CS726 – Advanced Machine Learning — Homework3

Name : Pulkit Katdare,
Roll no 130100006

1 Introduction

To come up with an with a Neural Network for Devnagri Character Recognition. I tried to implement a couple of Neural Network Architectures namely Multilayer Feedforward Network and Multi layer convolution networks. But I was getting maximum accuracy in covolutional network. I have implemented all the exercises mentioned in the assignment on this network.

Please note that the skeleton code on which I have tried to implement various exercises given in the assignment has been taken from [1]. Also as the dataset was too big for system I have done a pre processing step in the assignment where the image size was reduced from 320×320 to 28×28. This is included in the pre-processing code.

One of the interesting thins to note in this case is that the data sets given in the assignment was not sparse. In the sense that all the blank space given in the actual image data set was filled with white spaces instead of black ones like the MNIST dataset. It was noted that accuracy on the validation set with a white blank space was significantly lesser than($\sim 60\%$) as opposed to taking a contrast of the same image(this makes the image matrix sparse). We get a significantly higher accuracy($\sim 92\%$). This may be due to the fact that it takes a longer time and a larger dataset for the network to learn a non-sparse representation than a sparser one.

(a) **Effect of changing the hidden unit type from RelU to Sigmoid to Tanh**
We observe that from ReLu, tanh, and sigmoid. ReLu works the best. There are fewer fluctuations while using the ReLu hidden unit type. Secondly if we observe carefully then ReLu has a slightly greater accuracy than Tanh and Tanh has a slightly greater accuracy than sigmoid. As shown by the graph below,

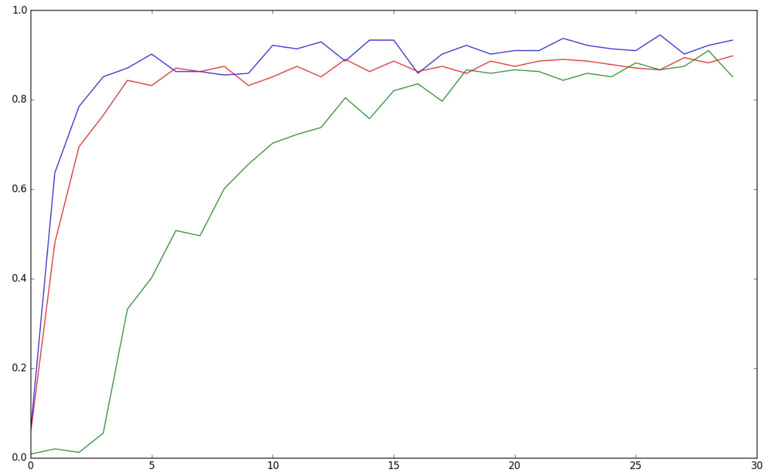

Figure 1: The blue line corresponds to RelU, red line corresponds to tanh, green line corresponds to sigmoid type hidden unit. The graph is shown for test accuracy for first 30 iteration

(b) **The accuracy on the validation dataset for increasing number of levels** We observe that by increasing the number of deep layers in the network there is some negligible changes in the accuracy of the network. Network with one hidden layer gives us an accuracy of (92%). Network with 2 hidden layer gives us a maximum accuracy of ($\sim$ 93%) while a layer with zero hidden layer gives a maximum accuracy of ($\sim$ 91%). Thus, there negligible difference between this type of changes in the number of hidden layer. However, A network with one hidden layer is still preferred.

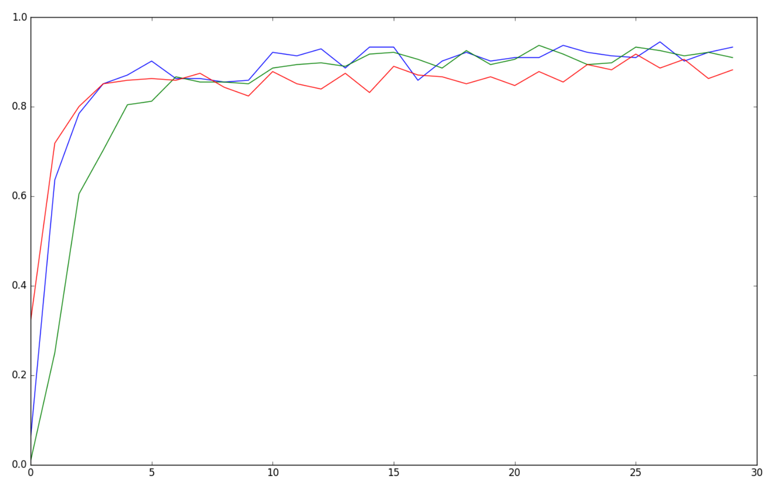

Figure 2: The blue line corresponds to 1 layer, red line corresponds to 0 layer, green line corresponds to 2 layer hidden unit. The graph is shown for test accuracy for first 30 iteration

(c) **Accuracy with increasing width for a fixed number of levels**
As we can see above there isn't much difference in the accuracy in changing the width of the

neurons in the hidden layer in the iteration. This may have been due to two reasons . One, I have tested this dataset for only 3 sample widths namely, 625,500,1000. So the negligible changes are seen due to this. Secondly, it is possible that larger width needs more time to train. Thus, due to these two reasons nothing concrete can be said about this.

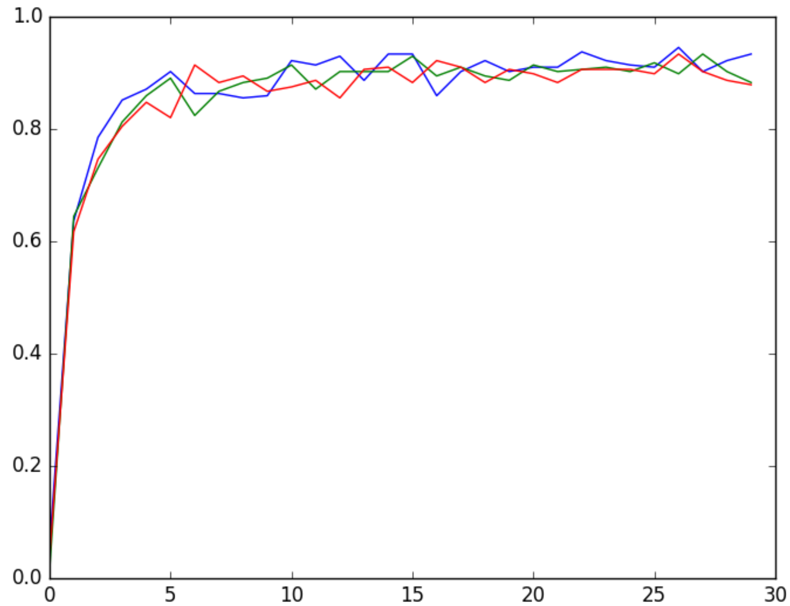

Figure 3: The blue line corresponds to 625 neuron layer, red line corresponds to 500 neurons layer, green line corresponds to 1000 neuron layer hidden unit. The graph is shown for test accuracy for first 30 iteration

(d) **Effect of different kinds of regularizers: L2 regularizer vs dropout**
We see that drop out does a better job than L2 regularization. Which is clear from the graph given below for both the types of network,

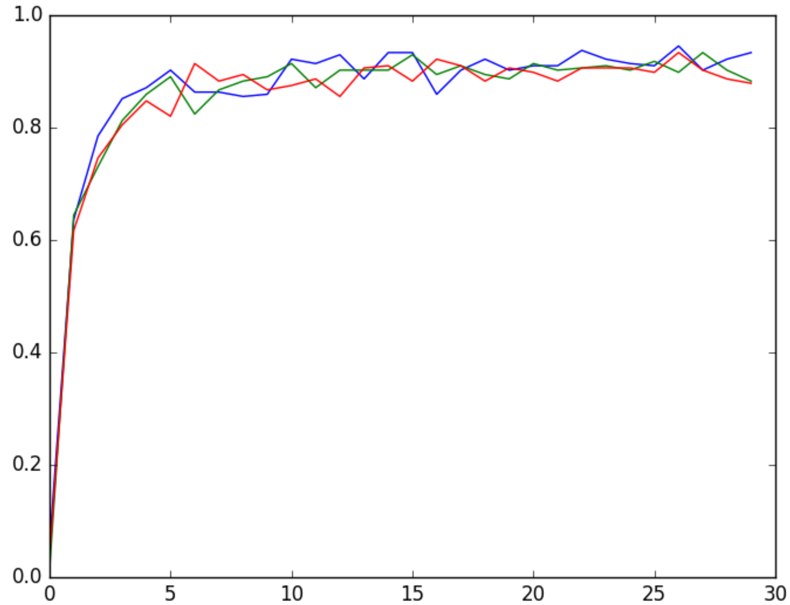

Figure 4: The blue line corresponds drop out, green line corresponds to L2 regularization The graph is shown for test accuracy for first 30 iteration

(e) **Accuracy with different learning rates**
We see that the accuracy is maximum for learning rate of 0.01 and is the optimal one as shown by the graph below,

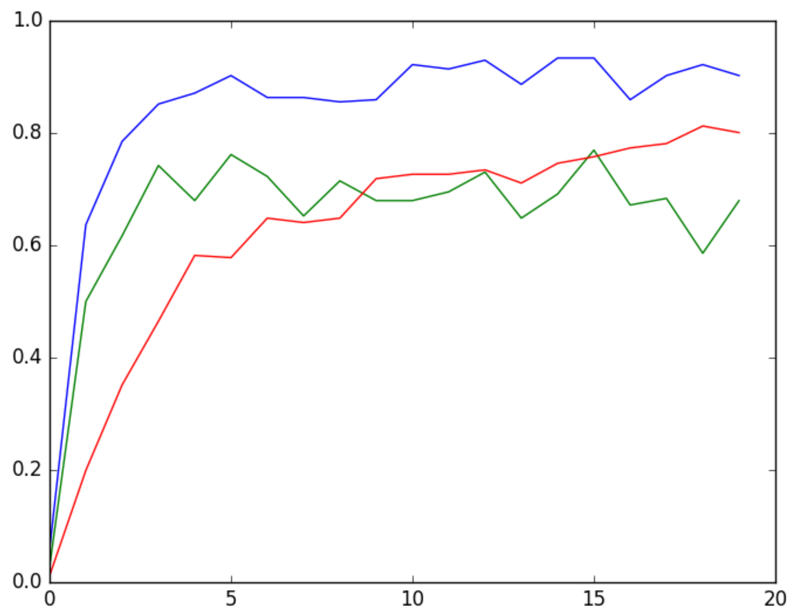

Figure 5: The blue line corresponds to 0.001 learning rate, red line corresponds to 0.0001 learning rate, green line corresponds to 0.01 learning rate. The graph is shown for test accuracy for first 20 iteration

2 References

1. github: https://github.com/nlintz/TensorFlow-Tutorials