Machine Learning
CSCI 5622 Fall 2017
Name: Chi Chen

Homework 3
Due Time Oct 13, 2017
CU identity key: chch4713

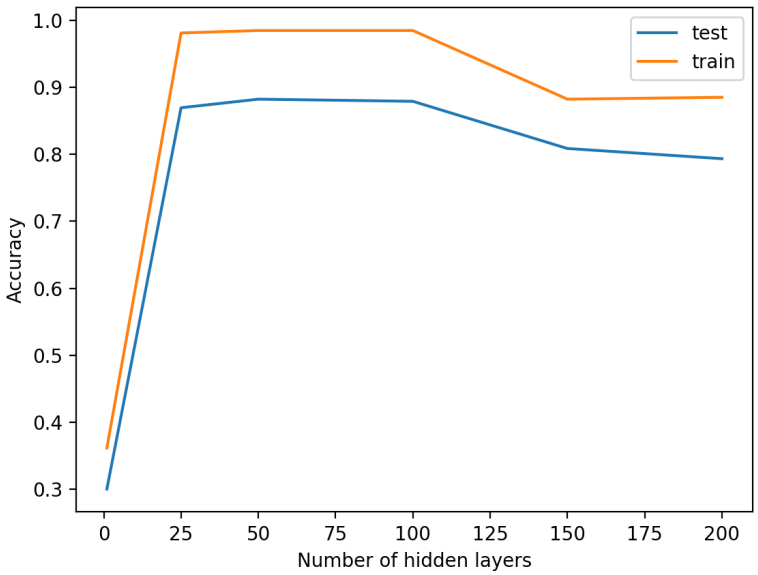
1. Back Propagation (35pts)

I. 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.

The structure of the tinyTOY dataset is [2, 30, 2], which means the input layer were 2 dimensions, the hidden layer were 30 dimensions and the output layer were 2 dimensions.

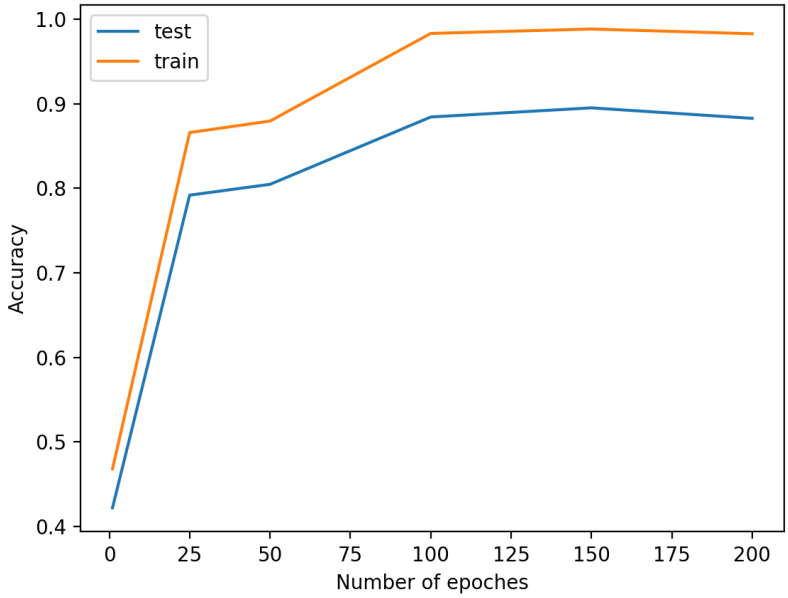
The structure of the tinyMNIST dataset is [196, 100, 10], which means the input layer were 196 dimensions, the hidden layer were 100 dimensions and the output layer were 10 dimensions.

II. 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)?



As you can see, the accuracy increased when numbers of hidden layers increased in first phase. But when the numbers of hidden layers went to large, the model might meet overfitting's problems. The accuracy started to decrease even number of hidden layers we're increasing.

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



In this figure, we could see that when the number of epochs increased, the accuracy will increased too. But it would stop to increase the performance if the epochs were already larger enough, and the epochs number would affect running time of the program, it should be a important issue to choose the proper number of epochs.

2. Keras CNN (35pts)

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

MaxPool2D: In this layer, it selected an area(2*2 in my program) and used the largest value in the matrix to stand for this matrix, so the size of output will be 1/4 of the input in my program.

Flatten: In this layer, input might be a matrix or a tensor, the output will be a 1*N matrix, the N is the length of the input. It is kind of like stretching the input into a single dimension matrix.

Dense: This layer controlled the hidden layer of the CNN, you can choose the activation function and the dimension of this layer.

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

First, I tried to increase the dimension of the hidden layers, it improve the accuracy a little but it took longer time too. Then I used the Dropout function to improve my

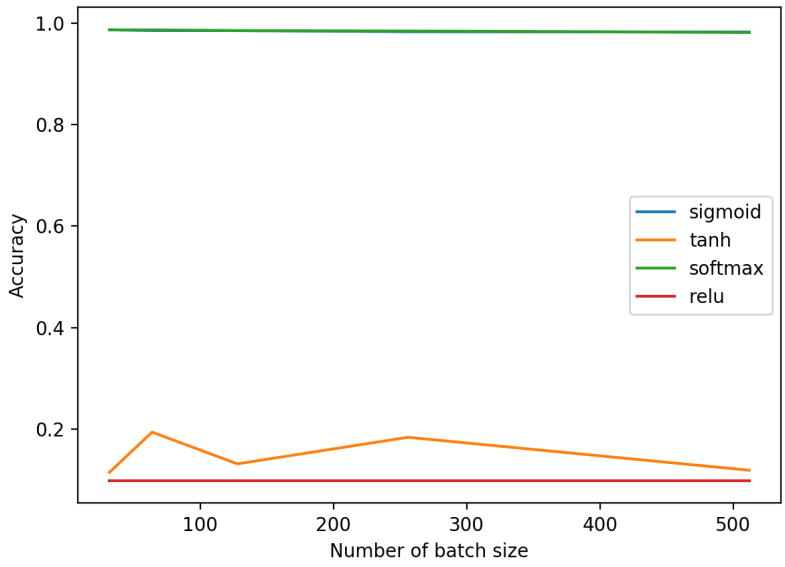
performance, it really helped. And finally I put the different activation function in hidden layers, it also useful and I got 98.38% in this phase.

```
Epoch 15/15
49999/49999 [==============================] - 37s - loss: 0.0440 - acc: 0.9860 - val_loss: 0.0529 - v
al_acc: 0.9838
```

However, after I finished this part, some classmates said there were some problems in normalization part, so I fixed that part, and I got 99.01% in the end.

```
Epoch 15/15
49999/49999 [==============================] - 40s - loss: 0.0093 - acc: 0.9973 - val_lo
ss: 0.0361 - val_acc: 0.9901
```

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



According to the figure, the “sigmoid” and “softmax” function both had almost 100% accuracy, and the “relu” and “tanh” function had really low accuracy. And the batch sizes didn’t really affect the accuracy a lot.

3. Keras RNN (30pts)

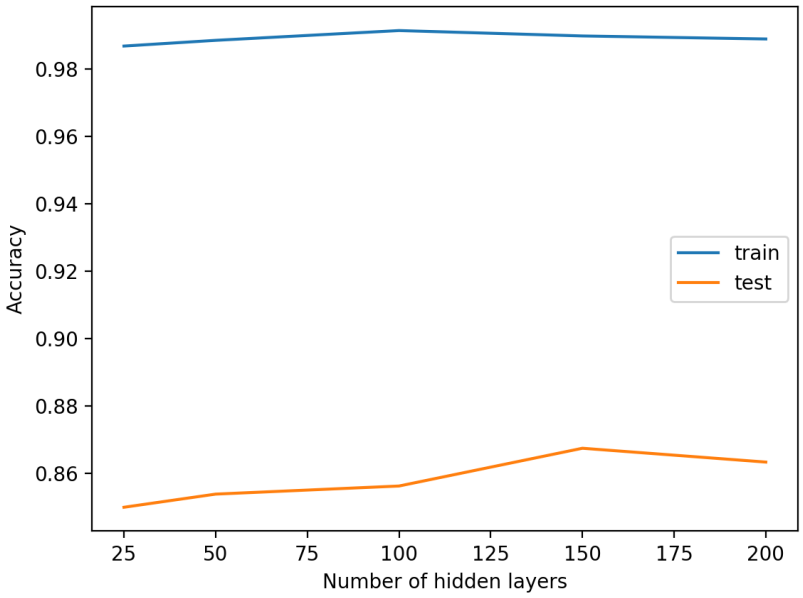
I. What is the purpose of the embedding layer? (Hint: think about the input and the output).

Since lots of the calculation were much more easier with vector, matrix or tensor, it was really important to transfer a word into a vector. That’s the main function of the

word embedding layers do, which transferring each word unit into a tensor. In my code, I used a set of 32 length matrix to stand for each word.

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

Generally, if you increase the dimension size of the hidden layer, the accuracy will be higher, but it will take longer time, either.



But as you can see, when the hidden layers increased to over 150, the overfitting problems might be caused, the accuracy would decrease when the hidden layers still increased.

III. Replace LSTM with GRU and compare their performance.

```
25000/25000 [==============================] - 1259s - loss: 0.1443 - acc: 0.9476 - val_loss:
 0.3377 - val_acc: 0.8810
```

```
25000/25000 [==============================] - 2290s - loss: 0.1022 - acc: 0.964
4 - val_loss: 0.3248 - val_acc: 0.8876
```

The first figure is the result of the LSTM with epoch equals to five, and the second figure is the result of using GRU with epoch equals five too. As you can see, the result of the GRU and LSTM didn't have large difference, but the GRU had better performance though.