







HW5 - LSTM

New York Stock Prediction

- Dataset description
- Goal
 - Predict New York Stock price
 - 希望透過”前N天的股票資料”去預測”隔天的股價”
- Content
 - 約85,0000筆資料來自於 501間公司
- Ref
 - <https://www.kaggle.com/dgawlik/nyse>

程式內容&檔案簡介

1. prices-split-adjusted.csv - 全部股票資料(不含STT)
2. STT.csv - STT股票資料
3. model.py - LSTM 模型架構
4. process.py - 包含資料分割、資料 normalize function
5. train_test.py - 模型訓練、測試

 _pycache_	2019/5/16 上午 0...	檔案資料夾	
 prices-split-adjusted	2019/5/16 上午 0...	Microsoft Excel 逗點分隔值檔案	53,239 KB
 STT	2019/5/16 上午 0...	Microsoft Excel 逗點分隔值檔案	110 KB
 models	2019/5/15 下午 1...	PY 檔案	2 KB
 preprocess	2019/5/16 上午 1...	PY 檔案	2 KB
 train_test	2019/5/16 上午 0...	PY 檔案	3 KB

程式內容說明

- process.py - 包含資料分割、資料 normalize function

```
13 def data_split(stock, seq_len):
14     amount_of_features = len(stock.columns) # 5
15     data = stock.as_matrix()
16     sequence_length = seq_len + 1 # index starting from 0
17     result = []
18
22     result = np.array(result)
23     row = round(0.85 * result.shape[0]) # 85% split
24
35     return [x_train, y_train, x_test, y_test]
```

data_split() 可將資料分割前85%為train、後15% 為test 以及透過 seq_len參數調整抓取前N天資料

Ex. seq_len = 15

那就會抓取前15天股票資料(X)，隔天資料(Y)

```
def normalize_data(df):
    min_max_scaler = preprocessing.MinMaxScaler()
    df['open'] = min_max_scaler.fit_transform(df.open.values.reshape(-1,1))
    df['close'] = min_max_scaler.fit_transform(df.close.values.reshape(-1,1))
    df['high'] = min_max_scaler.fit_transform(df.high.values.reshape(-1,1))
    df['low'] = min_max_scaler.fit_transform(df.low.values.reshape(-1,1))
    df['volume'] = min_max_scaler.fit_transform(df.volume.values.reshape(-1,1))
    return df
```

normalize_data() 可將讀取資料feature做normalize

Dataset Details - prices-split-adjusted.csv

	A	B	C	D	E	F	G
1	date	symbol	open	close	low	high	volume
2	2016/1/5	WLTW	123.43	125.84	122.31	126.25	2163600
3	2016/1/6	WLTW	125.24	119.98	119.94	125.54	2386400
4	2016/1/7	WLTW	116.38	114.95	114.93	119.74	2489500
5	2016/1/8	WLTW	115.48	116.62	113.5	117.44	2006300
6	2016/1/11	WLTW	117.01	114.97	114.09	117.33	1408600
7	2016/1/12	WLTW	115.51	115.55	114.5	116.06	1098000
8	2016/1/13	WLTW	116.46	112.85	112.59	117.07	949600
9	2016/1/14	WLTW	113.51	114.38	110.05	115.03	785300
10	2016/1/15	WLTW	113.33	112.53	111.92	114.88	1093700
11	2016/1/19	WLTW	113.66	110.38	109.87	115.87	1523500
12	2016/1/20	WLTW	109.06	109.3	108.32	111.6	1653900
13	2016/1/21	WLTW	109.73	110	108.32	110.58	944300
14	2016/1/22	WLTW	111.88	111.95	110.19	112.95	744900
15	2016/1/25	WLTW	111.32	110.12	110	114.63	703800

每筆資料包含日期、公司代號、五維feature (開/閉市股價、當天最高/最低股價、當天成交量)

作業流程

1. 讀取”STT.csv”
2. 抓取”公司(STT)股票資料”
3. 將其做normalize
4. 將”STT”資料分割
 - train, test 資料集
 - 前N天股票資料(X)和隔天預測股價(Y)
5. 訓練模型、測試模型

```
15 df = pd.read_csv("./STTcsv", index_col = 0)
16
17 STT = df[df.symbol == 'STT'].copy()
18 #print(GOOG)
19 STT.drop(['symbol'],1,inplace=True)
20 STT_new = normalize_data(STT)
21 #print(GOOG_new)
22 window = 15
23 X_train, y_train, X_test, y_test = data_split(STT_new, window)
```

訓練&測試資料

- 訓練資料：
 - 全部公司股票資料 and STT前85%的股票資料
- 測試資料：
 - STT後15%股票資料

作業要求

- 繳交時間: 5/29 11:59pm.
- 設計一個合理的方法，某A公司隔天預測股價，例如：
 - 用某A公司前15天的資料，預測第16天的結果
 - 或用某50家跟A同領域的公司前15天的資料，預測第16天的結果
 - 或用全部股市公司的資料，預測某A第16天的股價
 - 或用某A公司前30天的資料，預測第31天的結果
 -
- 設計你的LSTM模型 / 或直接用助教提供的
- 畫出訓練loss與測試accuracy curves
- 以測試資料每一天為單位：
 - 計算每天(預測股價-真實股價)²的總和，回報估測誤差