資料科學 - 作業六

tags: `Data Science`

系級：電機所碩一　姓名：楊冠彥　學號：R11921091

Kaggle Name: r11921091_累了只能照慣例努力清醒著

1. State all the hyperparameters you need for training (learning rate, #epochs, weight decay, moment, etc.) and how you tune them (10%)

1) 由於大學有訓練過mnist的資料集，不過當時是用tensorflow，而且dataset是直接用Kaggle Digit Recognizer ，所以有點小不一樣，因此我借鑒了以前的一些經驗，先將sample code `cnn.py` 的模型改掉(模型架構可見下題)，並把圖片直接旋轉7度，也就是做點data augmentation，optimizer部分先按照Sample Code使用SGD，並嘗試如下hyperparameters：

- num_epochs: 20
- train_batch_size: 64
- eval_batch_size: 64
- lr: 0.0005
- lr_gamma: 0.1
- lr_decay_step: 20
- weight_decay: 0.0005
- momentum: 0.9

  best epochs: 17

  Public Accuracy (on Kaggle): 0.98071

2)　接著按照之前調參數的經驗，batch size調低及epoch調高準確度蠻常會變好的，batch size調低是因為可讓model更晚出現overfitting，epoch調高這是訓練更多次，當然這兩個都會影響訓練時間，但作業spec沒規定要多久訓練完，所以我就調成如下的hyperparameters：

- num_epochs: 100
- train_batch_size: 8
- eval_batch_size: 8
- lr: 0.0005
- lr_gamma: 0.1
- lr_decay_step: 20
- weight_decay: 0.0005
- momentum: 0.9

  best epochs: 84

  Public Accuracy (on Kaggle): 0.98871

3)　再來按照之前寫CNN和NLP的經驗，猜測Optimizer換成AdamW應該能稍微提高準確度，參數部分除了momentumn沒用到之外，其餘都跟前面完全一樣，如下：

- num_epochs: 100
- train_batch_size: 8
- eval_batch_size: 8
- lr: 0.0005
- lr_gamma: 0.1
- lr_decay_step: 20
- weight_decay: 0.0005

  best epochs: 51

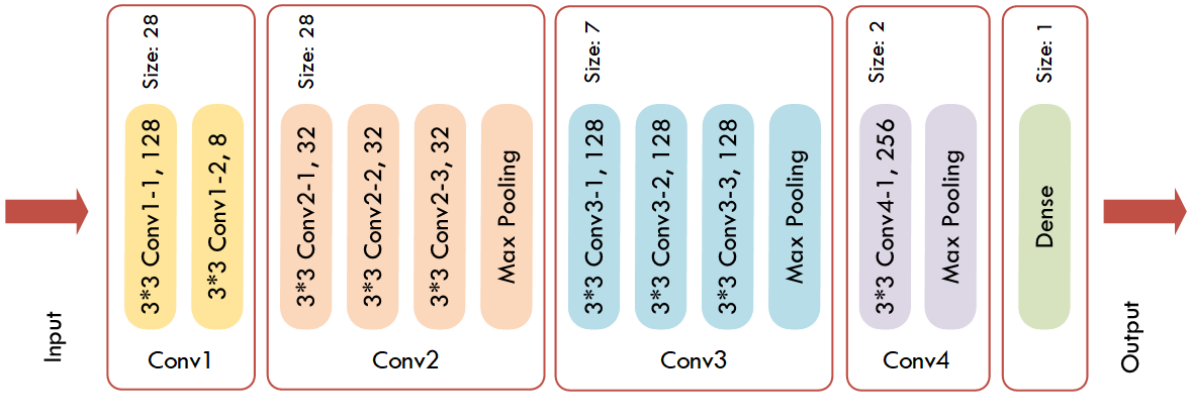
  Public Accuracy (on Kaggle): 0.99114

2. Show the structure of your best model. (hint : print(model)) (5%)

```
CNN(
  (conv1): Sequential(
    (0): Conv2d(3, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Dropout(p=0.4, inplace=False)
    (4): Conv2d(128, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  )
  (conv2): Sequential(
    (0): Conv2d(8, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU()
    (8): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (9): Conv2d(32, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (10): ReLU()
    (11): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (12): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (13): Dropout(p=0.4, inplace=False)
  )
  (conv3): Sequential(
    (0): Conv2d(32, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
```

```
    (2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU()
    (8): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (9): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (10): ReLU()
    (11): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (12): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (13): Dropout(p=0.4, inplace=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.4, inplace=False)
  )
  (fc): Sequential(
    (0): Linear(in_features=256, out_features=10, bias=True)
  )
)
```

3. Explain the design of your model and what you've observed. (10%)



圖一、模型架構圖 (圖片來源：自行繪製)

上圖我的模型示意圖。每個Convlution由convolution、Relu、Batch2Norm組成，並且在Conv1、Conv2、Conv3、Conv4這四個大層的結束都會做dropout。

我的模型有 11 個Convolution層和1個全連接層。除了Conv1和Conv2的連接，其他相鄰的Convolution層都通過Max Pooling連接，且每組包含一系列 3×3 卷積層，從 Conv2 的32 個增強到最後一組Conv4 的 256 個。至於前面會多放conv1是因為我偶然嘗試下，發現多放這個部分可以些微提高準確度。當我嘗試如下hyperparameter且圖片部分旋轉15度、optimizer採用SGD時，Kaggle準確度如表一所示。

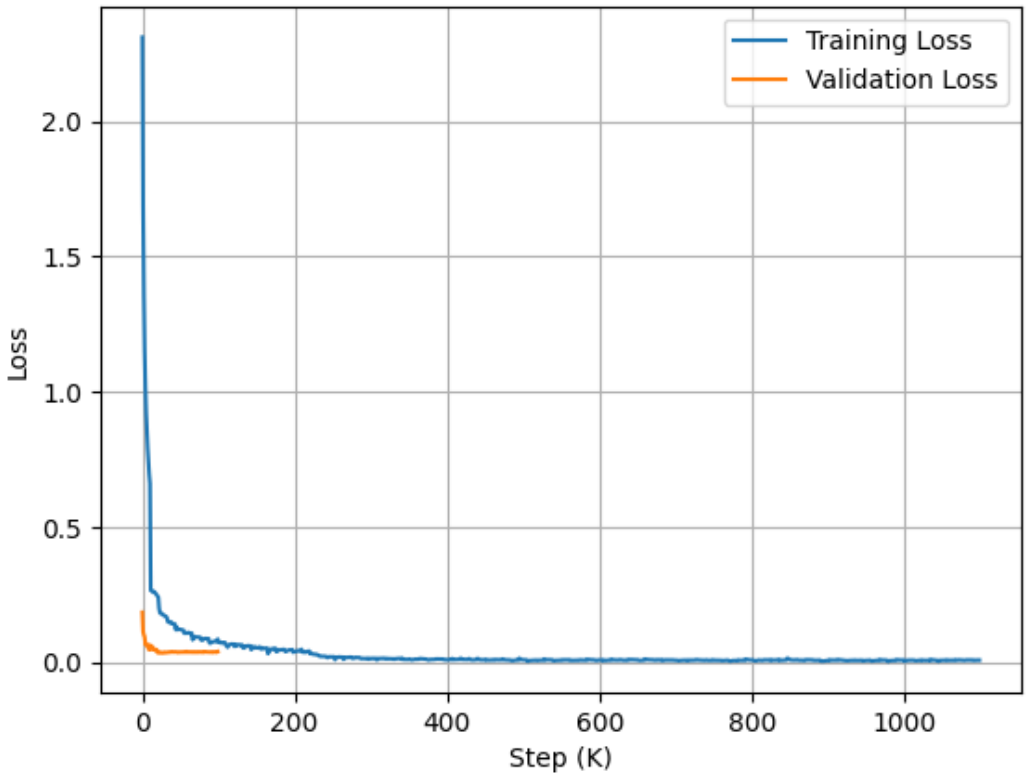
- num_epochs: 50
- train_batch_size: 8
- eval_batch_size: 8
- lr: 0.0005
- lr_gamma: 0.1
- lr_decay_step: 20
- weight_decay: 0.0005
- momentum: 0.9

表一、是否加入Conv1之準確度

	加入Conv1	**不加入Conv1**
Public Accuracy	0.986	0.98457

另外就是我發現這次的模型要盡量多疊一些convoltion層數，寬度則不需要拉到太大，256就差不多了，即讓模型變得比較高高瘦瘦。

4. Plot the learning curve during training (CrossEntropy Loss). (10%)



5. Plot the confusion matrix for validation set, and briefly explain what you've observed. (15%)

我發現實際是4的部分有蠻多預測為9的，9錯認為4也是第四多的，7和9也出現類似的狀況，猜測是因為9和4長得蠻像的，7和9也蠻像的。