National Tsing Hua University

Fall 2023 11210IPT 553000

Deep Learning in Biomedical Optical Imaging

Homework 2

Author One1

1謝瑋哲 清華大學動機系 碩士一年級 新竹 台灣

Student ID:112033645

1. 介紹

本報告是清華大學光電所陳鴻文教授開設的深度學習於生醫光學影像之應用課程，第二次的功課，本功課實驗了在單一資料源訓練次數、模型深度、學習率設定對於最終精度的影響，也將實驗訓練loss function以及驗證資料loss function做成圖表以對比差異。

1. 實驗

以下分別說明實驗設置依序

2.1 訓練次數

首先是訓練次數(epochs)分別為10、20、30、40、50、60，學習率皆為CosineAnnealingLR，最後下降至0，模型架構為簡單16層全連接的神經網路: nn.Flatten(),nn.Linear(256\*256\*1,64),nn.BatchNorm1d(64),nn.ReLU(),nn.Dropout(0.3),nn.Linear(64,64),nn.BatchNorm1d(64),nn.ReLU(),nn.Dropout(0.3),nn.Linear(64,64),nn.BatchNorm1d(64),nn.ReLU(),nn.Dropout(0.3),nn.Linear(64, 1)。

2.2 不同的學習率

接下來根據最好的訓練次數，我使用不同的學習率做比較，使用了StepLR作為比較，其他的參數與2.1相同。

2.3 模型深度

最後更改模型架構，將其變深以探討在更深的模型之下，訓練是否會更好，分別變深到20層與24層。

1. 實驗結果

3.1 本次實驗結果在訓練次數40次，學習率為CosineAnnealingLR，層數深度為16層時效果為最好，影響測試及最終準確率的超參數程度如下: 學習率>訓練次數>深度

1. 實驗圖表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名稱 | 測試結果 | epochs | Learning rate | Model depth |
| test\_10epochs | 75 | 10 | CosineAnnealingLR | 16 |
| test\_20epochs | 69.5 | 20 | CosineAnnealingLR | 16 |
| test\_30epochs | 68 | 30 | CosineAnnealingLR | 16 |
| test\_40epochs | 80 | 40 | CosineAnnealingLR | 16 |
| test\_50epochs | 74.5 | 50 | CosineAnnealingLR | 16 |
| test\_60epochs | 71.5 | 60 | CosineAnnealingLR | 16 |
| 40\_epochs\_lr\_scheduler2 | 75 | 40 | StepLR | 16 |
| bigger\_model\_1 | 73.5 | 40 | StepLR | 20 |
| bigger\_model\_2 | 74.25 | 40 | StepLR | 24 |

Table 1. 實驗結果

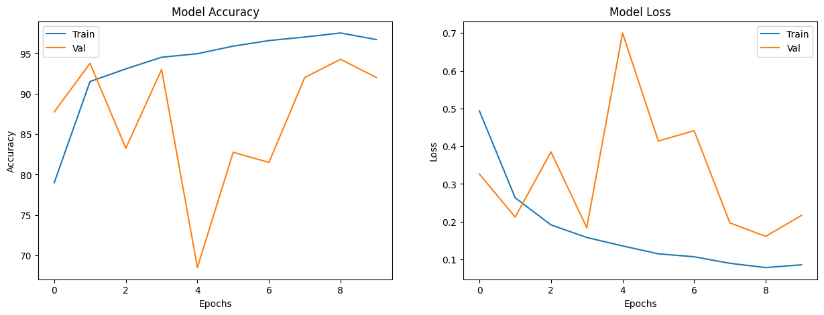


圖1、test\_10epochs loss function

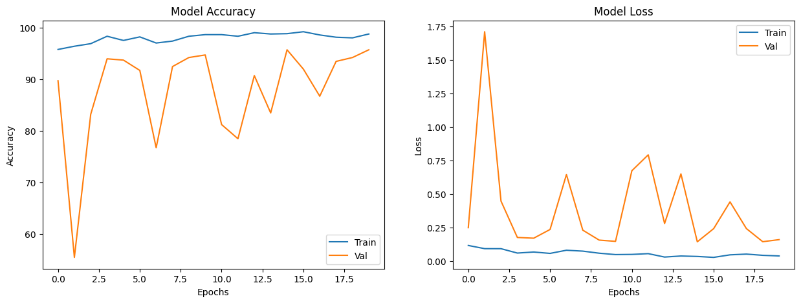


圖2、test\_20epochs loss function

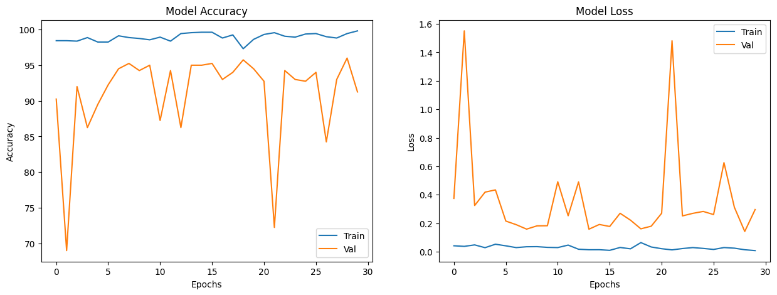


圖3、test\_30epochs loss function

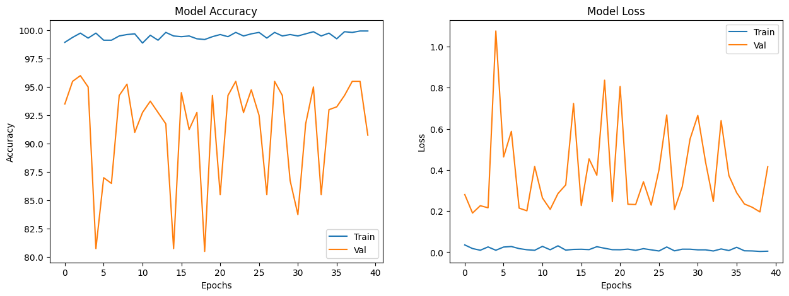


圖4、test\_40epochs loss function

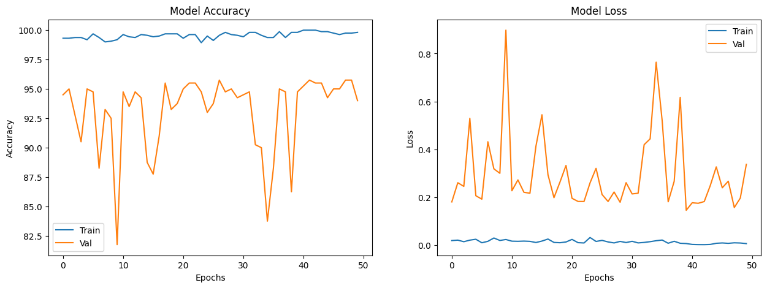


圖5、test\_50epochs loss function

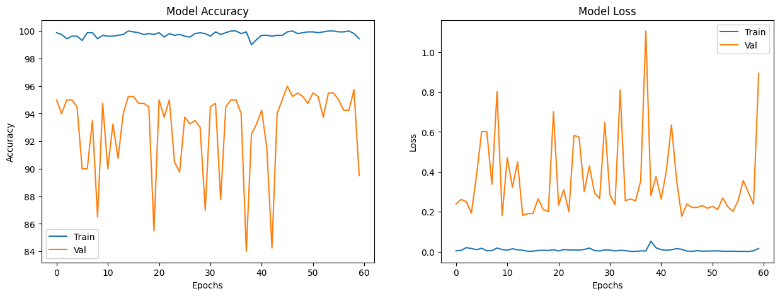


圖6、test\_60epochs loss function

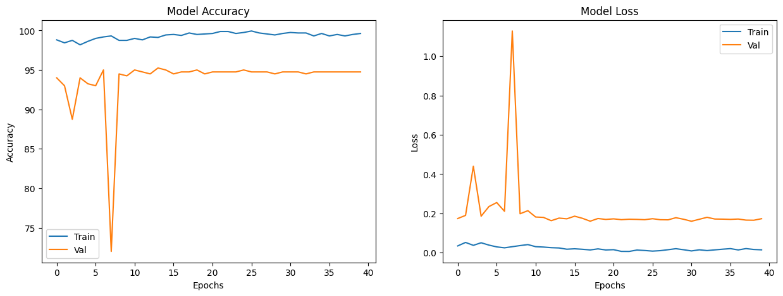


圖7、40\_epochs\_lr\_scheduler2 loss function

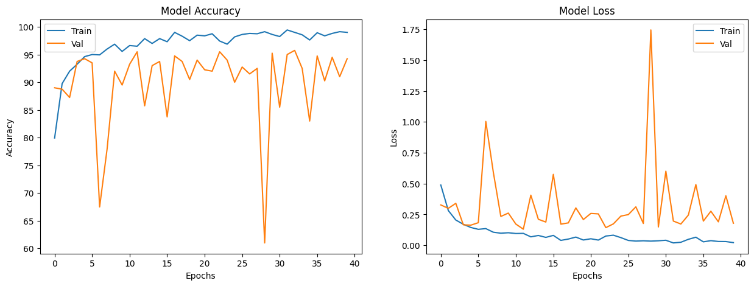


圖8、bigger\_model\_1 loss function

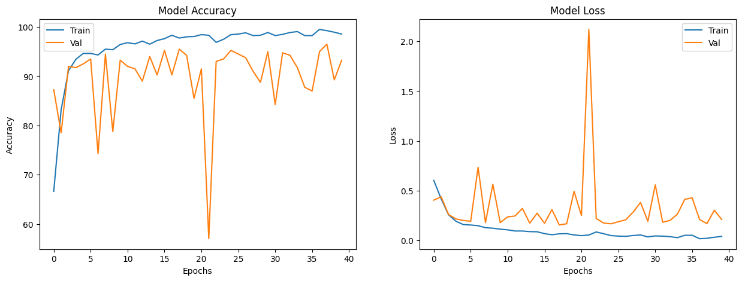


圖9、bigger\_model\_2 loss function

1. 心得

這次的實驗花太多的時間將整體的訓練model以及測試model轉換成class的形式以便於重複使用，而第一個訓練次數的實驗我發現我其實可以將model每訓練10次就儲存對應的權重下來而不是重新訓練，而最佳的形式也可以使用early stop來自動決定次數，而中間因為儲存loss function data格式有些混亂，導致後來讀取多花上不少的時間，如果未來要建立類似的模型，想我也要多加注意。