Deep-Learning-for-Autonomous-Driving

Lab2 : Image classification

ID : 310605007

Member : 鄭晴立

Department : 機器人學程

2022/7/19

目錄

表目錄

圖目錄

1. Discussion

1.1. Problem of class 0

利用 resnet 的 pretrained model 去測試，發現第 0 類是模型分不出來的關鍵，因此決定採用兩次驗證的方式，先區分第 0 類，再區分其他類。

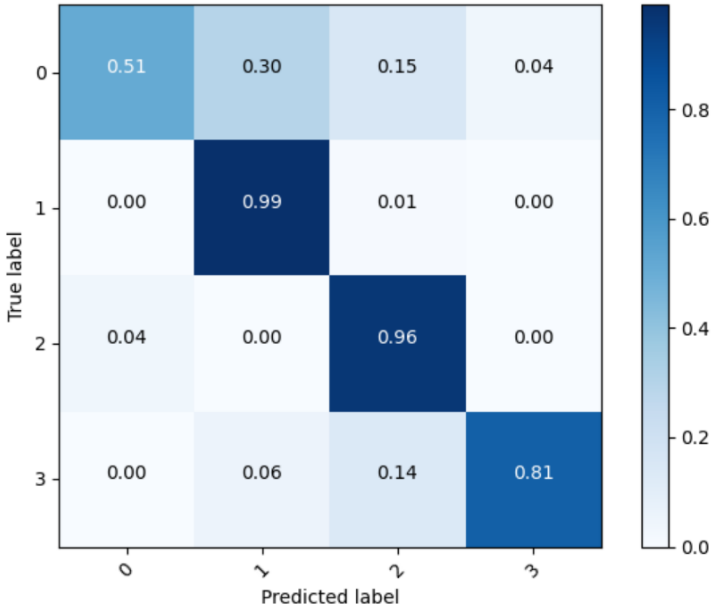

圖 一 第 0 類不易區分

1.2. Data augmentation

經觀察，水平、垂直翻轉及裁切皆能造出可用資料，而旋轉則須控制在 10 度以內，避免訓練多餘特徵。

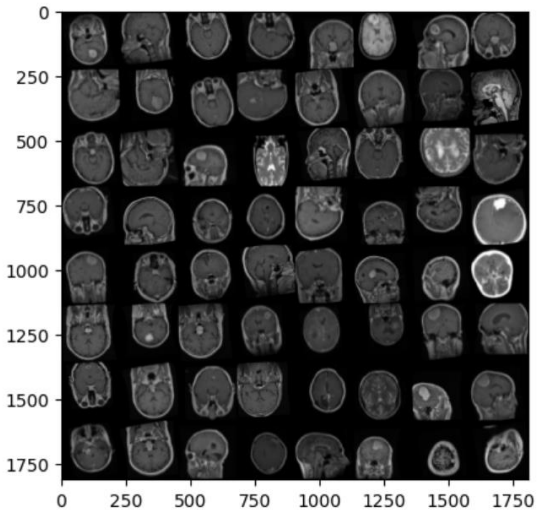

圖 二 Data augmentation

1.3. 重複的 Data

Testing data 及 Training data 有過多重複資料，以至於會有過多重複的訓練，且上傳 kaggle 會導致 testing accuracy 不一定能反映模型表現，例如，模型剛好對於重複的圖片有良好的辨識度，那這樣模型就會被高估，Testing data 與助教反映後，已經處理了，Training data 影響不大，但我還是把重複的圖片刪掉。

表 一 重複 Data 刪除前後數量

	Training Data				Testing Data
類別	glioma_tumor	meningioma_tumor	no_tumor	pituitary_tumor	All
刪除前	826	822	395	827	394
刪除後	818	814	327	810	314
刪除數	8	8	68	17	80

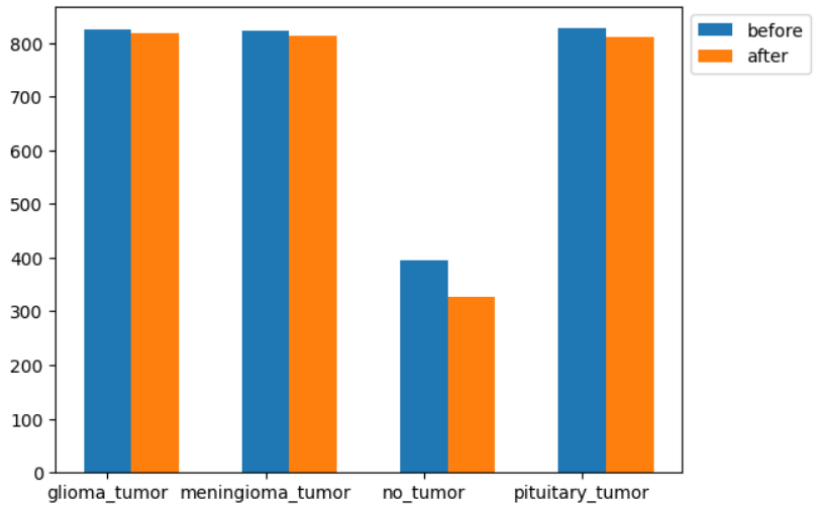

圖 三 重複 Data 刪除前後數量長條圖

1.4. Testing 與 training data 特色差異過大

由於 Training accuracy 及 validation accuracy 皆無法反映最終上傳至 kaggle 的 testing accuracy，其差異會到 10%以上。

2. Task1

由於 Task1 僅要求 Validation 的 Accuracy 因此問題不大，並不需要太多 epoch 就可以達到 Accuracy 要求。

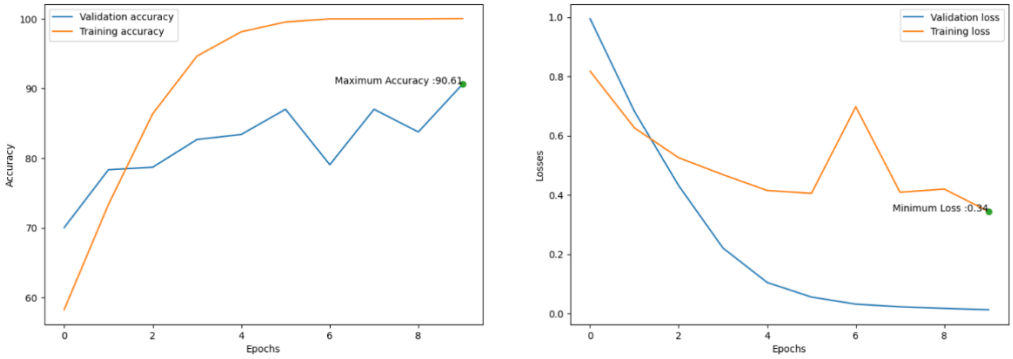

圖 四 Task 1 Accuracy 及 Loss

2.1. hyper parameter

表 二 Task 1 hyper parameter

Parameter	Neural Network
Net	ResNet18
Optimizer	Adam
Learning rate	0.00001
Epoch	10
Loss function	Cross Entropy Loss
Batch size	32

3. Task2

3.1. Learning rate schedule

由於 model 一開始不穩定，因此用較小的 Learning rate，隨著 loss 變動較小時，開始以較大的 Learning rate，到後期逐漸收斂時在小的 Learning rate 去靠近。

```
In [8]: def warmup_learning_rate(epoch):
            if epoch<15:
                return 0.00001 + (0.00003- 0.00001)*epoch/15
            elif epoch<40:
                return 0.00003
            else:
                return 0.00001

In [9]: lr=[]
        for i in range(50):
            lr.append(warmup_learning_rate(i))
        plt.figure(figsize=(18,6))
        plt.plot(lr)

Out[9]: [<matplotlib.lines.Line2D at 0x20ff2d98408>]
```

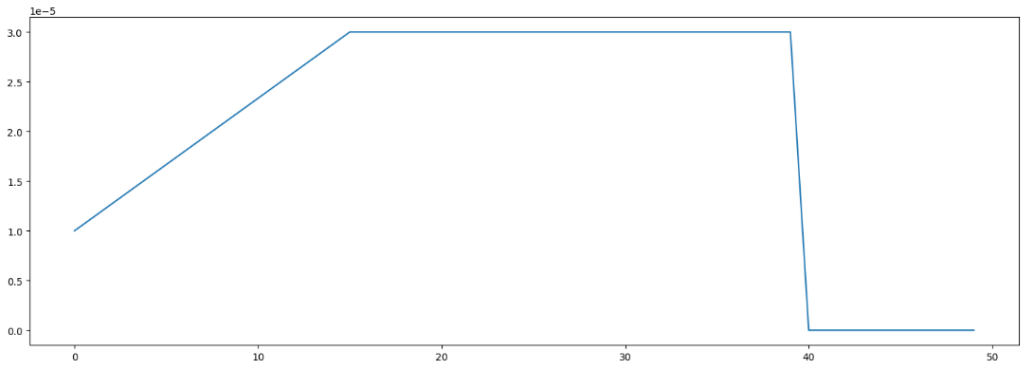

圖 五 Learning rate schedule

3.2. Hyper parameter

表 三 Task 2 hyper parameter

Parameter	Neural Network
Optimizer	Adam
Learning rate	0.00001
Epoch	500
Loss function	Cross Entropy Loss
Batch size	64

3.3. Net

參考網路上的資料，決定用 VGG Net 去嘗試，並調整 Dropout，必避免 overfitting。

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model_vgg = models.vgg16(pretrained=True)

## Modify last layer
num_ftrs = model_vgg.classifier[6].in_features
model_vgg.classifier[6] = nn.Linear(num_ftrs,4)
model_vgg.classifier[5] = nn.Dropout(p=0.7, inplace=False)
model_vgg.classifier[2] = nn.Dropout(p=0.7, inplace=False)
model_vgg
```

圖 六 model implement

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.6, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.6, inplace=False)
    (6): Linear(in_features=4096, out_features=4, bias=True)
  )
)
```

圖 七 model

3.4. Visualize feature maps

Pretrained model 的 feature map 清晰許多，其分類能力也較佳，不同尺度的 feature map 所抓取的特徵不一定能夠用普通視覺分辨，但明顯能看出不同尺度的作用。

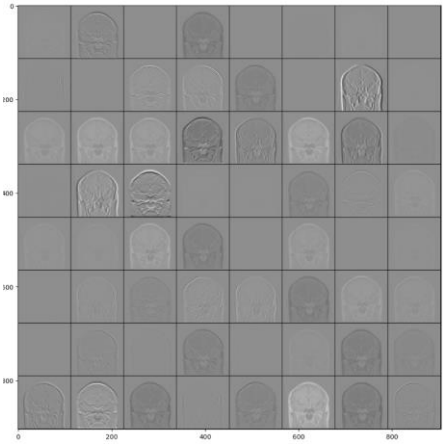

圖 八 pretrained model 的 feature map

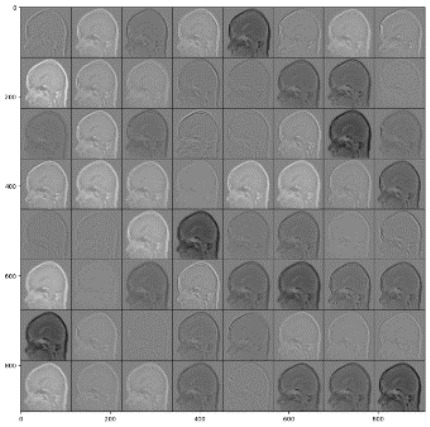

圖 九 non- pretrained model 的 feature map

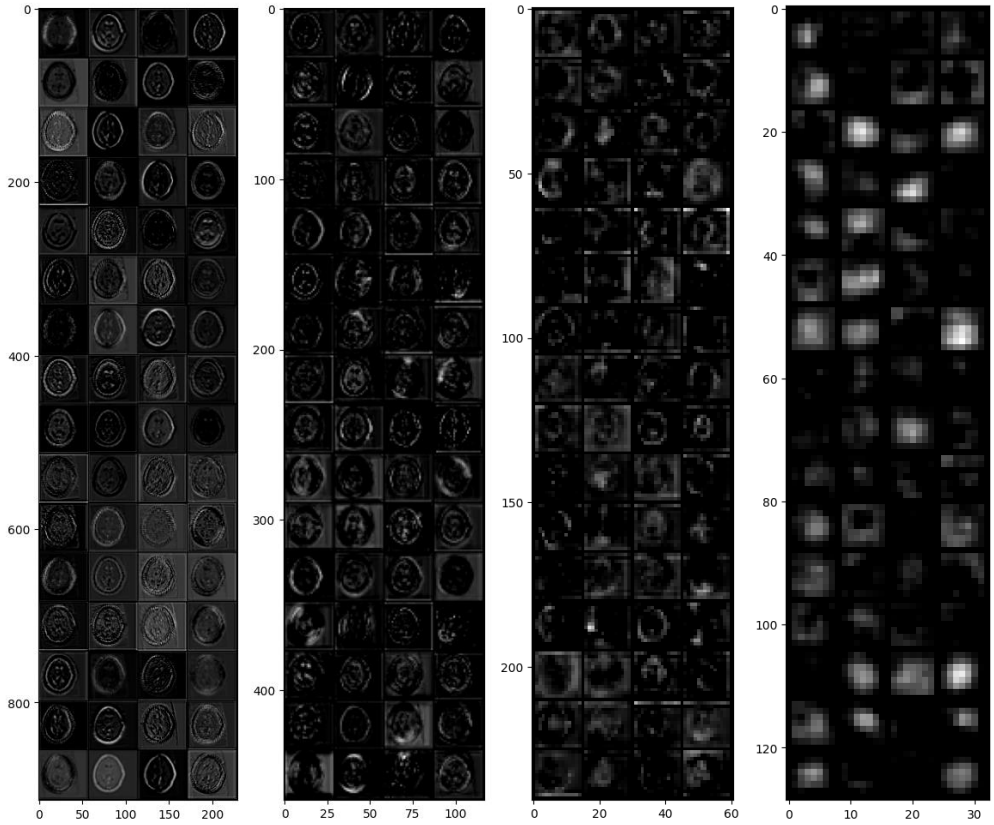

圖 十 不同 layer 的 feature map

3.5. Comparison with the pretrained model

Pretrained model 能使 accuracy 快速提升並收斂。

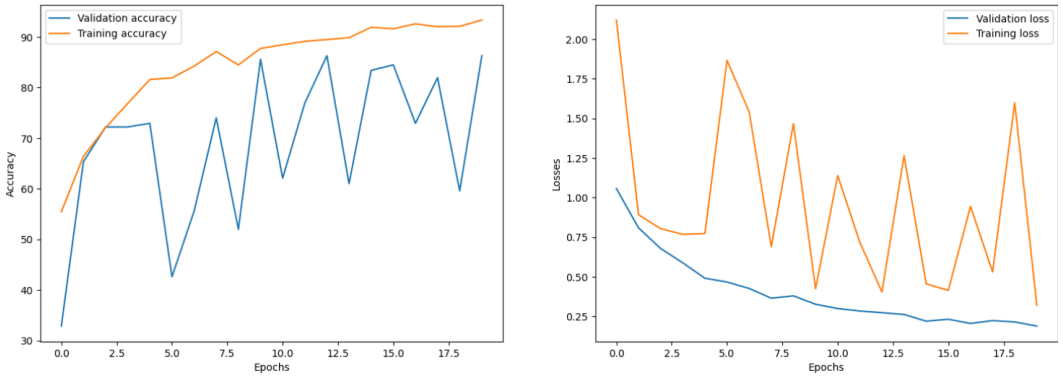

圖 十一 Non-Pretrained model 的 accuracy 和 Loss

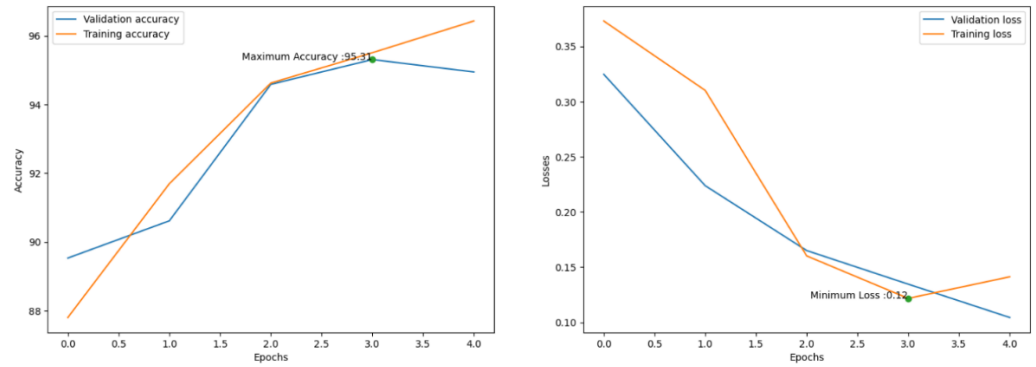

圖 十二 Pretrained model 的 accuracy 和 Loss

3.6. Task2 conclusion

本次的 Validation data 很難 test data 的 accuracy，但肉眼也無法分辨出 data 的問題，因此只能嘗試，唯一能確定是是第 0 類難以分辨，因次，分兩次分類能達到不錯的效果。