

Please read the information below **carefully** before checking or running the Code.

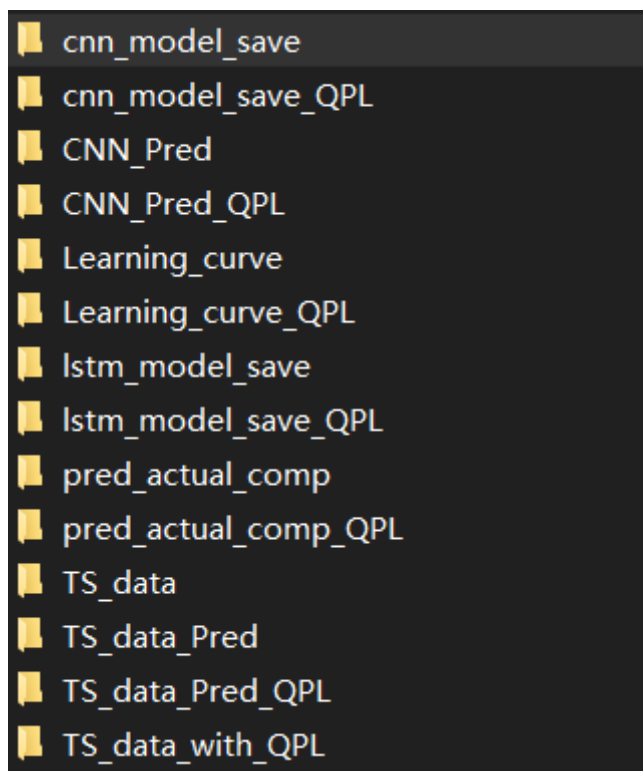
This folder contains the source code and the executable files of our forecast system. Our forecast system is written in *Python 3.7.6*. The system is implemented using the *Jupyter Notebook IDE*.

There are two folders inside:

1. Jupyter Notebook: This folder contains the source file and executable in Jupyter Notebook. The file format are .ipynb files, which can be opened and viewed using Jupyter Notebook. We recommend you use Jupyter Notebook to check and run the code, because this system is originally written in Jupyter, which can provide step by step experience. You should install Jupyter Notebook if you want to use it, the installation method can be searched on internet
2. Python Source Files: These are the .py format python source files that are exported by Jupyter Notebook, you can directly run them using the terminal.

Follow the steps below one by one to run this program:

1. Make should all these folders are created before you trying to run the code, or it will cause error (Already prepared):



2. Make sure that you have install these python packages: *pandas*, *numpy*, *torch*, *matplotlib*

3. The order of running code:

There are two sets of system's codes inside, one set is training the CNN with QPL and the other is training without QPL. All the files with postfix “_QPL” are related to the system where QPLs are used as training input. In this case, if you want to train the CNN with QPL, then the running order is:

- a) QPL_Calculation
- b) CNN_withQPL
- c) LSTM_withQPL

If you want to train the CNN without QPL, then the running order is:

- a) QPL_Calculation
- b) CNN
- c) LSTM

4. Run on Jupyter Notebook:

We use the **system with QPL as an example**:

Click the run bottom, then you can run the program step by step.



4.1 Calculate the QPL:

In this cell, you can set the product name and generate the excel file with QPL of this product

Load Data

```
product = 'AUDUSD'
fileName = product + '1440_Data.csv'

ts = pd.read_csv('TS_data/' + fileName)

ts = ts.dropna(axis=0, how='any')
#print(ts.isna().any())

print(ts.shape)

DT_CL = [i for i in ts.values[:, -7]]
DT_CL
```

You can also set the product list and calculate the QPL for a lot products by running the cell below “All in one”:

```
: product_list = ['AUDUSD', 'AUDCAD', 'AUDCHF', 'AUDNZD', 'EURCHF', 'GBPNZD', 'NZDCAD', 'NZDC
```

All in one

```
: #df_factor = sp.LoadData("ashare", 'S_DQ_CLOSE')
for k in range(len(product_list)):
    product = product_list[k]
    fileName = product+'1440_Data.csv'

    ts = pd.read_csv('TS_data/'+fileName)

    ts = ts.dropna(axis=0, how='any')
    #print(ts.isna().any())

    print(ts.shape)

    DT_CL = [i for i in ts.values[:,-7]]

    K = []

    from sklearn.metrics import mean_squared_error
```

The excel files with QPLs will be saved in folder ./TS_data_with_QPL

4.2 Use CNN to process the data with QPL

In this cell, the first product in the list is the product data you want to process (for example, process the data of AUDUSD):

Define Parameters

```
product_list = ['AUDUSD', 'AUDCHF', 'AUDCAD', 'NZDCHF', 'USDCHF', 'NZDCAD', 'EURCHF', 'AU

para_num = 4
ts_data_save = 'TS_data/'
model_save = 'cnn_model_save/'

# Hyper parameters
EPOCH = 5000
learning_rate = 1e-2

# Network Parameters
CNN_input_dim = 1
CNN_output_dim = 4
BATCH_SIZE = 5

train_ratio = 0.8
```

Run the code step by step, in this cell, the training will start and needs to wait for a while:

```
if __name__ == '__main__':
    model_name=product+'.pkl'
    train_loss, losses = train_model(train_X, model_name)
```

After the training, the result will be saved to folder ./CNN_Pred_QPL

4.3 Forward the processed data into LSTM:

In this cell, you can set the product you want to train:

DataLoader

```
]: product = 'NZDCHF'
ts_d = pd.read_csv(ts_data_add+product+'.csv')
# Change Header

#new_index = header
#ts_d.columns = new_index
ts_d = ts_d.dropna(axis=0, how='any')
# ts_d = ts_d.drop(['QPL1', 'QPL2', 'QPL3', 'QPL4', 'QPL5', 'QPL6', 'QPL7', 'QPL8', 'QPL9',
#                  'QPL11', 'QPL12', 'QPL13', 'QPL14', 'QPL15', 'QPL16', 'QPL17', 'QPL18',
#                  'QPL-5', 'QPL-6', 'QPL-7', 'QPL-8', 'QPL-9', 'QPL-10', 'QPL-11', 'QPL-
#                  'QPL-20', 'QPL-21'], axis=1)
# Minimize the first three days
n_samples = len(ts_d) - window_size+1
print(n_samples)
# data.shape: (len(ts_w)*4*4)
data = np.zeros((len(ts_d), window_size, input_dim), dtype=float)
```

Run the code step by step, in this cell, the training will start and needs to wait for a while:

```
if __name__ == '__main__':
    model_name=product+'.pkl'
    train_MSE, losses, train_RMSE = train_model(train_X, model_name)
```

After the training, the prediction result will be saved in ./TS_data_Pred_QPL

5. Run directly using python:

5.1 Open a terminal in the same folder with the .py file

5.2 As example, type **python CNN.py** to run the program

5.3 The parameters setting and training steps are similar to the steps above.