1. 利用keras 套件，在 tensorflow 上，使用 CNN 深度學習演算法對 MNIST 資料集進行分類，依序完成以下步驟及問題： (附圖並詳細說明)

(a) 建立 CNN 模型 (50%)

```
[39] model = Sequential()
     model.add(Conv2D(filters=25, kernel_size=(5), padding='same', input_shape=(28, 28, 1), activation='relu'))
     model.add(MaxPooling2D(pool_size=(2, 2)))
     model.add(Conv2D(filters=50, kernel_size=(3), padding='same', activation='relu'))
     model.add(MaxPooling2D(pool_size=(2, 2)))
     model.add(Dropout(0.5))
     model.add(Flatten())
     model.add(Dense(256, activation='relu'))
     model.add(Dropout(0.5))
     model.add(Dense(10, activation='softmax'))
```

(b) 承上題，

解釋所使用的 padding 方式為何?

Padding = same, which means that when padding the input such that the output has the same length as the original input.

所使用的激活函數為何?

Activation using Relu, which turns out that all the negative values are converted to zeros.

Dropout 值設為多少? (10%)

Dropout = 0.5. This can prevent overfitting, and the optimal value of ratio will be 0.5.

(c) model. Summary () 的結果 (5%)

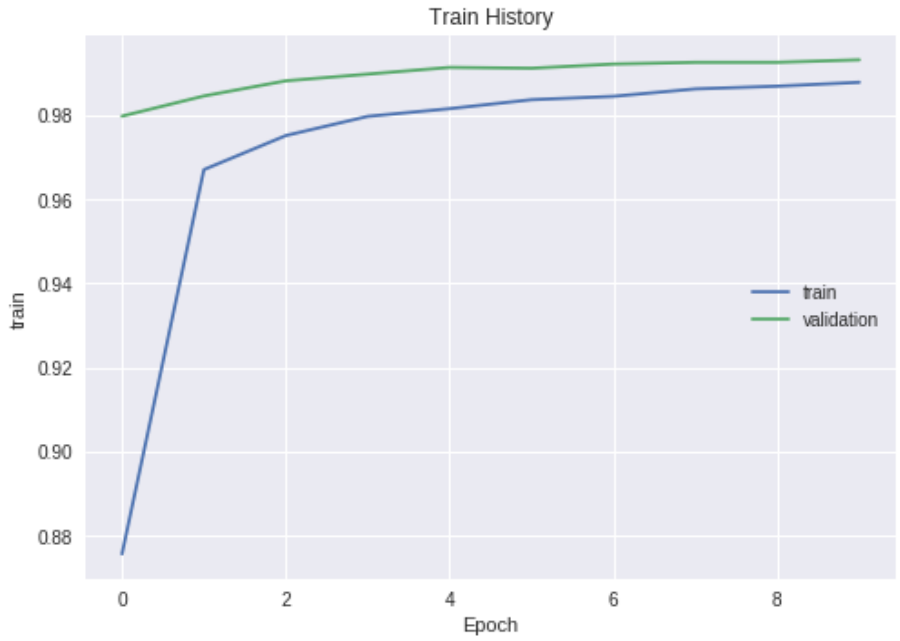
```
[40] model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_19 (Conv2D)           (None, 28, 28, 25)        650
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 14, 14, 25)        0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 14, 14, 50)        11300
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 7, 7, 50)          0
_________________________________________________________________
dropout_19 (Dropout)         (None, 7, 7, 50)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 2450)              0
_________________________________________________________________
dense_19 (Dense)             (None, 256)               627456
_________________________________________________________________
dropout_20 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_20 (Dense)             (None, 10)                2570
=================================================================
Total params: 641,976
Trainable params: 641,976
Non-trainable params: 0
_________________________________________________________________
```

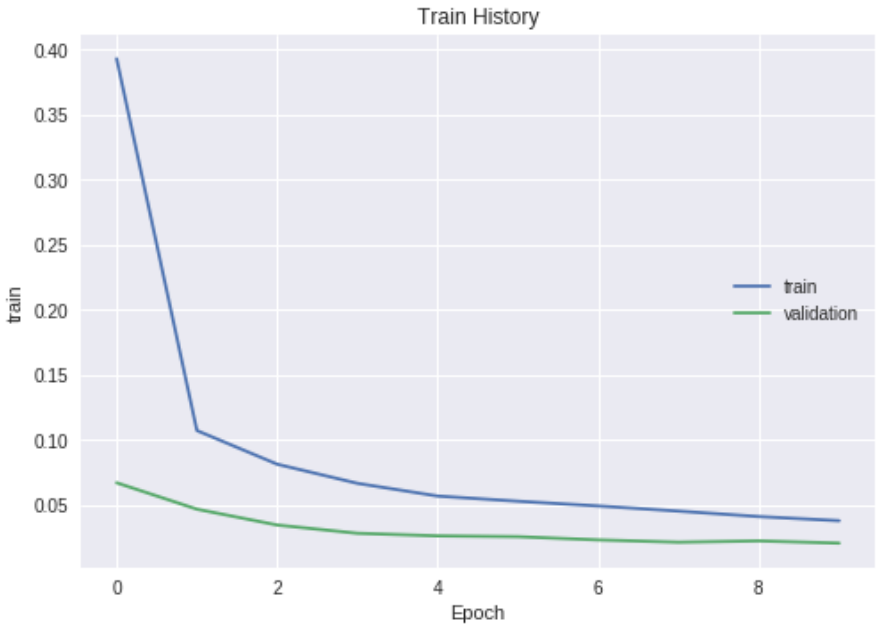
(d) 評估訓練結果 (5%)

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/10
 - 4s - loss: 0.3931 - acc: 0.8757 - val_loss: 0.0671 - val_acc: 0.9798
Epoch 2/10
 - 4s - loss: 0.1072 - acc: 0.9671 - val_loss: 0.0468 - val_acc: 0.9846
Epoch 3/10
 - 4s - loss: 0.0813 - acc: 0.9752 - val_loss: 0.0346 - val_acc: 0.9882
Epoch 4/10
 - 4s - loss: 0.0666 - acc: 0.9798 - val_loss: 0.0282 - val_acc: 0.9898
Epoch 5/10
 - 4s - loss: 0.0569 - acc: 0.9816 - val_loss: 0.0263 - val_acc: 0.9914
Epoch 6/10
 - 4s - loss: 0.0529 - acc: 0.9837 - val_loss: 0.0256 - val_acc: 0.9912
Epoch 7/10
 - 4s - loss: 0.0493 - acc: 0.9845 - val_loss: 0.0232 - val_acc: 0.9922
Epoch 8/10
 - 4s - loss: 0.0453 - acc: 0.9863 - val_loss: 0.0214 - val_acc: 0.9926
Epoch 9/10
 - 4s - loss: 0.0411 - acc: 0.9869 - val_loss: 0.0224 - val_acc: 0.9926
Epoch 10/10
 - 4s - loss: 0.0379 - acc: 0.9879 - val_loss: 0.0207 - val_acc: 0.9932
```

(e) 準確率視覺化 (5%)



(f) 誤差率視覺化 (5%)



(g) 整體模型的準確率為何? (10%)

```
[44] scores = model.evaluate(X_test, y_test)
     scores[1]
```

```
10000/10000 [==============================] - 1s 102us/step
0.9932
```

(h) 混淆矩陣 (10%)

```
(10000, 10)
```

predict	0	1	2	3	4	5	6	7	8	9
label										
0	975	0	0	0	0	0	3	1	1	0
1	0	1134	0	0	0	0	1	0	0	0
2	0	0	1027	0	1	0	0	4	0	0
3	0	0	1	1002	0	3	0	3	1	0
4	0	0	0	0	976	0	0	0	0	6
5	0	0	0	3	0	888	1	0	0	0
6	2	2	0	0	1	2	951	0	0	0
7	0	1	4	0	0	0	0	1022	1	0
8	1	0	2	1	0	0	0	2	966	2
9	0	3	0	0	4	5	0	3	3	991