

## 模型架構設計與說明

### Convolutional layers

conv1 :  $3 \times 32 \times 32 \rightarrow 16 \times 32 \times 32$  , 經 pooling 後變為  $16 \times 16 \times 16$  。

conv2 :  $16 \times 16 \times 16 \rightarrow 32 \times 16 \times 16$  , 經 pooling 後變為  $32 \times 8 \times 8$  。

### Pooling layer

使用 MaxPool2d , 每次降 2 倍。

### Fully Connected Layers

fc1 : 將展平後的輸入 ( $32 \times 8 \times 8$ ) 轉換為 512

fc2 :  $512 \rightarrow 256$  。

fc3 :  $256 \rightarrow 5$  (對應 5 個標籤)。

### 激活函數與其他層

使用 ReLU 激活函數。

使用 Dropout 防止過擬合。

使用 Softmax 將輸出轉換為機率分布。

```
x = self.conv1(x)
```

```
x = torch.relu(x)
```

```
x = self.pool(x)
```

```
x = self.conv2(x)
```

```
x = torch.relu(x)
```

```
x = self.pool(x)
```

```
x = x.view(-1, 32*8*8)
```

```
x = self.fc1(x)
```

```
x = torch.relu(x)
```

```
x = self.dropout(x)
```

```
x = self.fc2(x)
x = torch.relu(x)
x = self.dropout(x)

x = self.fc3(x)

x = self.softmax(x)
```

## 優化器與損失函數

SGD（隨機梯度下降）

CrossEntropyLoss

## 超參數設定及結果比較

學習率=0.001 批次大小 = 64

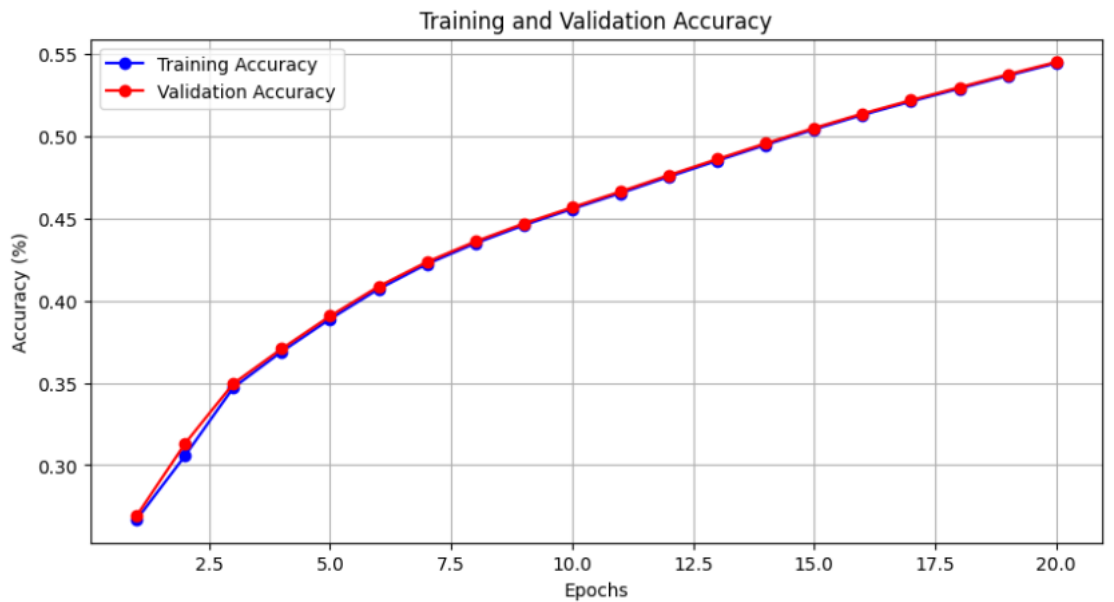
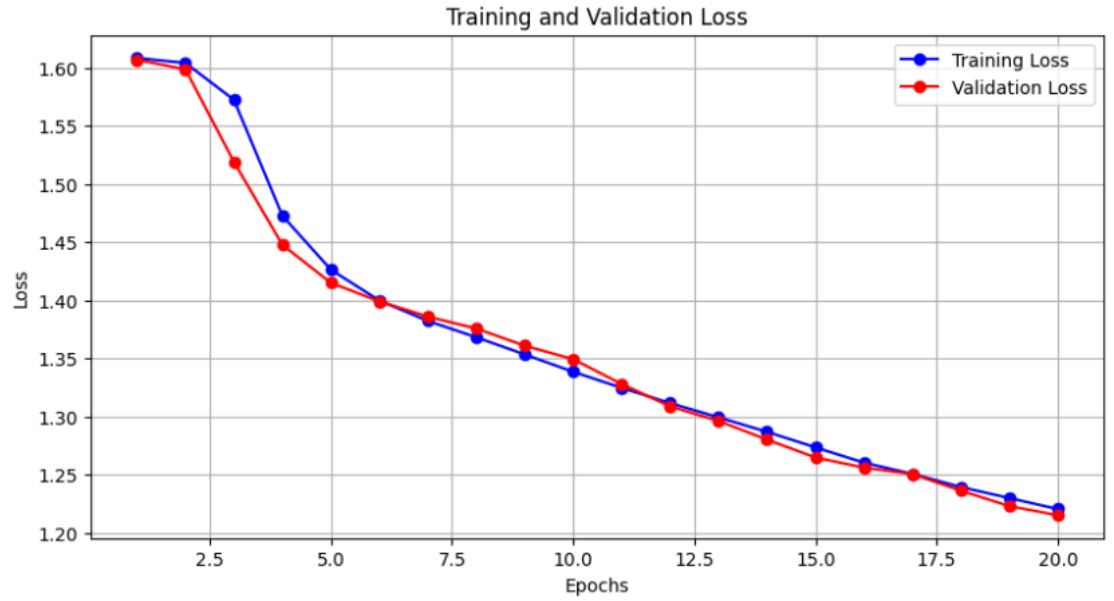
train Accuracy : 54%

Validation Accuracy : 54%

Epoch: 20

Training Loss: 1.2206700623109936

Validation Loss: 1.2153666107058525



學習率=0.01 批次大小 = 64

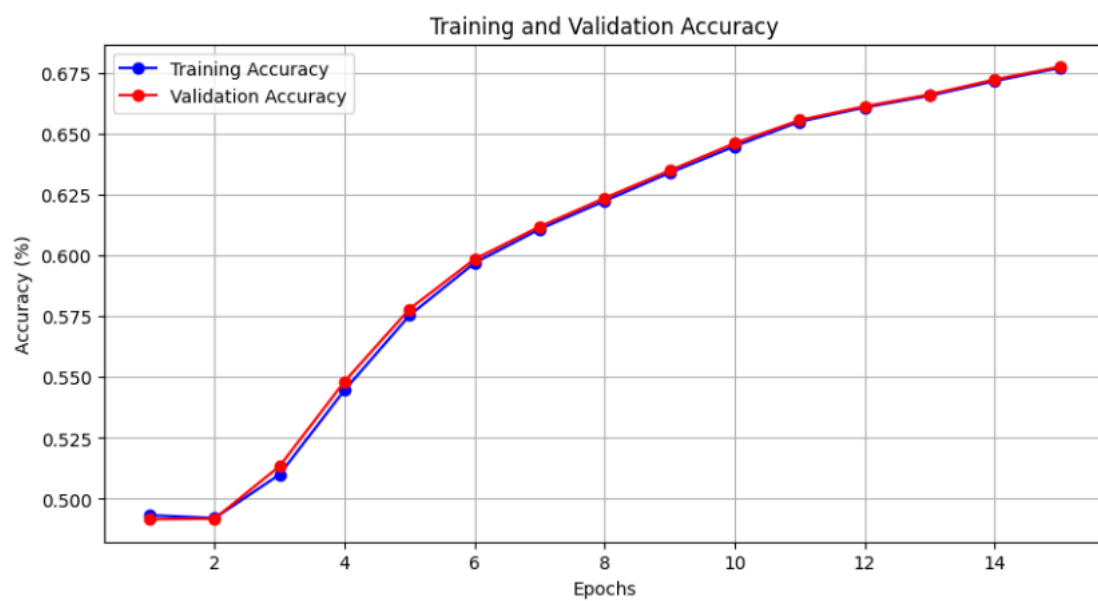
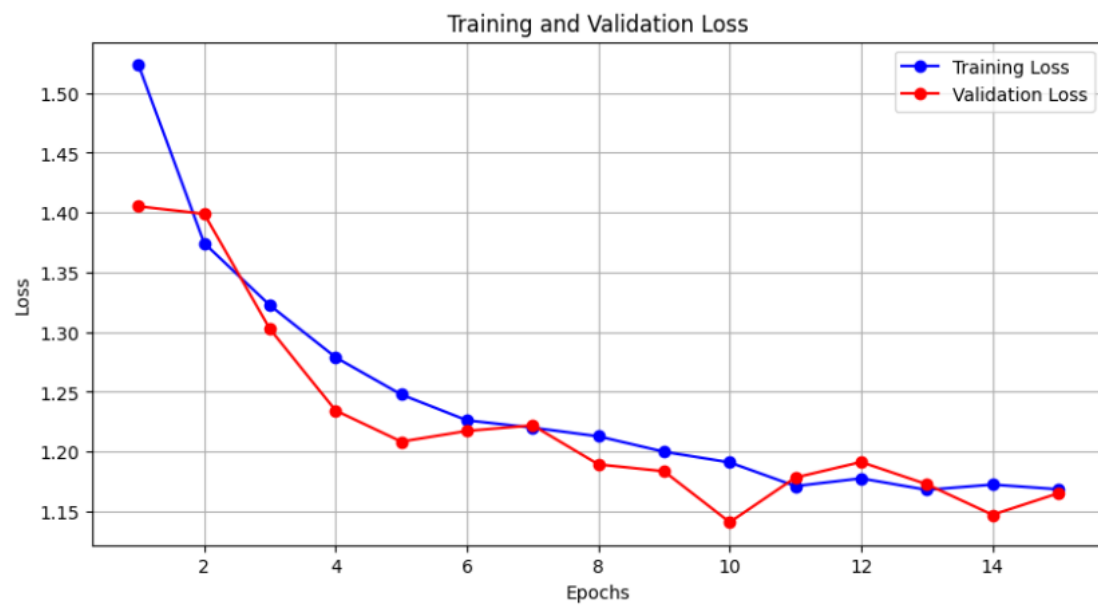
train Accuracy : 67%

Validation Accuracy : 67%

Epoch: 15

Training Loss: 1.1685377460494637

Validation Loss: 1.165133410781622



學習率=0.01 批次大小 = 32

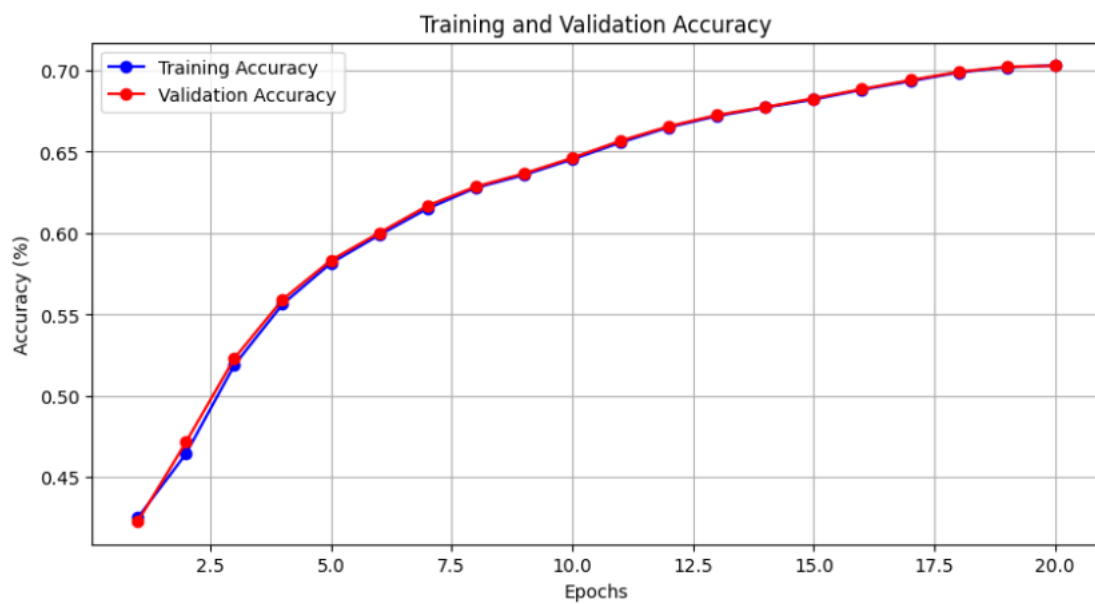
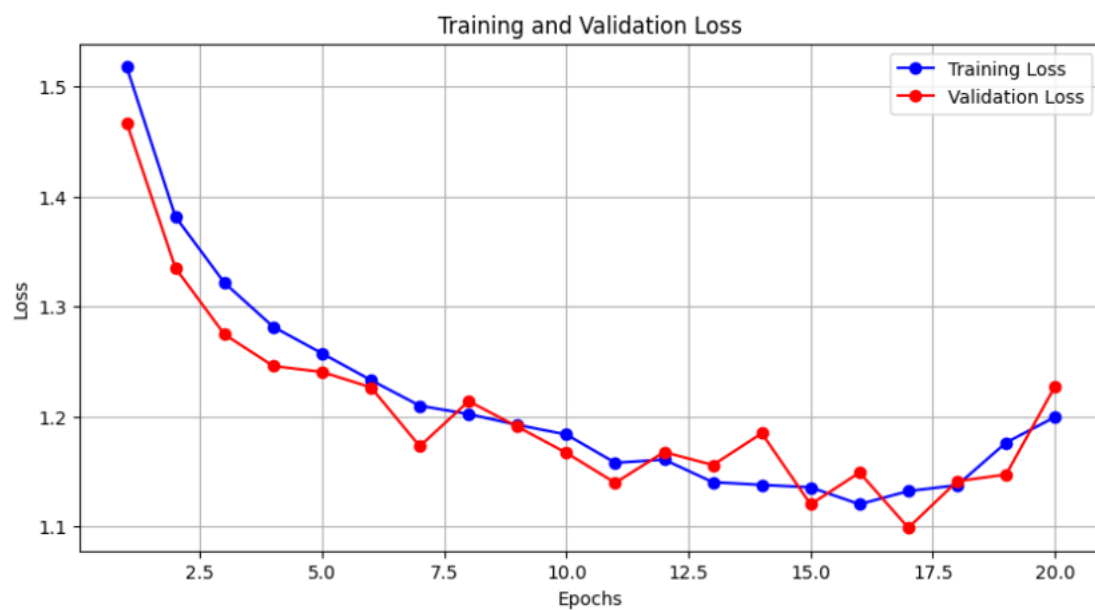
train Accuracy : 70%

Validation Accuracy : 70%

Epoch:20

Training Loss: 1.1993420573771

Validation Loss: 1.2272180241942405



學習率=0.001 批次大小 = 32

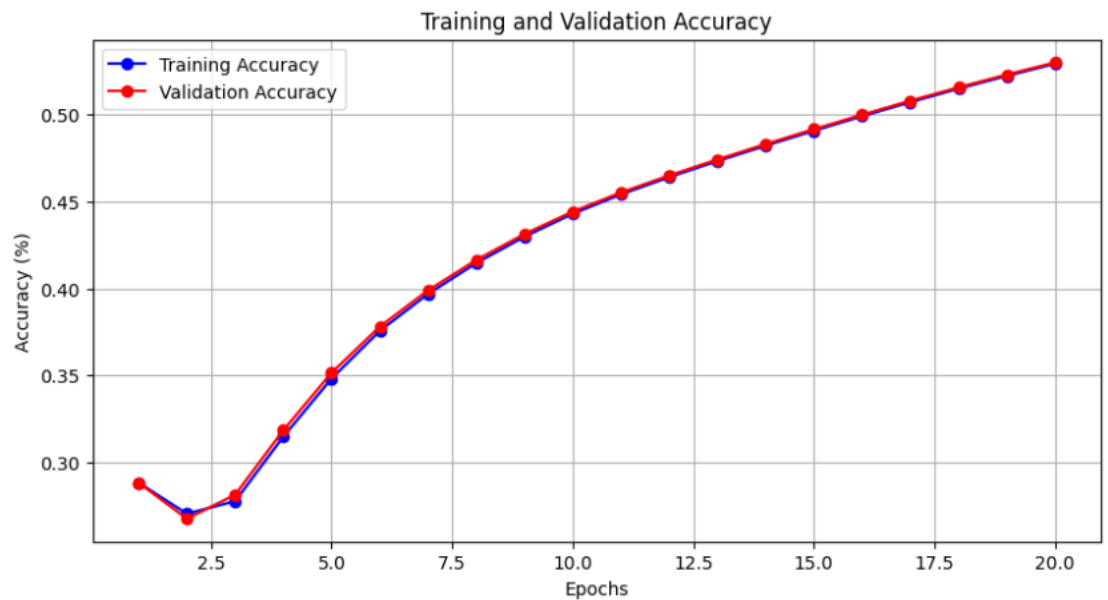
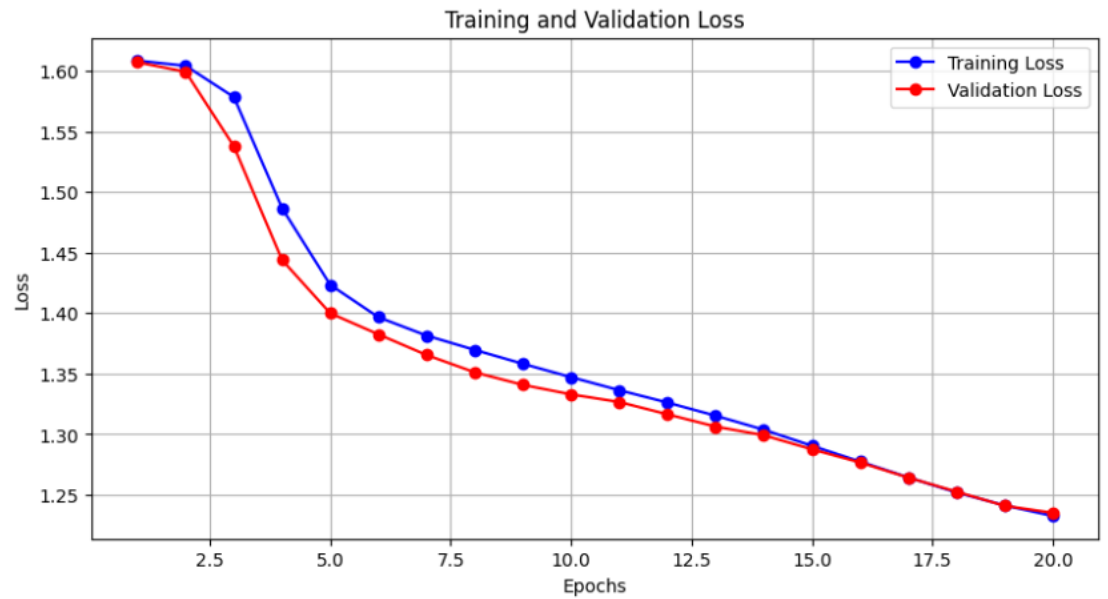
train Accuracy : 53%

Validation Accuracy : 53%

Epoch: 20

Training Loss: 1.2325558196604252

Validation Loss: 1.2351374534368516



Learning rate	Batch size	train Accuracy	Validation Accuracy	Training Loss	Validation Loss:	Best Epoch
0.001	32	53%	53%	1.2351	1.2351	20
	64	54%	54%	1.2207	1.2154	20
0.01	32	70%	70%	1.2272	1.2272	20
	64	67%	67%	1.1685	1.1651	15

分析：學習率對模型的訓練效果影響最大，學習率（0.01）的結果優於 0.001。學習率過低導致模型收斂速度慢，最終準確率與損失結果表現較差。學習率（0.01）收斂速度快，在批次大小(64)時，15 個 Epoch 就已達到（67%）的準確率，在批次大小(32)時訓練與驗證準確率達（70%）。

### 訓練與測試的準確率與損失值分析

學習率 = 0.001，批次大小 = 64

訓練準確率：54%，驗證準確率：54%

訓練損失：1.2207，驗證損失：1.2154

54%的準確率，模型可能未能充分學習數據特徵。

學習率過低，參數更新幅度過小，可能需要更多世代才能  
收斂。

學習率 = 0.01，批次大小 = 64

訓練準確率：67%，驗證準確率：67%

訓練損失：1.1685，驗證損失：1.1651

與學習率 0.001 相比，準確率提升很多，模型在數據特徵學習上更加  
充分。損失也降低，表示學習率適當，能加快收斂。

學習率 = 0.01，批次大小 = 32

訓練準確率：70%，驗證準確率：70%

訓練損失：1.1993，驗證損失：1.2272

批次大小從 64 降為 32，準確率進一步提高，小批次大小有助於捕捉  
更多數據細節。

學習率 = 0.001，批次大小 = 32

訓練準確率：53%，驗證準確率：53%

訓練損失：1.2326，驗證損失：1.2351

學習率低導致準確率與損失無法有效提升。

即使批次大小較小，模型也無法捕捉數據細節，仍停留在較低的準確率。

### **改進建議**

最佳的組合是，學習率 0.01、批次大小 32 的組合，能獲得高準確率。

使用動態學習率調整，避免高學習率在後期導致收斂不穩定。

使用更深層的模型，增加卷積層和全連接層，捕捉更多特徵。

增加迭代次數，讓模型可以繼續收斂。

### **心得與討論**

這次改使用 pytorch 來完成作業，比之前方便許多。低學習率下，模型不能有效的學習數據特徵，準確率低下，提高學習率後，準確率有所提升，但模型的準確率在不同配置下徘徊在 53%-70%，未能達到更高的準確率，模型可能存在學習能力不足的問題。CNN 架構相對簡單，僅包含兩層卷積和三層全連接層，可能無法捕捉圖像中的更多特徵或更複雜的樣式。