

- Model 各層 layer 的配置及原因

圖 1 為 baseline model 使用 3 層 CNN 和 3 層 NN 所組成，圖 2 為原本的 baseline model 在前 3 層加入 Batch Normalization layer，圖 3 為原本的 baseline model 移出掉一層 convolution layer，圖 4 為原本的 baseline model 再加上一層 convolution layer。

Baseline_Model ↓ Model: "sequential"	Model_with_BN ↓ Model: "sequential_1"																																																																											
<table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>conv2d (Conv2D)</td><td>(None, 28, 28, 32)</td><td>320</td></tr><tr><td>max_pooling2d (MaxPooling2D)</td><td>(None, 14, 14, 32)</td><td>0</td></tr><tr><td>conv2d_1 (Conv2D)</td><td>(None, 14, 14, 64)</td><td>18496</td></tr><tr><td>max_pooling2d_1 (MaxPooling2D)</td><td>(None, 7, 7, 64)</td><td>0</td></tr><tr><td>conv2d_2 (Conv2D)</td><td>(None, 7, 7, 64)</td><td>36928</td></tr><tr><td>max_pooling2d_2 (MaxPooling2D)</td><td>(None, 3, 3, 64)</td><td>0</td></tr><tr><td>flatten (Flatten)</td><td>(None, 576)</td><td>0</td></tr><tr><td>dense (Dense)</td><td>(None, 64)</td><td>36928</td></tr><tr><td>dense_1 (Dense)</td><td>(None, 64)</td><td>4160</td></tr><tr><td>dense_2 (Dense)</td><td>(None, 10)</td><td>650</td></tr></table> <div>Total params: 97,482 Trainable params: 97,482 Non-trainable params: 0</div>	Layer (type)	Output Shape	Param #	conv2d (Conv2D)	(None, 28, 28, 32)	320	max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0	conv2d_1 (Conv2D)	(None, 14, 14, 64)	18496	max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0	conv2d_2 (Conv2D)	(None, 7, 7, 64)	36928	max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 64)	0	flatten (Flatten)	(None, 576)	0	dense (Dense)	(None, 64)	36928	dense_1 (Dense)	(None, 64)	4160	dense_2 (Dense)	(None, 10)	650	<table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>conv2d_3 (Conv2D)</td><td>(None, 28, 28, 32)</td><td>320</td></tr><tr><td>batch_normalization (Batch Normalization)</td><td>(None, 28, 28, 32)</td><td>128</td></tr><tr><td>max_pooling2d_3 (MaxPooling2D)</td><td>(None, 14, 14, 32)</td><td>0</td></tr><tr><td>conv2d_4 (Conv2D)</td><td>(None, 14, 14, 64)</td><td>18496</td></tr><tr><td>batch_normalization_1 (Batch Normalization)</td><td>(None, 14, 14, 64)</td><td>256</td></tr><tr><td>max_pooling2d_4 (MaxPooling2D)</td><td>(None, 7, 7, 64)</td><td>0</td></tr><tr><td>conv2d_5 (Conv2D)</td><td>(None, 7, 7, 64)</td><td>36928</td></tr><tr><td>batch_normalization_2 (Batch Normalization)</td><td>(None, 7, 7, 64)</td><td>256</td></tr><tr><td>max_pooling2d_5 (MaxPooling2D)</td><td>(None, 3, 3, 64)</td><td>0</td></tr><tr><td>flatten_1 (Flatten)</td><td>(None, 576)</td><td>0</td></tr><tr><td>dense_3 (Dense)</td><td>(None, 64)</td><td>36928</td></tr><tr><td>dense_4 (Dense)</td><td>(None, 64)</td><td>4160</td></tr><tr><td>dense_5 (Dense)</td><td>(None, 10)</td><td>650</td></tr></table> <div>Total params: 98,122 Trainable params: 97,802 Non-trainable params: 320</div>	Layer (type)	Output Shape	Param #	conv2d_3 (Conv2D)	(None, 28, 28, 32)	320	batch_normalization (Batch Normalization)	(None, 28, 28, 32)	128	max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 32)	0	conv2d_4 (Conv2D)	(None, 14, 14, 64)	18496	batch_normalization_1 (Batch Normalization)	(None, 14, 14, 64)	256	max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 64)	0	conv2d_5 (Conv2D)	(None, 7, 7, 64)	36928	batch_normalization_2 (Batch Normalization)	(None, 7, 7, 64)	256	max_pooling2d_5 (MaxPooling2D)	(None, 3, 3, 64)	0	flatten_1 (Flatten)	(None, 576)	0	dense_3 (Dense)	(None, 64)	36928	dense_4 (Dense)	(None, 64)	4160	dense_5 (Dense)	(None, 10)	650
Layer (type)	Output Shape	Param #																																																																										
conv2d (Conv2D)	(None, 28, 28, 32)	320																																																																										
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0																																																																										
conv2d_1 (Conv2D)	(None, 14, 14, 64)	18496																																																																										
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0																																																																										
conv2d_2 (Conv2D)	(None, 7, 7, 64)	36928																																																																										
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 64)	0																																																																										
flatten (Flatten)	(None, 576)	0																																																																										
dense (Dense)	(None, 64)	36928																																																																										
dense_1 (Dense)	(None, 64)	4160																																																																										
dense_2 (Dense)	(None, 10)	650																																																																										
Layer (type)	Output Shape	Param #																																																																										
conv2d_3 (Conv2D)	(None, 28, 28, 32)	320																																																																										
batch_normalization (Batch Normalization)	(None, 28, 28, 32)	128																																																																										
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 32)	0																																																																										
conv2d_4 (Conv2D)	(None, 14, 14, 64)	18496																																																																										
batch_normalization_1 (Batch Normalization)	(None, 14, 14, 64)	256																																																																										
max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 64)	0																																																																										
conv2d_5 (Conv2D)	(None, 7, 7, 64)	36928																																																																										
batch_normalization_2 (Batch Normalization)	(None, 7, 7, 64)	256																																																																										
max_pooling2d_5 (MaxPooling2D)	(None, 3, 3, 64)	0																																																																										
flatten_1 (Flatten)	(None, 576)	0																																																																										
dense_3 (Dense)	(None, 64)	36928																																																																										
dense_4 (Dense)	(None, 64)	4160																																																																										
dense_5 (Dense)	(None, 10)	650																																																																										
圖 1	圖 2																																																																											
Model_with_shallower_layer ↓ Model: "sequential_2"	Model_with_deeper_layer ↓ Model: "sequential_3"																																																																											
<table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>conv2d_6 (Conv2D)</td><td>(None, 28, 28, 32)</td><td>320</td></tr><tr><td>max_pooling2d_6 (MaxPooling2D)</td><td>(None, 14, 14, 32)</td><td>0</td></tr><tr><td>conv2d_7 (Conv2D)</td><td>(None, 14, 14, 64)</td><td>18496</td></tr><tr><td>max_pooling2d_7 (MaxPooling2D)</td><td>(None, 7, 7, 64)</td><td>0</td></tr><tr><td>flatten_2 (Flatten)</td><td>(None, 3136)</td><td>0</td></tr><tr><td>dense_6 (Dense)</td><td>(None, 64)</td><td>200768</td></tr><tr><td>dense_7 (Dense)</td><td>(None, 64)</td><td>4160</td></tr><tr><td>dense_8 (Dense)</td><td>(None, 10)</td><td>650</td></tr></table> <div>Total params: 224,394 Trainable params: 224,394 Non-trainable params: 0</div>	Layer (type)	Output Shape	Param #	conv2d_6 (Conv2D)	(None, 28, 28, 32)	320	max_pooling2d_6 (MaxPooling2D)	(None, 14, 14, 32)	0	conv2d_7 (Conv2D)	(None, 14, 14, 64)	18496	max_pooling2d_7 (MaxPooling2D)	(None, 7, 7, 64)	0	flatten_2 (Flatten)	(None, 3136)	0	dense_6 (Dense)	(None, 64)	200768	dense_7 (Dense)	(None, 64)	4160	dense_8 (Dense)	(None, 10)	650	<table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>conv2d_8 (Conv2D)</td><td>(None, 28, 28, 32)</td><td>320</td></tr><tr><td>max_pooling2d_8 (MaxPooling2D)</td><td>(None, 14, 14, 32)</td><td>0</td></tr><tr><td>conv2d_9 (Conv2D)</td><td>(None, 14, 14, 64)</td><td>18496</td></tr><tr><td>max_pooling2d_9 (MaxPooling2D)</td><td>(None, 7, 7, 64)</td><td>0</td></tr><tr><td>conv2d_10 (Conv2D)</td><td>(None, 7, 7, 64)</td><td>36928</td></tr><tr><td>max_pooling2d_10 (MaxPooling2D)</td><td>(None, 3, 3, 64)</td><td>0</td></tr><tr><td>conv2d_11 (Conv2D)</td><td>(None, 3, 3, 64)</td><td>36928</td></tr><tr><td>max_pooling2d_11 (MaxPooling2D)</td><td>(None, 1, 1, 64)</td><td>0</td></tr><tr><td>flatten_3 (Flatten)</td><td>(None, 64)</td><td>0</td></tr><tr><td>dense_9 (Dense)</td><td>(None, 64)</td><td>4160</td></tr><tr><td>dense_10 (Dense)</td><td>(None, 64)</td><td>4160</td></tr><tr><td>dense_11 (Dense)</td><td>(None, 10)</td><td>650</td></tr></table> <div>Total params: 101,642 Trainable params: 101,642 Non-trainable params: 0</div>	Layer (type)	Output Shape	Param #	conv2d_8 (Conv2D)	(None, 28, 28, 32)	320	max_pooling2d_8 (MaxPooling2D)	(None, 14, 14, 32)	0	conv2d_9 (Conv2D)	(None, 14, 14, 64)	18496	max_pooling2d_9 (MaxPooling2D)	(None, 7, 7, 64)	0	conv2d_10 (Conv2D)	(None, 7, 7, 64)	36928	max_pooling2d_10 (MaxPooling2D)	(None, 3, 3, 64)	0	conv2d_11 (Conv2D)	(None, 3, 3, 64)	36928	max_pooling2d_11 (MaxPooling2D)	(None, 1, 1, 64)	0	flatten_3 (Flatten)	(None, 64)	0	dense_9 (Dense)	(None, 64)	4160	dense_10 (Dense)	(None, 64)	4160	dense_11 (Dense)	(None, 10)	650									
Layer (type)	Output Shape	Param #																																																																										
conv2d_6 (Conv2D)	(None, 28, 28, 32)	320																																																																										
max_pooling2d_6 (MaxPooling2D)	(None, 14, 14, 32)	0																																																																										
conv2d_7 (Conv2D)	(None, 14, 14, 64)	18496																																																																										
max_pooling2d_7 (MaxPooling2D)	(None, 7, 7, 64)	0																																																																										
flatten_2 (Flatten)	(None, 3136)	0																																																																										
dense_6 (Dense)	(None, 64)	200768																																																																										
dense_7 (Dense)	(None, 64)	4160																																																																										
dense_8 (Dense)	(None, 10)	650																																																																										
Layer (type)	Output Shape	Param #																																																																										
conv2d_8 (Conv2D)	(None, 28, 28, 32)	320																																																																										
max_pooling2d_8 (MaxPooling2D)	(None, 14, 14, 32)	0																																																																										
conv2d_9 (Conv2D)	(None, 14, 14, 64)	18496																																																																										
max_pooling2d_9 (MaxPooling2D)	(None, 7, 7, 64)	0																																																																										
conv2d_10 (Conv2D)	(None, 7, 7, 64)	36928																																																																										
max_pooling2d_10 (MaxPooling2D)	(None, 3, 3, 64)	0																																																																										
conv2d_11 (Conv2D)	(None, 3, 3, 64)	36928																																																																										
max_pooling2d_11 (MaxPooling2D)	(None, 1, 1, 64)	0																																																																										
flatten_3 (Flatten)	(None, 64)	0																																																																										
dense_9 (Dense)	(None, 64)	4160																																																																										
dense_10 (Dense)	(None, 64)	4160																																																																										
dense_11 (Dense)	(None, 10)	650																																																																										
圖 3	圖 4																																																																											

- BN layer 的比較，不同層數對於 accuracy 的影響

Compared w/ and w/o Batch Normalization Layer:

Model 的 training、setting set 切成 10:1。

加入 Batch Normalization layer 後，training loss、training accuracy、皆無顯著差異，推測應該是因為加入 Batch Normalization layer 前，training set 的準確度皆已經到達 0.99...的準確度，而 loss 也降至 0.012，已經達到很高的準確度和很低的 loss，所以即使加入 Batch Normalization layer 後表現可以再更進步的空間有限。

而 validation set 的部分，在加入 Batch Normalization layer 後，loss 有些微下降，下降的比例有略多於 training set，而 accuracy 也有略微提升一點，但整體來說差異並不大，應該也是因為在加入 Batch Normalization layer 之前準確度和 loss 接已經有很好的表現。

另外也有發現，在一次 epoch 訓練結束前，可以看到 accuracy 一度接近 0.996~0.998，因此推測如果把參數調整得更好的話，最後的 accuracy 應該可以再更高。

圖 5 為未加入 Batch Normalization layer 之前 baseline model 的結果，圖 6 為加入 Batch Normalization layer 後的結果。

```
Epoch 10/10  
1688/1688 [=====] - 6s 4ms/step - loss: 0.0120 - accuracy: 0.9963 - val_loss: 0.0524 - val_accuracy: 0.9902
```

↑ 圖 5(baseline model)

```
Epoch 10/10  
1688/1688 [=====] - 7s 4ms/step - loss: 0.0118 - accuracy: 0.9963 - val_loss: 0.0343 - val_accuracy: 0.9932
```

↑ 圖 6(model with BN layer)

Comparison w/ arbitrary layer of abovementioned CNN network:

為了比較 Model 不同層數之間的差異，分別移除了一層 convolution layer，和新增一層 convolution layer 和原本的 baseline model 做比較。

層數較少的 model 和原本的 baseline model 相比，在 training loss、training accuracy、validation loss、validation accuracy 上反而有更好的表現。圖 7 為移除一層 convolution layer 的 model 的 training 和 validation 結果。

層數較多的 model 和原本的 baseline model 相比，在 training loss、training accuracy、validation loss、validation accuracy 皆無明顯差異；在原本 baseline model 以表現相當優異的情況下，可以得知即使再加入更多層數，對於整體表現並無明顯差異。圖 8 為多加一層 convolution layer 的 model 的 training 和 validation 結果。

即是減少一層 convolution layer 所得到的表現有略高於 baseline model，但由上述結果可知，在原先 baseline model 以表現相當優異的情況下，層數減少與增加，對 model 影響並不大，但如果將 activation function 從 softmax 改成 sigmoid 將會導致 accuracy 顯著下降，是因為 sigmoid function 適用於二分法問

題。

```
Epoch 10/10  
1688/1688 [=====] - 7s 4ms/step - loss: 0.0078 - accuracy: 0.9973 - val_loss: 0.0463 - val_accuracy: 0.9912
```

圖 7(減少一層 convolution layer 的 model)

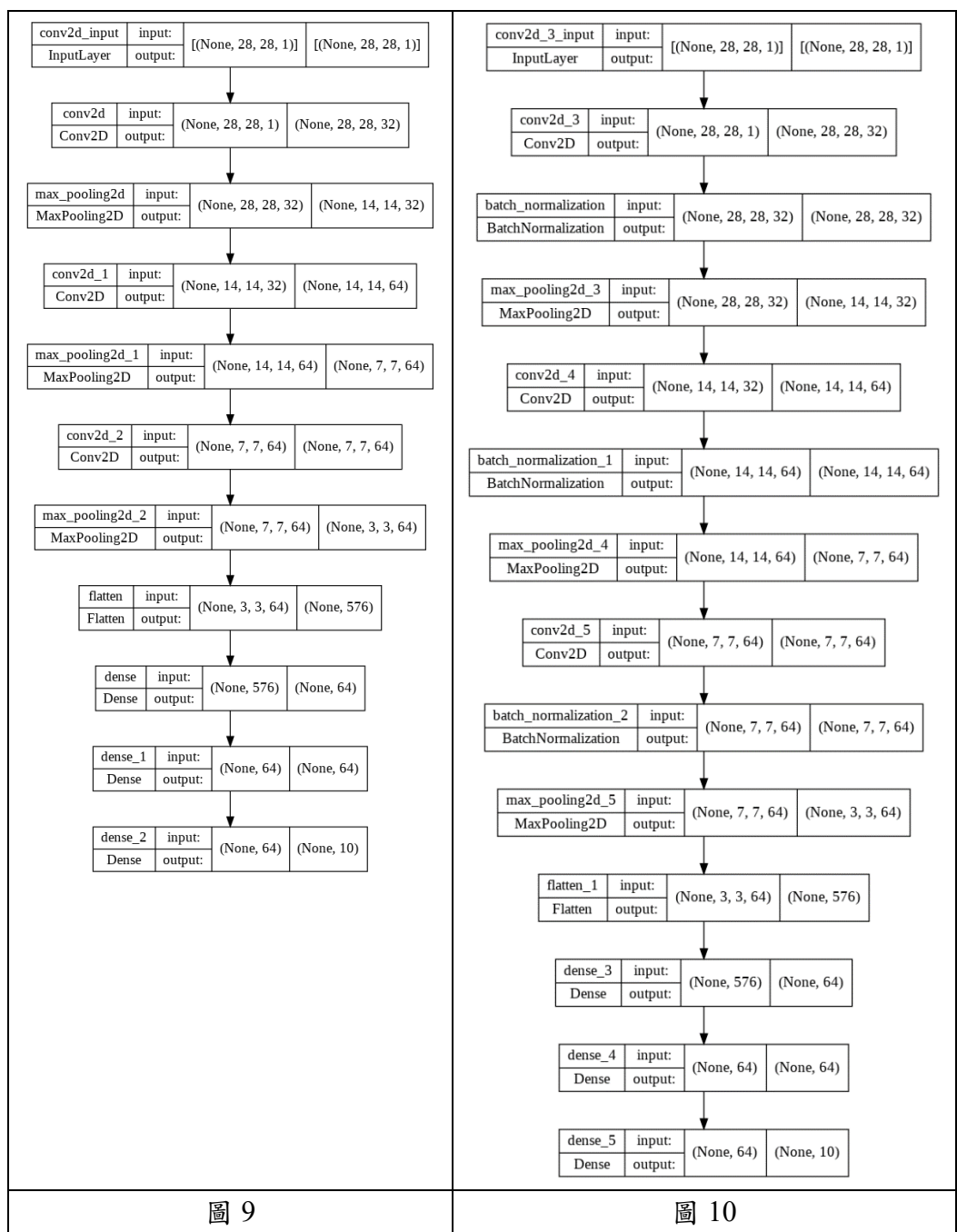
```
Epoch 10/10  
1688/1688 [=====] - 7s 4ms/step - loss: 0.0128 - accuracy: 0.9959 - val_loss: 0.0288 - val_accuracy: 0.9933
```

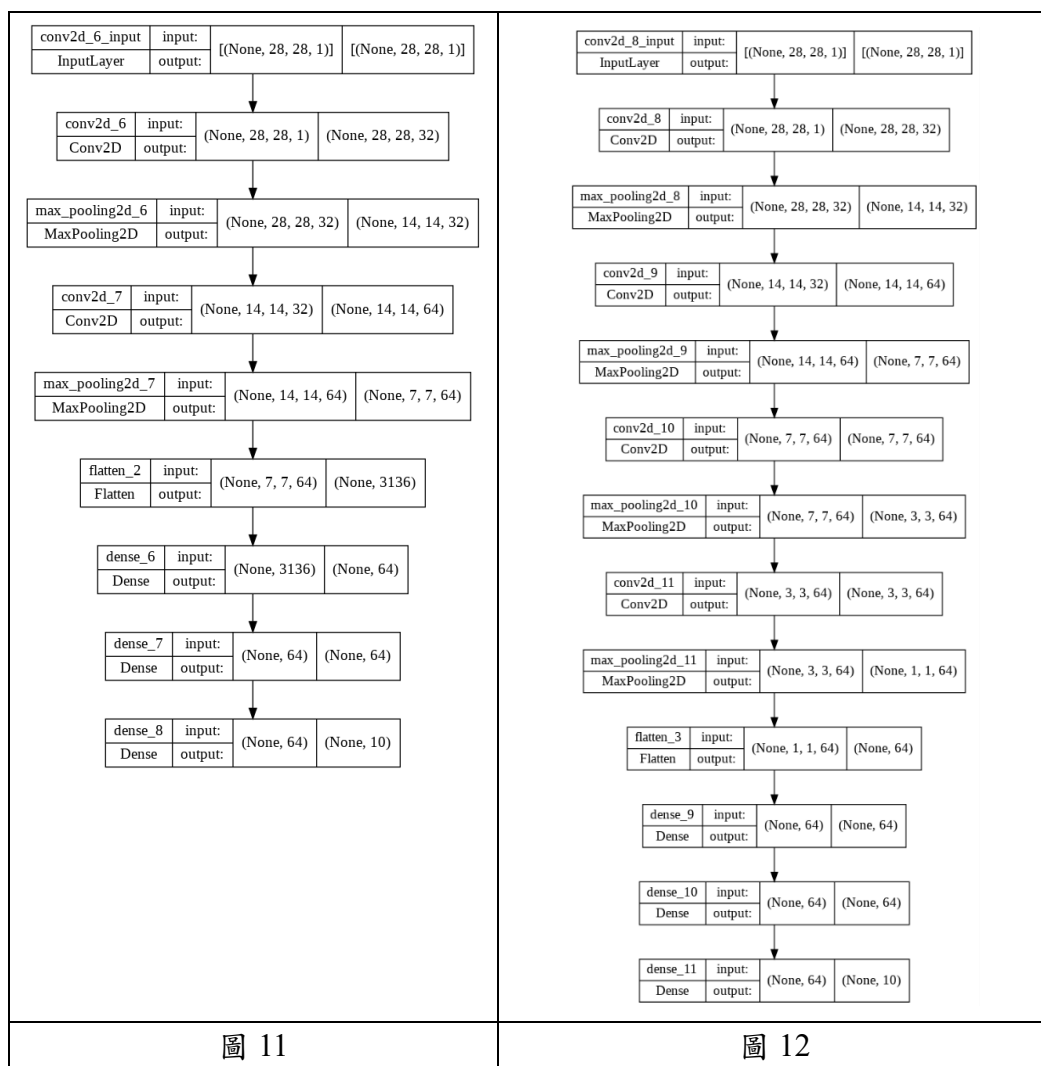
圖 8(增加一層 convolution layer 的 model)

- 各項 print 及 plot 輸出

Layer introduction:

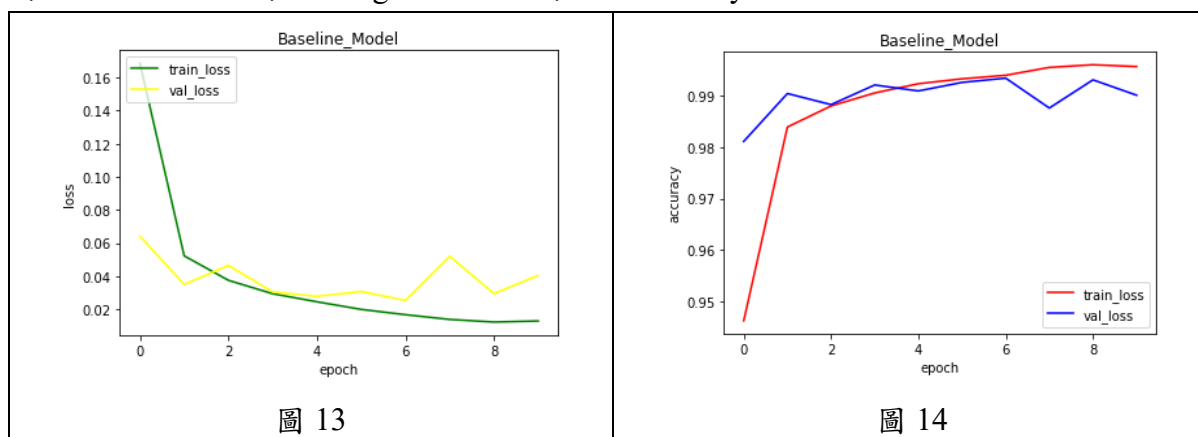
下列為個 model 之 layer introduction，圖 9 為 baseline model、圖 10 為加入 Batch Normalization layer 的 model、圖 11 為移除一層 convolution layer 的 model、圖 12 為新增一層 convolution layer 的 model。





loss and accuracy of models:

圖 13、14、15 為 baseline model 的 training/validation 的 loss/accuracy、圖 16、17、18 為有加入 Batch Normalization layer model 的 training/validation 的 loss/accuracy、圖 19、20、21 為移除一層 convolution layer 的 baseline model 的 training/validation 的 loss/accuracy、圖 22、23、24 為增加一層 convolution layer 的 baseline model 的 training/validation 的 loss/accuracy。



Test set loss of Baseline_Model is 0.05202106758952141
 Test set accuracy of Baseline_Model is 0.9868000149726868

圖 15

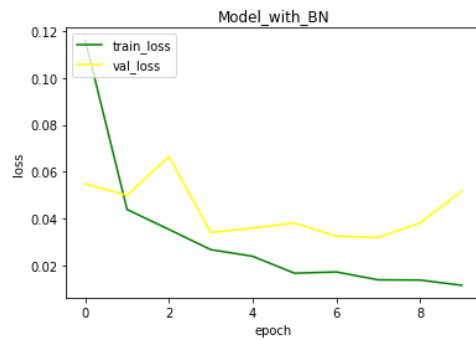


圖 16

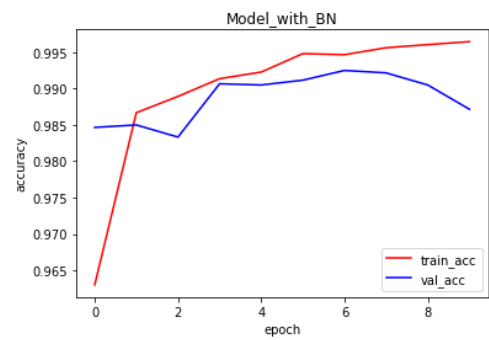


圖 17

Test set loss of Model_with_BN is 0.04637472331523895
 Test set accuracy of Model_with_BN is 0.9883999824523926

圖 18

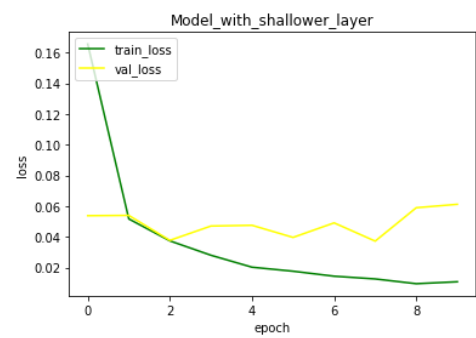


圖 19

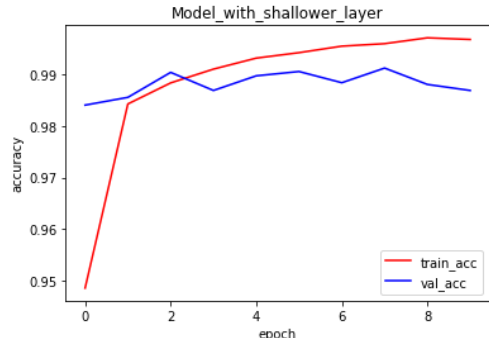


圖 20

Test set loss of Model_with_shallower_layer is 0.05822271853685379
 Test set accuracy of Model_with_shallower_layer is 0.9857000112533569

圖 21

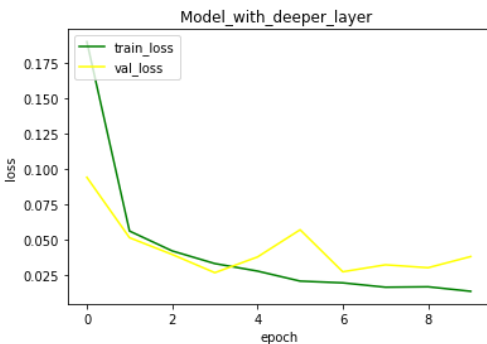


圖 22

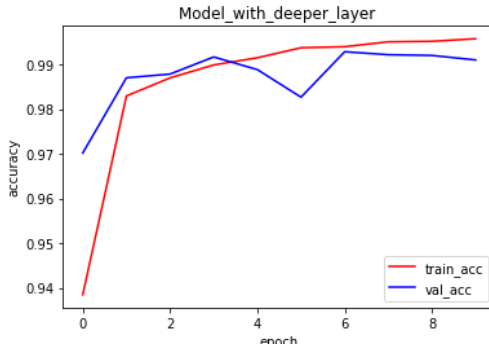


圖 23

Test set loss Model_with_deeper_layer is 0.038419902324676514
 Test set accuracy Model_with_deeper_layer is 0.9896000027656555

- 實作所遇到的困難及解決方法

這次的作業相較上次較簡單，比較多的是花時間熟悉 CNN 架構和背景知識。另外，其他比較花時間的則是在熟悉 tensorflow 的相關配置。