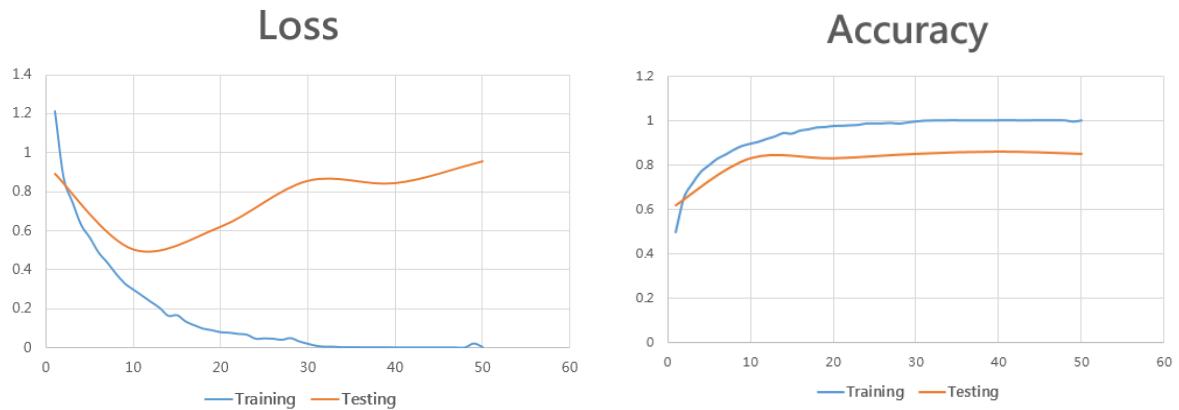


# 機器學習 作業一

405235035 資工三 王博輝

## 1. 給定參數的訓練結果

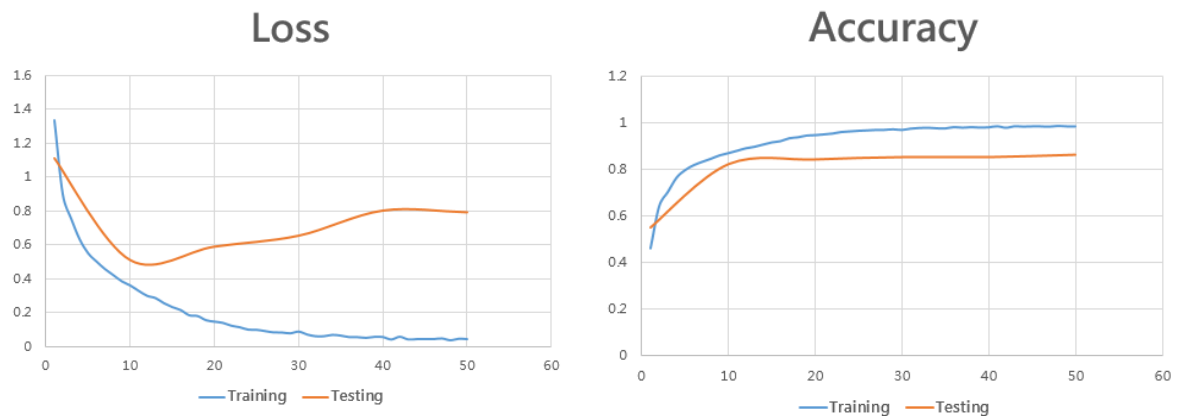
seed = 123, epochs = 50, batch\_size=32, SGD, lr = 0.01



最終結果:

```
Accuracy on the ALL test images: 85 %  
Testing loss: 0.9572  
Accuracy of buildings : 83 %  
Accuracy of forest : 97 %  
Accuracy of glacier : 80 %  
Accuracy of mountain : 80 %  
Accuracy of sea : 88 %  
Accuracy of street : 86 %
```

## 2. 自訂參數的訓練結果



最終結果:

```
Accuracy on the ALL test images: 86 %  
Testing loss: 0.7940  
Accuracy of buildings : 84 %  
Accuracy of forest : 98 %  
Accuracy of glacier : 85 %  
Accuracy of mountain : 80 %  
Accuracy of sea : 84 %  
Accuracy of street : 85 %
```

### 3. 問題與討論

- (1) 我閱讀了 train.py 和 test.py 裡的程式碼，從 train.py 中將能夠計算及印出 loss 和 accuracy 的程式碼加入 test.py 中，並且在 train.py 中利用迴圈直接讓 train.py 能夠在第 1、10、20、30、40、50 個 epoch 將 model 儲存並用使用測試資料來做測試，得到結果之後再利用 Excel 將 loss 和 accuracy 的結果畫出。
- (2) 在老師說明需要更改的五種參數：seed、epochs、batch\_size、SGD、lr，各個參數設定的討論如下。
  - i. 我認為更改 seed 不能夠有效地改善模型的準確度，因為在更改 seed 後，模型的權重仍是亂數產生，在這部分可能需要使用標準差為某值的常態分佈來做為權重的預設值才能夠有效的改善模型準確度，在這其中有一些演算法，如 Xavier 預設值和 He 預設值，搭配上 sigmoid 和 ReLU 等活化函數，據研究顯示能有效地增加模型準確率，所以我在這裡使用的 seed 為隨意設定的值。
  - ii. Epochs 為模型訓練的次數，我認為這次作業所使用的資料，做 50 次的 epoch 已相當足夠，所以在我的設定中 epoch 數並未更改，雖然跑 50 次 epoch 要跑世界久。
  - iii. Batch\_size 仍設定為 32，為了有效利用 GPU 的平行運算，原本我嘗試的將 batch\_size 改為 48，但發現顯示卡的記憶體使用 32 的 batch\_size 已是極限，所以並未向上或向下調整。
  - iv. SGD 為更新參數的最佳化演算法，在這我根據經驗及一些我所看到的研究論文，將 SGD 改為使用 Adam 的方法來最更新權重。
  - v. Lr 為更新參數的學習率，在我把更新參數的演算法改成 Adam 後，我利用 10 的冪次來找最佳的學習率，嘗試過使用 0.01、0.001、0.0001、0.00001 的 lr，發現除了 0.0001 和 0.00001 以外，模型的訓練結果都崩潰了，所以我再從 0.0001 的附近開始找最佳的 lr，因為訓練速度實在太慢，最後只能找到  $lr=0.0002$  來作為我的參數設定。

- (3) 總結來說，我認為以上的五種參數，能夠對模型的準確度產生較大影響的應該只有參數最佳化的方法(SGD、Adam)和 learning rate 的大小，所以我在對這次作業做測試時較著重於這兩個參數上。另外我發現訓練資料和測試資料之間的 loss 和 accuracy 相差許多，應該是發生了 over fitting，我也嘗試過修改 VGG-16 的 model，在其中加入了 Drop out 和 Batch normalization，但測試出來的結果並未改善，訓練準確度仍然可高達 96%、97%，但測試資料卻卡在 86%、87%上不去，詳細原因不明，希望助教能夠幫我解惑。
- (4) 在做這次作業的過程中，我實實在在地了解到機器學習為何近幾年才興起，因為機器學習是一個非常吃重電腦運算資源的技術，光是這次的作業，修改完參數，再跑完完整的一次 50 個 epoch 就要花上 24 小時的時間，而且這是使用 1080Ti+12GB 記憶體跑出的數據，如果是使用一般的筆記型電腦或比較低階的桌上型電腦，一定會跑得更痛苦。另外也發現 batch\_size 的設定，如果是使用 1050+2GB 記憶體的話，連 batch\_size=4 的設定都無法運作，起因似乎是因為顯示卡記憶體的消耗太大，換了顯示卡後，觀察發現 batch\_size=32 大約就要花掉 8GB 的記憶體，這也難怪 2GB 的記憶體會這麼不夠用了。最後還是想知道為甚麼這次作業的 over fitting 需要如何解決，VGG-16 應該是公信力相當夠的模型，為什麼在訓練這次的場景辨識時，會有無法有效提升準確度的問題。有可能是因為 epoch 太大的原因嗎？下次有機會的話，想嘗試看看每個 epoch 都做一次測試，來判斷是否是因為訓練太多才造成 over fitting。