

## Data Science HW6

R11922066 林庭安

### Problem: Convolutional Neural Network

1. State all the hyperparameters you need for training (learning rate, #epochs, weight decay, moment, etc.) and how you tune them

Hyperparameters:

- learning rate: 0.01
- num epochs: 30
- train batch size: 64
- eval batch size: 32
- momentum: 0.9
- weight decay: 0.0005

According to the introduction slide of this homework, training more epochs helps to pass the baseline. So I changed the number of epochs from the default 10 in the sample code to 30.

## 2. Show the structure of your best model.

```
CNN(
  (conv0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn0): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv1): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv3): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv4): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn4): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=25088, out_features=256, bias=True)
  (fc2): Linear(in_features=256, out_features=256, bias=True)
  (fc3): Linear(in_features=256, out_features=10, bias=True)
  (dropout): Dropout(p=0.3, inplace=False)
)
```

Layer (type)	Output Shape	Param #
=====		
Conv2d-1	[-1, 32, 28, 28]	864
BatchNorm2d-2	[-1, 32, 28, 28]	64
ReLU-3	[-1, 32, 28, 28]	0
Conv2d-4	[-1, 64, 28, 28]	18,432
BatchNorm2d-5	[-1, 64, 28, 28]	128
ReLU-6	[-1, 64, 28, 28]	0
MaxPool2d-7	[-1, 64, 14, 14]	0
Conv2d-8	[-1, 128, 14, 14]	73,728
BatchNorm2d-9	[-1, 128, 14, 14]	256
ReLU-10	[-1, 128, 14, 14]	0
MaxPool2d-11	[-1, 128, 7, 7]	0
Conv2d-12	[-1, 256, 7, 7]	294,912
BatchNorm2d-13	[-1, 256, 7, 7]	512
ReLU-14	[-1, 256, 7, 7]	0
Dropout-15	[-1, 256, 7, 7]	0
Conv2d-16	[-1, 512, 7, 7]	1,179,648
BatchNorm2d-17	[-1, 512, 7, 7]	1,024
ReLU-18	[-1, 512, 7, 7]	0
Dropout-19	[-1, 512, 7, 7]	0
Linear-20	[-1, 256]	6,422,784
ReLU-21	[-1, 256]	0
Dropout-22	[-1, 256]	0
Linear-23	[-1, 256]	65,792
ReLU-24	[-1, 256]	0
Dropout-25	[-1, 256]	0
Linear-26	[-1, 10]	2,570
=====		
Total params: 8,060,714		
Trainable params: 8,060,714		
Non-trainable params: 0		
=====		

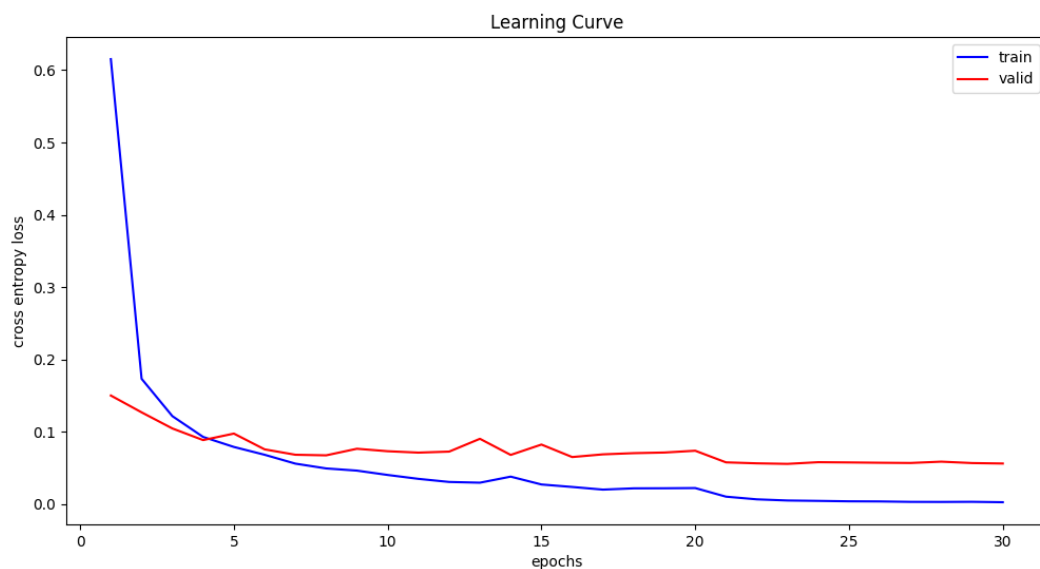
3. Explain the design of your model and what you've observed.

Initially, I tried not to change the model in the sample code, but only to increase the number of epochs, but this did not have good accuracy.

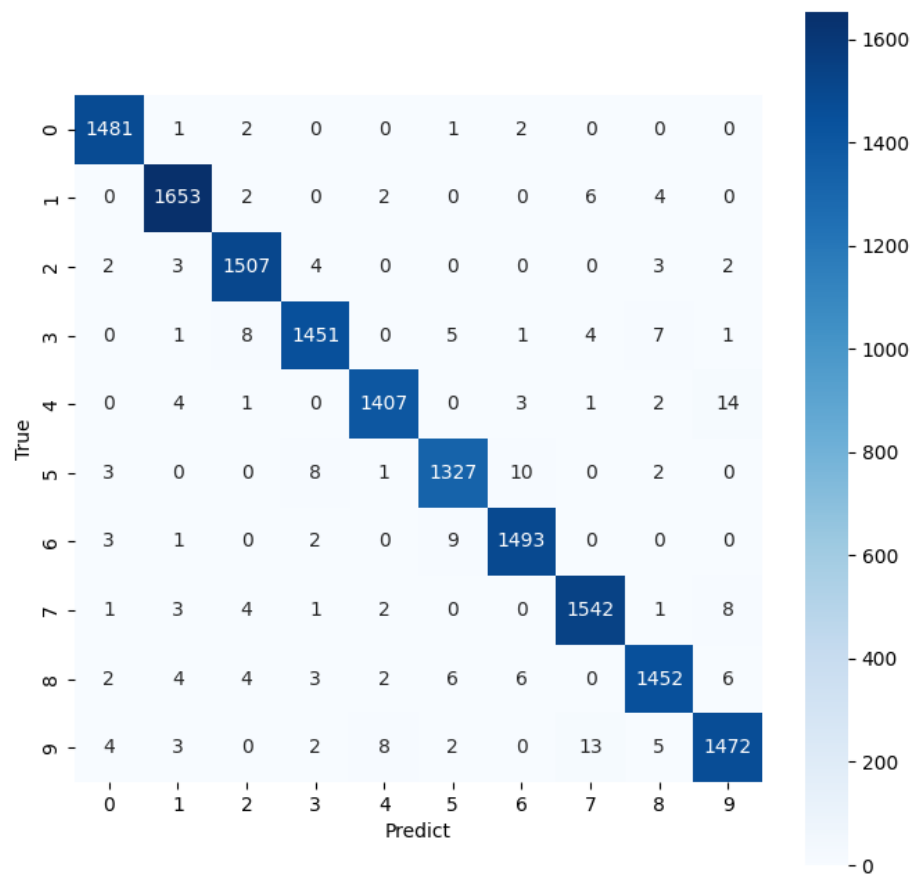
Therefore, I implemented a deeper model and added dropouts after some layers to prevent overfitting.

As TA mentioned in the homework slide, a deeper model and increasing dropout and batch layers can improve accuracy.

4. Plot the learning curve during training (CrossEntropy Loss).



5. Plot the confusion matrix for validation set, and briefly explain what you've observed.



According to the confusion matrix, 4 is easily predicted as 9, and 9 is easily predicted as 7.

I think the complex color of the dataset may be the reason for the wrong prediction.