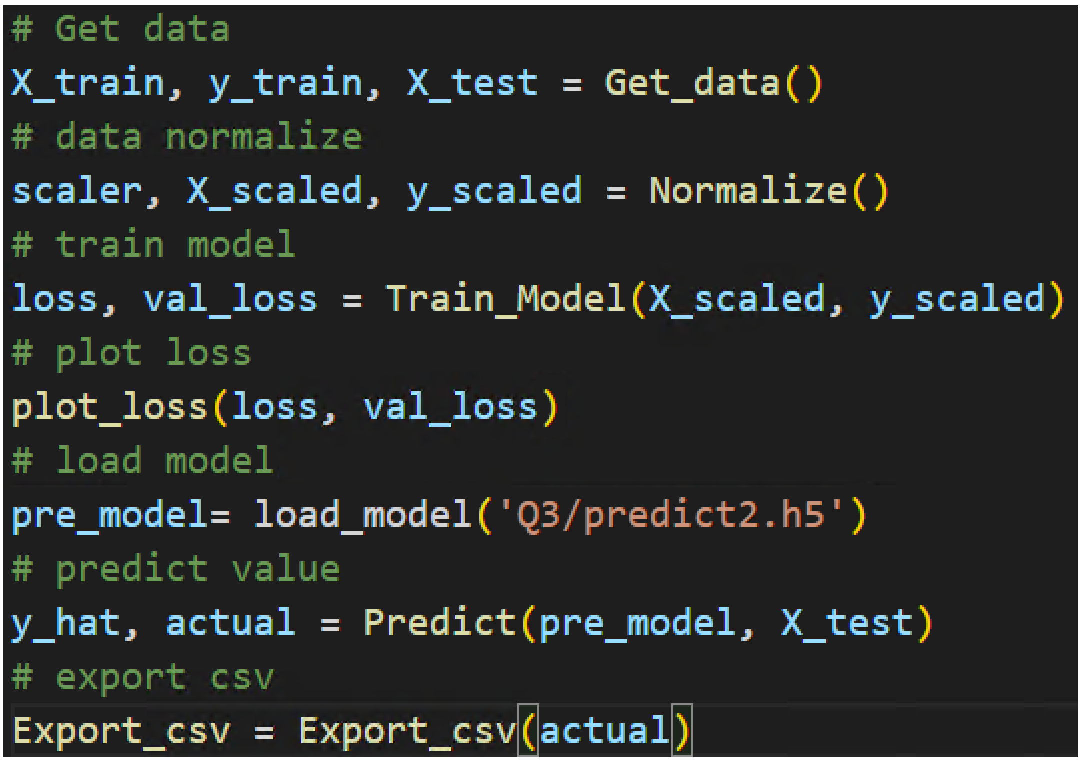
Second Programming Exam.
Deep Learning, 2022

姓名：吳佳展　　學號：110318517

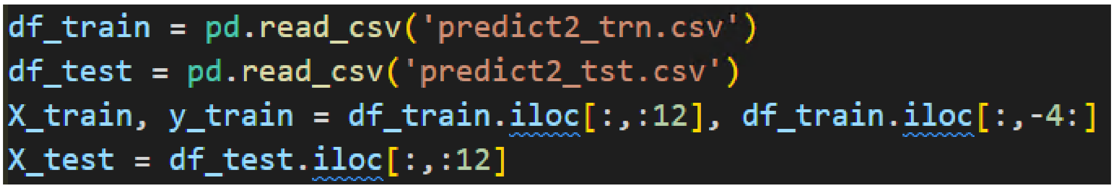
此題的答案的整體架構為 7 大步驟分別為：a.讀取資料 b.資料正規畫 c.設計模型架構 d.訓練模型 e.劃出訓練模型的 loss f.使用模型進行產出預測值 g.將其測試集資料和預測值合併匯出 csv 檔案，詳細步驟下方將會逐一說明各項過程。

整體架構圖：



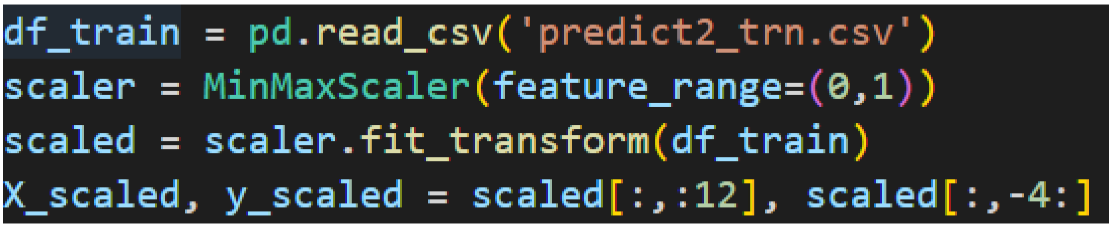
```
# Get data
X_train, y_train, X_test = Get_data()
# data normalize
scaler, X_scaled, y_scaled = Normalize()
# train model
loss, val_loss = Train_Model(X_scaled, y_scaled)
# plot loss
plot_loss(loss, val_loss)
# load model
pre_model= load_model('Q3/predict2.h5')
# predict value
y_hat, actual = Predict(pre_model, X_test)
# export csv
Export_csv = Export_csv(actual)
```

1. 第 1 步先讀取“predict2_trn.csv”和“predict2_tst.csv”的檔案當作 train data 和 Test_data。



```
df_train = pd.read_csv('predict2_trn.csv')
df_test = pd.read_csv('predict2_tst.csv')
X_train, y_train = df_train.iloc[:,:12], df_train.iloc[:,-4:]
X_test = df_test.iloc[:,:12]
```

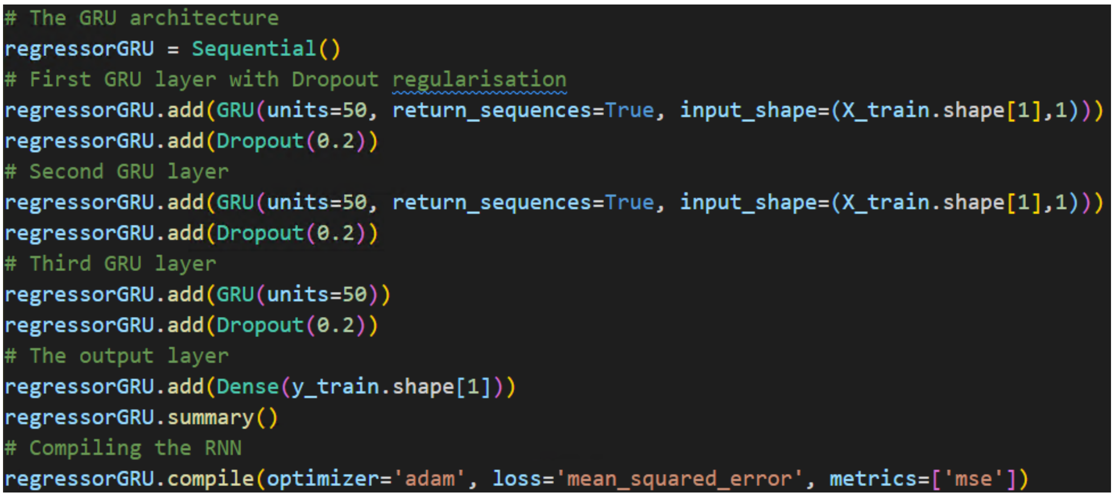
2. 第 2 步驟將 train data 和 test data 做標準化(normalize)



```
df_train = pd.read_csv('predict2_trn.csv')
scaler = MinMaxScaler(feature_range=(0,1))
scaled = scaler.fit_transform(df_train)
X_scaled, y_scaled = scaled[:,:12], scaled[:,-4:]
```

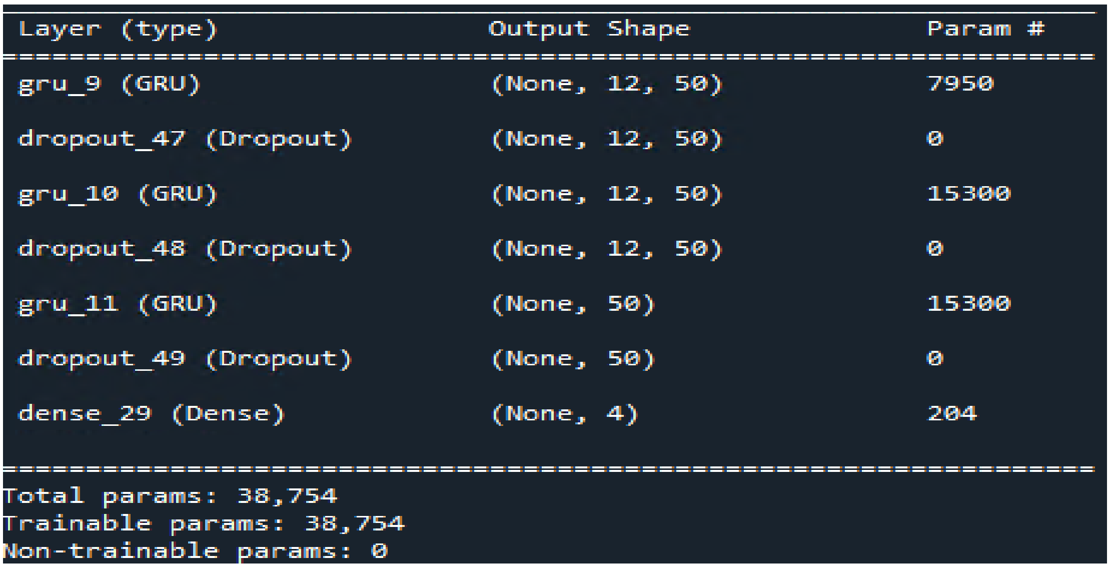
3. 第三步驟設計 GRU 模型架構所供 training data 訓練。

GRU 模型架構程式如下圖所示



```
# The GRU architecture
regressorGRU = Sequential()
# First GRU layer with Dropout regularisation
regressorGRU.add(GRU(units=50, return_sequences=True, input_shape=(X_train.shape[1],1)))
regressorGRU.add(Dropout(0.2))
# Second GRU layer
regressorGRU.add(GRU(units=50, return_sequences=True, input_shape=(X_train.shape[1],1)))
regressorGRU.add(Dropout(0.2))
# Third GRU layer
regressorGRU.add(GRU(units=50))
regressorGRU.add(Dropout(0.2))
# The output layer
regressorGRU.add(Dense(y_train.shape[1]))
regressorGRU.summary()
# Compiling the RNN
regressorGRU.compile(optimizer='adam', loss='mean_squared_error', metrics=['mse'])
```

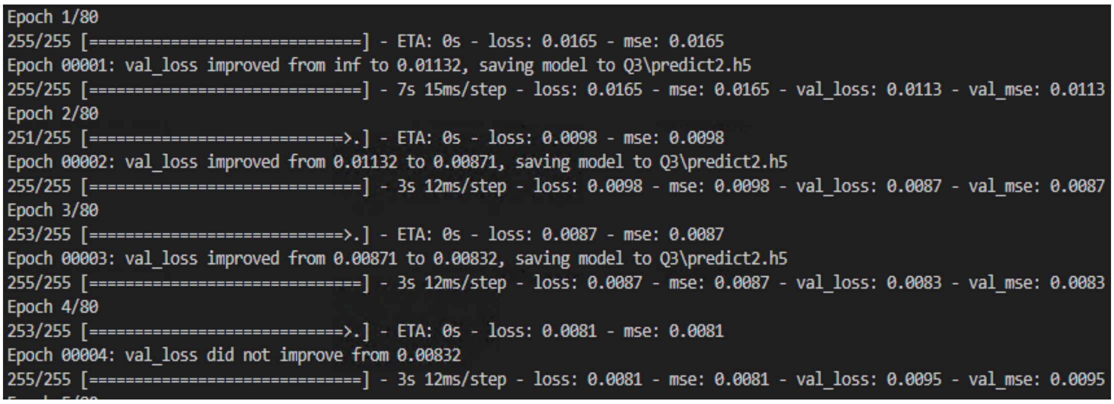
LSTM 模型架構及參數如下圖所示



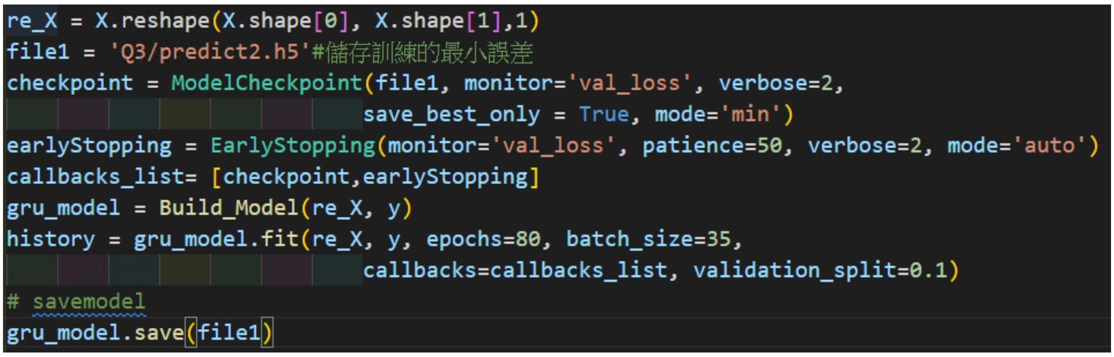
```
 Layer (type)                Output Shape              Param #
=================================================================
 gru_9 (GRU)                 (None, 12, 50)            7950

 dropout_47 (Dropout)        (None, 12, 50)            0

 gru_10 (GRU)                (None, 12, 50)            15300

 dropout_48 (Dropout)        (None, 12, 50)            0

 gru_11 (GRU)                (None, 50)                15300

 dropout_49 (Dropout)        (None, 50)                0

 dense_29 (Dense)            (None, 4)                 204

=================================================================
Total params: 38,754
Trainable params: 38,754
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	MSE	adam	80	35

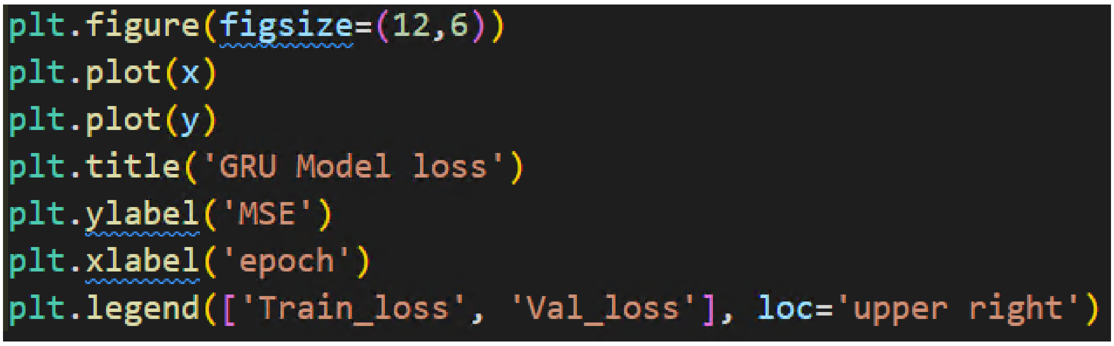


```
Epoch 1/80
255/255 [==============================] - ETA: 0s - loss: 0.0165 - mse: 0.0165
Epoch 00001: val_loss improved from inf to 0.01132, saving model to Q3\predict2.h5
255/255 [==============================] - 7s 15ms/step - loss: 0.0165 - mse: 0.0165 - val_loss: 0.0113 - val_mse: 0.0113
Epoch 2/80
251/255 [============================>.] - ETA: 0s - loss: 0.0098 - mse: 0.0098
Epoch 00002: val_loss improved from 0.01132 to 0.00871, saving model to Q3\predict2.h5
255/255 [==============================] - 3s 12ms/step - loss: 0.0098 - mse: 0.0098 - val_loss: 0.0087 - val_mse: 0.0087
Epoch 3/80
253/255 [============================>.] - ETA: 0s - loss: 0.0087 - mse: 0.0087
Epoch 00003: val_loss improved from 0.00871 to 0.00832, saving model to Q3\predict2.h5
255/255 [==============================] - 3s 12ms/step - loss: 0.0087 - mse: 0.0087 - val_loss: 0.0083 - val_mse: 0.0083
Epoch 4/80
253/255 [============================>.] - ETA: 0s - loss: 0.0081 - mse: 0.0081
Epoch 00004: val_loss did not improve from 0.00832
255/255 [==============================] - 3s 12ms/step - loss: 0.0081 - mse: 0.0081 - val_loss: 0.0095 - val_mse: 0.0095
```

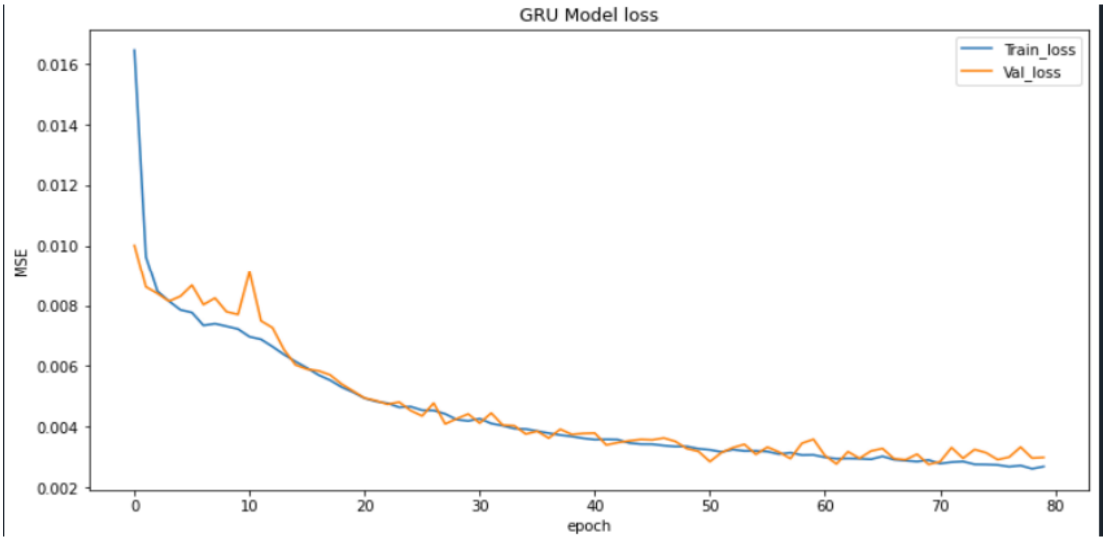


```
re_X = X.reshape(X.shape[0], X.shape[1],1)
file1 = 'Q3/predict2.h5'#儲存訓練的最小誤差
checkpoint = ModelCheckpoint(file1, monitor='val_loss', verbose=2,
                            save_best_only = True, mode='min')
earlyStopping = EarlyStopping(monitor='val_loss', patience=50, verbose=2, mode='auto')
callbacks_list= [checkpoint,earlyStopping]
gru_model = Build_Model(re_X, y)
history = gru_model.fit(re_X, y, epochs=80, batch_size=35,
                            callbacks=callbacks_list, validation_split=0.1)
# savemodel
gru_model.save(file1)
```

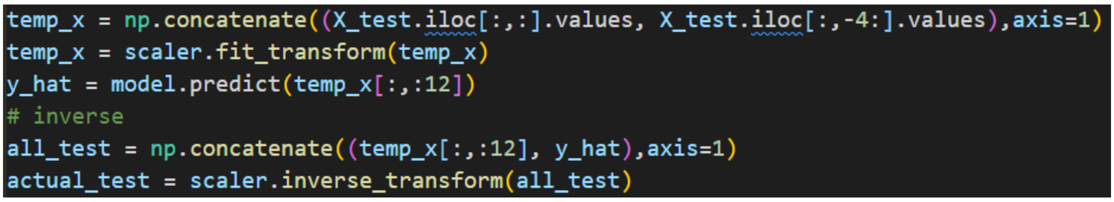
5. 第五步驟為將所訓練過程的 learn error 用圖表所示，判斷是否有 overfitting 或是 underfitting 的狀態。



```
plt.figure(figsize=(12,6))
plt.plot(x)
plt.plot(y)
plt.title('GRU Model loss')
plt.ylabel('MSE')
plt.xlabel('epoch')
plt.legend(['Train_loss', 'Val_loss'], loc='upper right')
```

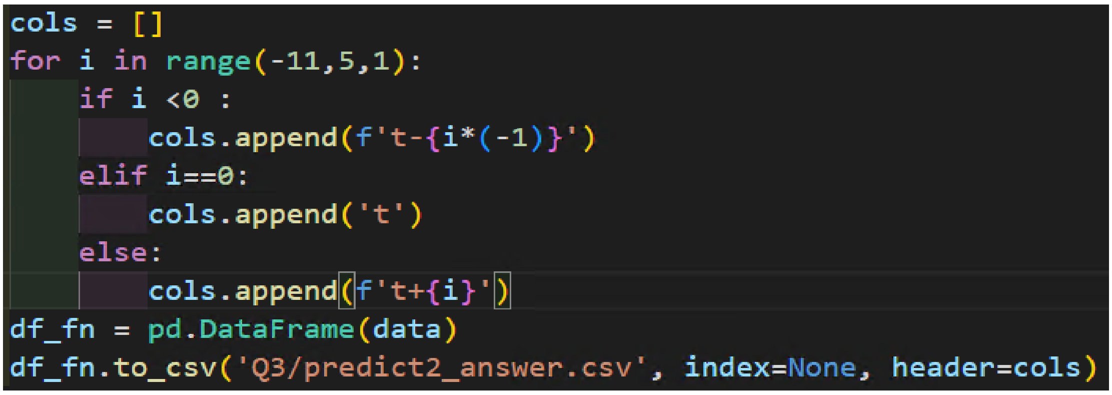


6. 第六步驟將 test_data 使用第 4 步驟所生成的 model 進行預測，進而產生出預測值(t+1,t+2,t+3,t+4,t+5,t+6)。



```
temp_x = np.concatenate((X_test.iloc[:,:].values, X_test.iloc[:,-4:].values),axis=1)
temp_x = scaler.fit_transform(temp_x)
y_hat = model.predict(temp_x[:,:12])
# inverse
all_test = np.concatenate((temp_x[:,:12], y_hat),axis=1)
actual_test = scaler.inverse_transform(all_test)
```

7. 第七步驟為將 test_data 和預測值組合起來輸出成“predict2_answer.csv”。



```
cols = []
for i in range(-11,5,1):
    if i <0 :
        cols.append(f't-{i*(-1)}')
    elif i==0:
        cols.append('t')
    else:
        cols.append(f't+{i}')
df_fn = pd.DataFrame(data)
df_fn.to_csv('Q3/predict2_answer.csv', index=None, header=cols)
```