

## 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：

```
# read csv data
X_train, y_train, X_test = Get_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('Q2/predict1.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 “predict1\_trn.csv” 和 “predict1\_tst.csv” 的檔案當作 train data 和 Test\_data。

```
df_train = pd.read_csv('predict1_trn.csv')
df_test = pd.read_csv('predict1_tst.csv')
X_train, y_train = df_train.iloc[:, :24], df_train.iloc[:, -6:]
X_test = df_test.iloc[:, :24]
```

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

```
df_train = pd.read_csv('predict1_trn.csv')
scaler = MinMaxScaler(feature_range=(0,1))
scaled = scaler.fit_transform(df_train)
X_scaled, y_scaled = scaled[:, :24], scaled[:, -6:]
```

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

LSTM 模型架構程式如下圖所示

```
model = Sequential()
model.add(LSTM(50, return_sequences = True, input_shape = (X.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(50, return_sequences = True))
model.add(Dropout(0.2))
model.add(LSTM(50))
model.add(Dropout(0.2))
model.add(Dense(y.shape[1]))
model.summary()
model.compile(loss = 'mse', optimizer='adam', metrics=['mse']) #rmsprop
```

LSTM 模型架構及參數如下圖所示

```
Model: "sequential_19"
Layer (type)                Output Shape                Param #
=====
lstm_42 (LSTM)               (None, 24, 50)             10400
dropout_44 (Dropout)         (None, 24, 50)             0
lstm_43 (LSTM)               (None, 24, 50)             20200
dropout_45 (Dropout)         (None, 24, 50)             0
lstm_44 (LSTM)               (None, 50)                 20200
dropout_46 (Dropout)         (None, 50)                 0
dense_28 (Dense)             (None, 6)                  306
=====
Total params: 51,106
Trainable params: 51,106
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	90	30

```
Epoch 1/90
1655/1657 [=====>.] - ETA: 0s - loss: 0.0068 - mae: 0.0604
Epoch 00001: val_loss improved from inf to 0.00224, saving model to Q2\predict1.h5
1657/1657 [=====] - 42s 23ms/step - loss: 0.0068 - mae: 0.0604 - val_loss: 0.0022 - val_mae: 0.0371
Epoch 2/90
1656/1657 [=====>.] - ETA: 0s - loss: 0.0034 - mae: 0.0445
Epoch 00002: val_loss improved from 0.00224 to 0.00182, saving model to Q2\predict1.h5
1657/1657 [=====] - 36s 22ms/step - loss: 0.0034 - mae: 0.0445 - val_loss: 0.0018 - val_mae: 0.0333
Epoch 3/90
1655/1657 [=====>.] - ETA: 0s - loss: 0.0022 - mae: 0.0350
Epoch 00003: val_loss improved from 0.00182 to 0.00084, saving model to Q2\predict1.h5
1657/1657 [=====] - 37s 22ms/step - loss: 0.0022 - mae: 0.0350 - val_loss: 8.3583e-04 - val_mae: 0.0223
Epoch 4/90
1655/1657 [=====>.] - ETA: 0s - loss: 0.0015 - mae: 0.0288
Epoch 00004: val_loss improved from 0.00084 to 0.00073, saving model to Q2\predict1.h5
1657/1657 [=====] - 36s 22ms/step - loss: 0.0015 - mae: 0.0288 - val_loss: 7.2879e-04 - val_mae: 0.0199
Epoch 5/90
1657/1657 [=====] - ETA: 0s - loss: 0.0013 - mae: 0.0266
Epoch 00005: val_loss improved from 0.00073 to 0.00071, saving model to Q2\predict1.h5
1657/1657 [=====] - 37s 22ms/step - loss: 0.0013 - mae: 0.0266 - val_loss: 7.0945e-04 - val_mae: 0.0197
Epoch 6/90
1657/1657 [=====] - ETA: 0s - loss: 0.0012 - mae: 0.0249
Epoch 00006: val_loss improved from 0.00071 to 0.00066, saving model to Q2\predict1.h5
1657/1657 [=====] - 36s 22ms/step - loss: 0.0012 - mae: 0.0249 - val_loss: 6.6435e-04 - val_mae: 0.0192
```

```

re_X = X.reshape(X.shape[0], X.shape[1],1)
file1 = 'Q2/predict1.h5'
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]
lstm_model = Build_Model(re_X, y)
history = lstm_model.fit(re_X, y, epochs=90, batch_size=30,
                        callbacks=callbacks_list, validation_split=0.1)
# savemodel
lstm_model.save(file1)

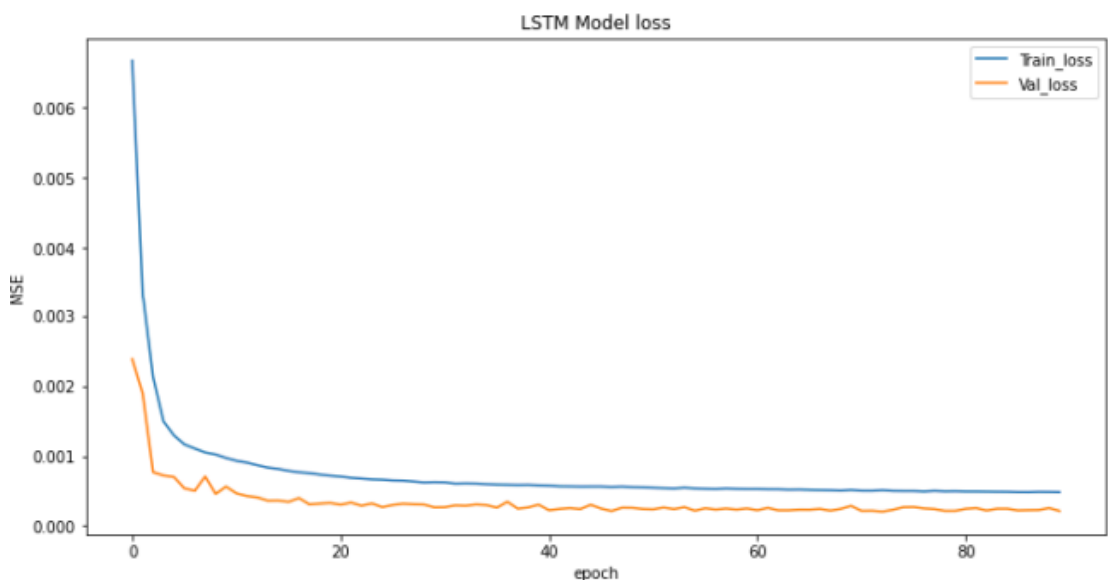
```

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('LSTM 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[:, -6:].values), axis=1)
temp_x = scaler.fit_transform(temp_x)
y_hat = model.predict(temp_x[:, :24])
# inverse
all_test = np.concatenate((temp_x[:, :24], 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 “predict1\_answer.csv” 。

```
cols = []
for i in range(-23, 7, 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('Q2/predict1_answer.csv', index=None, header=cols)
```