Machine Learning #4

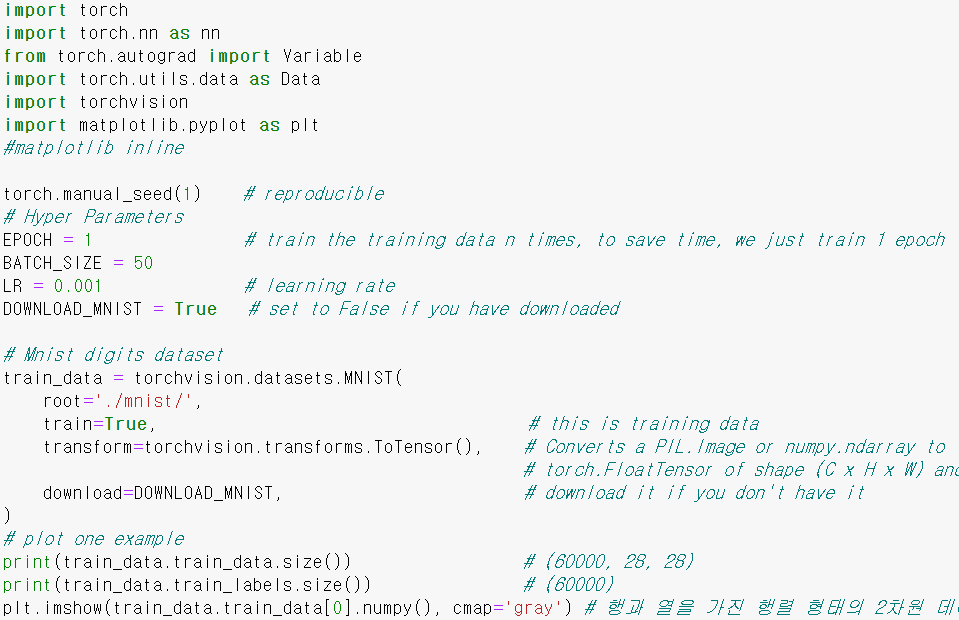
학번 : 2015112227

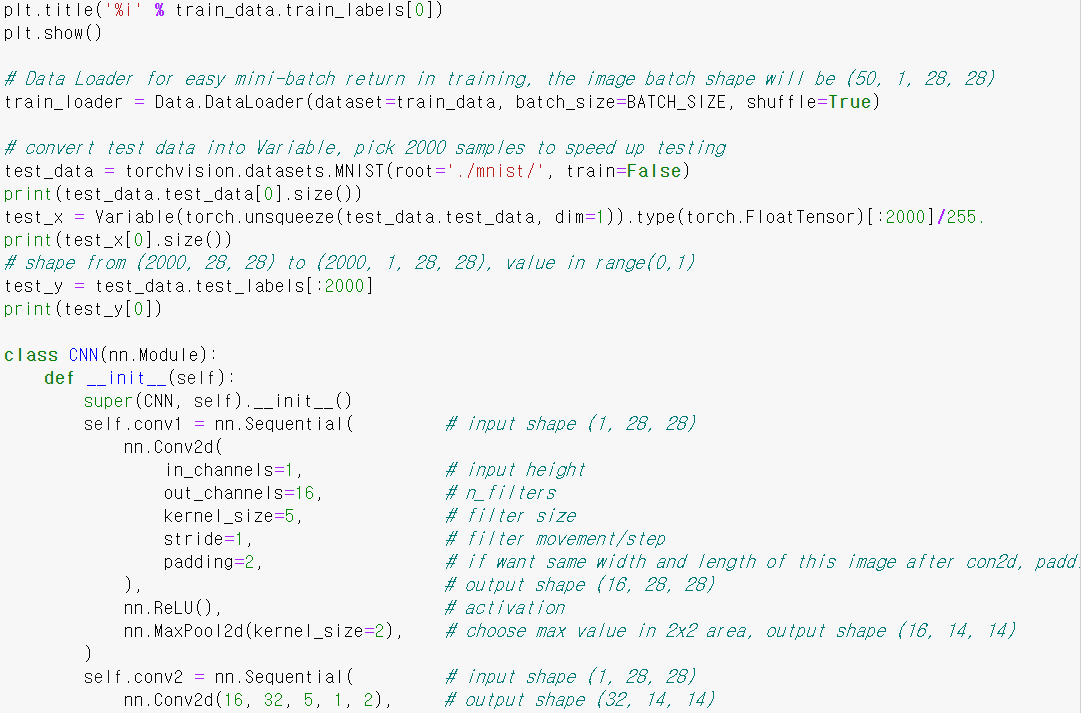
이름 : 조일현

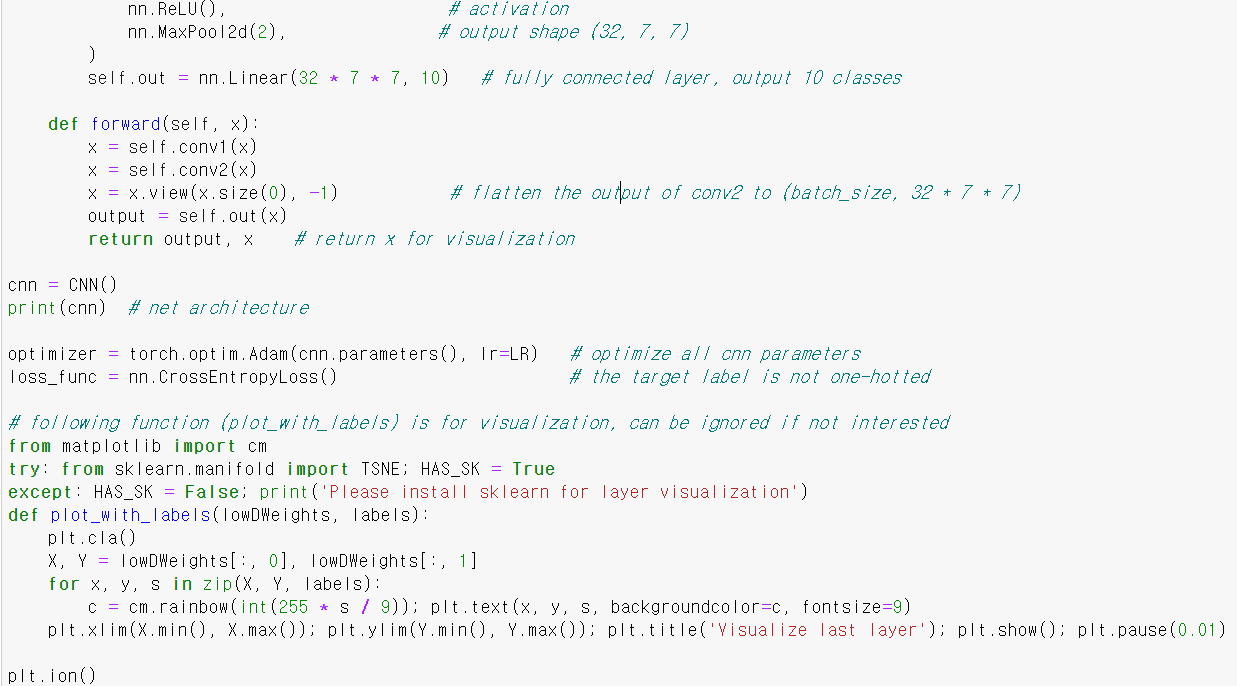
1. CNN & MNIST

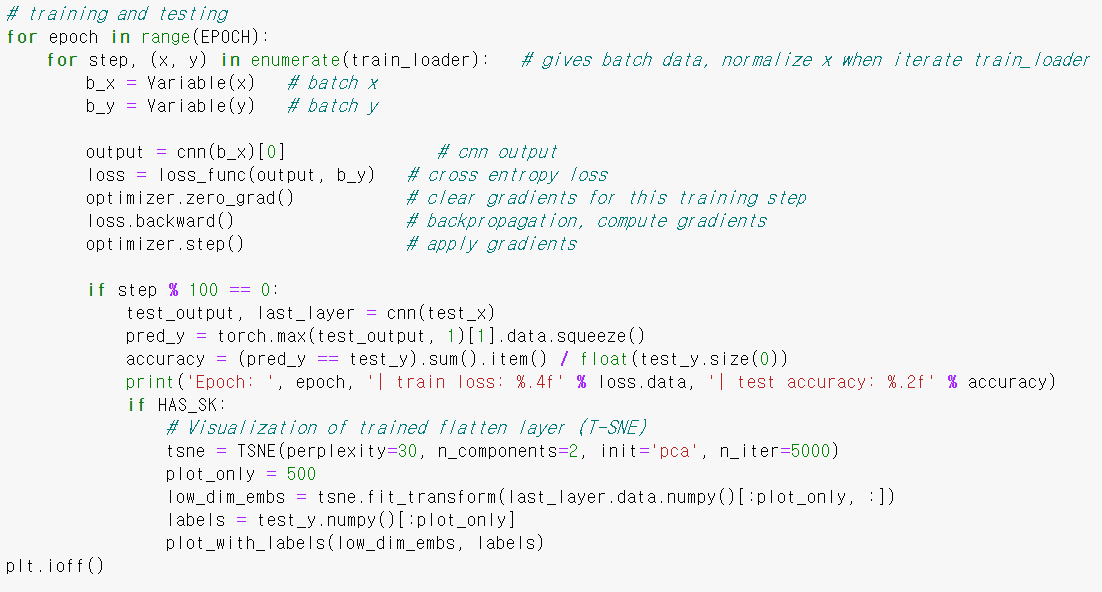
1) refer to the CNN\_Mnist program in e-class

<PROGRAM CODE>

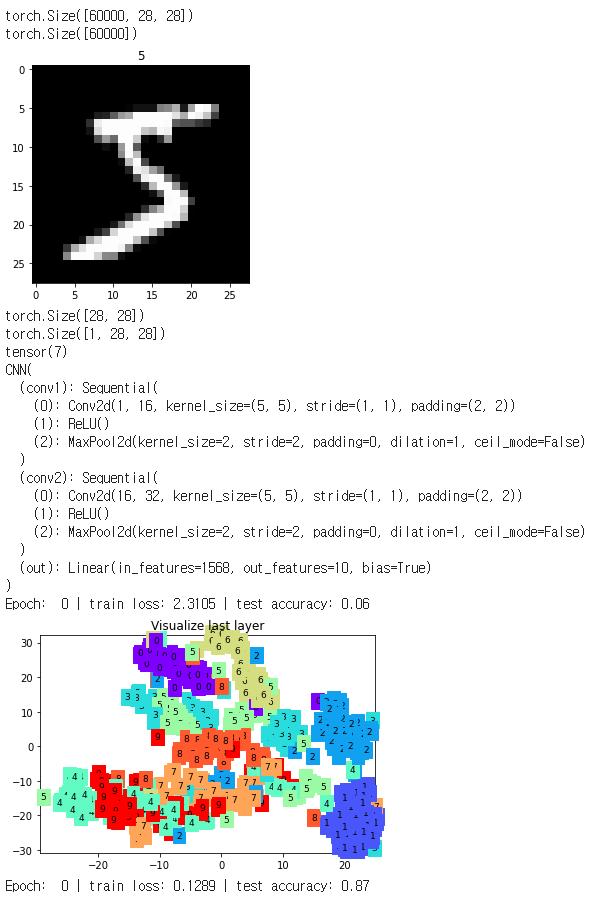


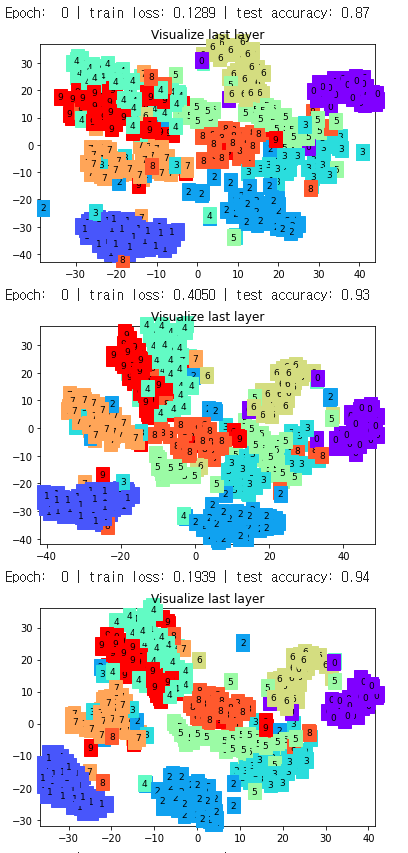


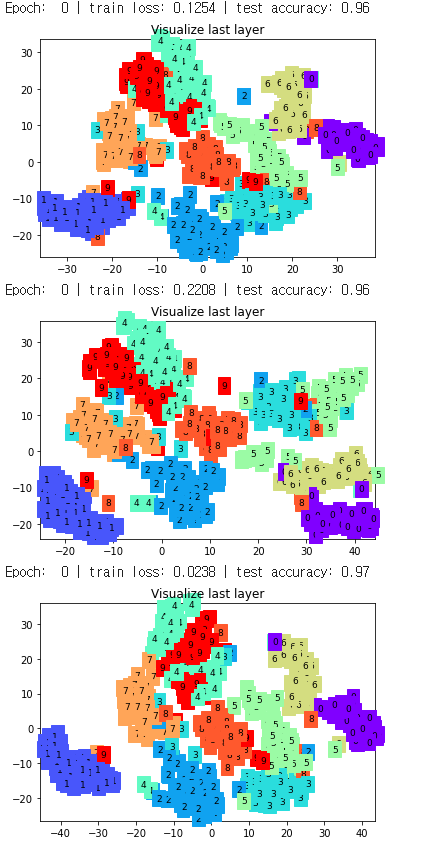


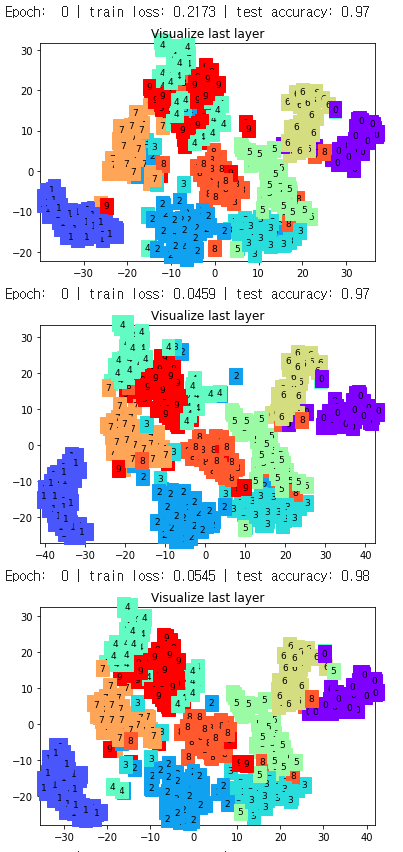


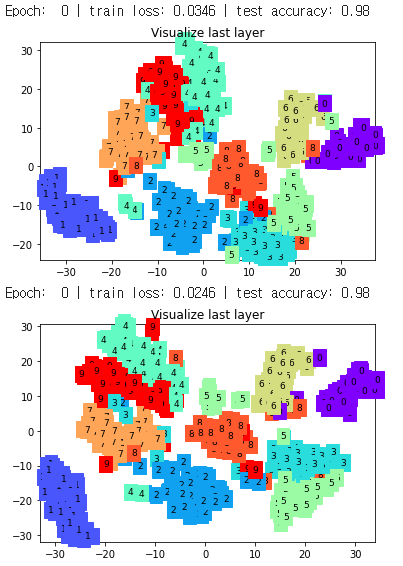
<RESULT>









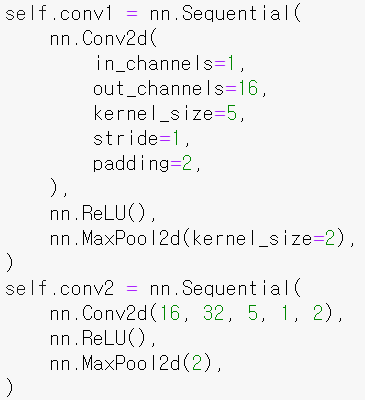


2) change the current kernel size of the program to different size. (Change

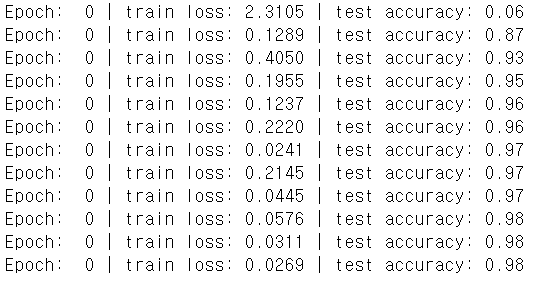
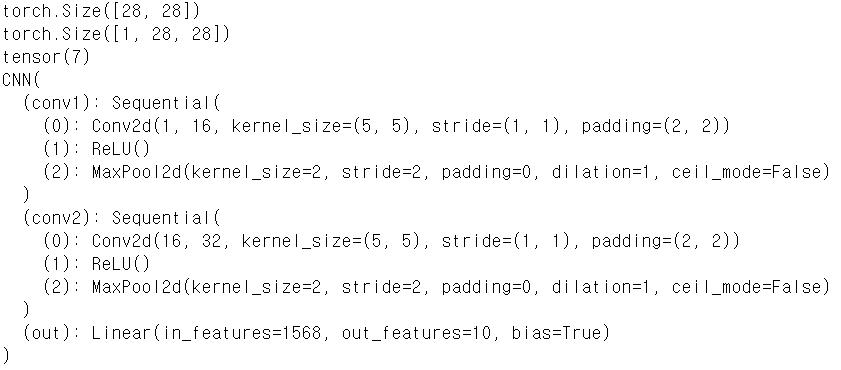
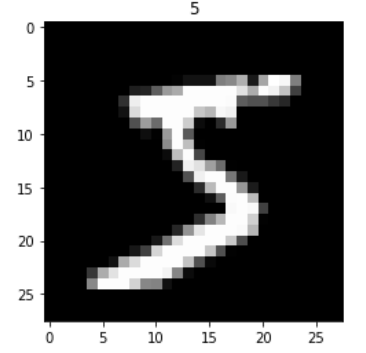
‘kernel\_size’ parameter of ‘Conv2D’ function.) Repeat this three times and compare the results

<PROGRAM CODE>

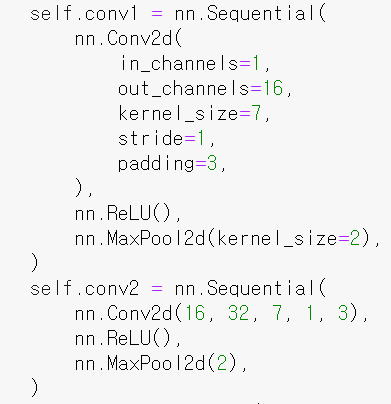
-kernel\_size = 5



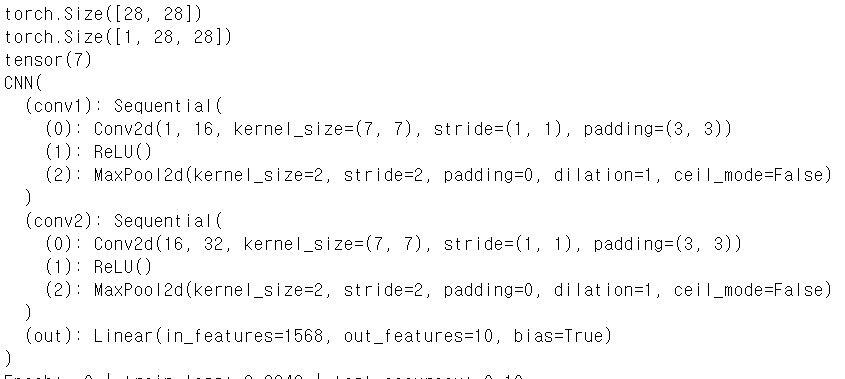
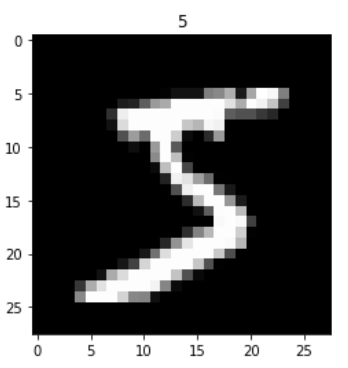
<Result>

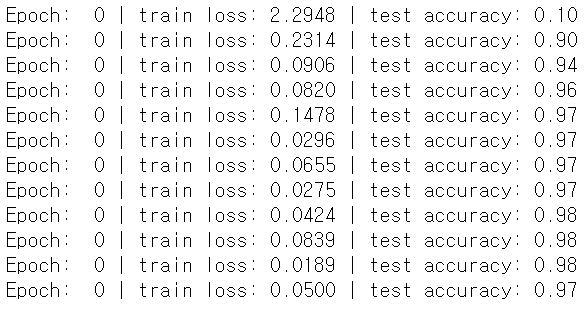


-kernel\_size = 7

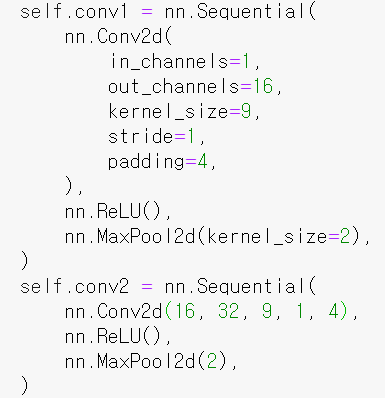


<Result>

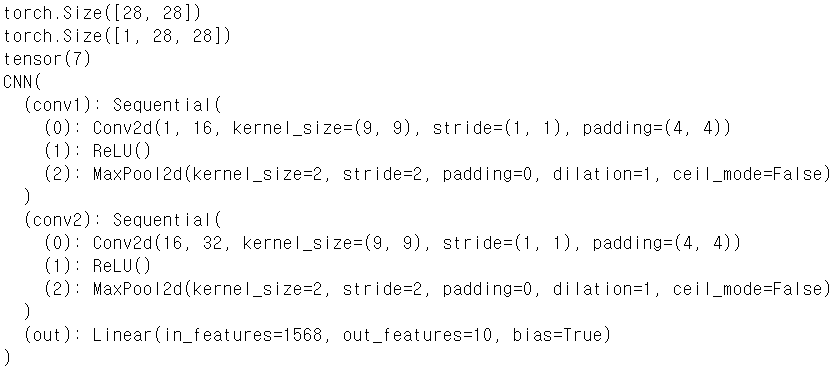
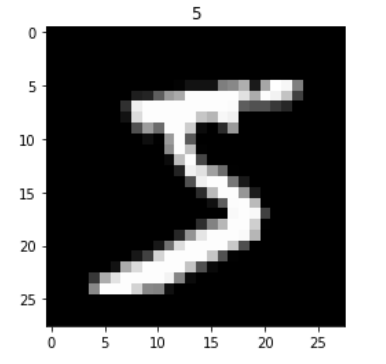


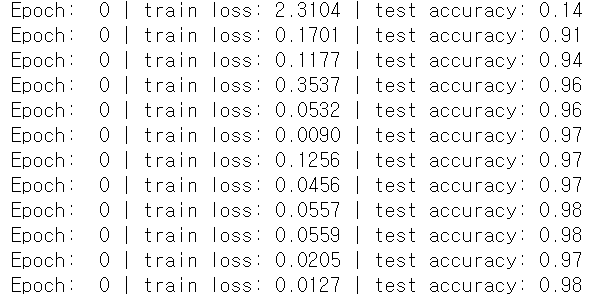


-Kernel\_size = 9

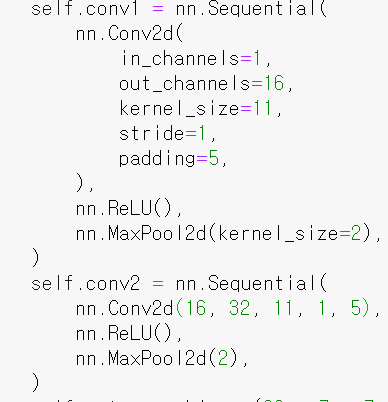


<Result>

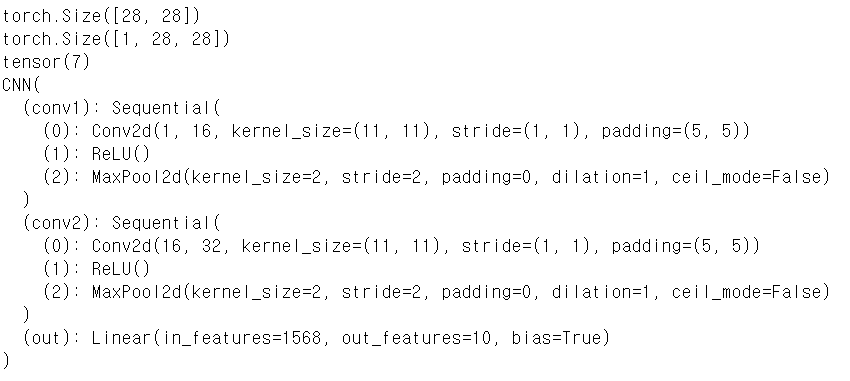
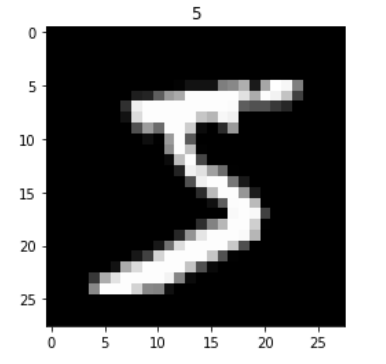


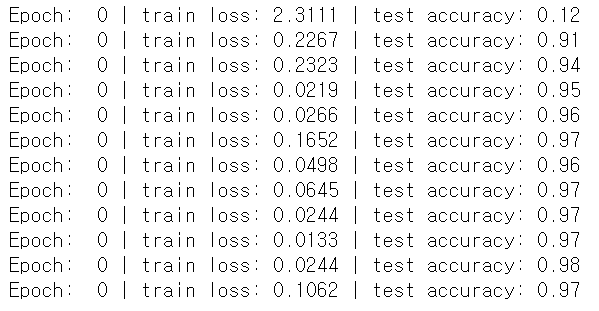


-Kernel\_size = 11



<Result>

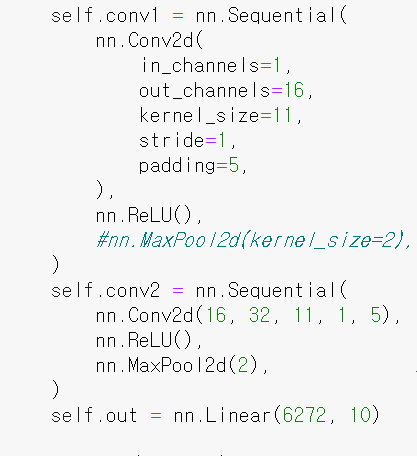




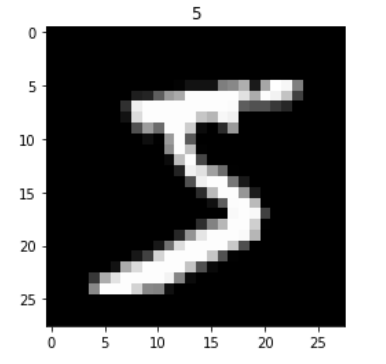
3) remove pooling layer in the program (you can remove ‘MaxPool2D’ function)

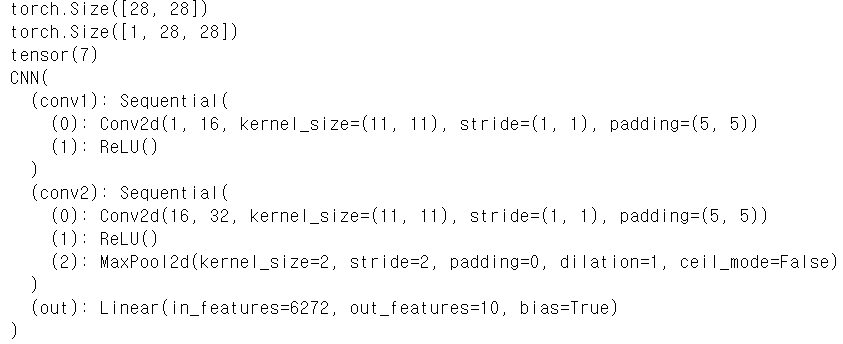
and compare the results

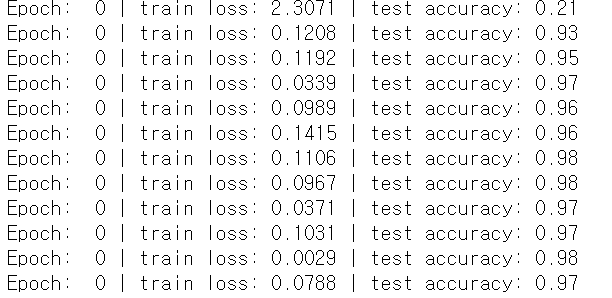
<PROGRAM CODE>



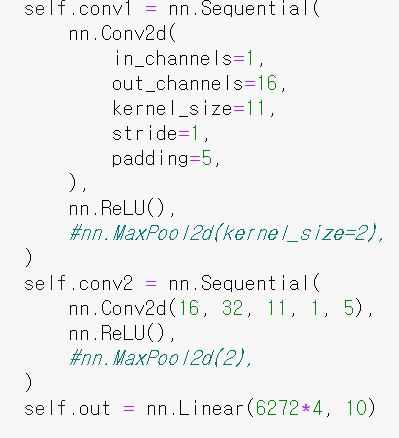
<Result>



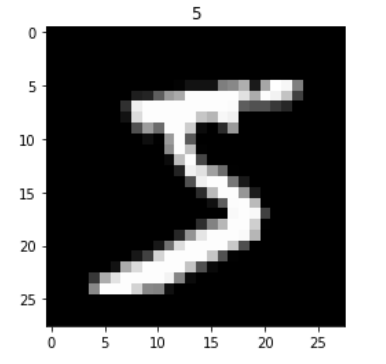


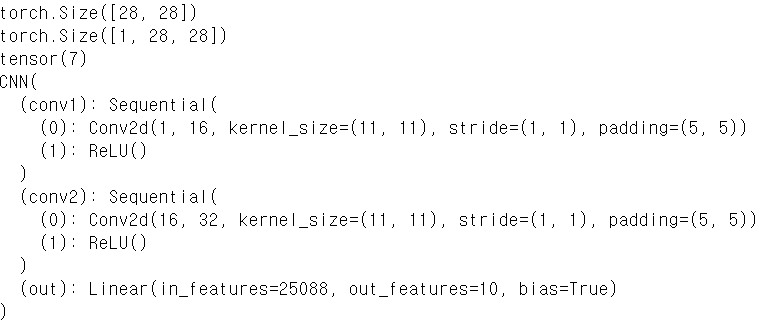


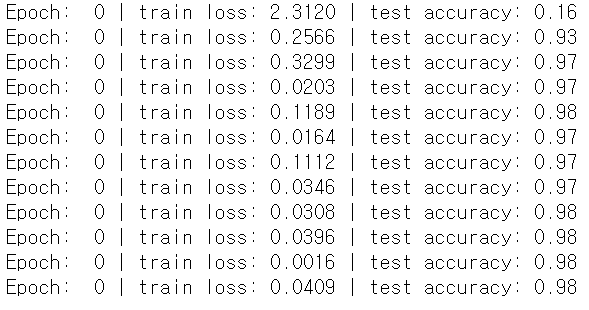
<Program code>



<Result>



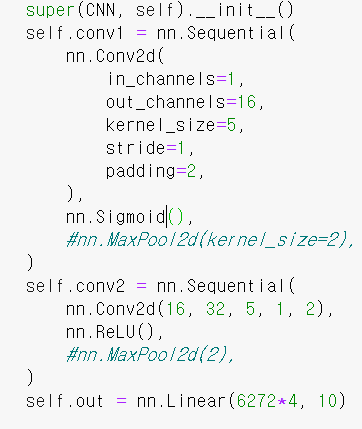




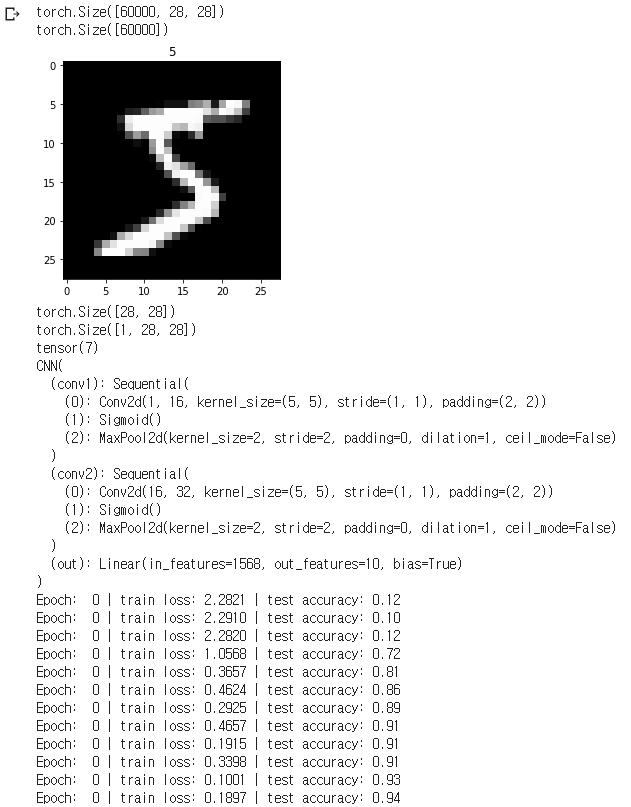
4) change the current activation function to other activation function (e.g. sigmoid, tanh, etc). You can do so by nn.Sigmoid() to nn.ReLU(), nn.Tanh(), etc) Repeat this three times and compare the results.

<PROGRAM CODE>

nn.Sigmoid()

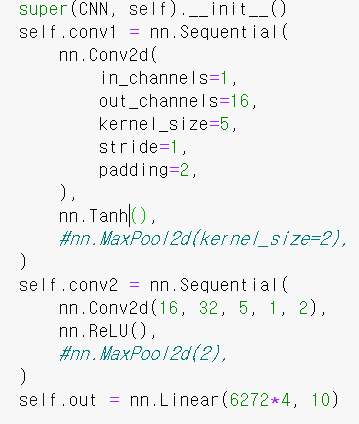


<Result>

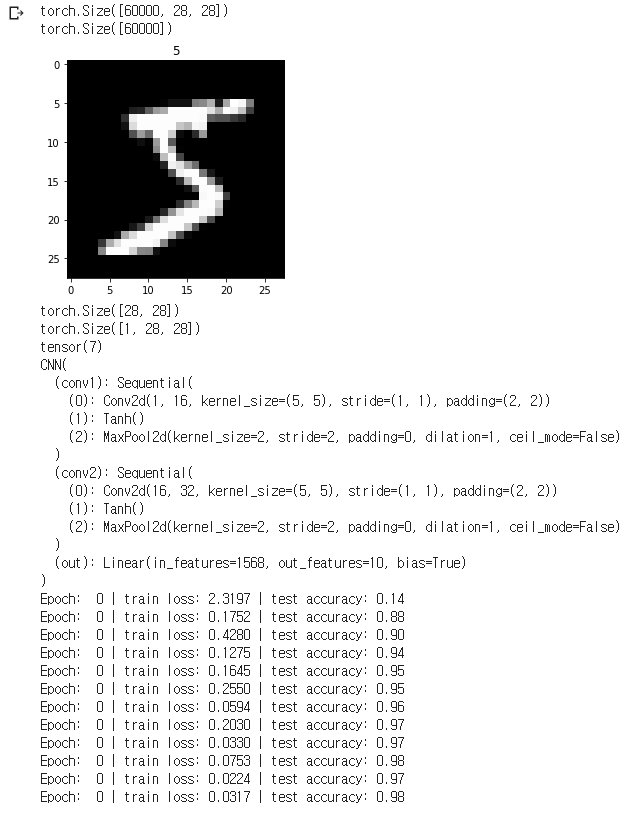


<PROGRAM CODE>

nn.Tanh()

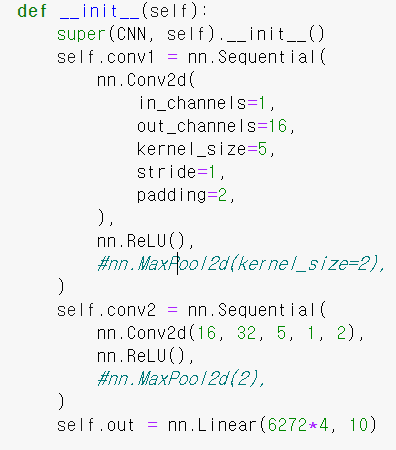


<Result>

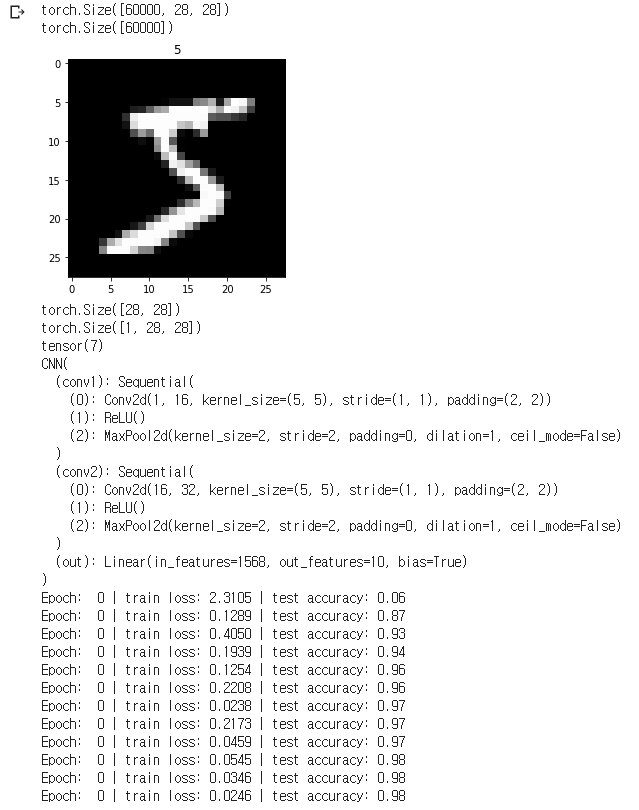


<PROGRAM CODE>

nn.ReLU()



<Result>



5) change the current optimization method to other optimization methods (e.g.

adam, adaGrad, RMSProp, adaDelta, etc). You can use torch.optim.Adam, etc.

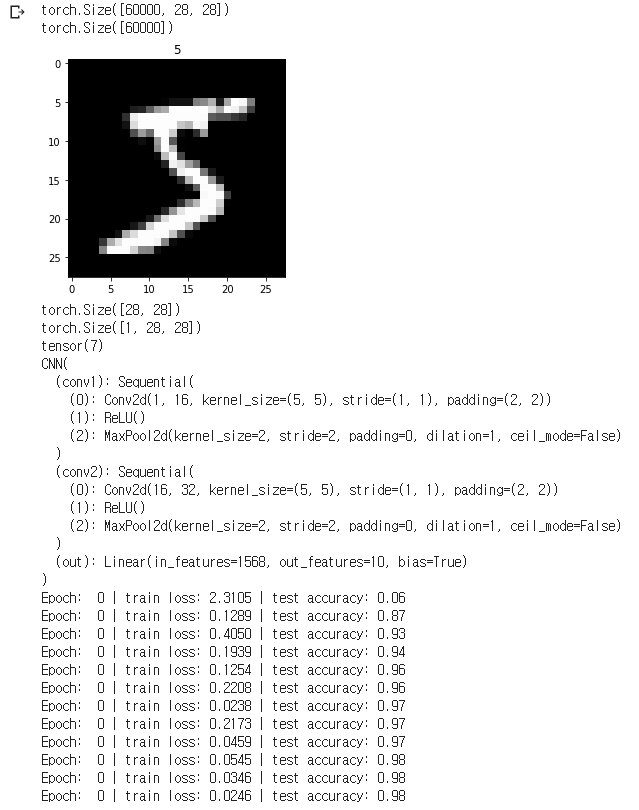
Repeat this three times and compare the results

<PROGRAM CODE>

Adadelta

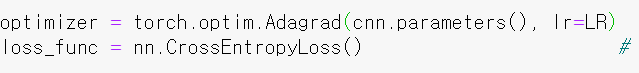
C:\Users\hyun\Desktop\머러과제\10.PNG

<Result>

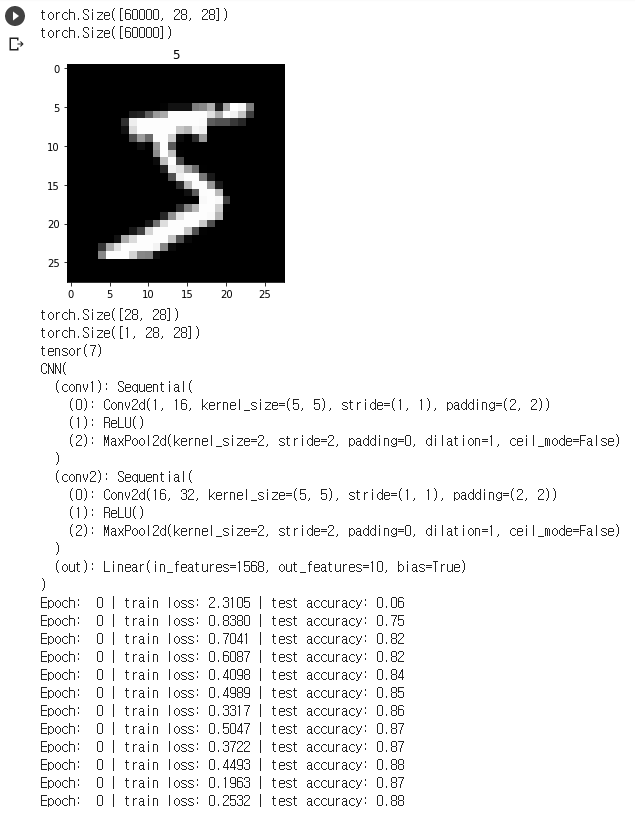


<PROGRAM CODE>

AdaGrad

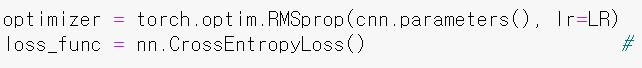


<Result>

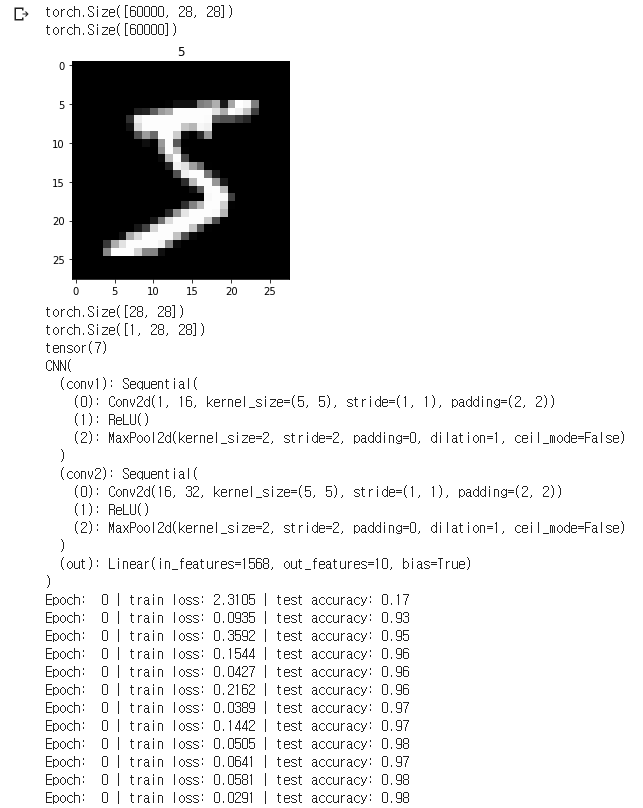


<PROGRAM CODE>

RMSprop



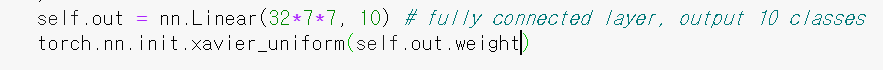
<Result>



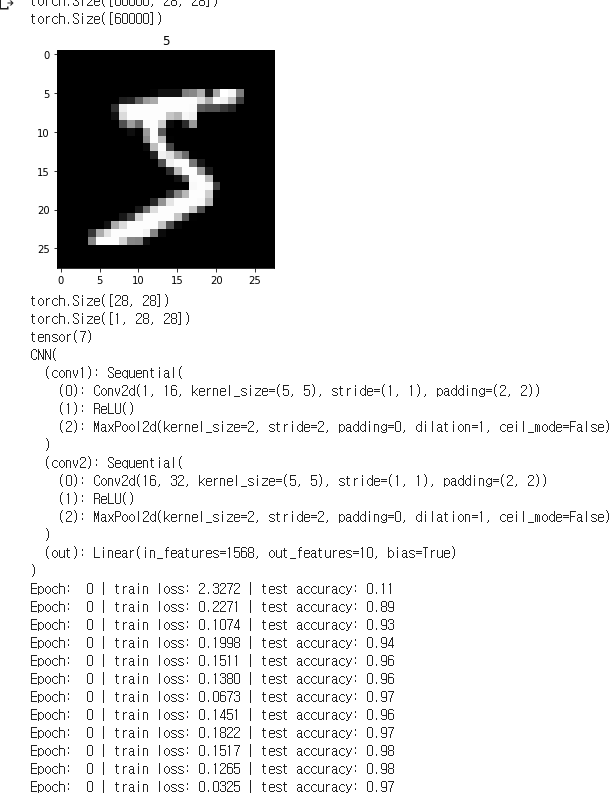
6) now add the Xavier weight initialization method and compare the results. (use

torch.nn.init.xavier\_uniform)

<PROGRAM CODE>



<RESULT>

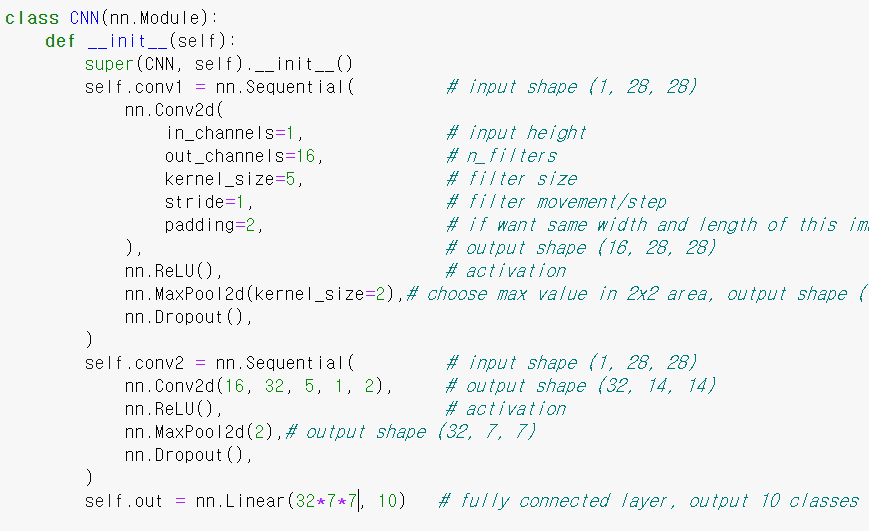


7) choose ONE other parameters of CNN program (e.g. number of hidden nodes,

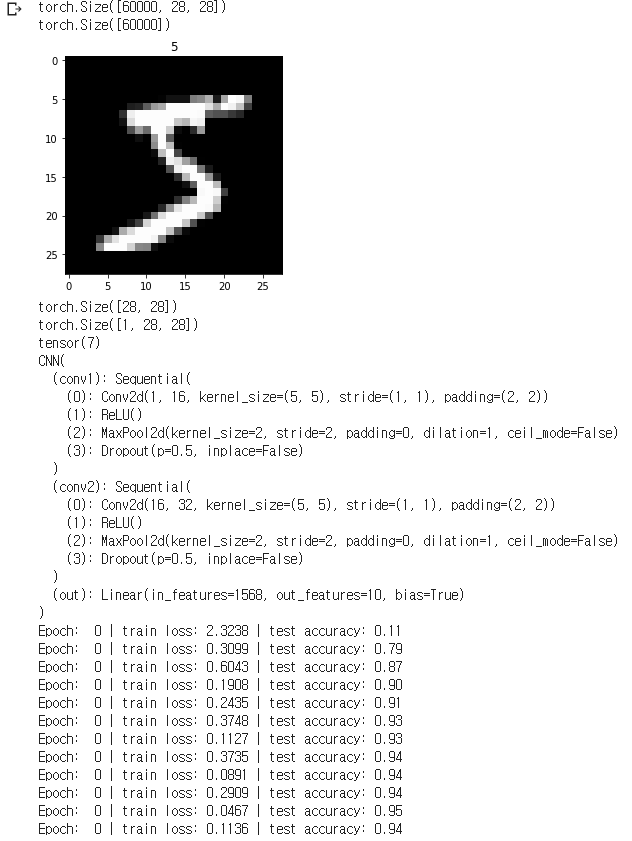
dropout, epochs, batch normalization, etc). Change the value of this parameter and

compare the results.

<PROGRAM CODE>



<RESULT>



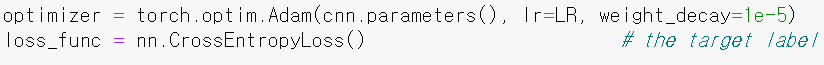
8) (\*optional\*) choose Adam optimization method and use L2 (ridge) regularization

method this time. You can do so by setting ‘weight\_decay’ value in optimization

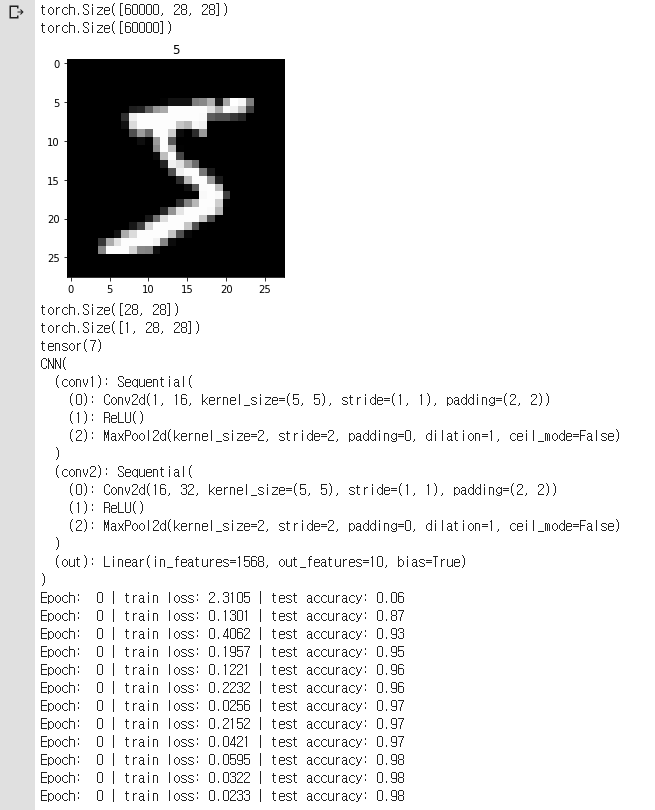
method (torch.optim.Adam) to a tiny number (e.g. 1e-5). Compare the results of

using regularization.

<PROGRAM CODE>



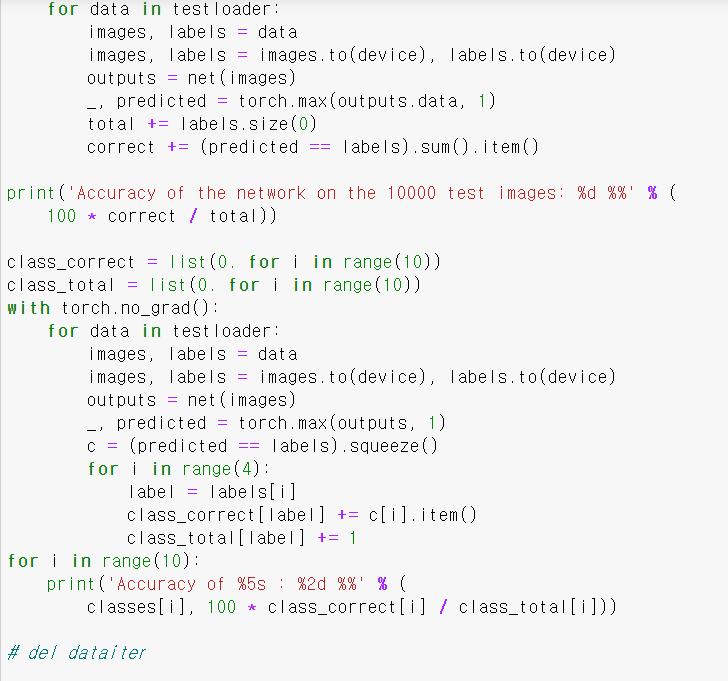
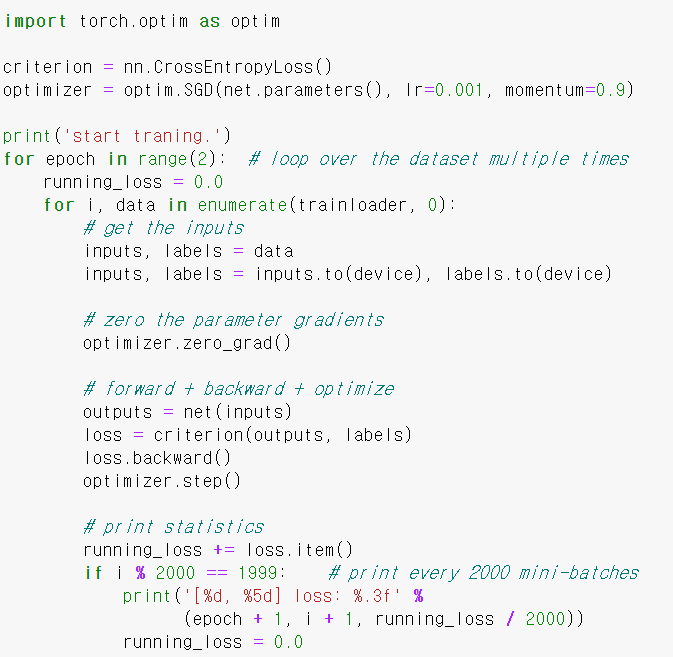
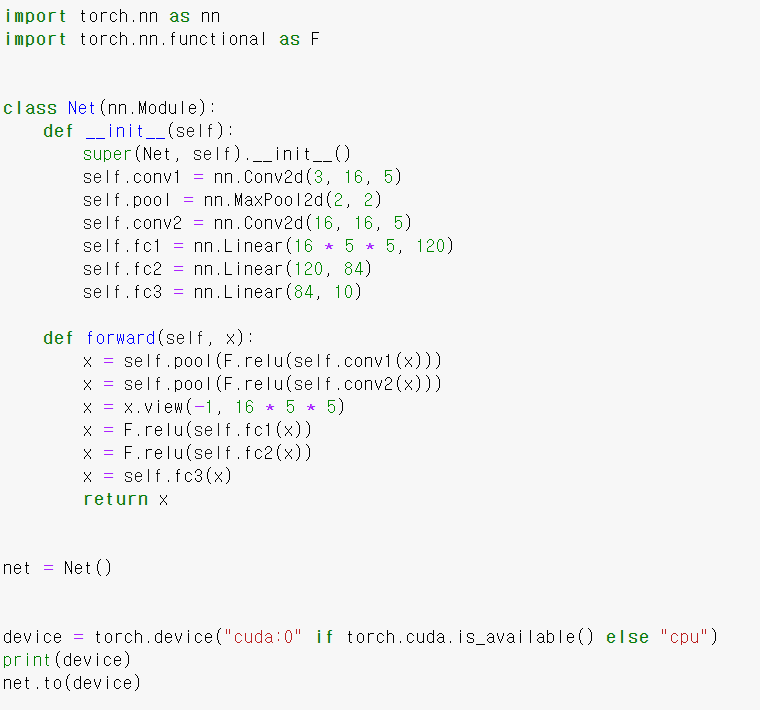
<RESULT>



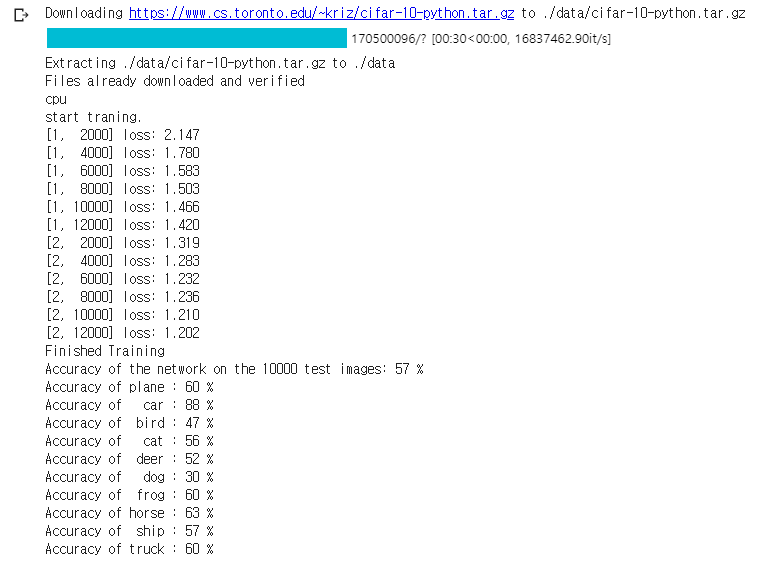
2. CNN & CIFAR-10

1) refer to the CNN\_Cifar-10 program in e-class

<PROGRAM CODE>



<RESULT>

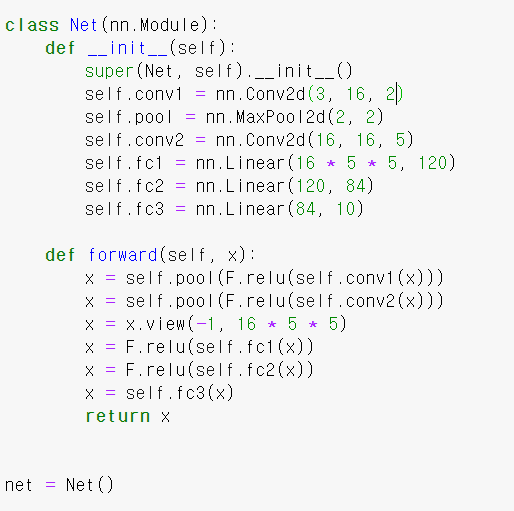


2) change the current kernel size of the program to different size. (Change

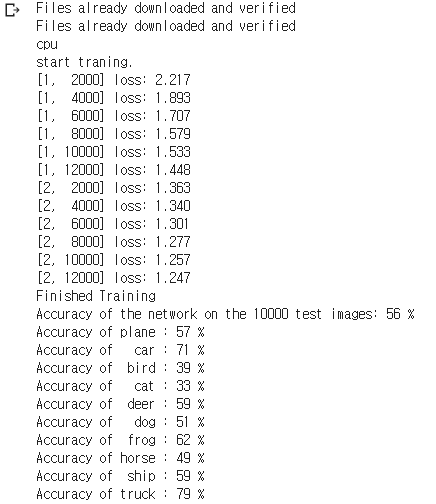
‘kernel\_size’ parameter of ‘Conv2D’ function.) Repeat this three times and compare the results.

kernel\_size=2

<PROGRAM CODE>

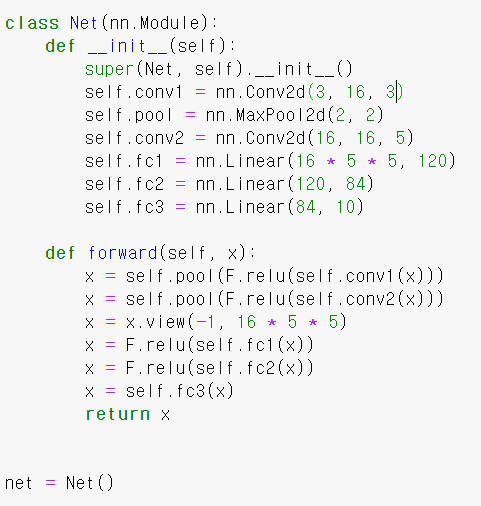


<RESULT>

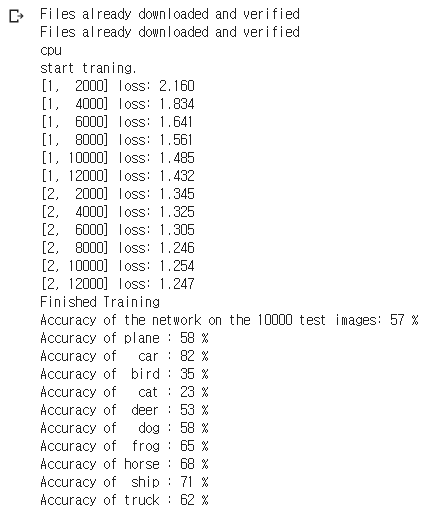


kernel\_size=3

<PROGRAM CODE>

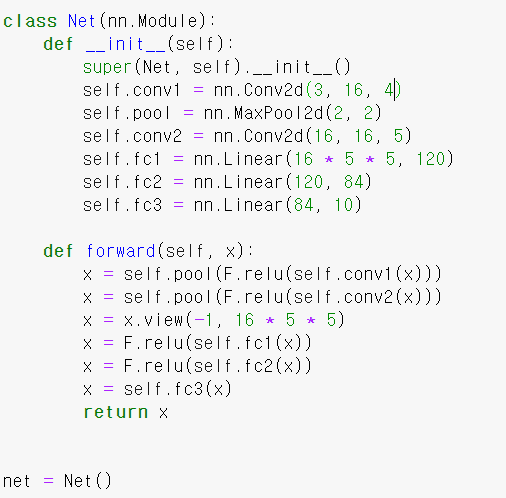


<RESULT>

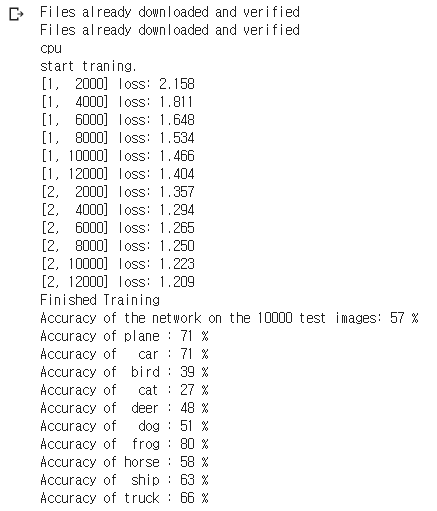


<PROGRAM CODE>

kernel\_size=4



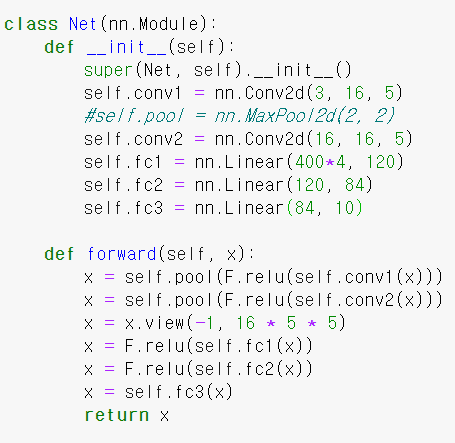
<RESULT>



3) remove pooling layer in the program (you can remove ‘MaxPool2D’ function)

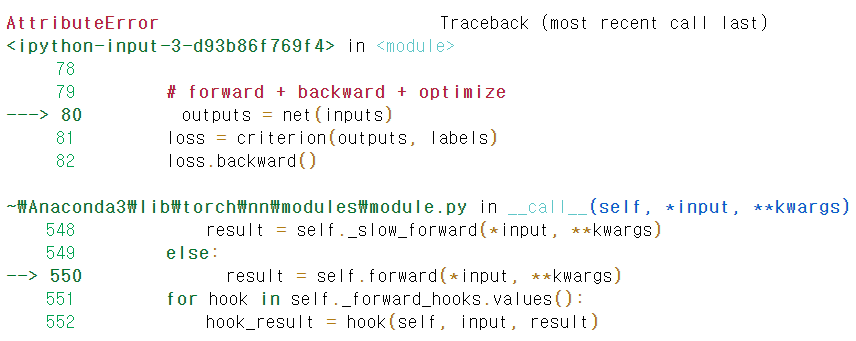
and compare the results

<PROGRAM CODE>



<RESULT>

(설정 값을 바꿨지만 오류가 떴다..)



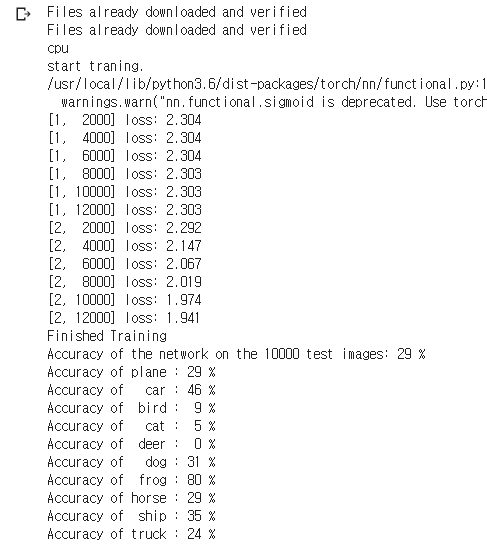
4) change the current activation function to other activation function (e.g. sigmoid, tanh, etc). You can do so by nn.Sigmoid() to nn.ReLU(), nn.Tanh(), etc) Repeat this three times and compare the results.

nn.Sigmoid()

<PROGRAM CODE>

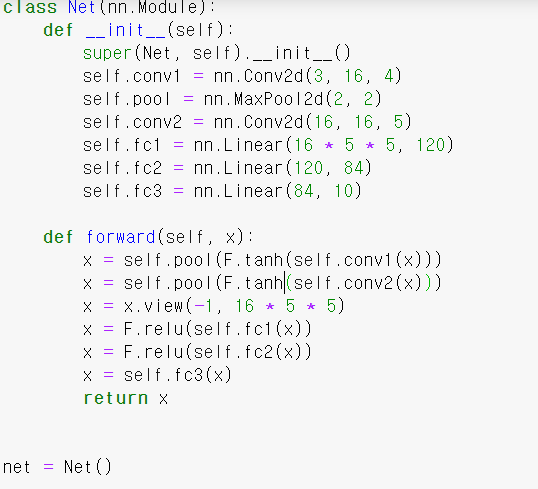


<RESULT>

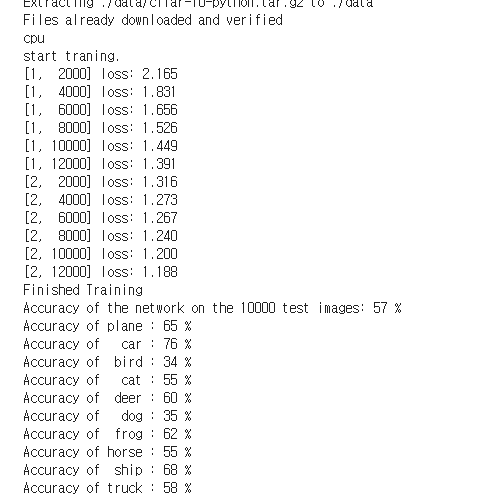


nn.Tanh()

<PROGRAM CODE>

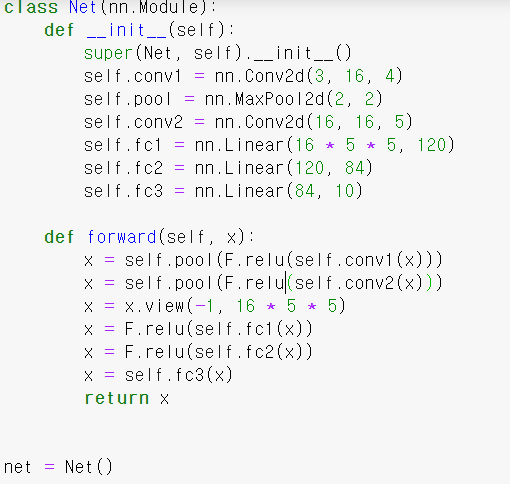


<RESULT>

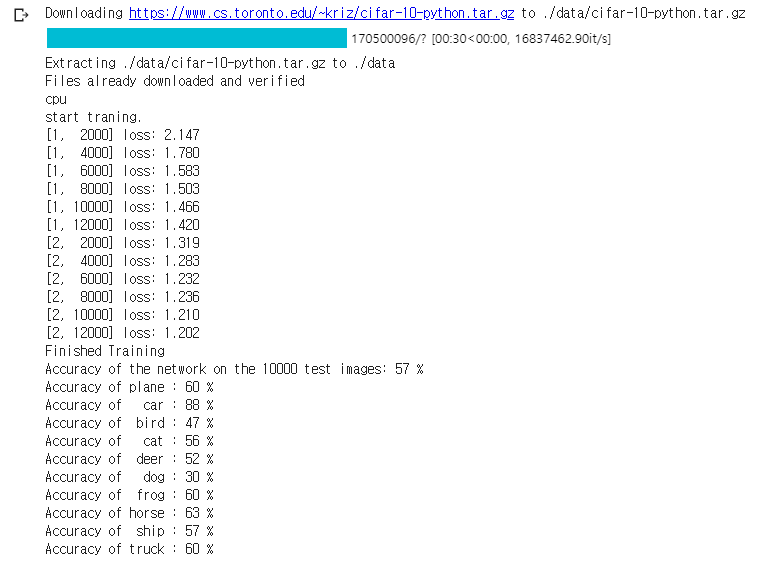


nn.relu()

<PROGRAM CODE>



<RESULT>



5) change the current optimization method to other optimization methods (e.g.

adam, adaGrad, RMSProp, adaDelta, etc). You can use torch.optim.Adam, etc.

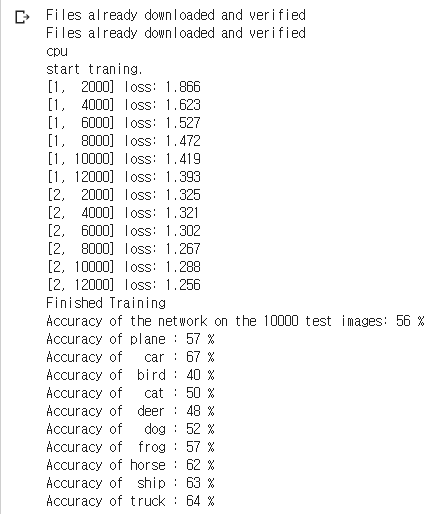
Repeat this three times and compare the results

Adam

<PROGRAM CODE>



<RESULT>

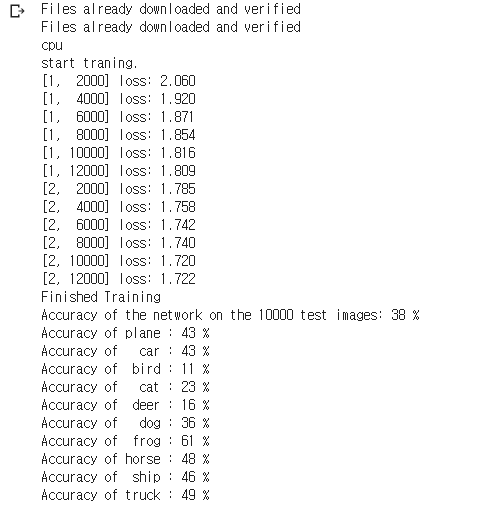


AdaGrad

<PROGRAM CODE>



<RESULT>

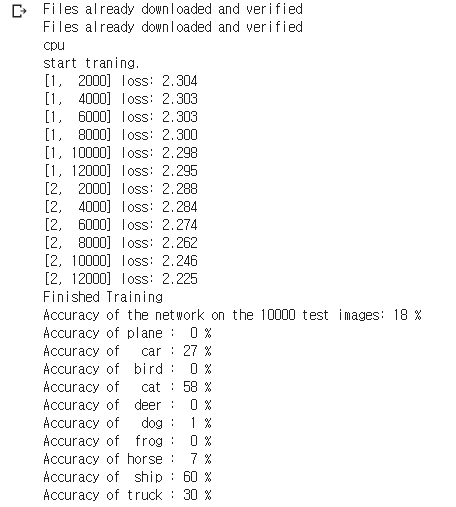


Adadelta

<PROGRAM CODE>



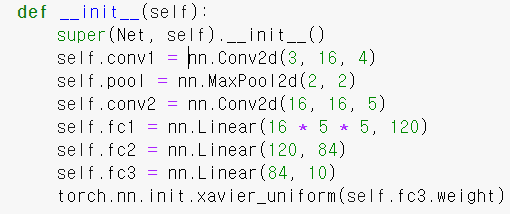
<RESULT>



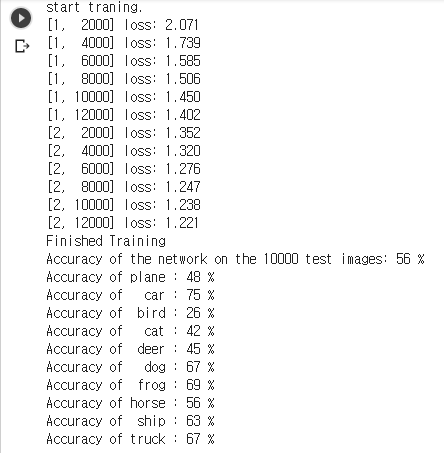
6) now add the Xavier weight initialization method and compare the results. (use

torch.nn.init.xavier\_uniform)

<PROGRAM CODE>



<RESULT>

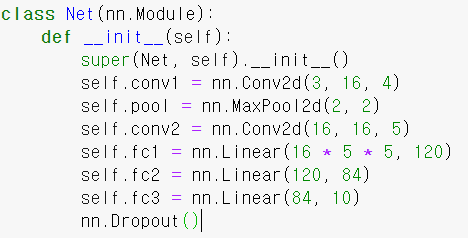


7) choose ONE other parameters of CNN program (e.g. number of hidden nodes,

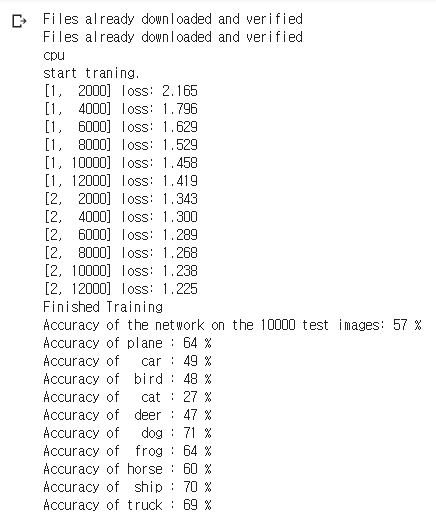
dropout, epochs, batch normalization, etc). Change the value of this parameter and

compare the results.

<PROGRAM CODE>



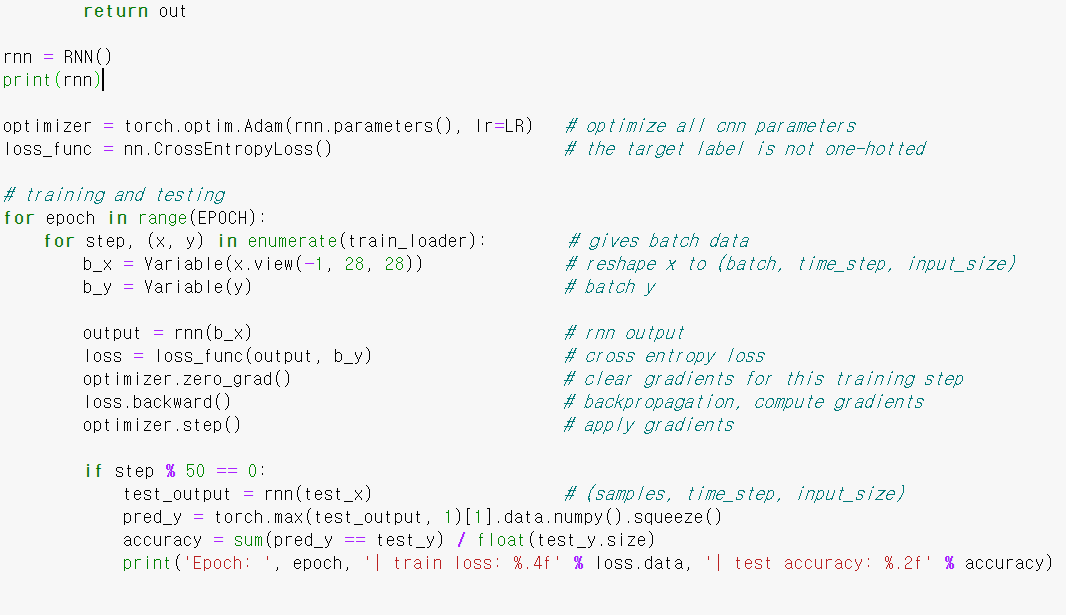
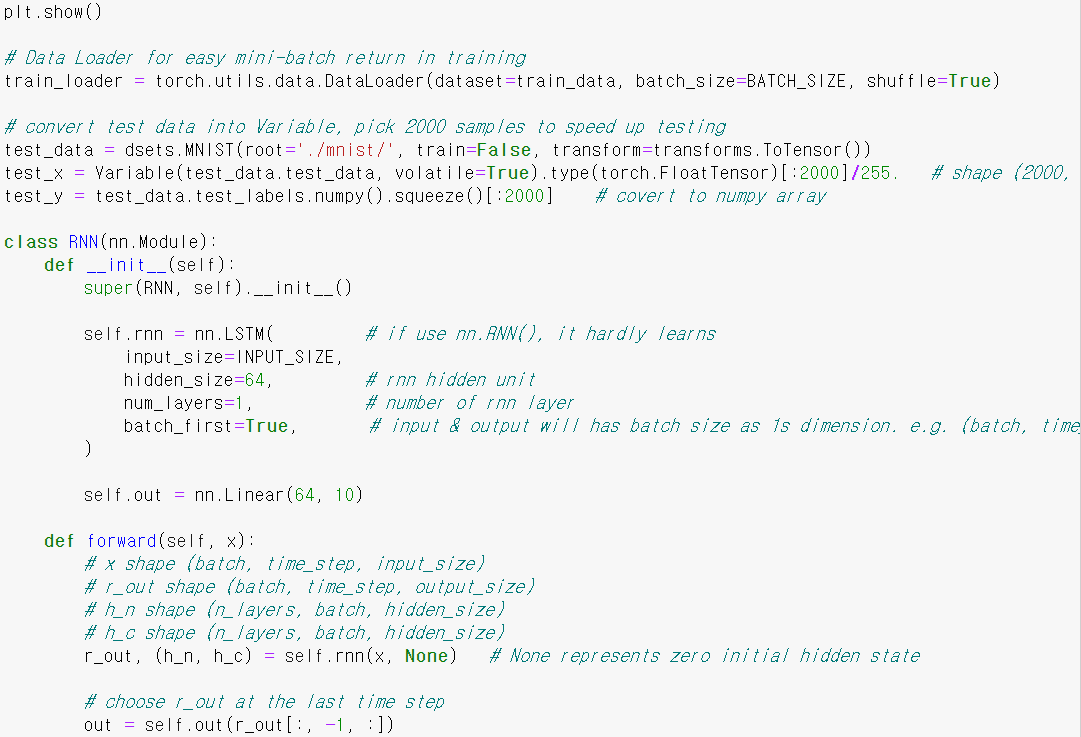
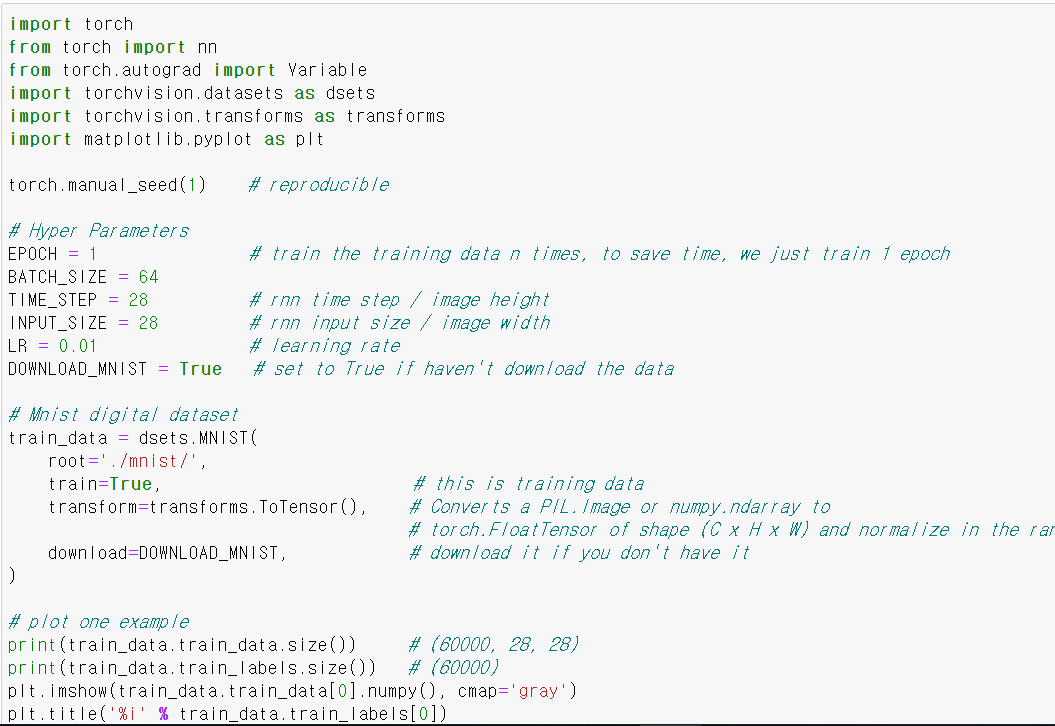
<RESULT>



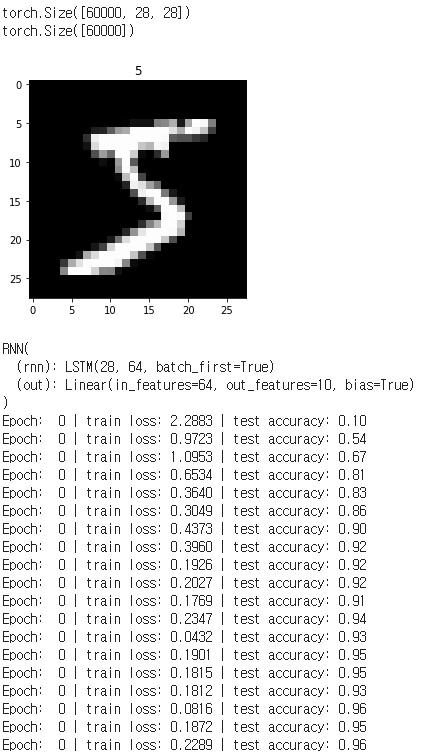
3. RNN & Mnist

1) refer to the RNN\_Mnist program in e-class

<PROGRAM CODE>



<RESULT>

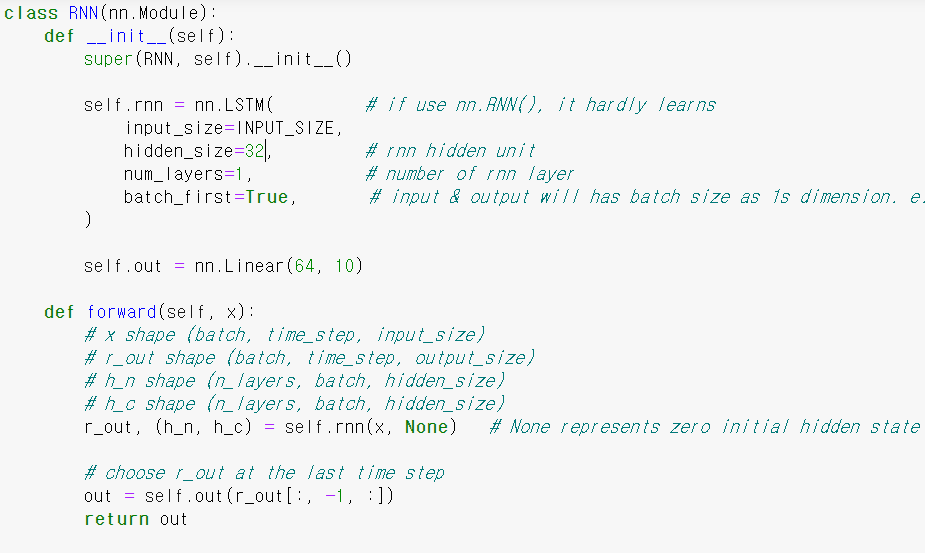


2) change the number of hidden nodes in the program three times and compare

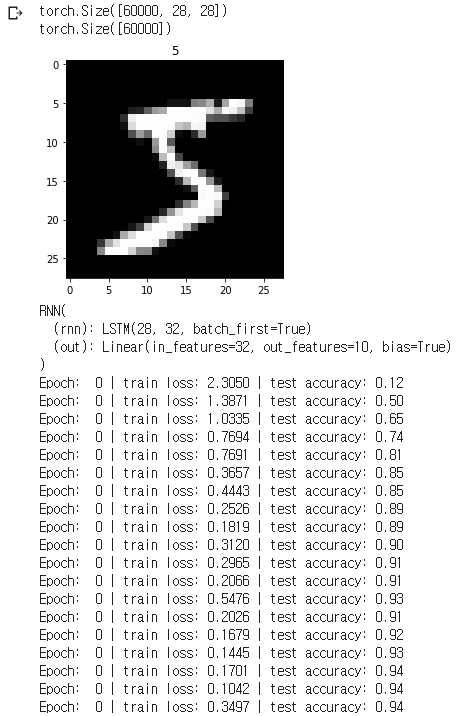
the results.

hidden\_size=32

<PROGRAM CODE>

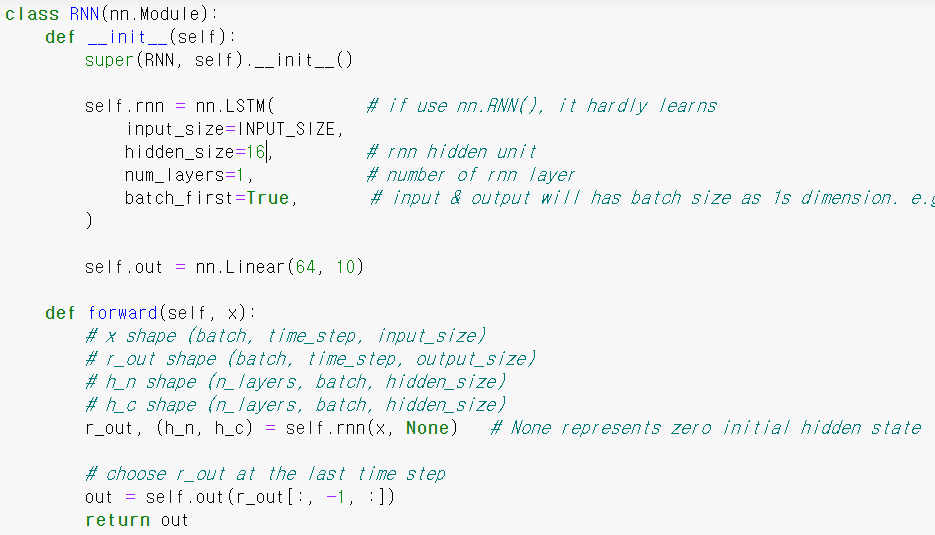


<RESULT>

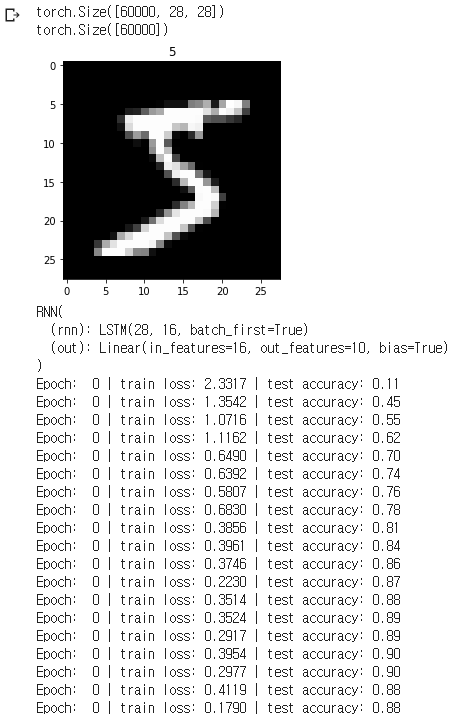


hidden\_size=16

<PROGRAM CODE>

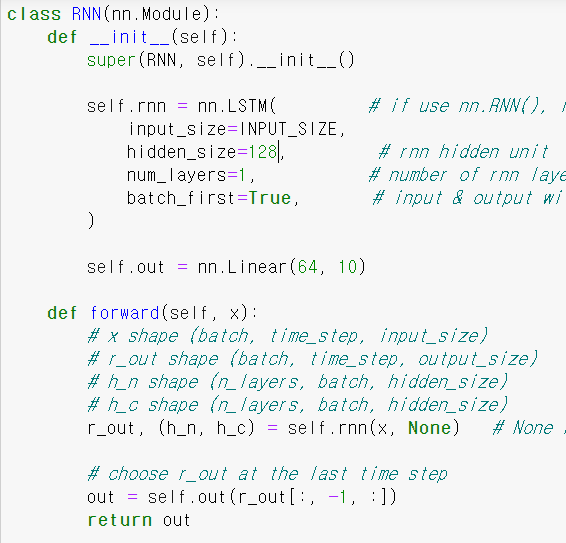


<RESULT>

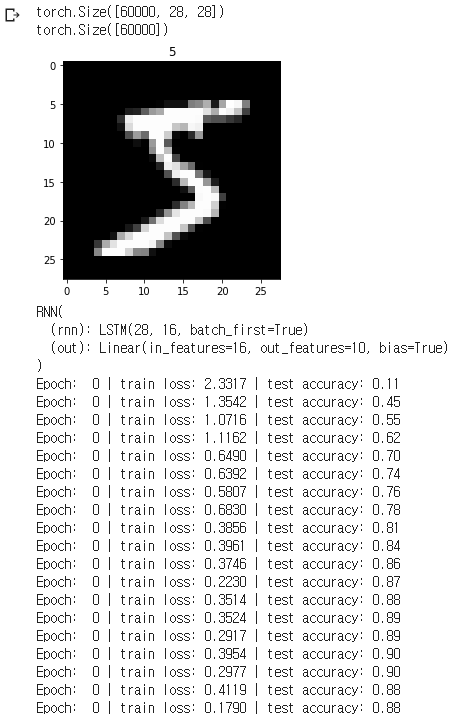


hidden\_size=128

<PROGRAM CODE>



<RESULT>



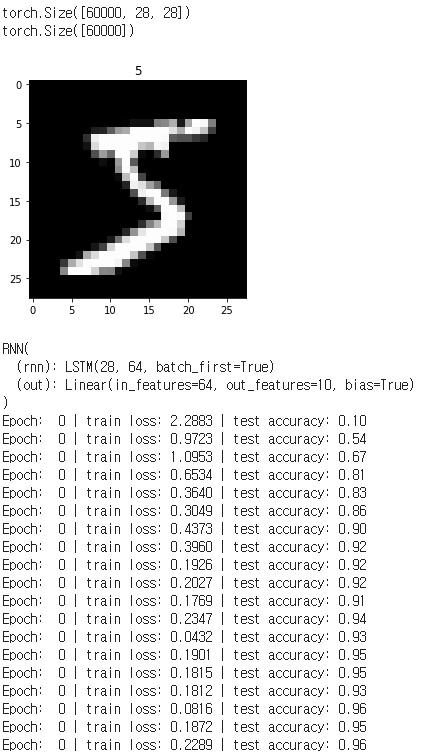
3) change the current optimization method to other optimization methods.

Adam

<PROGRAM CODE>



<RESULT>

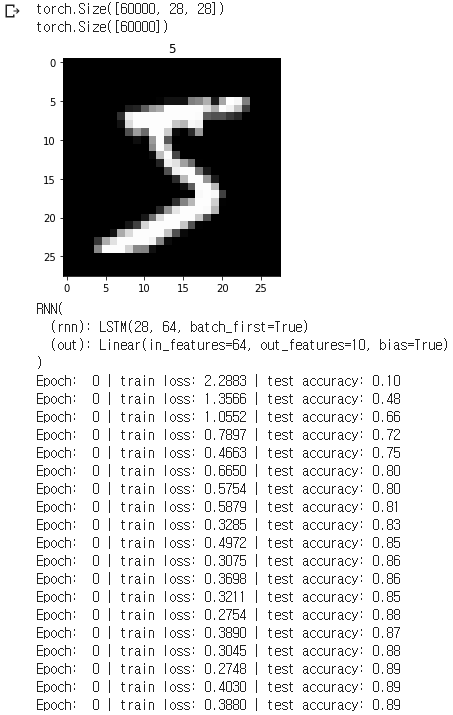


AdaGrad

<PROGRAM CODE>

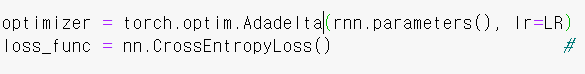


<RESULT>

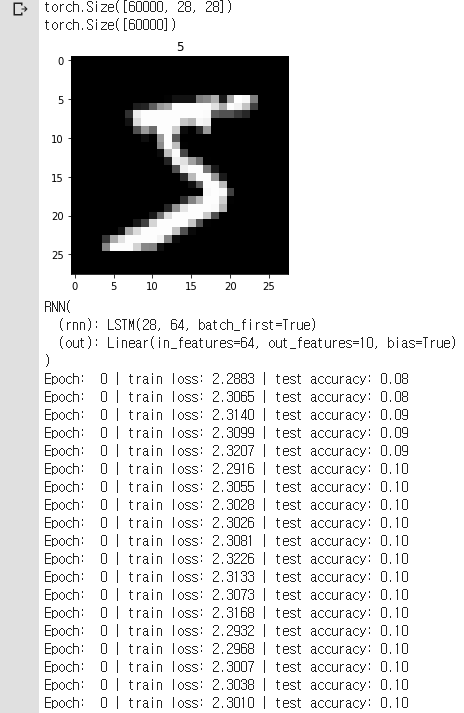


AdaDelta

<PROGRAM CODE>



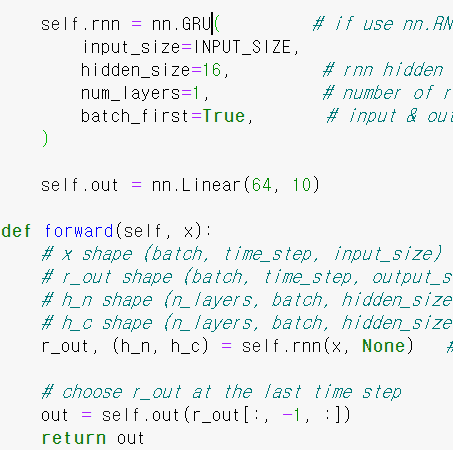
<RESULT>



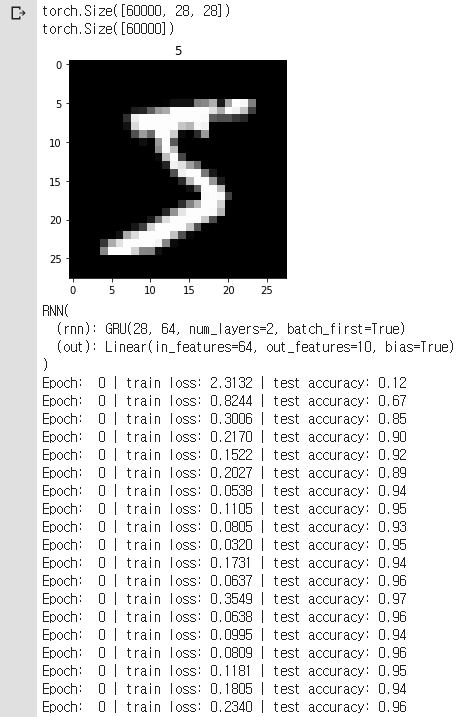
4) change LSTM to GRU (or vice versa). Compare the results.

GRU

<PROGRAM CODE>



<RESULT>

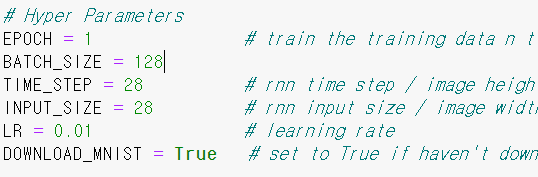


5) choose ONE other parameters of RNN program (e.g. batch\_size, epochs, etc).

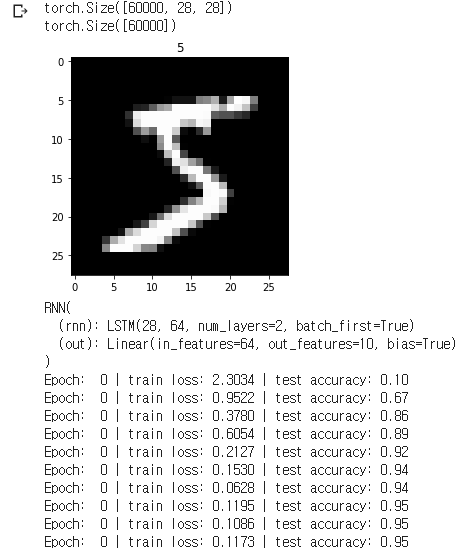
Change the value of this parameter and compare the results.

Change the value of batch\_size (128)

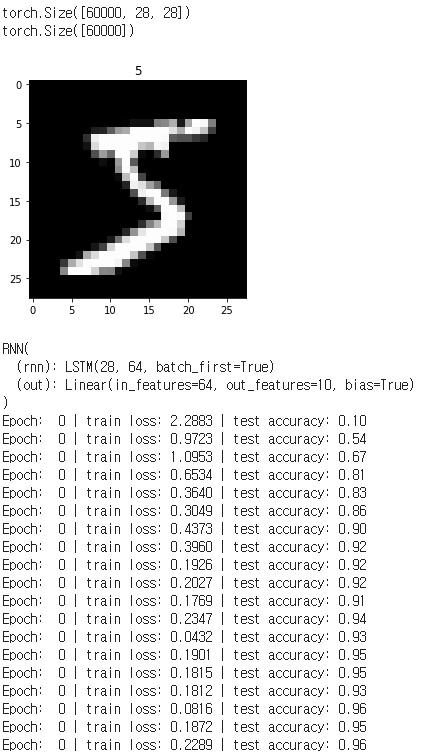
<PROGRAM CODE>



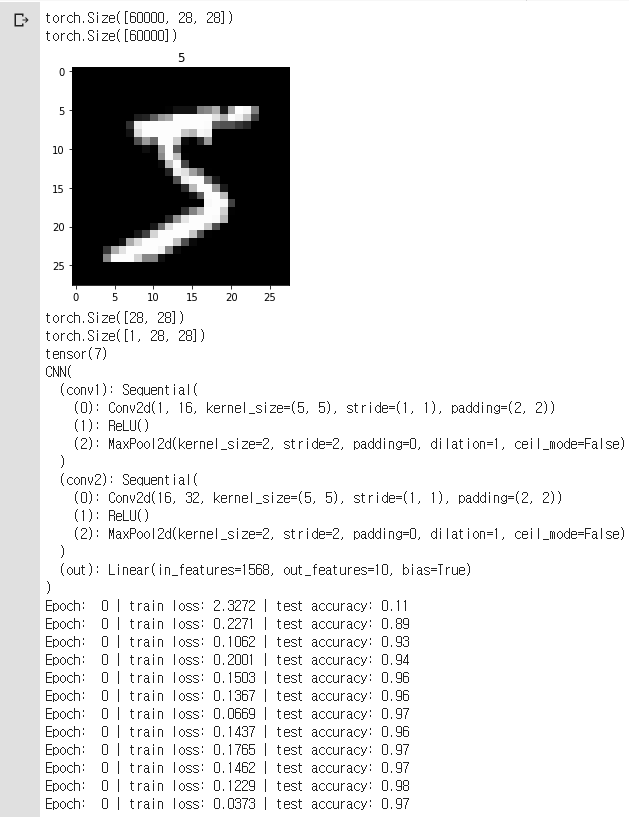
<RESULT>



6) compare the accuracy of RNN for Mnist with that of CNN.<RNN>



<CNN>

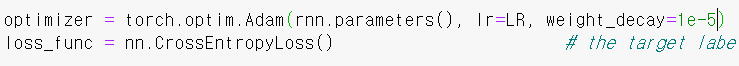


7) (\*optional\*) choose Adam optimization method and use L2 (ridge) regularization

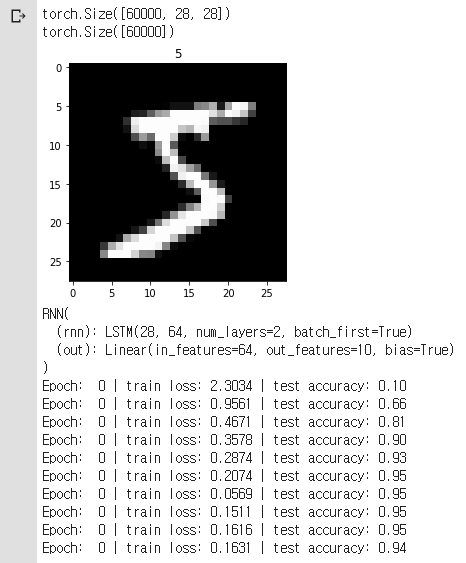
method this time. You can do so by setting ‘weight\_decay’ value in optimization

method to a tiny number (e.g. 1e-5). Compare the results of using regularization.

<PROGRAM CODE>

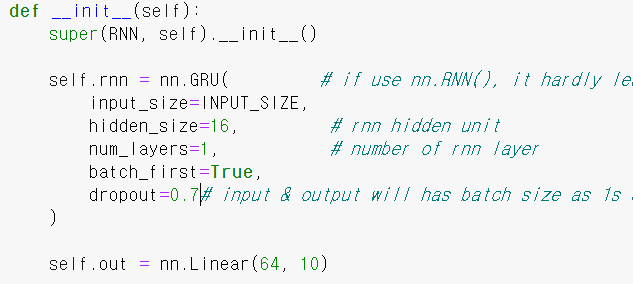


<RESULT>



8) (\*optional\*) use dropout technique and compare the results.

<PROGRAM CODE>



<RESULT>

