

深度學習HW2:影像分類

AI二 B1228005 胡樂麒

一、模型架構說明

1.1 CNN的主要架構如下：

Conv1: 3輸入通道→32輸出通道, 3×3卷積核, padding=1

Conv2: 32→64通道, 3×3卷積核, padding=1

Conv3: 64→128通道, 3×3卷積核, padding=1

Conv4: 128→128通道, 3×3卷積核, padding=1

Conv1~4我都使用了BatchNorm跟ReLU, 進行正規化, 減少不同特徵尺度帶來的偏差, 並用ReLU來提升模型的非線性表達能力。

1.2 池化層：

使用self.pool = torch.nn.MaxPool2d (kernel_size=2, stride=2)
因為stride=2所以為了防止抓取不到特徵所以設定(padding=1)。

1.3 全連接層：

FC1: 輸入 $128 \times 2 \times 2 = 512$ 維, 輸出256維隱藏層, 使用ReLU激活。

在FC1後加入Dropout(), 降低過擬合。

FC2: 256維→10維輸出(對應10個類別)。

1.4 設計考量：

使用3×3卷積核, 能以較少參數量來捕捉有效的局部特徵。

逐步增加通道數，從32→64→128，配合池化層提升特徵的表達能力。

二、模型效能與改進

2.1 準確率

訓練準確率：約89%

驗證準確率：約85%

測試準確率：86.28%

2.2 資料、模型改進與前處理

數據增強：

訓練時加入RandomHorizontalFlip()

對訓練資料中的圖片進行隨機水平翻轉模擬現實中的圖片，使模型能學會辨識不同方向的圖像。

RandomCrop(32, padding=4)

先在圖片四周填補 4 像素的邊界(padding)，再隨機裁切出 32x32 大小的區塊。

Normalize(mean=[0.5]*3, std=[0.5]*3): 標準化像素值至 [-1, 1] 範圍。

模型訓練優化：

Adam: 調整學習率，穩定收斂。

學習率調整：

`scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=10, gamma=0.5)`

(此方法是ChatGPT告訴我的)

StepLR是用來調整學習率。設定每訓練10個epoch，學習率乘以0.5。因此在後面的訓練可以使用較小的學習率來減少損失的波動，幫助模型更穩定地收斂。

`criterion = torch.nn.CrossEntropyLoss(label_smoothing=0.1)`

(此處參數調成0.1是我問ChatGPT它建議的)

用0.1而非原本的1、0，讓模型訓練時更彈性提高泛化能力。

增加訓練輪次：從原本10個epoch增加到50個epoch。

架構調整：

在框架基礎上增加了一層卷積(conv4)。

2.3 改進效果

最大的改變就是準確率的提升，因為我進行完資料翻轉、增加CNN的層數和epoch=50之後準確率就一直在80%上下，後來使用這個scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=10, gamma=0.5)後才突破85%。

三、較差情況分析

3.1 樣本類型

- 圖片解析度低、資訊量不足。
- 物品類別外觀相近，如“貓 vs 狗”。
- CIFAR-10中每張圖僅32×32像素，對小細節的操作較為困難。

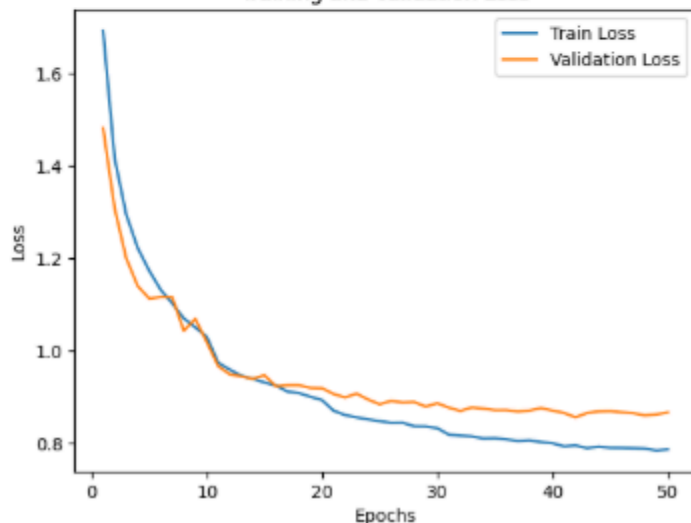
四、自由發揮

85%準確率

過程中我增加了一層CNN(Conv4)和資料增強的方法(圖片翻轉,隨機裁切), 學習率調整與調整交叉熵參數等等方法才完成。

Epoch 1/50, Train Loss: 1.6928, Train Acc: 0.4444, Val Loss: 1.4821, Val Acc: 0.5549
Epoch 2/50, Train Loss: 1.4165, Train Acc: 0.5948, Val Loss: 1.3084, Val Acc: 0.6431
Epoch 3/50, Train Loss: 1.2953, Train Acc: 0.6568, Val Loss: 1.2005, Val Acc: 0.6909
Epoch 4/50, Train Loss: 1.2223, Train Acc: 0.6956, Val Loss: 1.1397, Val Acc: 0.7226
Epoch 5/50, Train Loss: 1.1730, Train Acc: 0.7217, Val Loss: 1.1128, Val Acc: 0.7341
Epoch 6/50, Train Loss: 1.1314, Train Acc: 0.7438, Val Loss: 1.1158, Val Acc: 0.7332
Epoch 7/50, Train Loss: 1.1022, Train Acc: 0.7555, Val Loss: 1.1155, Val Acc: 0.7369
Epoch 8/50, Train Loss: 1.0696, Train Acc: 0.7731, Val Loss: 1.0427, Val Acc: 0.7659
Epoch 9/50, Train Loss: 1.0501, Train Acc: 0.7821, Val Loss: 1.0685, Val Acc: 0.7609
Epoch 10/50, Train Loss: 1.0295, Train Acc: 0.7901, Val Loss: 1.0181, Val Acc: 0.7818
Epoch 11/50, Train Loss: 0.9733, Train Acc: 0.8187, Val Loss: 0.9656, Val Acc: 0.8041
Epoch 12/50, Train Loss: 0.9585, Train Acc: 0.8260, Val Loss: 0.9483, Val Acc: 0.8119
Epoch 13/50, Train Loss: 0.9452, Train Acc: 0.8315, Val Loss: 0.9438, Val Acc: 0.8115
Epoch 14/50, Train Loss: 0.9387, Train Acc: 0.8328, Val Loss: 0.9388, Val Acc: 0.8169
Epoch 15/50, Train Loss: 0.9307, Train Acc: 0.8381, Val Loss: 0.9464, Val Acc: 0.8178
Epoch 16/50, Train Loss: 0.9240, Train Acc: 0.8415, Val Loss: 0.9225, Val Acc: 0.8235
Epoch 17/50, Train Loss: 0.9109, Train Acc: 0.8455, Val Loss: 0.9247, Val Acc: 0.8248
Epoch 18/50, Train Loss: 0.9077, Train Acc: 0.8476, Val Loss: 0.9247, Val Acc: 0.8210
Epoch 19/50, Train Loss: 0.8998, Train Acc: 0.8523, Val Loss: 0.9191, Val Acc: 0.8280
Epoch 20/50, Train Loss: 0.8925, Train Acc: 0.8558, Val Loss: 0.9183, Val Acc: 0.8279
Epoch 21/50, Train Loss: 0.8696, Train Acc: 0.8641, Val Loss: 0.9058, Val Acc: 0.8323
Epoch 22/50, Train Loss: 0.8601, Train Acc: 0.8698, Val Loss: 0.8983, Val Acc: 0.8377
Epoch 23/50, Train Loss: 0.8552, Train Acc: 0.8726, Val Loss: 0.9066, Val Acc: 0.8273
Epoch 24/50, Train Loss: 0.8506, Train Acc: 0.8741, Val Loss: 0.8937, Val Acc: 0.8366
Epoch 25/50, Train Loss: 0.8470, Train Acc: 0.8774, Val Loss: 0.8835, Val Acc: 0.8427
Epoch 26/50, Train Loss: 0.8431, Train Acc: 0.8789, Val Loss: 0.8907, Val Acc: 0.8372
Epoch 27/50, Train Loss: 0.8434, Train Acc: 0.8780, Val Loss: 0.8873, Val Acc: 0.8399
Epoch 28/50, Train Loss: 0.8359, Train Acc: 0.8832, Val Loss: 0.8883, Val Acc: 0.8375
Epoch 29/50, Train Loss: 0.8356, Train Acc: 0.8814, Val Loss: 0.8785, Val Acc: 0.8477
Epoch 30/50, Train Loss: 0.8317, Train Acc: 0.8850, Val Loss: 0.8857, Val Acc: 0.8411
Epoch 31/50, Train Loss: 0.8178, Train Acc: 0.8904, Val Loss: 0.8763, Val Acc: 0.8434
Epoch 32/50, Train Loss: 0.8155, Train Acc: 0.8901, Val Loss: 0.8690, Val Acc: 0.8471
Epoch 33/50, Train Loss: 0.8141, Train Acc: 0.8915, Val Loss: 0.8767, Val Acc: 0.8421
Epoch 34/50, Train Loss: 0.8094, Train Acc: 0.8942, Val Loss: 0.8740, Val Acc: 0.8481
Epoch 35/50, Train Loss: 0.8099, Train Acc: 0.8941, Val Loss: 0.8708, Val Acc: 0.8454
Epoch 36/50, Train Loss: 0.8073, Train Acc: 0.8958, Val Loss: 0.8708, Val Acc: 0.8484
Epoch 37/50, Train Loss: 0.8042, Train Acc: 0.8952, Val Loss: 0.8679, Val Acc: 0.8517
Epoch 38/50, Train Loss: 0.8054, Train Acc: 0.8953, Val Loss: 0.8699, Val Acc: 0.8480
Epoch 39/50, Train Loss: 0.8014, Train Acc: 0.8979, Val Loss: 0.8751, Val Acc: 0.8472
Epoch 40/50, Train Loss: 0.7992, Train Acc: 0.9003, Val Loss: 0.8699, Val Acc: 0.8465
Epoch 41/50, Train Loss: 0.7924, Train Acc: 0.9028, Val Loss: 0.8647, Val Acc: 0.8495
Epoch 42/50, Train Loss: 0.7942, Train Acc: 0.9007, Val Loss: 0.8553, Val Acc: 0.8550
Epoch 43/50, Train Loss: 0.7886, Train Acc: 0.9033, Val Loss: 0.8645, Val Acc: 0.8483
Epoch 44/50, Train Loss: 0.7918, Train Acc: 0.9022, Val Loss: 0.8679, Val Acc: 0.8485
Epoch 45/50, Train Loss: 0.7894, Train Acc: 0.9034, Val Loss: 0.8685, Val Acc: 0.8512
Epoch 46/50, Train Loss: 0.7891, Train Acc: 0.9032, Val Loss: 0.8660, Val Acc: 0.8483
Epoch 47/50, Train Loss: 0.7885, Train Acc: 0.9029, Val Loss: 0.8643, Val Acc: 0.8512
Epoch 48/50, Train Loss: 0.7875, Train Acc: 0.9051, Val Loss: 0.8595, Val Acc: 0.8502
Epoch 49/50, Train Loss: 0.7835, Train Acc: 0.9054, Val Loss: 0.8615, Val Acc: 0.8553
Epoch 50/50, Train Loss: 0.7855, Train Acc: 0.9045, Val Loss: 0.8657, Val Acc: 0.8525

Training and Validation Loss





秒



```
accuracy_score(test_labels, predictions)
```



0.8628