range of hidden nodes would be considered to see how the performance is changing according to the number of hidden nodes. The hidden node ranges include [3,5,7,9].

```
Confusion matrix is:
[[ 40.   0.]
 [ 20.  60.]]
Percentage Correct:  83.3333333333
```

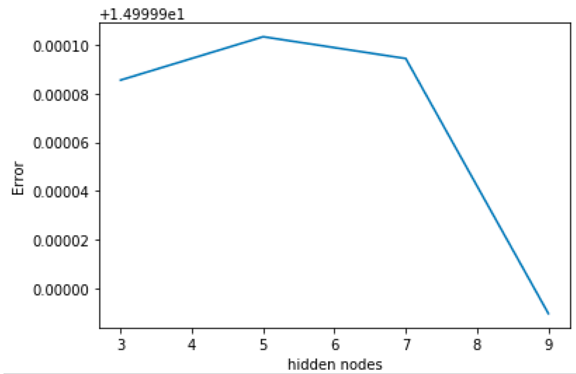

Figure 3: learning rate: 0.01, hidden nodes: [3,5,7,9], niterations :1

The above plot suggests that nine hidden nodes could achieve better results in comparison with the other values. This result confirms what was found in the previous section with 1000000 iterations (Figure 2.a). However, there is one concern that making the model too complicated is safe or not. As there is not a separate validation data and the training data is small, it cannot be confirmed whether increasing the number of hidden nodes would not lead to overfitting. Therefore, there is a need to expand the data to cross-validate and deciding the number of hidden nodes which is working correctly on training and test dataset.

References:

1. https://medium.freecodecamp.org/how-to-pick-the-best-learning-rate-for-your-machine-learning-project-9c28865039a8