School of Computer Science

Connectionist Computing

Report for Assignment

Xiaosheng Liang - 15211913

Question1&2:

The code was run in jupyter. Here are prediction outputs after training and y in dataset.

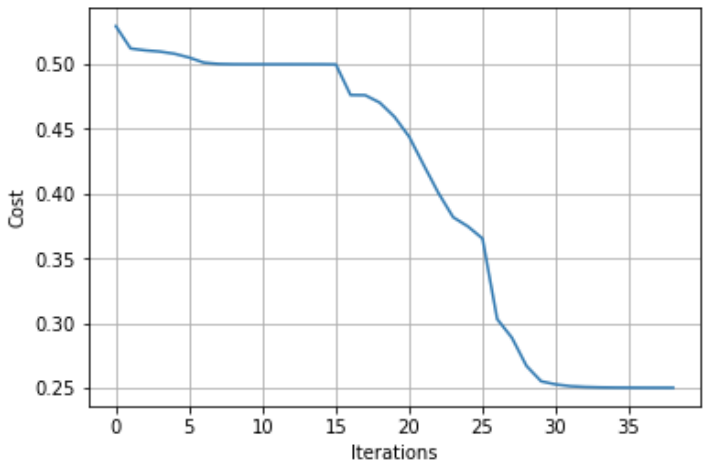
```
In [248]: print (NN.forward(X))

[[  3.88386611e-05]
 [  9.99991957e-01]
 [  4.99982062e-01]
 [  4.99998989e-01]]

In [249]: print (y)

[[ 0.]
 [ 1.]
 [ 1.]
 [ 0.]]
```

The cost calculated by given prediction outputs and y. It shows after 30 iterations, the cost is set to be 0, which means prediction outputs and y is getting closed.



Question3:

Understanding of Multi Layered Perceptron：

- Input the data into network whose weights are random;
- Get the predition of Y and calculate the cost function which is difference between predition and Y in dataset
- Calculate minimal cost and corresponding weights using backpropagation method
- input the test data into network whose weights are obtained before and get

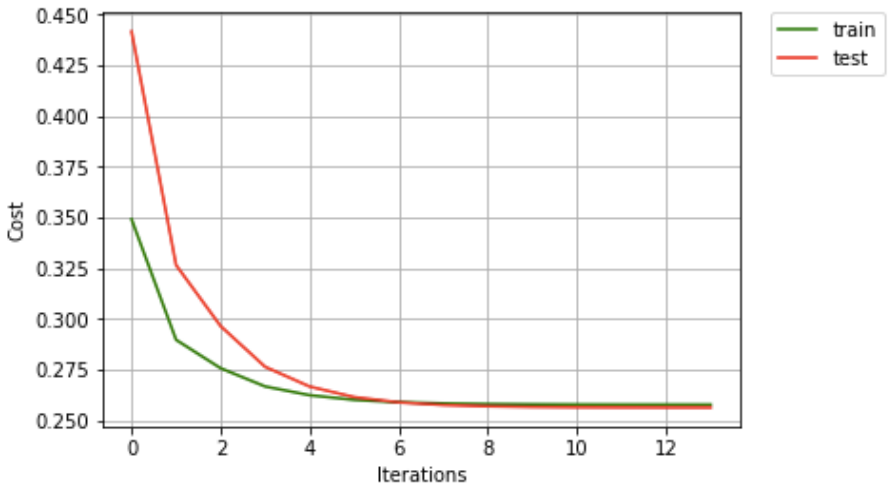
accurate prediction

Result of training:

```
In [258]: print (NN.feedforward(testX))
[[  5.67514578e-05]
 [  4.43362644e-05]
 [  2.57736024e-05]
 [  6.19167770e-06]
 [  1.20189711e-05]
 [  2.08965318e-05]
 [  1.23332072e-04]
 [  6.28457875e-07]
 [  1.44264197e-05]
 [  2.25070409e-05]]

In [259]: print (testY)
[[ 0.0401262 ]
 [ 0.06759616]
 [-0.3356125 ]
 [ 0.81438776]
 [ 0.87272564]
 [-0.61840449]
 [ 0.36435506]
 [-0.57980794]
 [ 0.00720828]
 [-0.75655464]]
```

Results of cost function in cost and training data.

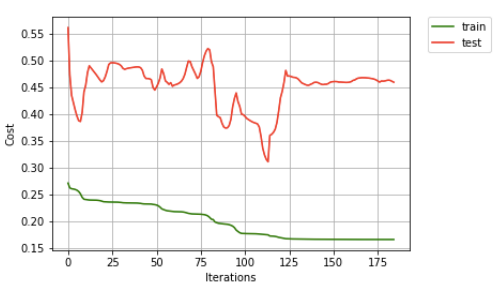


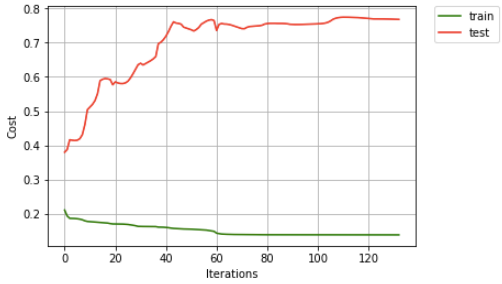
Question4:

Although each calculating step is pretty straight forward, it's still relatively easy to make mistakes. Code with incorrectly implemented gradients may cause mistakes. After all, there might be some subtle problems in the codes, a good

solution is to test the gradient computation part of code. In addition, Overfitting is another problem.

In the processing of training, there are lots of mistakes happened. The results of prediction is not stable, costs are as follow:





After some methods were tried, the results have been improved. Increasing the number of hidden units or number of input data are able to improve the problem.