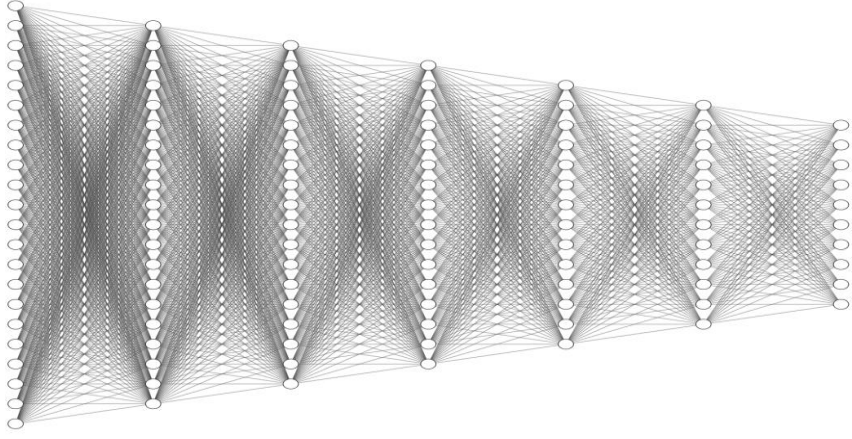
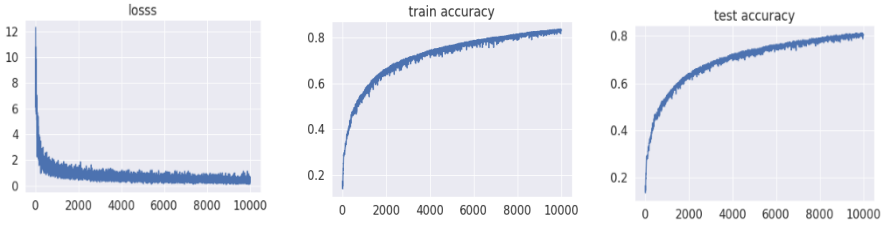
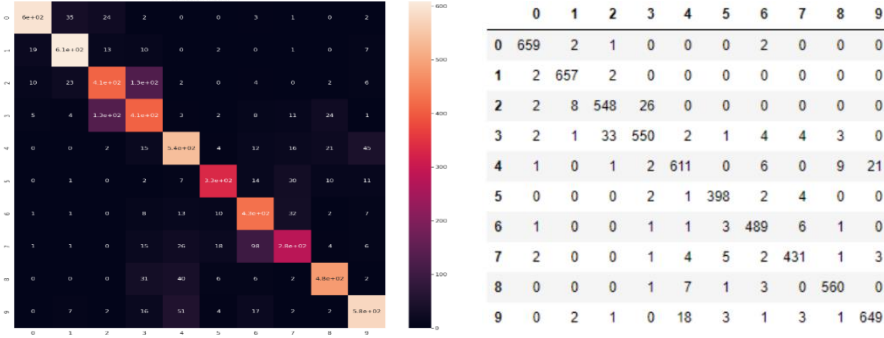


## 1. DNN - MNIST dataset :

### ■ Network Architecture and Performance with Confusion Matrix:

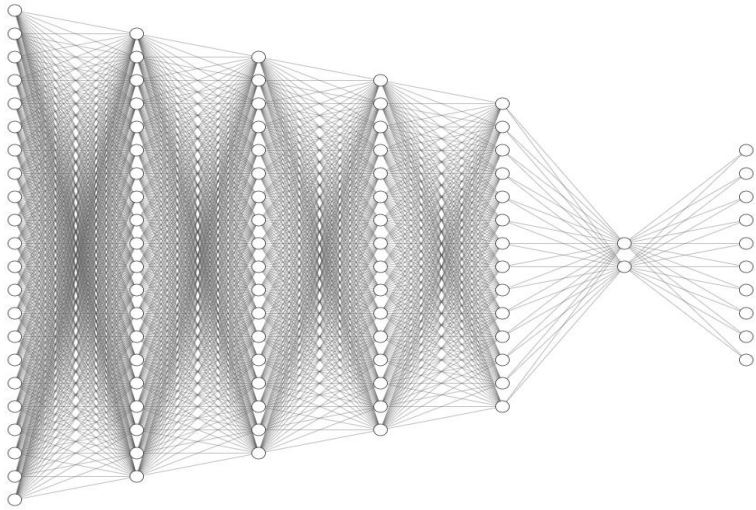
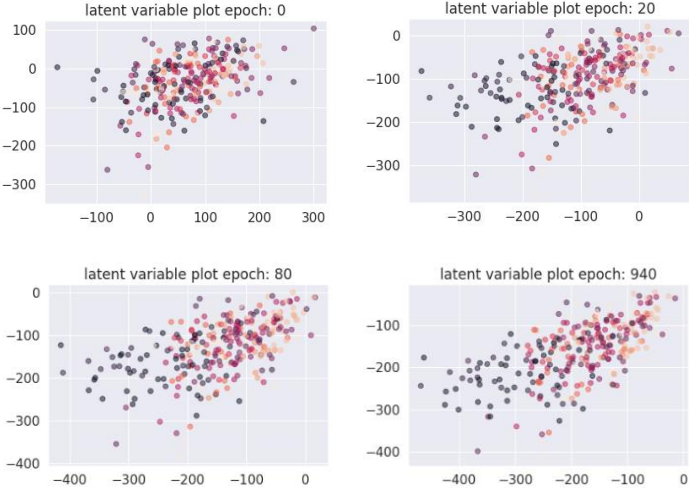
Network Architecture	<p>(784,392,196,98,49,24,10)</p> <p>(未按實際繪製，僅輸出層與層數保持一致)</p> 		
參數	Batch_size = 30	learning_rate = 0.0001	max_epoch = 10000
Performance			
Confusion Matrix			
Analysis	<ol style="list-style-type: none"> <li>由 Loss 與 Training Accuracy、Test Accuracy 可知此模型可保持一定的預測能力，但是模型到高的 Epoch 時皆產生了 Training 的 Variation，此現象可藉由加入一些 Regularization 的手法減緩。</li> <li>由 Confusion Matrix 可觀察到一些有趣的現象，2 與 3 的結果顯示出兩者容易搞混，甚至出現幾乎相同的分布，由助教在講義中提供的結果可以確認此現象。顯示出 2、3 僅有一小撇之分，所以難以區別兩者。而 4 與 9 兩者也十分相似，因此各有近 50 例的混淆。</li> </ol>		

■ Perform Randn and Zero initialization :

因為參數設成 0，造成 Gradient Decent 無法運作，參數不能更新。

```
epoch:0.0000 Loss : 2.3026 training_acc : 0.1073 test_acc : 0.1012
epoch:1.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:2.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:3.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:4.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:5.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:6.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:7.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:8.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:9.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
epoch:10.0000 Loss : 2.3026 training_acc : 0.1227 test_acc : 0.1146
```

■ Evolution Of Latent Feature :

Network Architecture	<p>(784,392,196,98,49,2,10)</p> <p>(未按實際繪製，僅輸出層、層數、倒數第二層保持一致)</p> 			
參數	Batch_size = 3000	learning_rate = 0.001	max_epoch = 1000	Test 的點數 :300
Evolution Of Latent Variable				
Analysis	<p>從 EPOCH = 0 到 EPOCH = 20 最能看得出點數分布的變化，而剩餘的 EPOCH 如 80 與 940 各個往群聚的中心靠近。</p>			

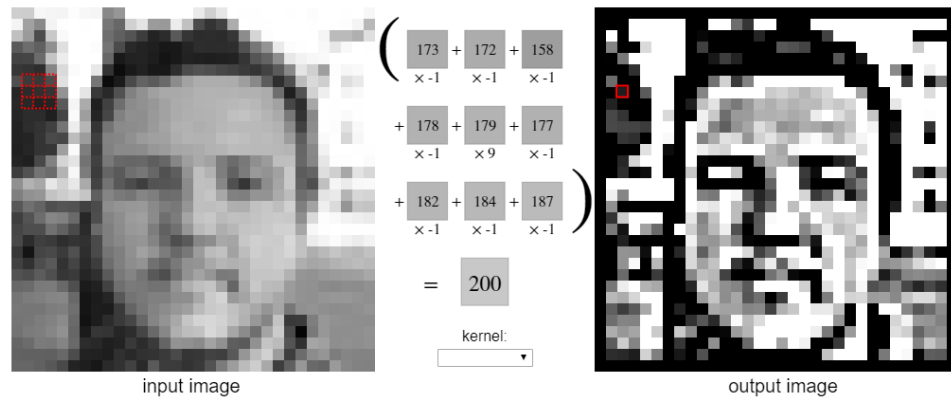
## 2. CNN – Medical Mask Dataset :

### ■ Preprocessing Method : (此段 code 在 CNN\_training.py 內)

#### 1. Crop :

利用 CSV 檔提供的圖形位置將每個頭像的 Frame 抓下來。

#### 2. 先用 Kernel 掃過 : (此 kernel 為 High pass filter , 有銳化效果)

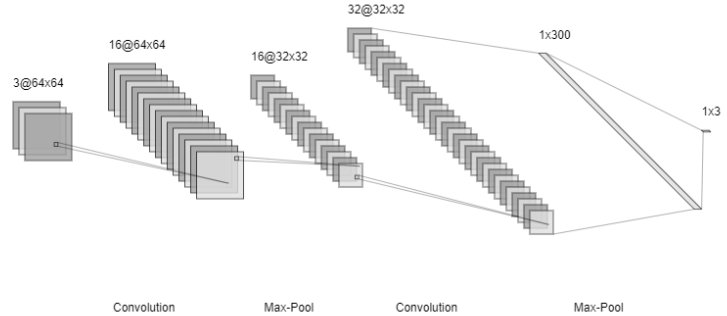
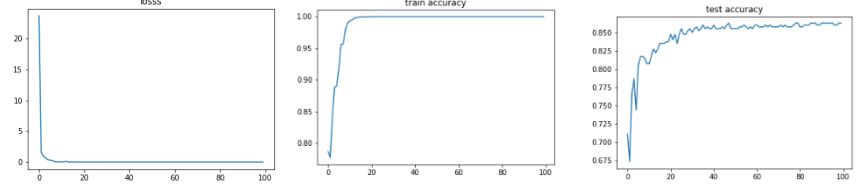


Kernel 的設計如下所示 :

-1	-1	-1
-1	9	-1
-1	-1	-1

#### 3. 再 Resize 成(1,3,64,64)

■ Network Architecture and Performance:

Network Architecture	<p>(因 Linear 層數量過大，實際上應為 8192 而非 300，但難以匯出圖形，僅以 1x300 表示)</p>  <p>Convolution      Max-Pool      Convolution      Max-Pool</p>		
參數	Batch_size = 50	learning_rate = 0.0001	max_epoch = 100
Performance			
模型預測能力	<pre> bad test accuracy: 0.71  bad train accuracy: 1.00  good test accuracy: 0.96  good train accuracy: 1.00  none test accuracy: 0.18  none train accuracy: 1.00</pre> <p>(使用 training_statistic.py 統計的)</p>		
Analysis	<ol style="list-style-type: none"> <li>1. Loss 可幾乎完美降到 0，Training Accuracy 飽和在 1，有點 overfitting，但 Test accuracy 在 0.8 後就上不太去了。</li> <li>2. None 的預測效果最差，原因在於 None 與 Bad 有 Overlapping 的現象，加上 None 的數量過少，所以預測的效果最差。</li> <li>3. 而 Bad 的預測效果比 Good 差的原因推測是訓練母數沒有 Good 多，所以預測的效果沒有 Good 好。</li> </ol>		



■ Show some examples and Result : (使用 label\_image.py 繪製的)

