DEEP LEARNING assignment 2

Linear Classifer

黃子軒  
 *Institute of Data Science*  
 *National Cheng Kung University*Tainan, Taiwan  
RE6111058@gs.ncku.edu.tw

# Introduction

本次作業內容為(I) 一個適用於變量輸入通道的卷積模塊(II)設計一個有2~4層的CNN神經網絡模型並做圖片分類，此次的資料集為mini-ImageNet。

# Data

本次作業使用的資料為圖像資料，訓練集中的資料量為63325筆資料，共50個類別，而測試集資料為450筆，共50個類別，而驗證集為450筆。共50個類別，整理結果如下表所示。

|  |  |  |
| --- | --- | --- |
| 資料集名稱 | 總影像數 | 包含類別數 |
| Train | 63325 | 50 |
| Test | 450 | 50 |
| Validation | 450 | 50 |

# Designing a Convolution Module for Variable Input Channels

第一個作業為設計一個首先適用於變量輸入通道的卷積模塊，我們需要先設計一個類似於convolution2d的layer，使其可以被我們放在CNN模型中的第一層，並連接後面的layer。因此，我選擇使用最基本的作法，我設計的layer只擁有最基本的功能，像是設定kernel size，in\out channel，並且能夠做forward不然無法計算loss。

# Designing a Two-Layer Network for Image Classification

為了達成與ResNet34差不多表現的模型，我設計了一個有4層CNN的模型，其中引入了residual block，用以提升模型的表現。定義好了殘差塊後，我的模型四層layer第一層為卷積層，使用batch normal做歸一化，再來是使用ReLU激活，做後是maxpooling，下一層為殘差層，共有三個layer，第一個layer為3個殘差塊，第二個為4個殘差塊，第三個為6個殘差塊下一層為全局平均池化層以及全連接層。

# Model Performance(I)

針對第一部分的模型，配適好後的結過如下表

|  |  |  |
| --- | --- | --- |
| Model | Test Accuracy | Validation  Accuracy |
| 自製CNN | 0.02 | 0.02 |

會導致這個原因的發生第一個可能是因為我的epoch只有設定成10，如果多跑一點的話效果應該會更好，第二點是因為我設計的部分並沒有放入太多的layer進去，因此在特徵提取的部分沒有做得很好，在沒有特別多有用特徵的情況下，模型的表現的確會很差，這點可以跟下一題的模型做出比較。

# Model Performance(II)

針對第二部分，首先我們先建構了一個標準的ResNet34，接著我們自製一個CNN，但是在包含輸入輸出層後只能有6個layer的前提下，引入了殘差層的構想，希望可以藉此方法提升模型辨識率，藉以提高準確率。

而模型的表現如下:

|  |  |  |
| --- | --- | --- |
| Model | Test  Loss | Test Accuracy  Accuracy |
| 自製CNN | 3.3159 | 0.2111 |
| ResNet34 | 3.5081 | 0.1444 |

可以看到，在同樣的epoch下，引入了殘差層後，模型的正確率相較前一題來說好很多，而自製的CNN模型因為有殘差層的關係，且有做過相關的超參數調整，即便在epoch相同下，其表現也是相對來說不錯的。

# Conclusion

。雖然有引入了殘差層的設計，但是在只能夠使用最多四層的其他layer的情況下，我無法將我的模型準確率提升到超過50%的正確率，歸咎下來，我認為原因出在我對於image的處理，除了基本的超參數外，我並沒有特別針對激活函數還有歸一化作調整，但是這些都有可能是讓正確率提高的方法，雖然可能沒辦法一舉突破50%，但是突破30%應該不是問題才對。另外一個問題是，有可能原因出自於epoch的設定，畢竟我兩個部份的設計，epoch都只有設定到10而已，而epoch，也的確是非常可能會影響到模型的一項設定，但由於電腦性能的問題，我不得不放棄將epoch調高，因為此前曾經試著想要跑30的epoch，電腦卻因此自動關機，所以我才只能跑最基礎的10而已，希望未來有機會的話，我可以再向著個設計挑戰看看，期許自己寫的模型可以創造更好的表現。

# GitHub Link

https://github.com/H24061024/DL\_HW\_1