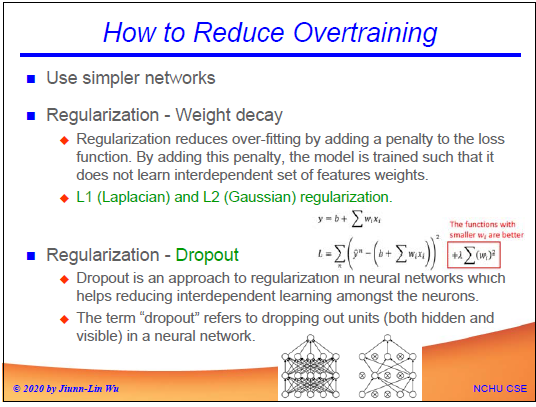
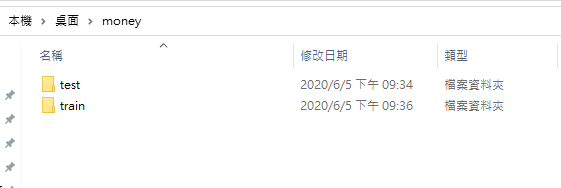
7108056022 陳冠霖

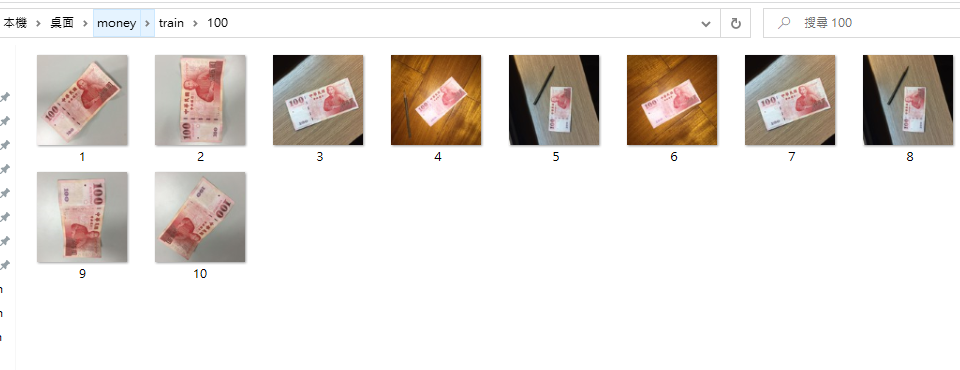
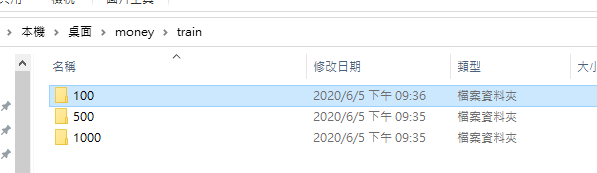
作法:



第一步:

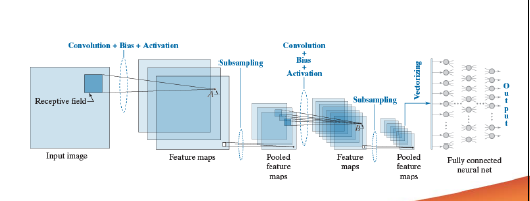
將一開始拿到的資料包含100、500、1000元的影像總共60張，劃分為2個資料夾分別代表訓練資料(train)和測試資料(test)，裡面每個資料夾都各10筆資料，並在這2個資料夾底下再分為3個資料夾，資料夾名稱分別為各個圖片類別的名稱分別為100、500、1000。





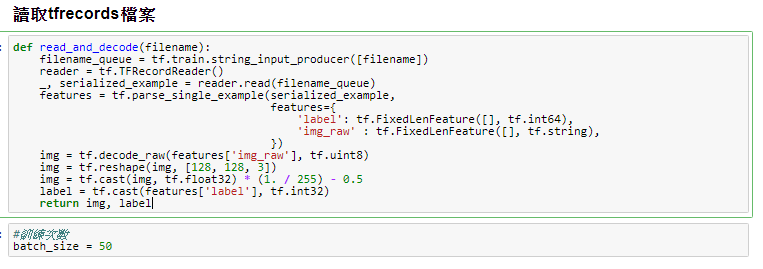
第二步:

建立起CNN(卷積神經網路（Convolutional Neural Network)）的架構，將準備好的資料製作tfrecords檔案，tfrecord會根據你選擇輸入檔案的類，自動給每一類打上同樣的標籤，接著讀取tfrecords檔案，將圖片和標籤讀出，使用卷積神經網路訓練，定義好卷積神經網路的結構，卷機層、池化層、全連線層……等，進行訓練得到學習率。

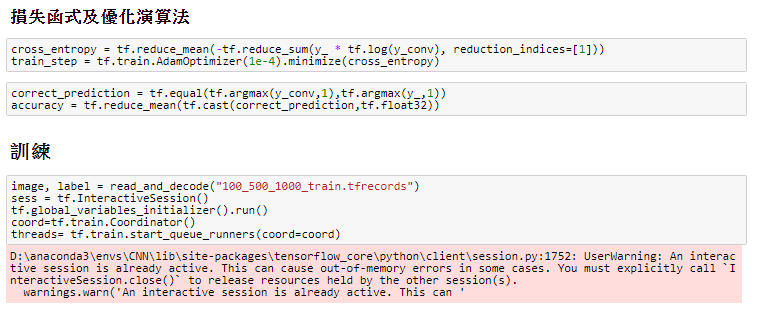


程式(CNN架構):







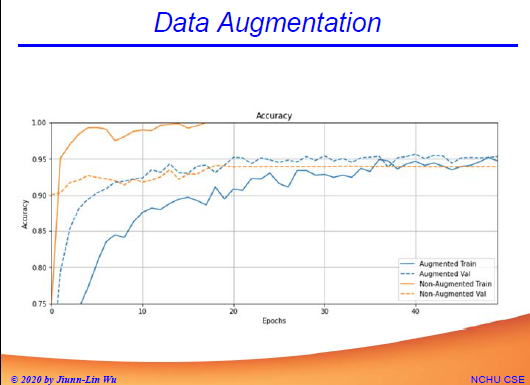


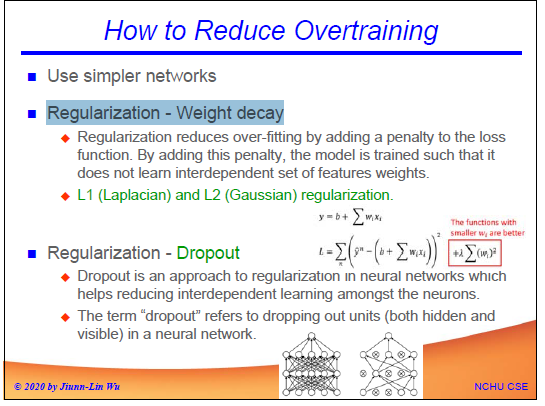
第三部:

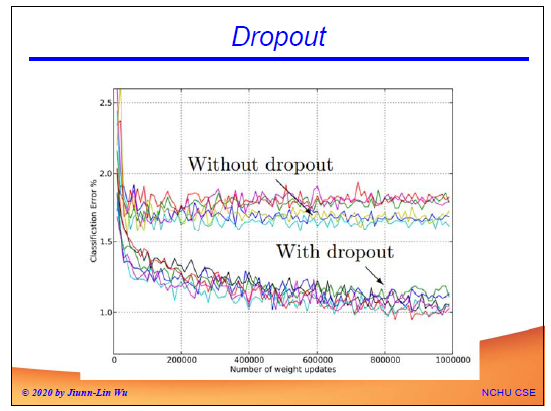
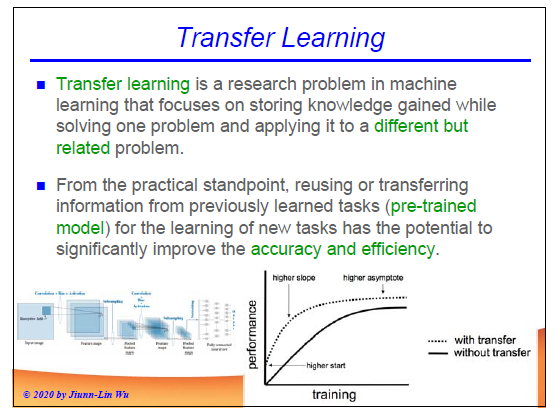
由於訓練資料少，可能造成overtraining導致學習率非常低(0.2)，所以利用課本中提到的4種方法，來減少Overtraining發生的機率，分別為Data Augmentation、Regularization-Weight decay、Regularization-Dropout、Transfer Learning。

這個部分我使用Data Augmentation和Regularization-Dropout，來避免 overfitting。



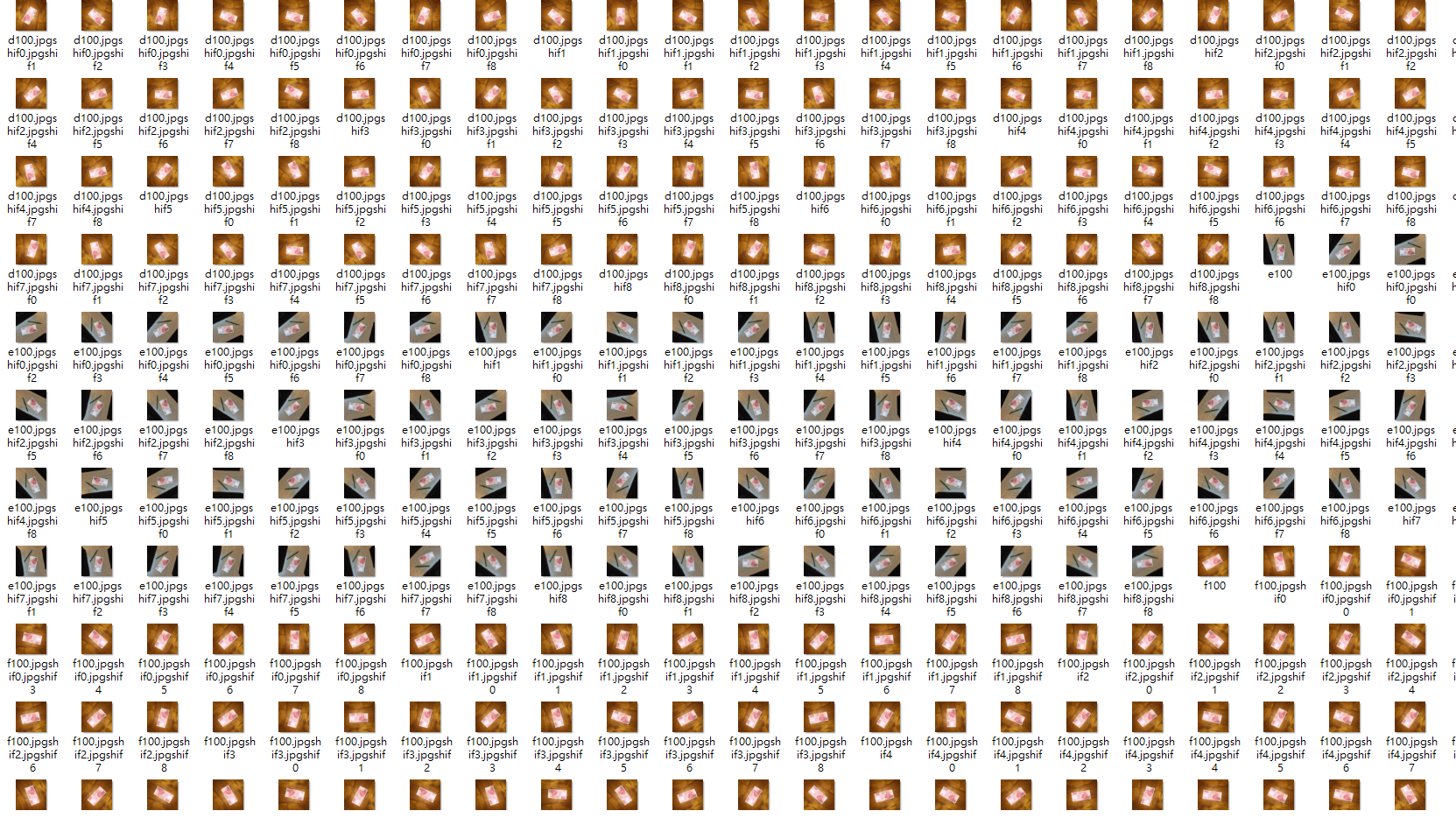




Data Augmentation**:**

一開始訓練的資料只有30筆(每個資料夾10筆)，所以利用Data Augmentation，將原來擁有的30張影像變成2730張影像(每個資料夾910張影像)，利用將原本的30張影像做旋轉隨機生成出新的影像，這裡我做了2次Data Augmentation，也就是每個資料夾從10變成100張接著再做一次變成910張。



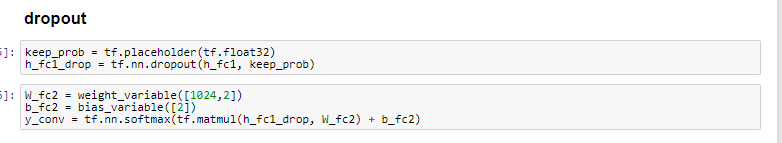
程式(Data Augmentation):

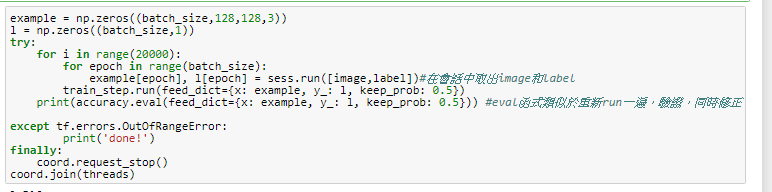


Regularization-Dropout:

這個方式就是在訓練的過程中，減少神經元的數量，制定一個門檻值，刪去一些神經元，避免overfitting，是神經網絡正規化的一種方法，有助於減少神經元之間的相互依賴學習，dropout的意思就是在神經網路中是藏起來卻可見。

程式(Regularization-Dropout)





結論:

由於設備問題，無法達到太高的訓練次數，礙於本身的電腦的顯示卡不提供Cuda，所以無法用GPU來訓練資料，只能使用CPU，但CPU訓練資料的速度慢，導致無法將訓練次數拉高，所以在寫作業時，我的訓練次數都是控制在500次，在一開始沒有使用Data Augmentation和Dropout的狀況下，訓練得出來的學習率非常的浮動最低有0.2而最高有0.6，在經過Data Augmentation和Dropout後，訓練得出的學習率相對穩定都有0.5以上，但是由於設備問題訓練次數無法使用太高，導致學習率還是不高，原因在於訓練次數太低。

