Amir Kashipazha
CSCI 5622 HW 3 Analysis

1.2 Analysis

1. What is the structure of your neural network (for both tinyTOY and tinyMNIST dataset)? Show the dimensions of the input layer, hidden layer and output layer.

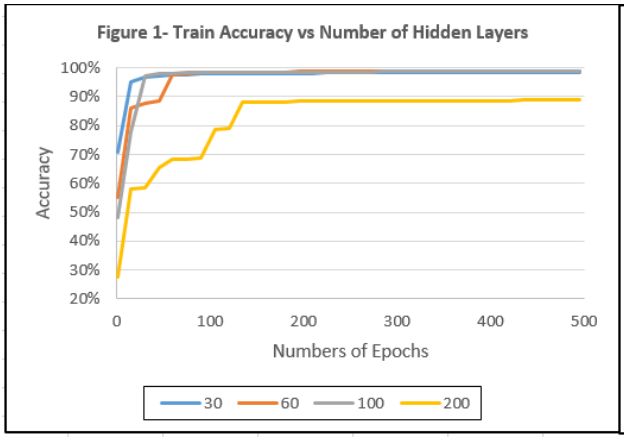
Table 1 shows structure of the neural network of both tinyTOY and tinyMNIST.

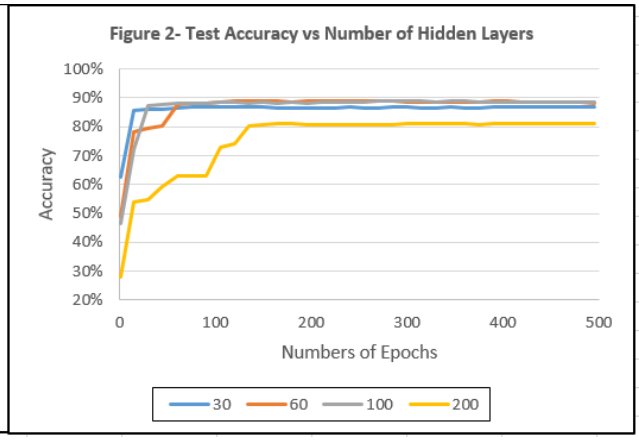
Dataset - Layer	Input	Hidden	Output
tinyTOY	2	30	2
tinyMNIST	196	100	10

Table 1: Dimensions of input layer, hidden layer and output layer

2. What the role of the size of the hidden layer on train and test accuracy (plot accuracy vs. size of hidden layer using tinyMNIST dataset)?

Figures 1 and 2 show the effect of size of hidden layer on training and test datasets. As these Figures show with 30 hidden layers, accuracy begins with larger numbers for both train and test. The highest accuracy was for 100 and 60 hidden layers. The difference in accuracy between 60 and 100 hidden layers was smallest for 150 epochs. Even though we got higher accuracy with 60 and 100 hidden layers for both test and train datasets, the accuracy differences between them with 30 hidden layer was less than 2 percent from epoch 150. The Figures also indicate that using large numbers of hidden layers can hurt accuracy as we can see for 200 hidden layers. Therefore, we need to iterate to find the optimum number of accuracy for our model.

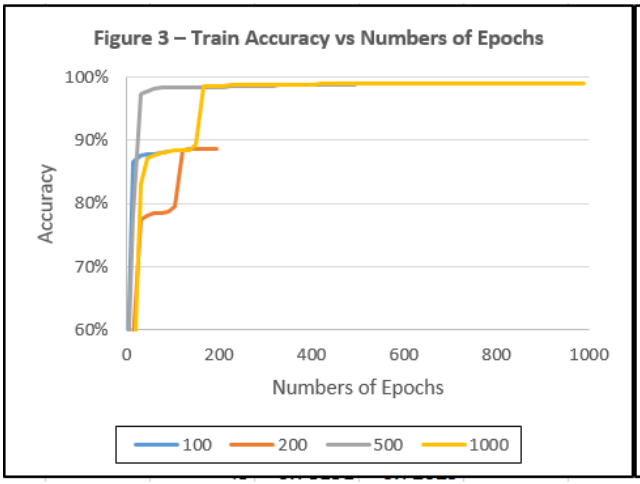


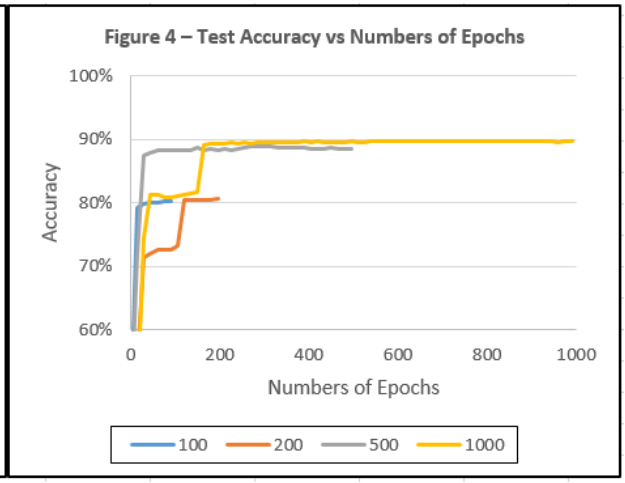


3. How does the number of epochs affect train and test accuracy (plot accuracy vs. epochs using tinyMINST dataset)?

In general accuracy increases when the number of epochs increase. Figures 3 and 4 show the change of accuracy versus number for 100, 200, 500, and 1000 epochs. The number of hidden layers for these tests was 100. As these Figures show, accuracy improves with larger numbers of epochs. This can be seen when the number of epochs increased from 200 to 500. The accuracy increased 10% with this change. However, the accuracy did not increase after 500 epochs. These

Figures also show that with using different numbers of epochs, we get a slightly different pattern of accuracy chance than can be seen for 500 and 1000.





2.2 Analysis

1. Point out at least three layer types you used in your model. Explain what are they used for.

I used the following layers for my model:

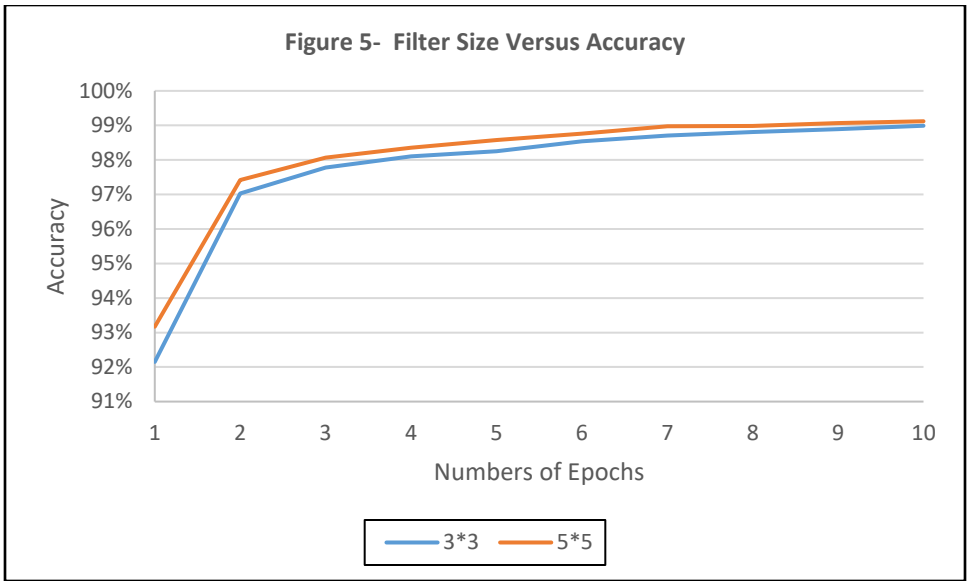
Convolutional Layer: This layer applies convolution operation to the input and then it passes the results to next layer. The parameters of the layer contains sets of filters. The filters have a small fixed size, such as 3*3 or 5*5 pixels. The filters extend through the input. In forward pass, the filters compute the dot product of the entries of filter and input at any position. As filter moves over the width and height of the image, it produces a 2D activation map which correspond to the results of the filter at each position. Then the result will pass to the next layer. This process helps the network learn visual features.

Pooling Layer: This layer reduces the number of parameters and computations in the network progressively by reducing spatial size of representation. The pooling combines the neuron output cluster at one layer into a single neuron in the next layer. Pooling layers mostly used a filter size of 2*2 with a stride of 2. I used **Max Pooling** for this homework which uses the maximum value of each of the clusters of neurons at the prior layer.

Dropout Layer: In general fully connected layers that contain most of the parameters are prone to overfitting. The dropout is one method to reduce overfitting. At each stage of this process each node could drop out form the network with a probability of 1-p or not with probability of p.

2. How did you improve your model for higher accuracy?

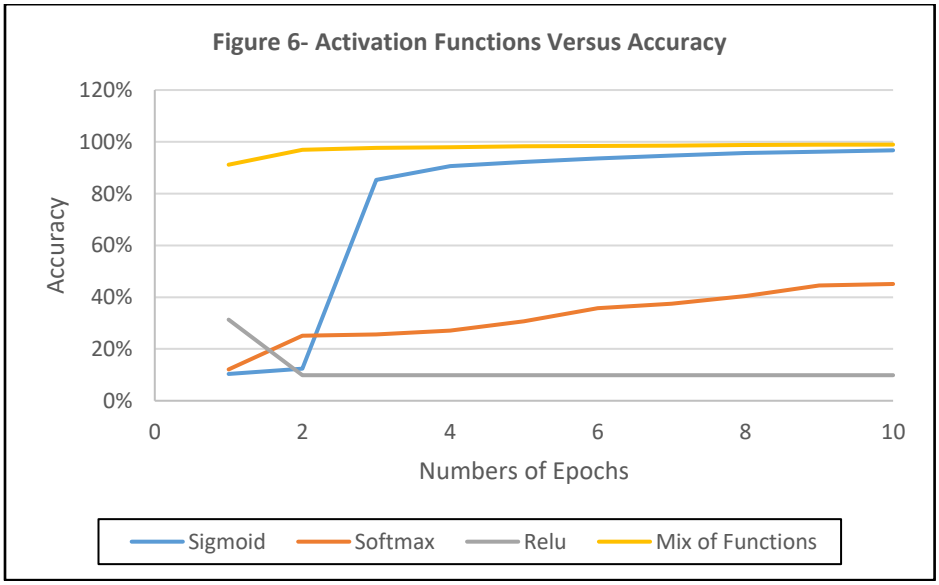
First I used 3*3 and 5*5 filter size in order to see their affect on accuracy. Figure 5 shows how accuracy improved with increasing size of the filter. I used different numbers of batch sizes and found that 128 is the best batch size for this model (see Figure 6). I also tried various activation functions and combinations of them. Results of this effort can be seen in Figure 7.

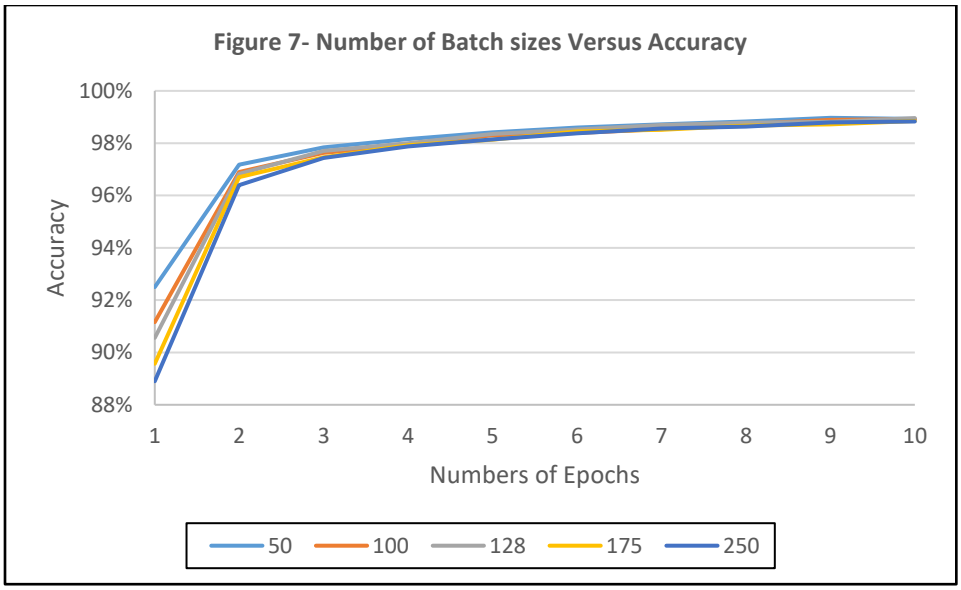


3. Try different activation functions and batch sizes. Show the corresponding accuracy.

Figure 6 shows how various activation functions affect accuracy of the model. For this model I used Sigmoid, Softmax, Relu, and a mix of them. The mix for Figure 6 is the one that has the largest accuracy. I submitted this one. As we can see, the mix activation function hast the best accuracy, followed by sigmoid.

Figure 7 shows how accuracy changes with the number of batch sizes. As Figure 7 shows, the accuracy is higher for first epoch for smaller batch sizes and then the accuracy gets pretty close from epoch 3 to 10. The accuracy was also fairly close for batch sizes at epoch 10. The largest accuracy was for batch size 128 between other batch sizes. So 128 could be the optimum number of batch size for this model.





3.2 Analysis

1. What is the purpose of the embedding layer? (Hint: think about the input and the output).

The embedding layer is used to improve accuracy of the model. It uses a low-dimensional vector instead of one vector to represent words. This leads similar words with similar meaning to have similar vectors. These vectors are a form of pre-training. So by using the embedding layer we tell the network which words are the same so the network needs to learn less about the language. We can use the pre-train vector if we do not have lots of data since it lets the network generalize unseen words.

2. What is the effect of the hidden dimension size in LSTM?

The size of the hidden dimensions affects the accuracy. But it does not mean that the larger number of hidden dimensions leads to the larger number of accuracy. This can be seen in Figure 8. As shown, the accuracy was larger for 128 hidden dimensions, which is larger than 64 and 256 hidden dimensions. This means that 128 could be the optimum number of hidden dimensions for this model.

3. Replace LSTM with GRU and compare their performance.

Figure 9 shows how accuracy changes when LSTM was replaced with GRU. According to Figure 9, accuracy is larger for GRU from epochs 1 to 7 and then their accuracy gets closer from epochs 8 to 15. Accuracy of GRU is slightly larger than LSTM.

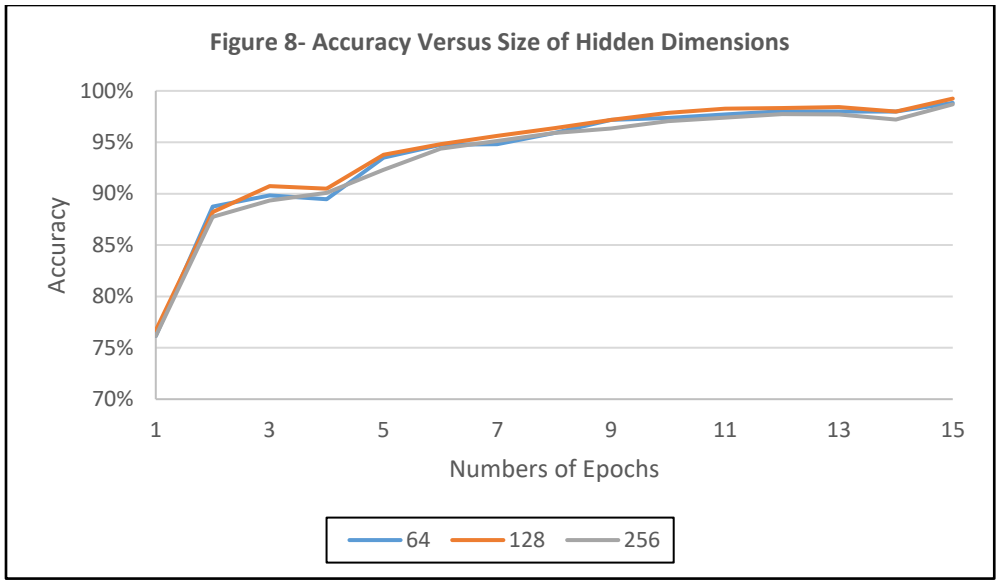


Figure 8- Accuracy Versus Size of Hidden Dimensions
100%
95%
90%
85%
80%
75%
70%
Accuracy
1
3
5
7
9
11
13
15
Numbers of Epochs
64
128
256

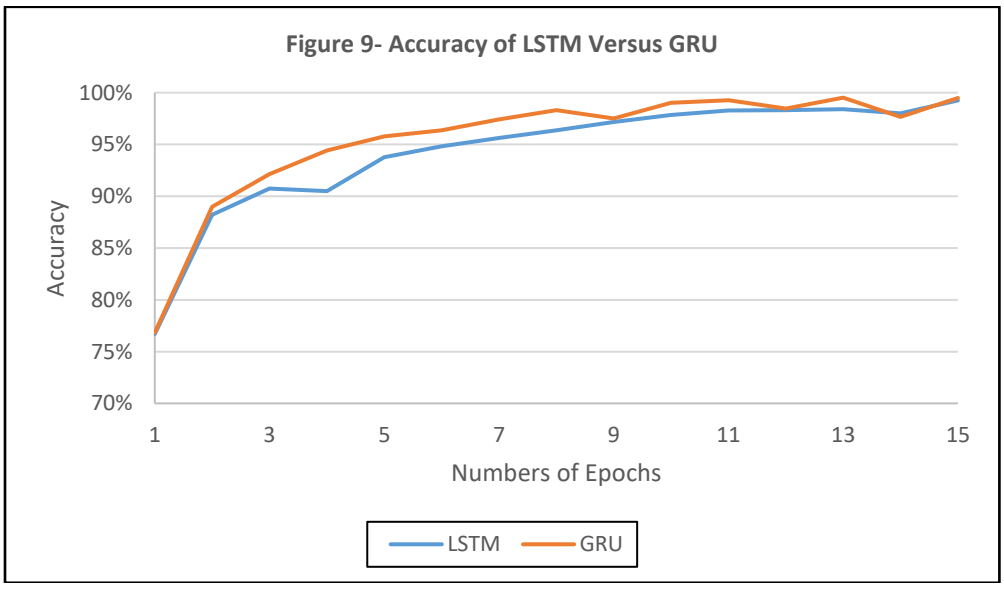


Figure 9- Accuracy of LSTM Versus GRU
100%
95%
90%
85%
80%
75%
70%
Accuracy
1
3
5
7
9
11
13
15
Numbers of Epochs
LSTM
GRU