Model 1

- Model layer : 5
- Full connect layer : 3
- Activation function : ReLU
- Optimizer function : SGD
- Nesterov 加速算法
- Learning rate : 0.013

```
self.conv1 = nn.Conv2d(3, 128, kernel_size=3, stride=1)
self.conv2 = nn.Conv2d(128, 256, kernel_size=3, stride=1)
self.conv3 = nn.Conv2d(256, 256, kernel_size=3, stride=1)
self.conv4 = nn.Conv2d(256, 512, kernel_size=3, stride=1)
self.conv5 = nn.Conv2d(512, 512, kernel_size=3, stride=1)

self.relu = nn.ReLU()
self.maxpool = nn.MaxPool2d(kernel_size=2)
self.dropout = nn.Dropout(p=0.2)
self.fc1 = nn.Linear(12800, 5000)
self.fc2 = nn.Linear(5000, 1000)
self.fc3 = nn.Linear(1000, 100)
```

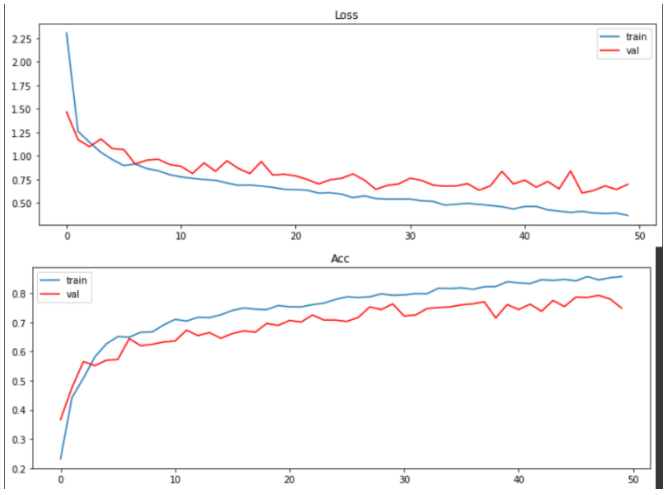
```
out = self.conv1(x)
out = self.relu(out)
out = self.maxpool(out)
out = self.conv2(out)
out = self.relu(out)
out = self.maxpool(out)
out = self.conv3(out)
out = self.relu(out)
out = self.maxpool(out)
out = self.conv4(out)
out = self.relu(out)
out = self.maxpool(out)
out = self.conv5(out)
out = self.relu(out)
out = self.maxpool(out)
out = out.view(out.size(0), -1)
out = self.fc1(out)
out = self.dropout(out)
out = self.fc2(out)
out = self.fc3(out)
```

```
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.013, momentum=0.9, nesterov=True)
```

執行結果 ：

- Epoch : 50

```
  Val Acc: 0.750869   Val Loss: 0.676238
========== Epoch 36 ==========
Train Acc: 0.818663 Train Loss: 0.494711
  Val Acc: 0.760139   Val Loss: 0.703355
========== Epoch 38 ==========
Train Acc: 0.822303 Train Loss: 0.471639
  Val Acc: 0.770568   Val Loss: 0.680050
========== Epoch 40 ==========
Train Acc: 0.839510 Train Loss: 0.434469
  Val Acc: 0.761298   Val Loss: 0.700583
========== Epoch 42 ==========
Train Acc: 0.833223 Train Loss: 0.461547
  Val Acc: 0.762457   Val Loss: 0.665460
========== Epoch 44 ==========
Train Acc: 0.843812 Train Loss: 0.410773
  Val Acc: 0.775203   Val Loss: 0.648574
========== Epoch 46 ==========
Train Acc: 0.842488 Train Loss: 0.408668
  Val Acc: 0.786790   Val Loss: 0.603924
========== Epoch 48 ==========
Train Acc: 0.845797 Train Loss: 0.386227
  Val Acc: 0.792584   Val Loss: 0.680107
========== Epoch 50 ==========
Train Acc: 0.857048 Train Loss: 0.365693
  Val Acc: 0.748552   Val Loss: 0.696548
```



- Kaggle 測試結果 ：0.77076

Model 2

- Model layer : 4
- Full connect layer : 3
- Activation function : ReLU
- Optimizer function : SGD + Schedular
- Nesterov 加速算法
- Learning rate : 0.0015

```
self.conv1 = nn.Conv2d(3, 128, 3, stride=1)
self.conv2 = nn.Conv2d(128, 256, 3, stride=1)
self.batch2 = nn.BatchNorm2d(256)
self.conv3 = nn.Conv2d(256, 256, 3, stride=1)
self.conv4 = nn.Conv2d(256, 512, 3, stride=1)
self.batch4 = nn.BatchNorm2d(512)

self.relu = nn.ReLU()
self.maxpool = nn.MaxPool2d(kernel_size=2)
self.dropout = nn.Dropout(p=0.2)
self.fc1 = nn.Linear(73728, 1024)
self.fc2 = nn.Linear(1024, 128)
self.fc3 = nn.Linear(128, 5)
```

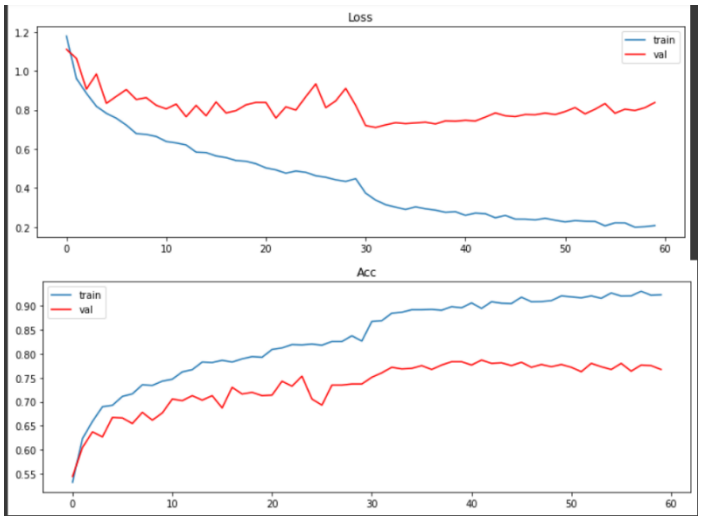
```
out = self.conv1(x)
out = self.relu(out)
out = self.maxpool(out)
out = self.conv2(out)
out = self.relu(out)
out = self.maxpool(out)
out = self.batch2(out)
out = self.conv3(out)
out = self.relu(out)
out = self.maxpool(out)
out = self.conv4(out)
out = self.relu(out)
out = self.maxpool(out)
out = self.batch4(out)
out = out.view(out.size(0), -1)
out = self.fc1(out)
out = self.fc1(out)
out = self.dropout(out)
out = self.fc2(out)
out = self.fc3(out)
```

```
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.0015, momentum=0.9, nesterov=True)
scheduler = optim.lr_scheduler.StepLR(optimizer, step_size=30, gamma=0.1)
```

執行結果 :

- Epoch : 60
- Kaggle 測試結果 : 0.80730

```
========== Epoch 46 ==========
Train Acc: 0.917604 Train Loss: 0.241173
  Val Acc: 0.782155   Val Loss: 0.766018
========== Epoch 48 ==========
Train Acc: 0.908339 Train Loss: 0.237112
  Val Acc: 0.777520   Val Loss: 0.775631
========== Epoch 50 ==========
Train Acc: 0.920251 Train Loss: 0.235471
  Val Acc: 0.777520   Val Loss: 0.776720
========== Epoch 52 ==========
Train Acc: 0.916281 Train Loss: 0.233442
  Val Acc: 0.762457   Val Loss: 0.812326
========== Epoch 54 ==========
Train Acc: 0.915288 Train Loss: 0.229346
  Val Acc: 0.772885   Val Loss: 0.803708
========== Epoch 56 ==========
Train Acc: 0.919921 Train Loss: 0.221836
  Val Acc: 0.779838   Val Loss: 0.783130
========== Epoch 58 ==========
Train Acc: 0.929848 Train Loss: 0.199752
  Val Acc: 0.776362   Val Loss: 0.797153
========== Epoch 60 ==========
Train Acc: 0.922568 Train Loss: 0.207922
  Val Acc: 0.767092   Val Loss: 0.838216
```



Model 1 與 Model 2 的差異

1. Model Layer 減少 (5 → 4)
2. 加入 2 層 batch，更新參數
3. Optimizer 搭配 schedular
4. Learning rate 調小 (0.013 → 0.0015)