Yuxin Sun
Deep Learning
Homework 2
02/06/2022
The code of the assignment is
https://colab.research.google.com/drive/1am2sqKDwN0lxG-lehjWUWJcgqFOJpfM6?usp=sharing

Step1

a.

Table1

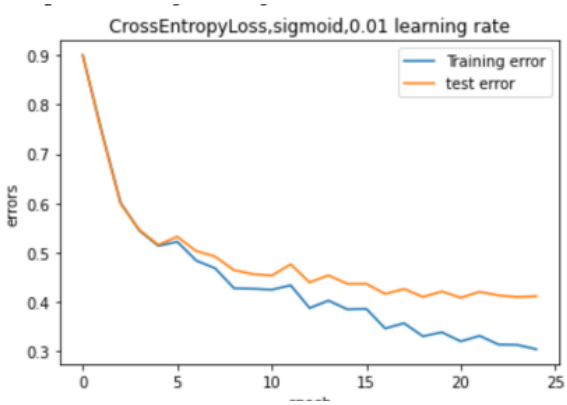


Table2

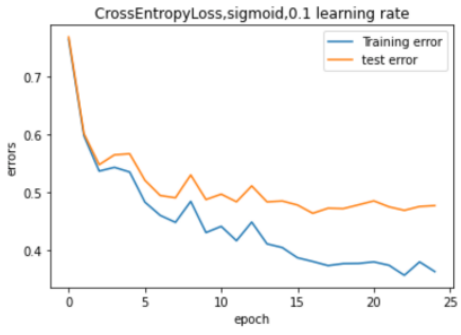


Table3

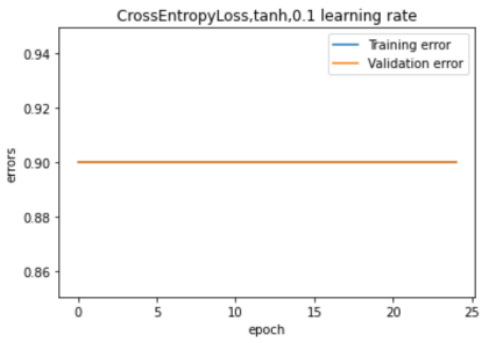


Table4

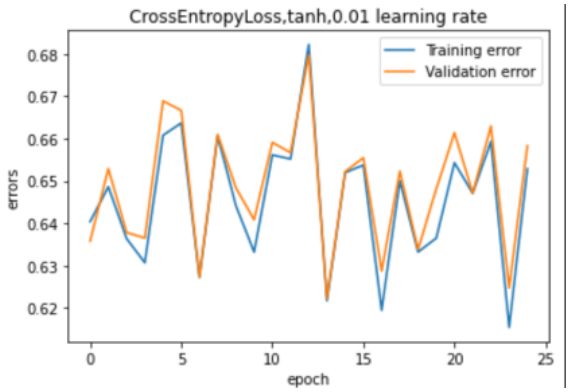


Table5

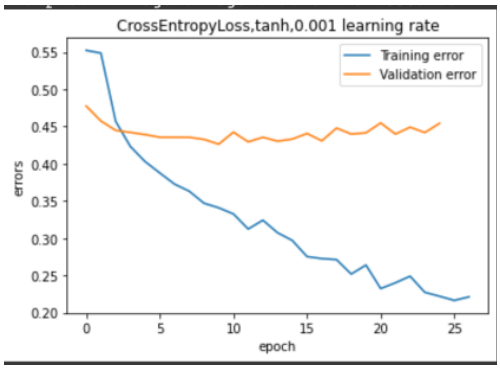


Table6

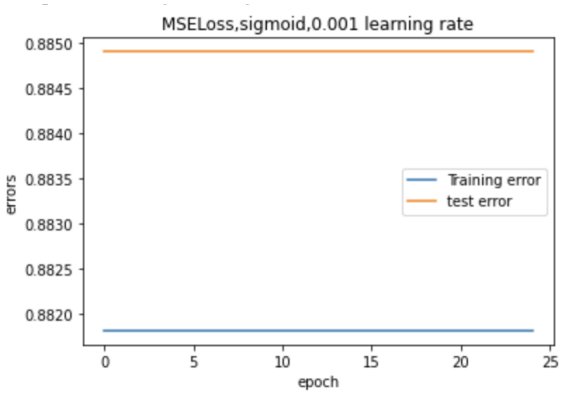


Table7

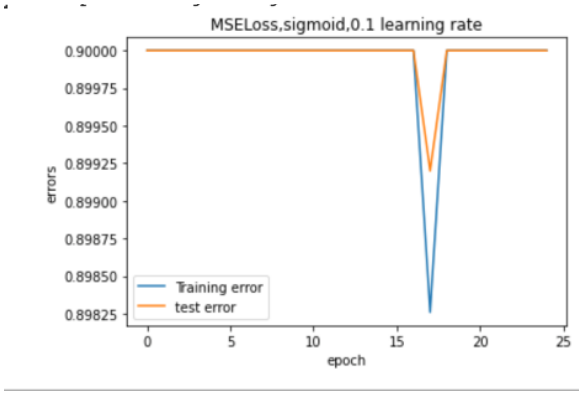


Table8

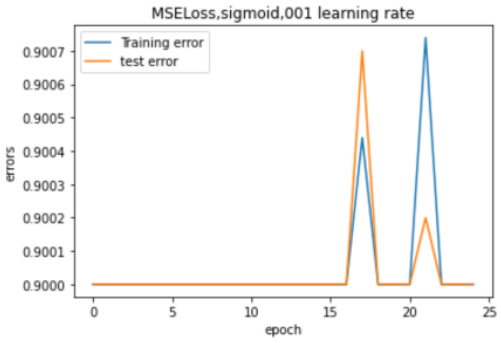


Table9

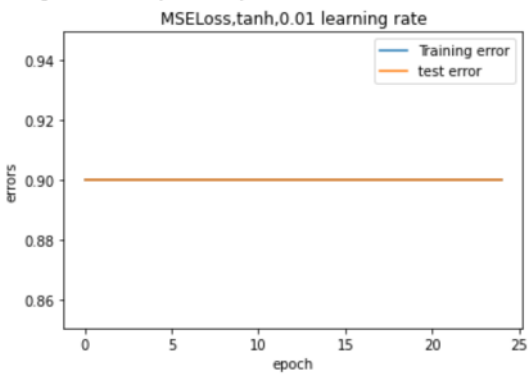


Table10

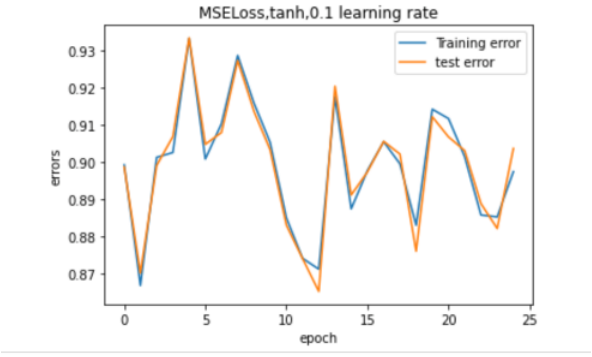


Table11

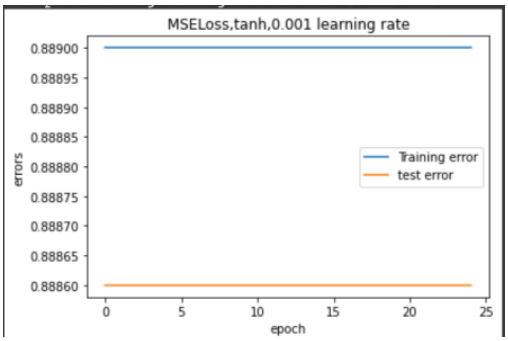
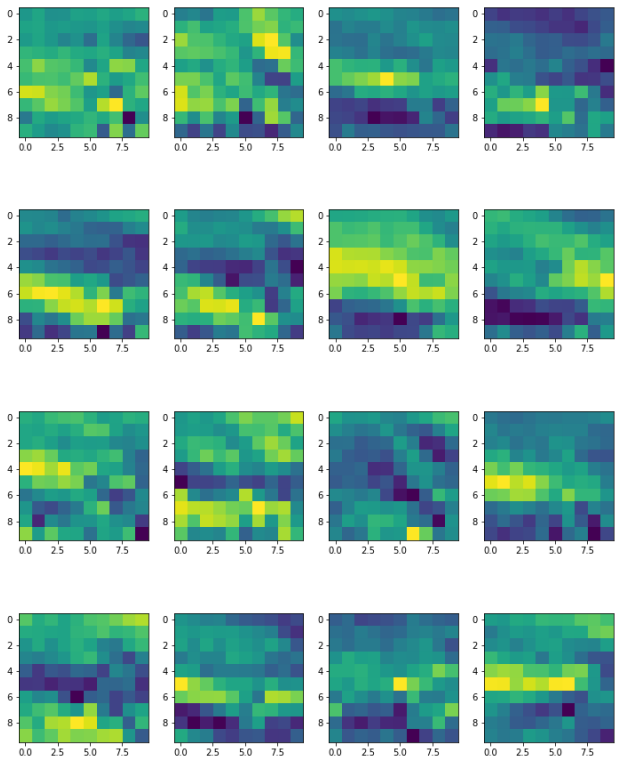


Table12

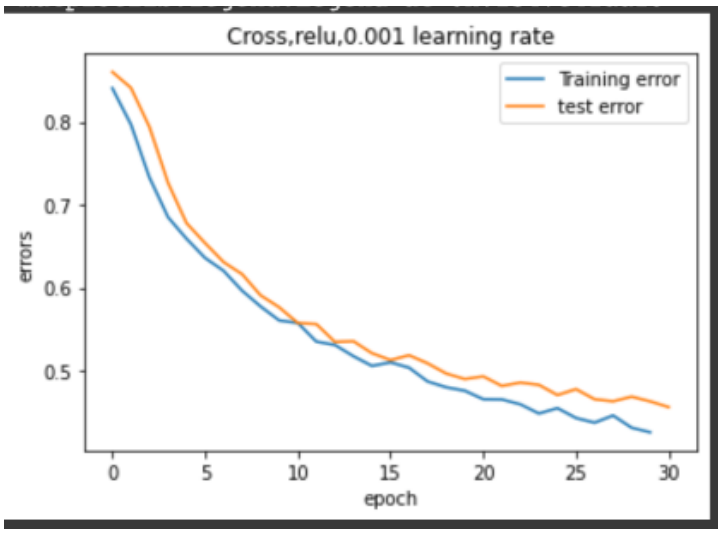
For the step1, I tested learning rate 0.1, 0.01, 0.001; activation function = sigmoid, Tanh; loss function = MSE, cross-entropy, and get the plots of training error vs epoch and test error vs epoch (from table1 to table 16). The best fit is cross-entropy, sigmoid with 0.01 learning rate(table 1), the validation and train error are low and the accuracy is increased to 70% after 25 epochs of training. Interesting thing is that all of MSELoss gets bad accuracy(less than 20%). I think it is because of outliers, the squaring part of the function magnifies the error. And the MSELoss is not suitable for Lenet.

B. The images at the last convolution layer



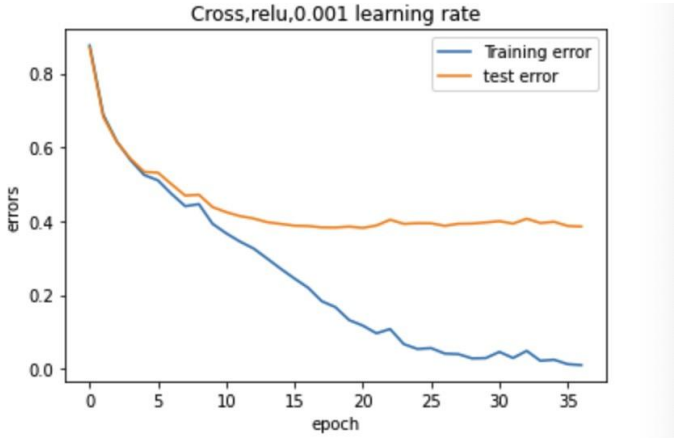
From the images, it shows that the 32x32 images are separated into 16 pictures(10x10), because the images are 32x32x3 -> 16x10x10(after two convolutions and a pool).

Step 2



In step 2, I use cross-entropy, relu and 0.001 learning rate and get a good fit model. Compared with cross-entropy, sigmoid, 0.001 learning rate(table1), the rule is better than sigmoid for the beginning, after 10 epochs, the sigmoid is better. I think it is because relu resulting in a faster training process and convergence. And a 3x3 kennel size shared more frequently than the 5x5 kernel.

Step3



In step3, it will cost almost 10m/epoch. It takes 6 hour to test 35 epochs. It is because the stride is 1 and the last convolutions send a 93312 (128x27x27). It will train the data one by one. It will take a lot of time. Compared to Step3 and Step2, errors are significantly reduced, but the overfitting is more pronounced. In this step, the model gets a very low training error(almost 0!). I think it is because the size of the graph changes very little, so the loss is smaller. However, the test error is higher than the training error, it is overfitting. I think it is because there are too many convolutions layers.