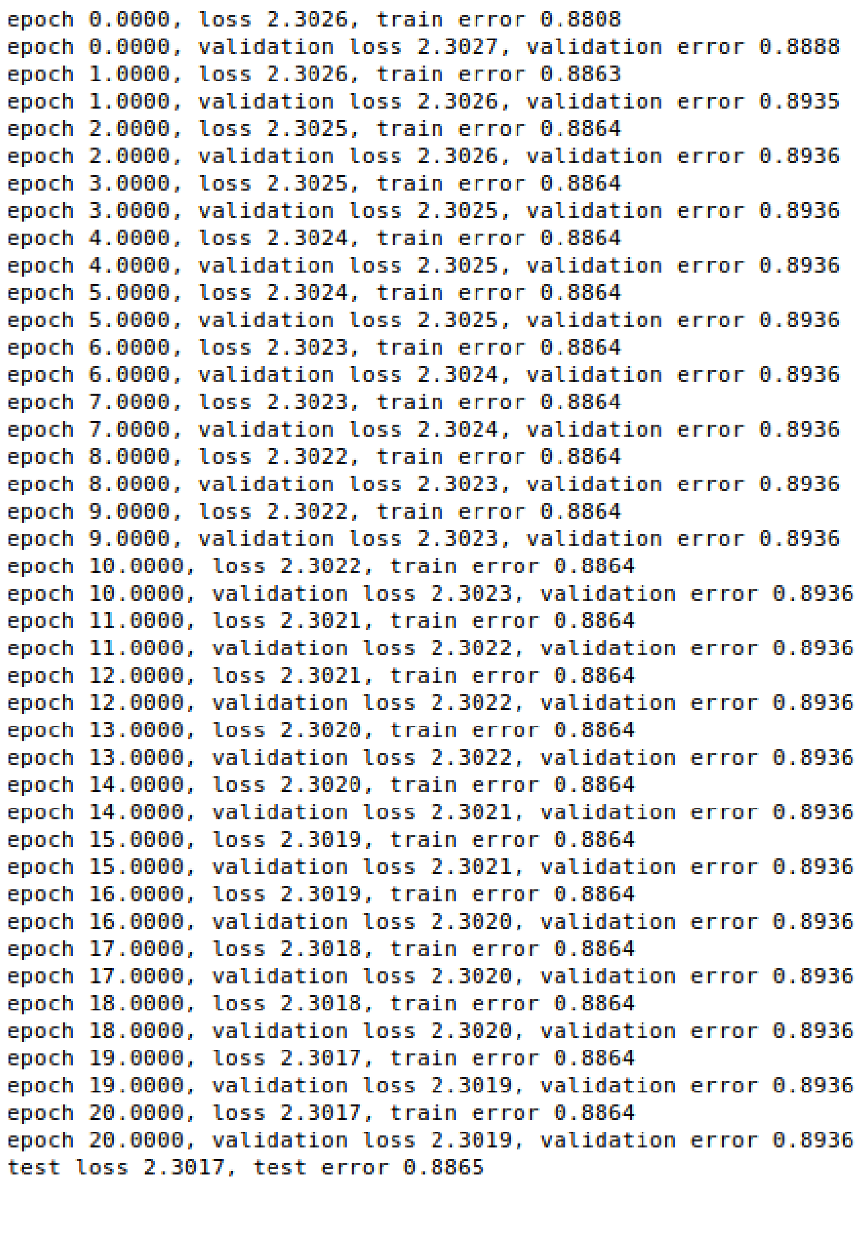
Usman Ahmed
Deep Learning Lab
Exercise-1 Report

This was an excellent exercise which covered the basics of neural network. The code was well structured and helpful and allowed us freedom to work on only the essential parts and not implement everything from scratch. The forward propagation was the easier part of the exercise. Back propagation however was a bit tricky for me to implement.

I started off this exercise by coding the activation functions for forward propagation and fprop method in the FullyConnectedLayer class. After that I moved on to implement input_grad for linear and softmax output. I then proceeded to implement bprop and then gradient descent and stochastic gradient descent. Finally I implemented gradient checking.

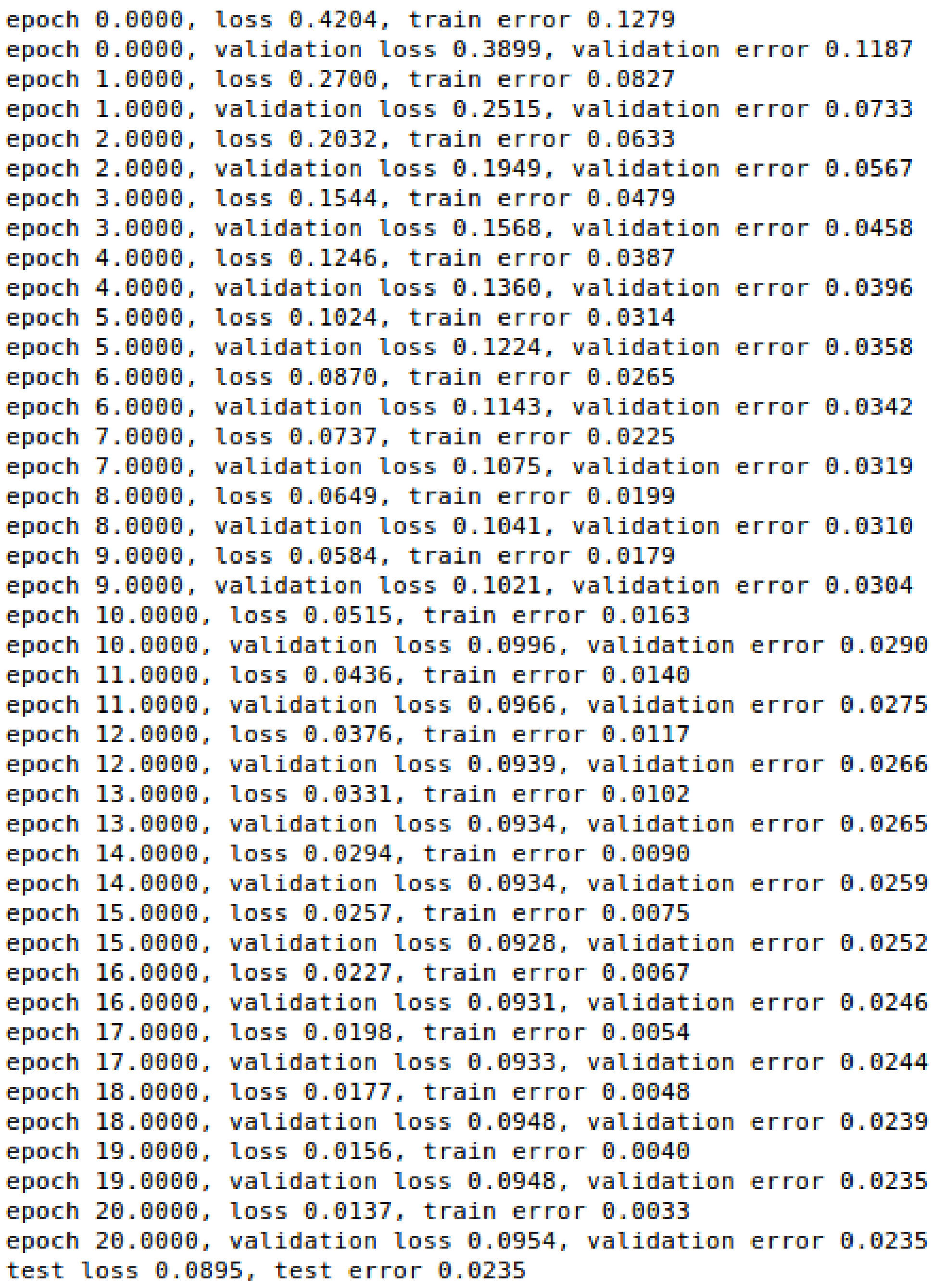
I then set up a neural network consisting of three fully connected layers. I played around with the network by setting different activation functions and units for different layers. Finally I settled for 100, 180 and 10 units respectively and activation function of relu and tanh because I noticed better accuracy for these values.

Initially I was using gradient descent and I got the following results.



```
epoch 0.0000, loss 2.3026, train error 0.8808
epoch 0.0000, validation loss 2.3027, validation error 0.8888
epoch 1.0000, loss 2.3026, train error 0.8863
epoch 1.0000, validation loss 2.3026, validation error 0.8935
epoch 2.0000, loss 2.3025, train error 0.8864
epoch 2.0000, validation loss 2.3026, validation error 0.8936
epoch 3.0000, loss 2.3025, train error 0.8864
epoch 3.0000, validation loss 2.3025, validation error 0.8936
epoch 4.0000, loss 2.3024, train error 0.8864
epoch 4.0000, validation loss 2.3025, validation error 0.8936
epoch 5.0000, loss 2.3024, train error 0.8864
epoch 5.0000, validation loss 2.3025, validation error 0.8936
epoch 6.0000, loss 2.3023, train error 0.8864
epoch 6.0000, validation loss 2.3024, validation error 0.8936
epoch 7.0000, loss 2.3023, train error 0.8864
epoch 7.0000, validation loss 2.3024, validation error 0.8936
epoch 8.0000, loss 2.3022, train error 0.8864
epoch 8.0000, validation loss 2.3023, validation error 0.8936
epoch 9.0000, loss 2.3022, train error 0.8864
epoch 9.0000, validation loss 2.3023, validation error 0.8936
epoch 10.0000, loss 2.3022, train error 0.8864
epoch 10.0000, validation loss 2.3023, validation error 0.8936
epoch 11.0000, loss 2.3021, train error 0.8864
epoch 11.0000, validation loss 2.3022, validation error 0.8936
epoch 12.0000, loss 2.3021, train error 0.8864
epoch 12.0000, validation loss 2.3022, validation error 0.8936
epoch 13.0000, loss 2.3020, train error 0.8864
epoch 13.0000, validation loss 2.3022, validation error 0.8936
epoch 14.0000, loss 2.3020, train error 0.8864
epoch 14.0000, validation loss 2.3021, validation error 0.8936
epoch 15.0000, loss 2.3019, train error 0.8864
epoch 15.0000, validation loss 2.3021, validation error 0.8936
epoch 16.0000, loss 2.3019, train error 0.8864
epoch 16.0000, validation loss 2.3020, validation error 0.8936
epoch 17.0000, loss 2.3018, train error 0.8864
epoch 17.0000, validation loss 2.3020, validation error 0.8936
epoch 18.0000, loss 2.3018, train error 0.8864
epoch 18.0000, validation loss 2.3020, validation error 0.8936
epoch 19.0000, loss 2.3017, train error 0.8864
epoch 19.0000, validation loss 2.3019, validation error 0.8936
epoch 20.0000, loss 2.3017, train error 0.8864
epoch 20.0000, validation loss 2.3019, validation error 0.8936
test loss 2.3017, test error 0.8865
```

After Implementing stochastic gradient descent I used it as descent type. I noticed that there was a huge difference in accuracy. Stochastic gradient descent produced much better results. The results also got much better with increased epoch. Following are the results I got by using stochastic gradient descent



```
epoch 0.0000, loss 0.4204, train error 0.1279
epoch 0.0000, validation loss 0.3899, validation error 0.1187
epoch 1.0000, loss 0.2700, train error 0.0827
epoch 1.0000, validation loss 0.2515, validation error 0.0733
epoch 2.0000, loss 0.2032, train error 0.0633
epoch 2.0000, validation loss 0.1949, validation error 0.0567
epoch 3.0000, loss 0.1544, train error 0.0479
epoch 3.0000, validation loss 0.1568, validation error 0.0458
epoch 4.0000, loss 0.1246, train error 0.0387
epoch 4.0000, validation loss 0.1360, validation error 0.0396
epoch 5.0000, loss 0.1024, train error 0.0314
epoch 5.0000, validation loss 0.1224, validation error 0.0358
epoch 6.0000, loss 0.0870, train error 0.0265
epoch 6.0000, validation loss 0.1143, validation error 0.0342
epoch 7.0000, loss 0.0737, train error 0.0225
epoch 7.0000, validation loss 0.1075, validation error 0.0319
epoch 8.0000, loss 0.0649, train error 0.0199
epoch 8.0000, validation loss 0.1041, validation error 0.0310
epoch 9.0000, loss 0.0584, train error 0.0179
epoch 9.0000, validation loss 0.1021, validation error 0.0304
epoch 10.0000, loss 0.0515, train error 0.0163
epoch 10.0000, validation loss 0.0996, validation error 0.0290
epoch 11.0000, loss 0.0436, train error 0.0140
epoch 11.0000, validation loss 0.0966, validation error 0.0275
epoch 12.0000, loss 0.0376, train error 0.0117
epoch 12.0000, validation loss 0.0939, validation error 0.0266
epoch 13.0000, loss 0.0331, train error 0.0102
epoch 13.0000, validation loss 0.0934, validation error 0.0265
epoch 14.0000, loss 0.0294, train error 0.0090
epoch 14.0000, validation loss 0.0934, validation error 0.0259
epoch 15.0000, loss 0.0257, train error 0.0075
epoch 15.0000, validation loss 0.0928, validation error 0.0252
epoch 16.0000, loss 0.0227, train error 0.0067
epoch 16.0000, validation loss 0.0931, validation error 0.0246
epoch 17.0000, loss 0.0198, train error 0.0054
epoch 17.0000, validation loss 0.0933, validation error 0.0244
epoch 18.0000, loss 0.0177, train error 0.0048
epoch 18.0000, validation loss 0.0948, validation error 0.0239
epoch 19.0000, loss 0.0156, train error 0.0040
epoch 19.0000, validation loss 0.0948, validation error 0.0235
epoch 20.0000, loss 0.0137, train error 0.0033
epoch 20.0000, validation loss 0.0954, validation error 0.0235
test loss 0.0895, test error 0.0235
```