

視訊串流與追蹤 HW1

311551096 錢承

1. How to reproduce your result:

- (1) 利用提供的環境檔建立環境並且啟用環境：

```
conda env create -f environment.yml
```

```
conda activate lab1
```

- (2) 執行 test.py

```
python test.py
```

使用的環境和套件包含 python 3.8.13、pytorch 1.12.1、numpy

1.23.3、matplotlib 3.6.1、pandas 1.5.0、PIL、glob、re

2. Number of Model parameters:

```
num = sum(p.numel() for p in vgg.parameters() if p.requires_grad)
```

Number of Model parameters: 9247

3. Explain model structure:

包含 4 個 block，每個 block 都有 convolution、batch normalize、leaky ReLU，而 convolution 的 channel 分別為 4、8、16、32。

```
self.c1 = nn.Sequential(  
    nn.Conv2d(3, 4, 3, 1, 1),  
    nn.BatchNorm2d(4),  
    nn.LeakyReLU(),  
)  
  
self.c2 = nn.Sequential(  
    nn.Conv2d(4, 8, 3, 1, 1),  
    nn.BatchNorm2d(8),  
    nn.LeakyReLU(),  
)  
  
self.c3 = nn.Sequential(  
    nn.Conv2d(8, 16, 3, 1, 1),  
    nn.BatchNorm2d(16),  
    nn.LeakyReLU(),  
)  
  
self.c4 = nn.Sequential(  
    nn.Conv2d(16, 32, 3, 1, 1),  
    nn.BatchNorm2d(32),  
    nn.LeakyReLU(),  
)
```

在每一個 block 之後，會進行 Max pool。

```
self.mp = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
```

再來會將其攤平，分組進行全連接產生一個 channel 為 3 的 node。最後會進行全連接，將維度調整為 batch size x 10，經過 softmax 後輸出。

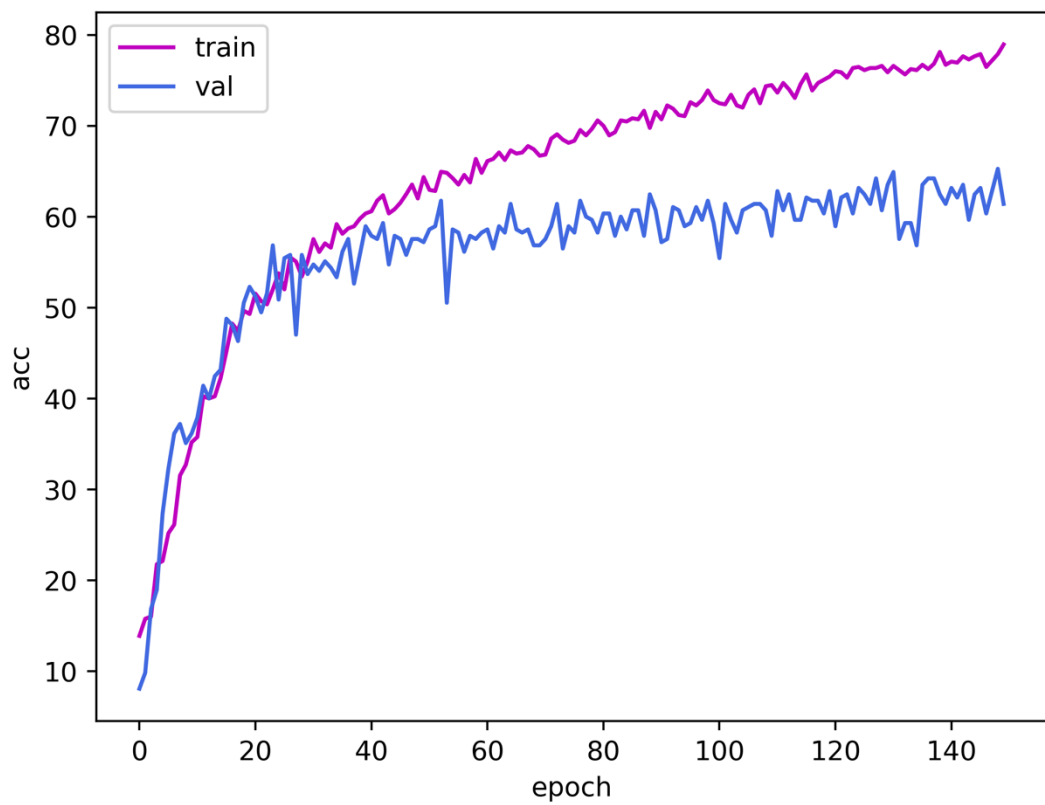
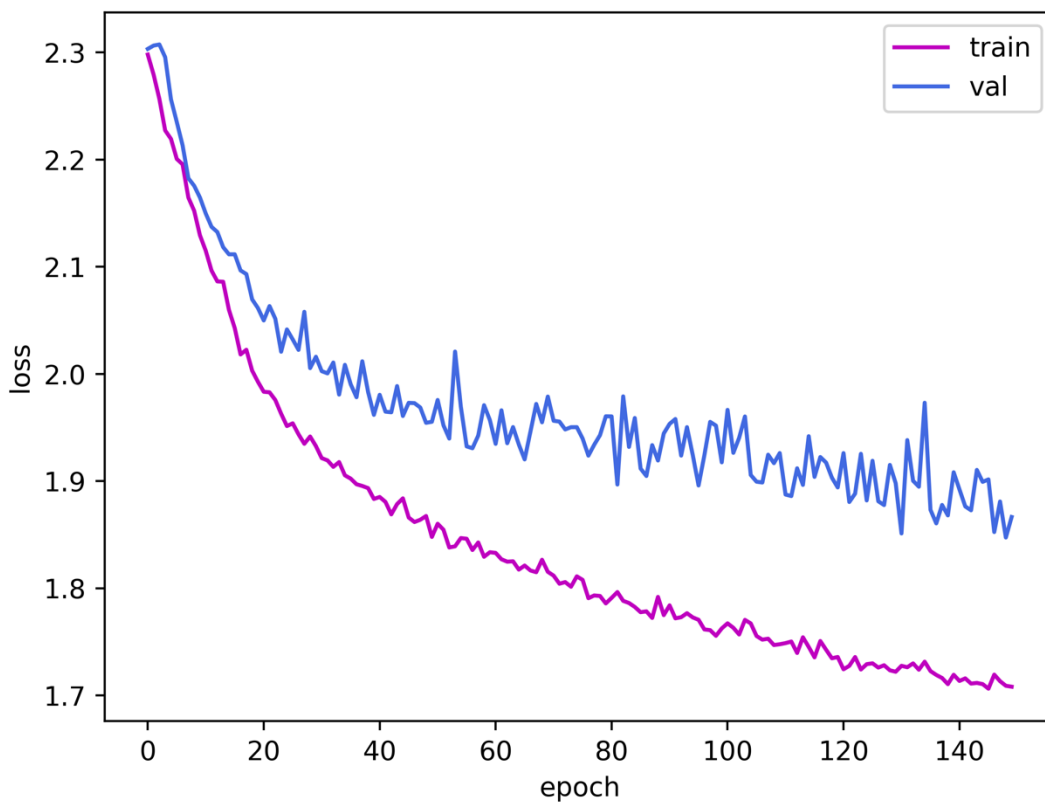
```
self.fc = nn.Sequential(  
    nn.Conv2d(1,3,(1,896),896),  
    nn.LeakyReLU(),  
    nn.Conv2d(3,10,(1,7)),  
    nn.ReLU(),  
    nn.Softmax(dim = 1)  
)
```

Loss function 採用 Cross Entropy Loss，optimizer 則採用 Adam。

```
optimizer = torch.optim.Adam(vgg.parameters(), lr=hparams['lr'])  
criterion = torch.nn.CrossEntropyLoss()
```

4. Results:

在訓練 20 個 epoch 之後，可以看到 loss 有顯著的下降。而準確率的部分，也是在 20 個 epoch 之後逐漸穩定在 60%左右。



5. Problems encountered and discussion

在設計架構時，若使用一般的全連接，會造成參數量都在百萬等級。為了降低參數量，我對全連接層做了改動，我是對其分組後再全連接，這樣可以有效的降低參數量，達到 9000 左右。