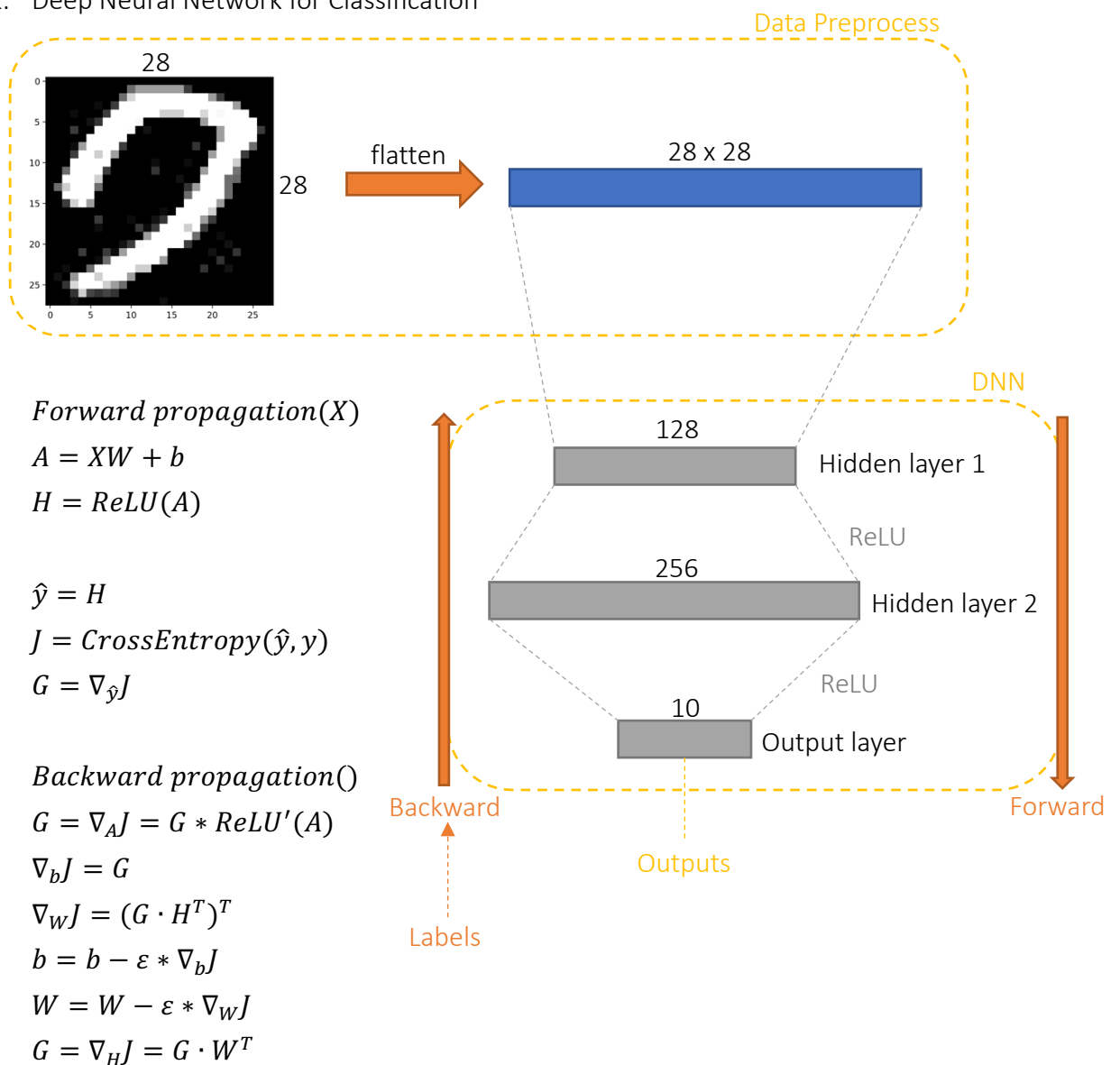
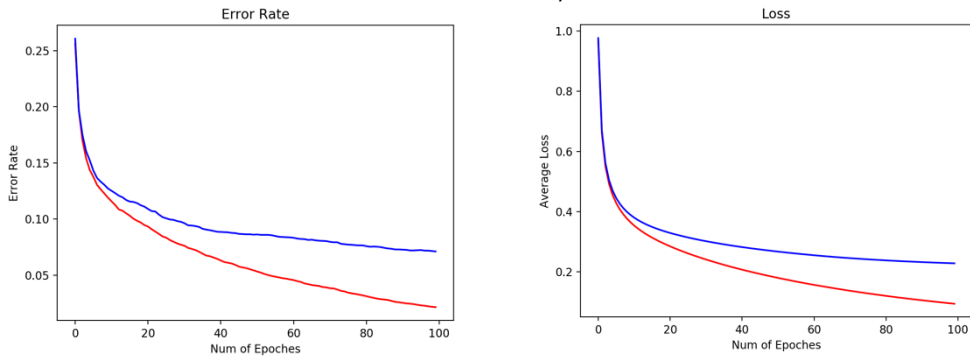


1. Deep Neural Network for Classification

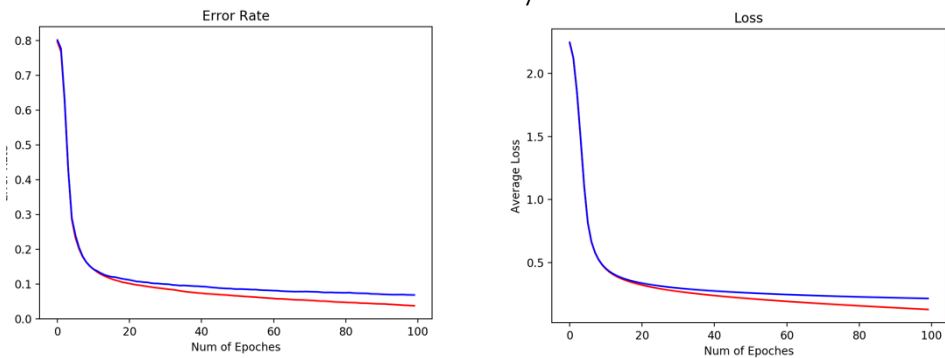


- a. 利用以上模型，並且以 random initialization 開始，分別計算訓練集(紅)和測試集(藍)每一 epoch 的 error rate 和 loss。可以發現訓練集下降較快，可以猜測若繼續訓練可能會 overfitting。我再利用上述模型並且再加上一層隱藏層，一樣計算 error 和 loss，可以發現模型越複雜下降越快速，但是也不能讓模型過度複雜，否則可能會難以收斂。

2 hidden layers



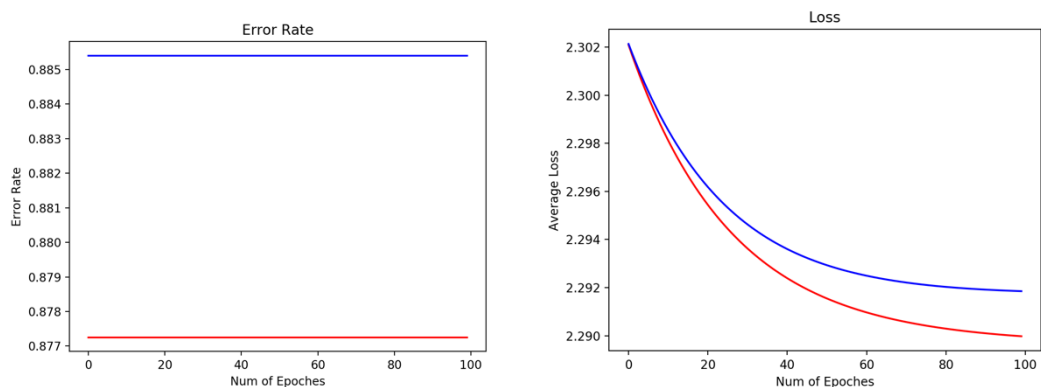
3 hidden layers



- b. 比較 Zero initialization 和 Random Initialization。一樣利用上述的模型，但是改成 zero initialization 來實作。可以發現 zero initialization 計算的 error rate 維持不變，那是因為 $W = 0, b = 0$

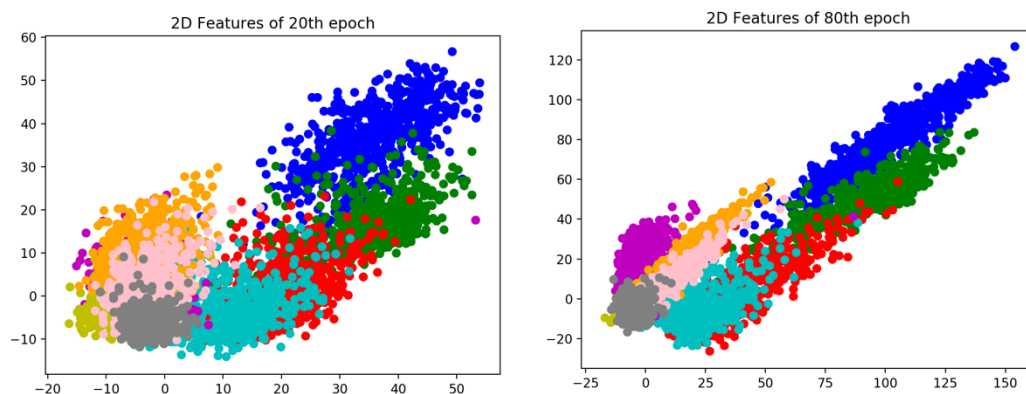
$$A = XW + b = 0$$

並且通過 ReLU 再微分的話，gradient 也會為 0，權重就不會改變。所以要避免使用 zero initialization。

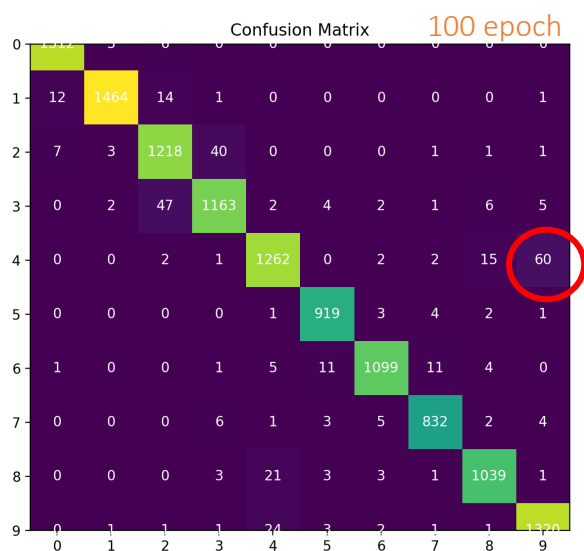
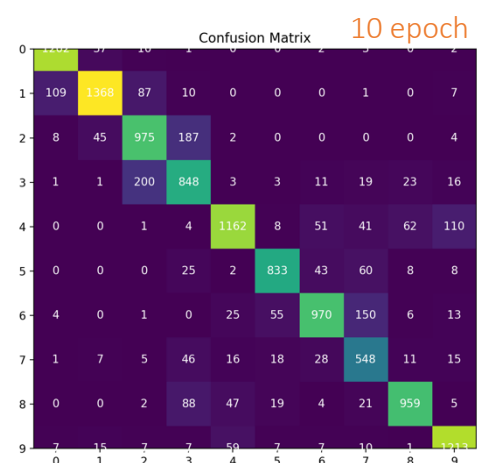
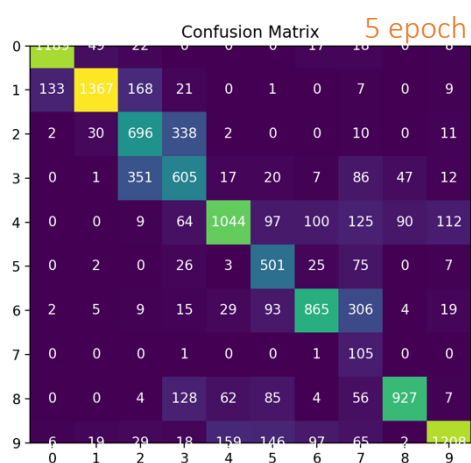


- c. 一樣利用上述模型，但在 output layer 前加上一層只有 2 個 nodes 的 hidden layer，觀察測試集分別在訓練 20 個 epoch 後和訓練 80 個 epoch 之後在這 2 nodes 上的分佈。可以發現訓練越多次，模型會讓相同 label 的資料更集中，更聚集在一起，資料分布更為確切，可以讓在做分類時

更容易找出分界線。

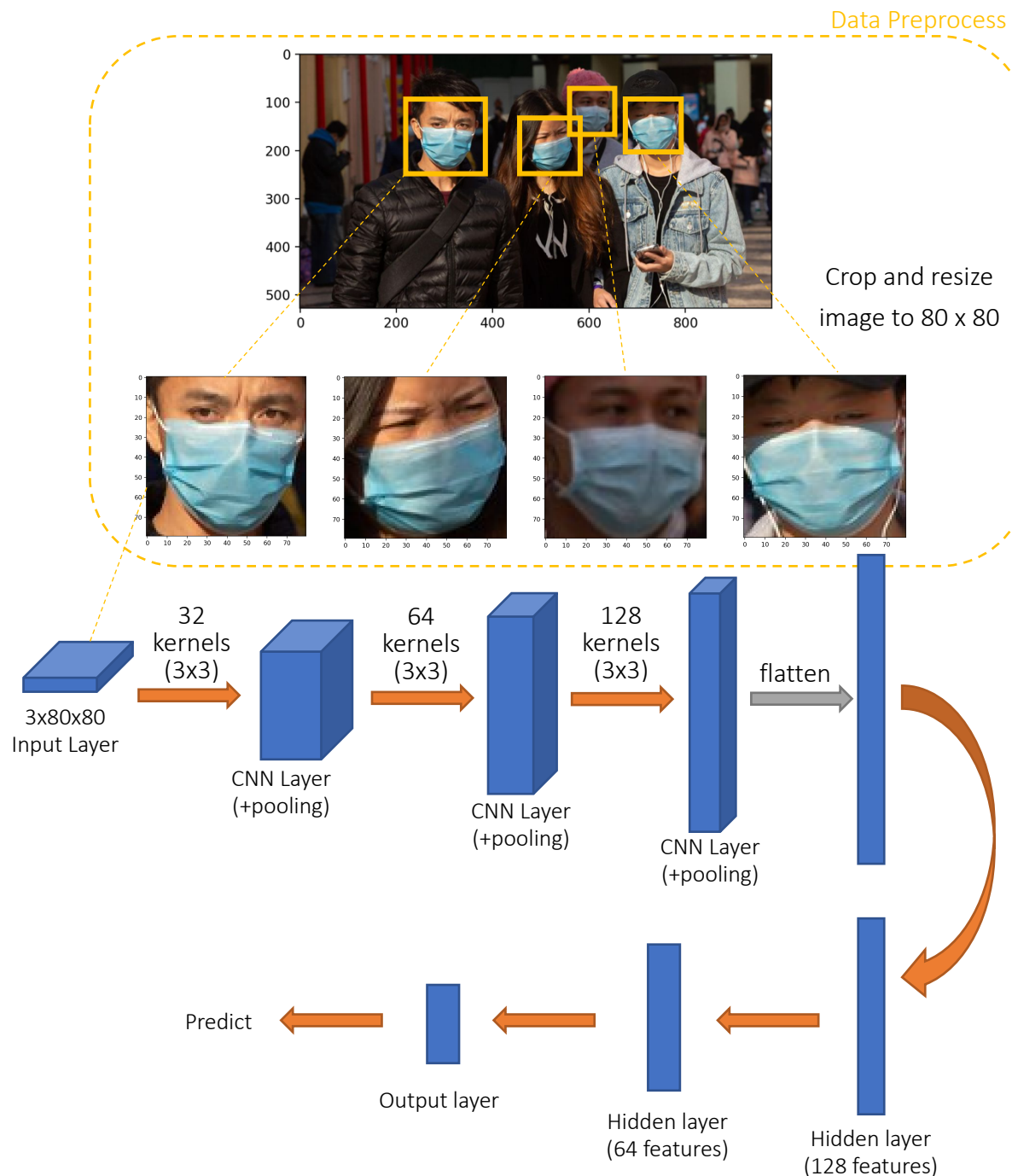


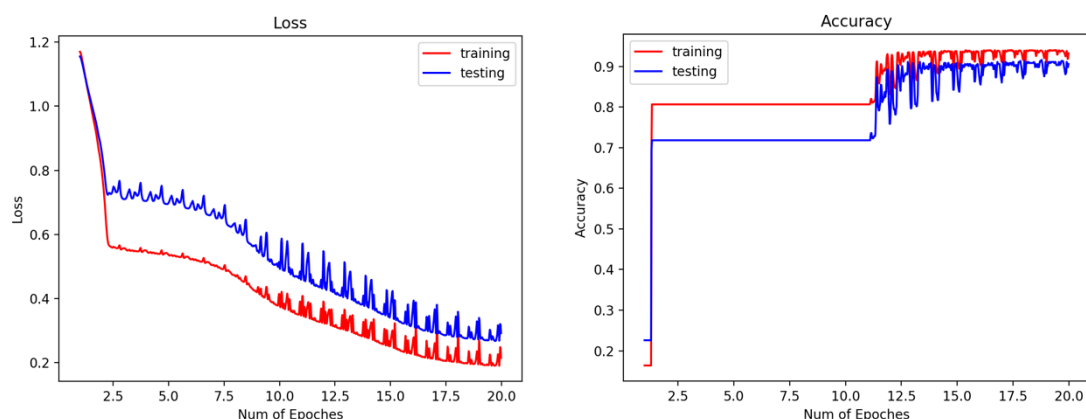
- d. Confusion Matrix。從 5 epoch 到 10 epoch 到 100 epoch 的結果可以看出訓練得越多，訓練集的結果會越趨於正確，集中於對角線。也可以發現在預測 9 時，得到錯誤的 outputs 多集中在預測成是 4。



2. Convolutional Neural Network for Image Recognition

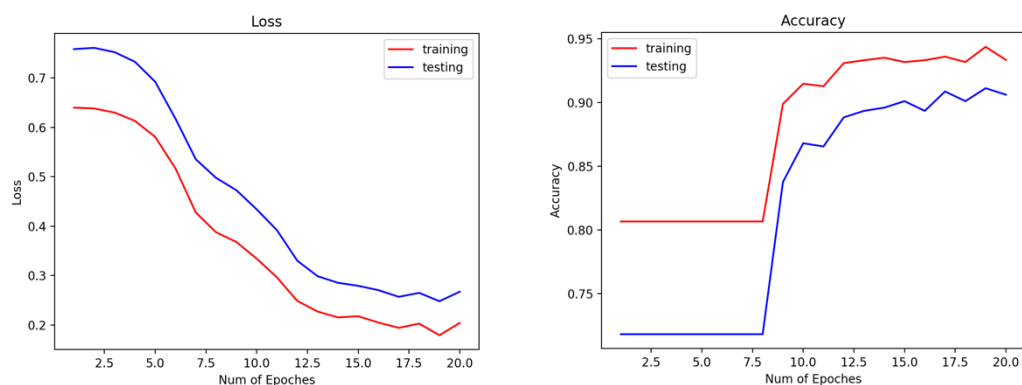
先將照片中人臉擷取出來，並全部 resize 到 80x80 固定的大小，方便之後模型參數的設定。我設計這次的模型包含 3 層的 convolutional layer 其中 max pooling stride=2 作降維，再接上 3 層的 fully connected layer 並使用 ReLU activation function，最後輸出層使用 softmax activation function，而 cost function 使用的是 cross entropy，使用 stochastic gradient decent 做最佳化。





Class	Train acc	Test acc
good	96.7%	96.1%
none	1.0%	0%
bad	96.5%	96.6%

上述的結果，雖然 accuracy 和 loss 都有正確下降，但因為資料的不平衡，對於 none 的類別正確率只有不到 1%。其中最主要是因為在切 mini batch 時，因為資料的不平衡，可能常常沒切到 none 的類別，使得 mini batch 裡只有 good 和 bad 類別，讓模型沒訓練到 none 類別。所以我重新訓練一個模型，並且在切 mini batch 裡，一定會有一定量的 none 類別，訓練結果 none 類別的正確率大幅提升。



class	Train acc	Test acc
Good	94.2%	92.3%
None	60.6%	59.1%
Bad	95.0%	91.0%