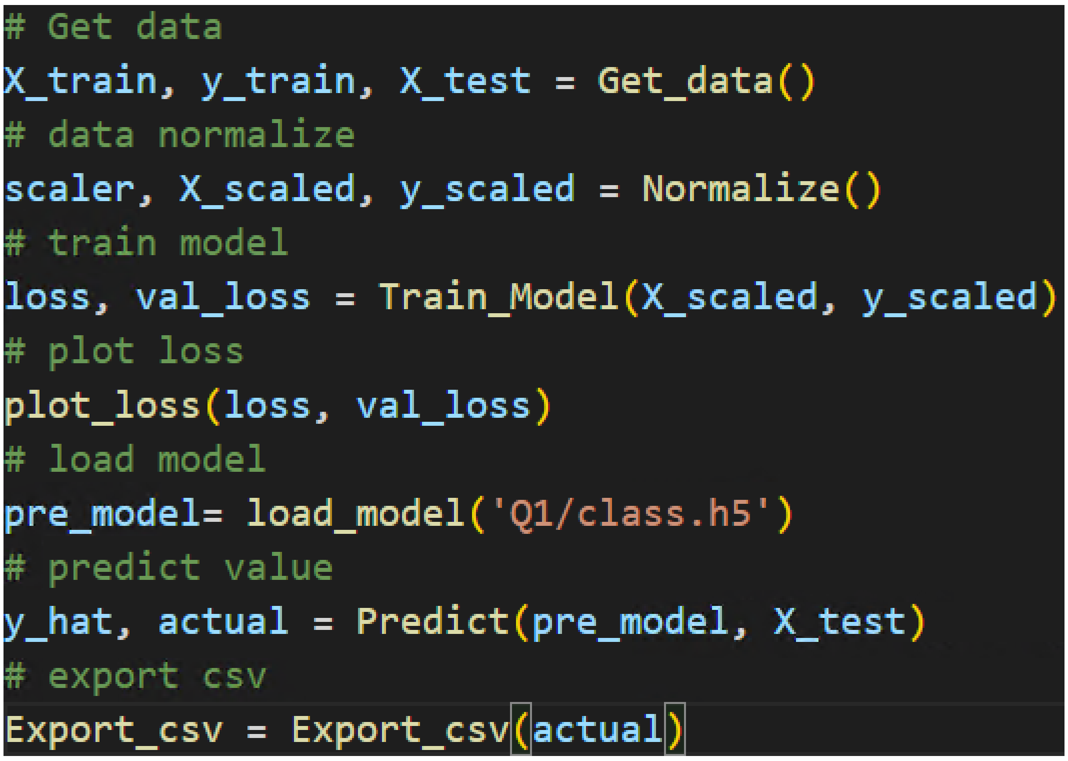
Second Programming Exam.
Deep Learning, 2022

姓名：吳佳展　　學號：110318517

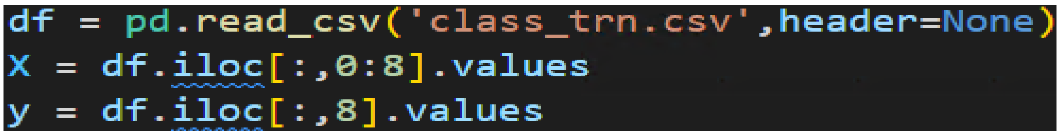
第一題的答案的整體架構為 7 大步驟分別為：a.讀取資料 b.資料正規畫 c.設計模型架構 d.訓練模型 e.劃出訓練模型的 loss f.使用模型進行產出預測值 g.將其測試集資料和預測值合併匯出 csv 檔案，詳細步驟下方將會逐一說明各項過程。

整體架構圖：

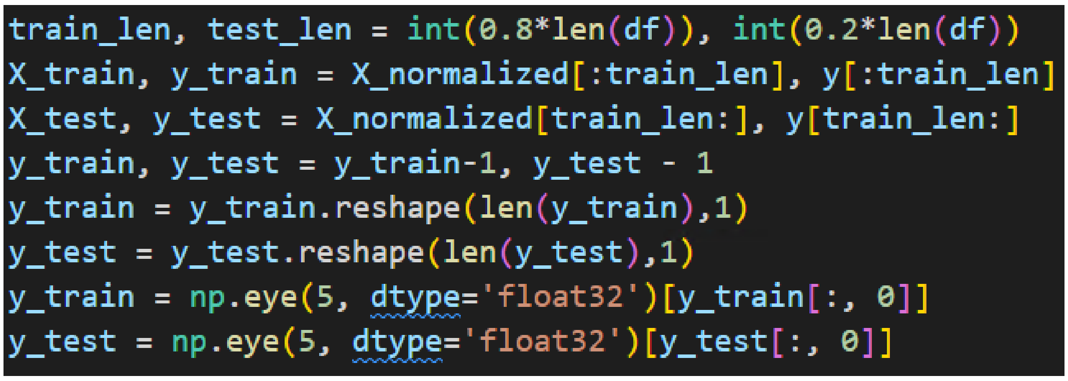


```
# Get data
X_train, y_train, X_test = Get_data()
# data normalize
scaler, X_scaled, y_scaled = Normalize()
# train model
loss, val_loss = Train_Model(X_scaled, y_scaled)
# plot loss
plot_loss(loss, val_loss)
# load model
pre_model= load_model('Q1/class.h5')
# predict value
y_hat, actual = Predict(pre_model, X_test)
# export csv
Export_csv = Export_csv(actual)
```

1. 第 1 步先讀取“class_trn.csv”和“class_tst.csv”的檔案當作 train data 和 Test_data。

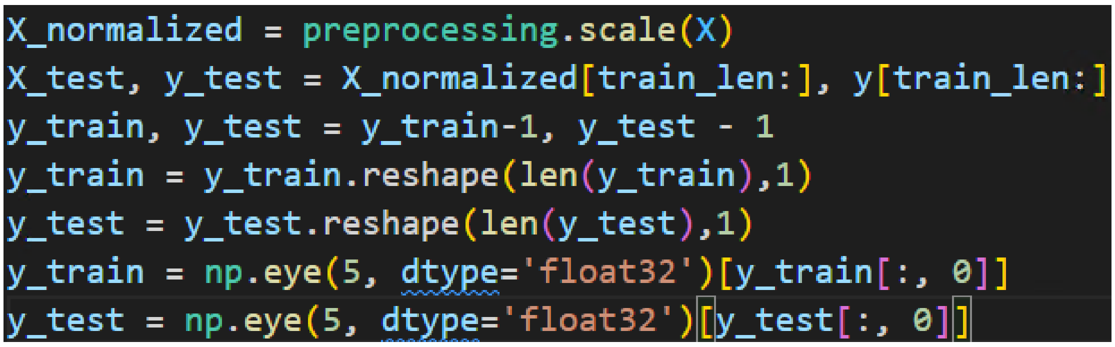


```
df = pd.read_csv('class_trn.csv',header=None)
X = df.iloc[:,0:8].values
y = df.iloc[:,8].values
```



```
train_len, test_len = int(0.8*len(df)), int(0.2*len(df))
X_train, y_train = X_normalized[:train_len], y[:train_len]
X_test, y_test = X_normalized[train_len:], y[train_len:]
y_train, y_test = y_train-1, y_test - 1
y_train = y_train.reshape(len(y_train),1)
y_test = y_test.reshape(len(y_test),1)
y_train = np.eye(5, dtype='float32')[y_train[:, 0]]
y_test = np.eye(5, dtype='float32')[y_test[:, 0]]
```

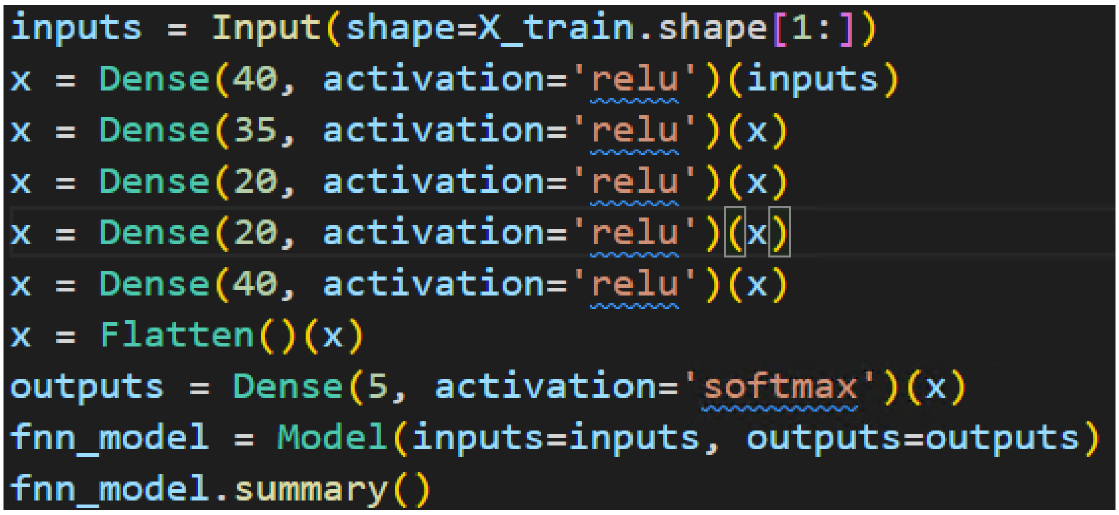
2. 第 2 步驟將 train data 和 test data 做標準化(normalize)



```
X_normalized = preprocessing.scale(X)
X_test, y_test = X_normalized[train_len:], y[train_len:]
y_train, y_test = y_train-1, y_test - 1
y_train = y_train.reshape(len(y_train),1)
y_test = y_test.reshape(len(y_test),1)
y_train = np.eye(5, dtype='float32')[y_train[:, 0]]
y_test = np.eye(5, dtype='float32')[y_test[:, 0]]
```

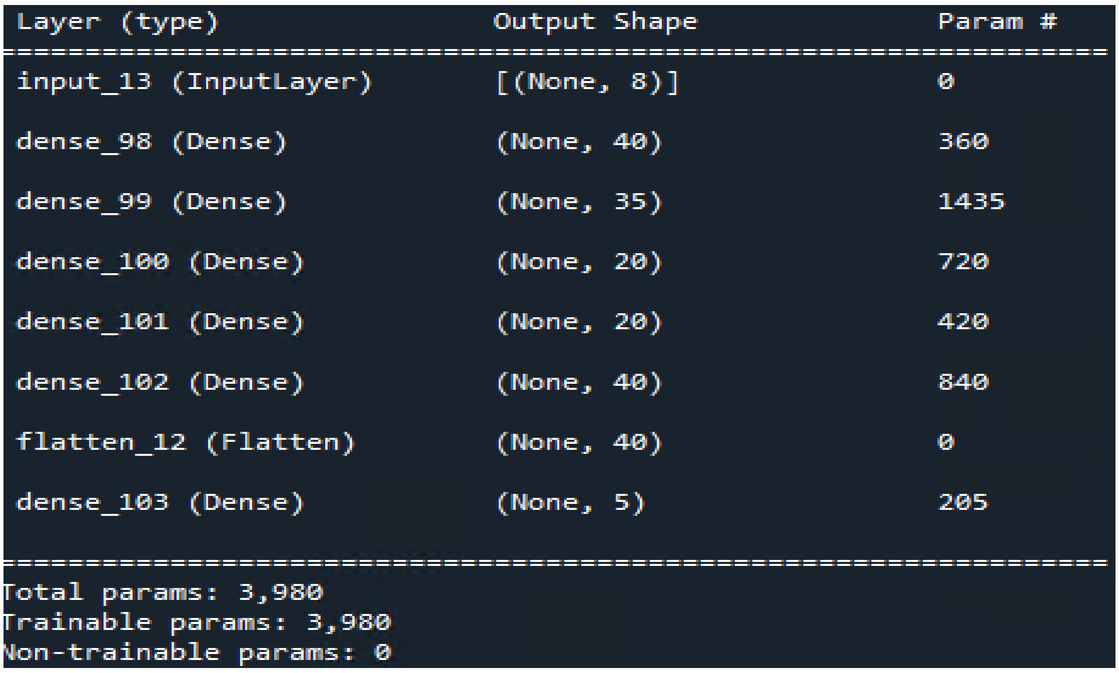
3. 第三步驟設計 FNN 模型架構所供 training data 訓練。

FNN 模型架構程式如下圖所示



```
inputs = Input(shape=X_train.shape[1:])
x = Dense(40, activation='relu')(inputs)
x = Dense(35, activation='relu')(x)
x = Dense(20, activation='relu')(x)
x = Dense(20, activation='relu')(x)
x = Dense(40, activation='relu')(x)
x = Flatten()(x)
outputs = Dense(5, activation='softmax')(x)
fnn_model = Model(inputs=inputs, outputs=outputs)
fnn_model.summary()
```

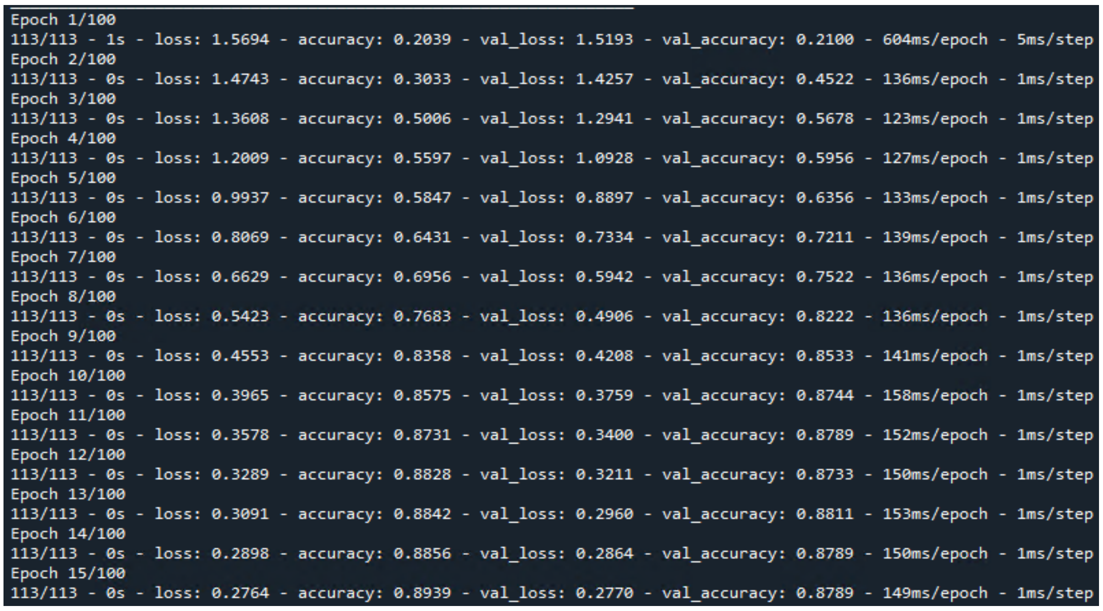
FNN 模型架構及參數如下圖所示



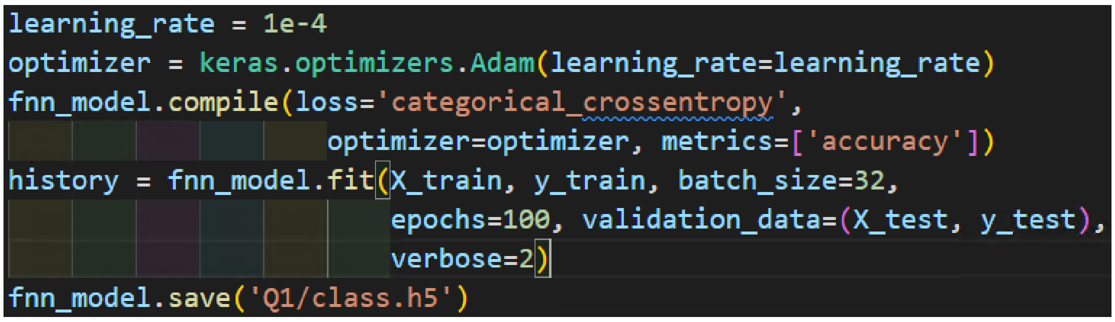
```
 Layer (type)                Output Shape              Param #
=================================================================
 input_13 (InputLayer)       [(None, 8)]               0

 dense_98 (Dense)            (None, 40)                360

 dense_99 (Dense)            (None, 35)                1435

 dense_100 (Dense)           (None, 20)                720

 dense_101 (Dense)           (None, 20)                420

 dense_102 (Dense)           (None, 40)                840

 flatten_12 (Flatten)        (None, 40)                0

 dense_103 (Dense)           (None, 5)                 205

=================================================================
Total params: 3,980
Trainable params: 3,980
Non-trainable params: 0
```

4. 第四步驟將 TRAINING 資料放入模型訓練，並將模型存取，而模型訓練的參數如下表所示。

參數名稱	Loss	Optimizer	Epochs	Batch_size
所使用參數	categorical_crossentropy	adam	100	32

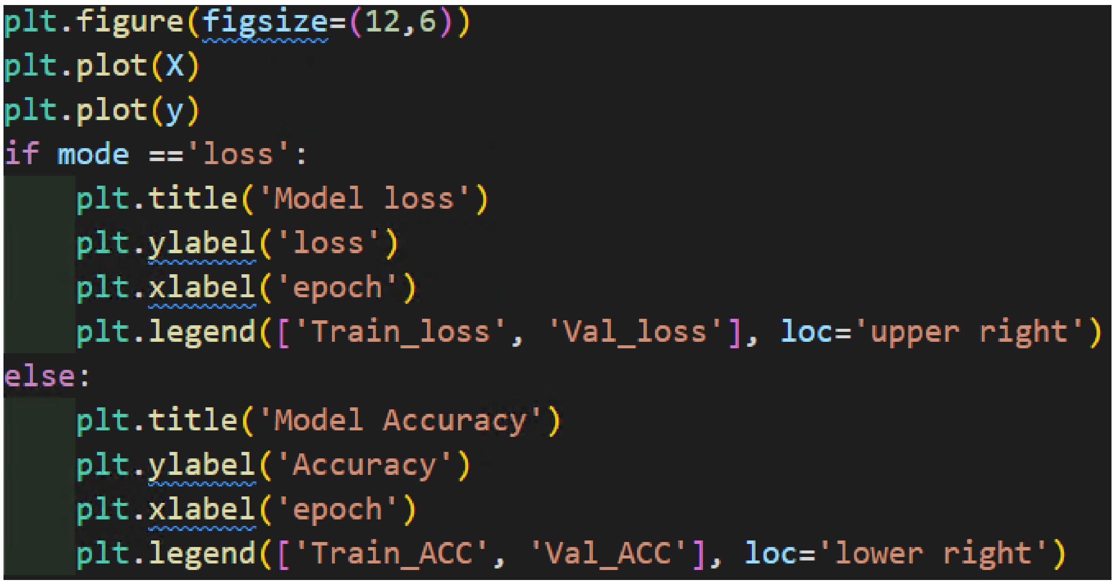


```
Epoch 1/100
113/113 - 1s - loss: 1.5694 - accuracy: 0.2039 - val_loss: 1.5193 - val_accuracy: 0.2100 - 604ms/epoch - 5ms/step
Epoch 2/100
113/113 - 0s - loss: 1.4743 - accuracy: 0.3033 - val_loss: 1.4257 - val_accuracy: 0.4522 - 136ms/epoch - 1ms/step
Epoch 3/100
113/113 - 0s - loss: 1.3608 - accuracy: 0.5006 - val_loss: 1.2941 - val_accuracy: 0.5678 - 123ms/epoch - 1ms/step
Epoch 4/100
113/113 - 0s - loss: 1.2009 - accuracy: 0.5597 - val_loss: 1.0928 - val_accuracy: 0.5956 - 127ms/epoch - 1ms/step
Epoch 5/100
113/113 - 0s - loss: 0.9937 - accuracy: 0.5847 - val_loss: 0.8897 - val_accuracy: 0.6356 - 133ms/epoch - 1ms/step
Epoch 6/100
113/113 - 0s - loss: 0.8069 - accuracy: 0.6431 - val_loss: 0.7334 - val_accuracy: 0.7211 - 139ms/epoch - 1ms/step
Epoch 7/100
113/113 - 0s - loss: 0.6629 - accuracy: 0.6956 - val_loss: 0.5942 - val_accuracy: 0.7522 - 136ms/epoch - 1ms/step
Epoch 8/100
113/113 - 0s - loss: 0.5423 - accuracy: 0.7683 - val_loss: 0.4906 - val_accuracy: 0.8222 - 136ms/epoch - 1ms/step
Epoch 9/100
113/113 - 0s - loss: 0.4553 - accuracy: 0.8358 - val_loss: 0.4208 - val_accuracy: 0.8533 - 141ms/epoch - 1ms/step
Epoch 10/100
113/113 - 0s - loss: 0.3965 - accuracy: 0.8575 - val_loss: 0.3759 - val_accuracy: 0.8744 - 158ms/epoch - 1ms/step
Epoch 11/100
113/113 - 0s - loss: 0.3578 - accuracy: 0.8731 - val_loss: 0.3400 - val_accuracy: 0.8789 - 152ms/epoch - 1ms/step
Epoch 12/100
113/113 - 0s - loss: 0.3289 - accuracy: 0.8828 - val_loss: 0.3211 - val_accuracy: 0.8733 - 150ms/epoch - 1ms/step
Epoch 13/100
113/113 - 0s - loss: 0.3091 - accuracy: 0.8842 - val_loss: 0.2960 - val_accuracy: 0.8811 - 153ms/epoch - 1ms/step
Epoch 14/100
113/113 - 0s - loss: 0.2898 - accuracy: 0.8856 - val_loss: 0.2864 - val_accuracy: 0.8789 - 150ms/epoch - 1ms/step
Epoch 15/100
113/113 - 0s - loss: 0.2764 - accuracy: 0.8939 - val_loss: 0.2770 - val_accuracy: 0.8789 - 149ms/epoch - 1ms/step
```

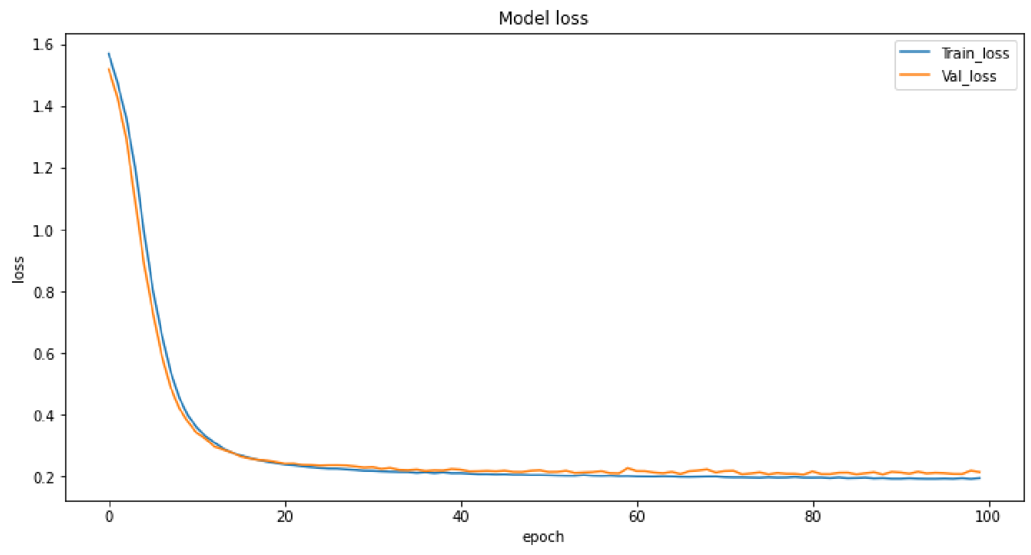


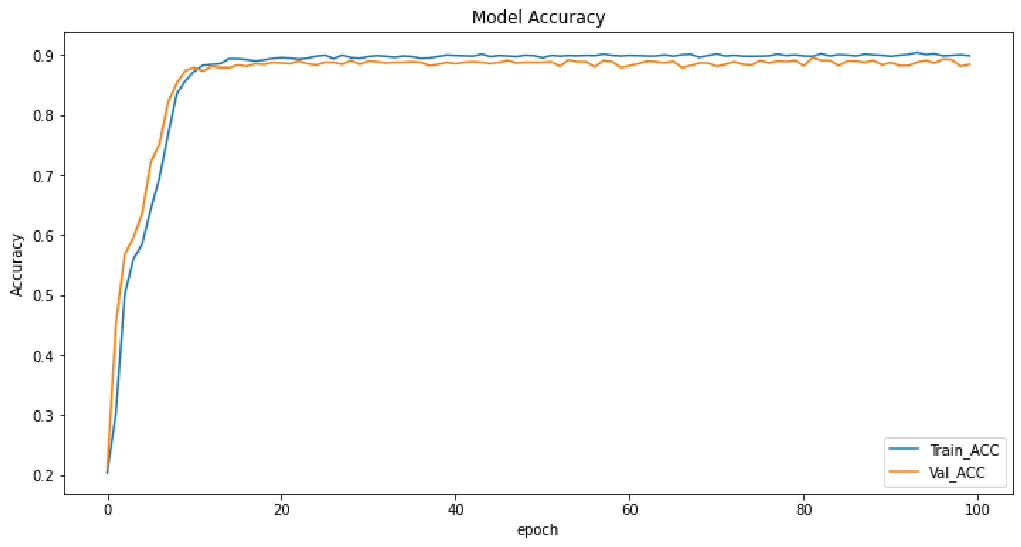
```python
learning_rate = 1e-4
optimizer = keras.optimizers.Adam(learning_rate=learning_rate)
fnn_model.compile(loss='categorical_crossentropy',
                  optimizer=optimizer, metrics=['accuracy'])
history = fnn_model.fit(X_train, y_train, batch_size=32,
                        epochs=100, validation_data=(X_test, y_test),
                        verbose=2)
fnn_model.save('Q1/class.h5')
```

5. 第五步驟為將所訓練過程的 learn errorr 及 Accuracy 用圖表所示，判斷是否有 overfitting 或是 underfitting 的狀態。

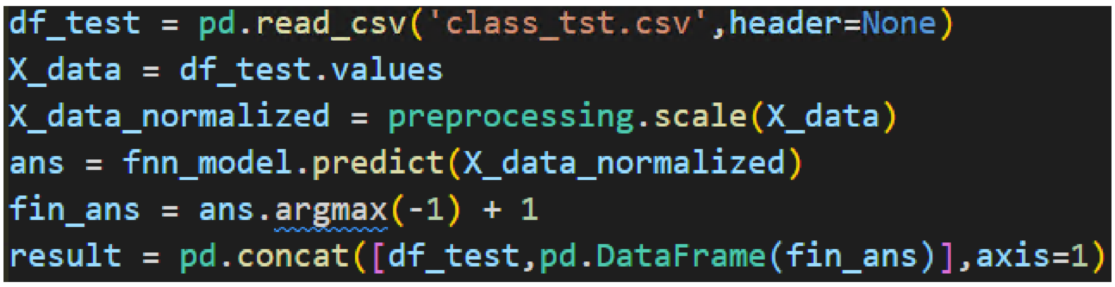


```python
plt.figure(figsize=(12,6))
plt.plot(X)
plt.plot(y)
if mode =='loss':
    plt.title('Model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['Train_loss', 'Val_loss'], loc='upper right')
else:
    plt.title('Model Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('epoch')
    plt.legend(['Train_ACC', 'Val_ACC'], loc='lower right')
```



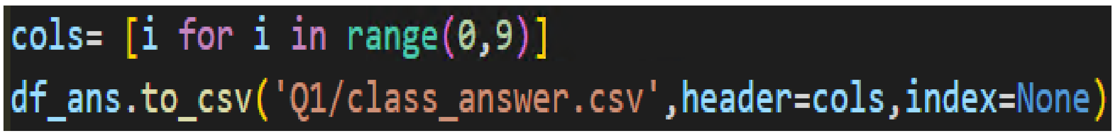


6. 第六步驟將 test_data 使用第 4 步驟所生成的 model 進行預測，進而產生出預測值。



```
df_test = pd.read_csv('class_tst.csv',header=None)
X_data = df_test.values
X_data_normalized = preprocessing.scale(X_data)
ans = fnn_model.predict(X_data_normalized)
fin_ans = ans.argmax(-1) + 1
result = pd.concat([df_test,pd.DataFrame(fin_ans)],axis=1)
```

7. 第七步驟為將 test_data 和預測值組合起來輸出成“class_answer.csv”。



```
cols= [i for i in range(0,9)]
df_ans.to_csv('Q1/class_answer.csv',header=cols,index=None)
```