STOCK TRADING STRATEGY BASED ON NEURAL NETWORKS

HKU MSCCS 3035249227

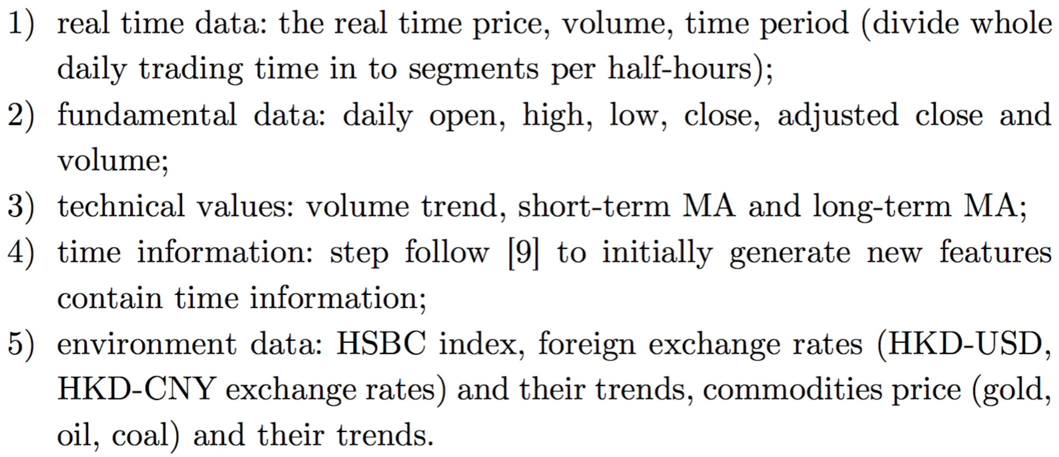
ZHAN HUI

CONTENT

1. Purpose
2. System structure
3. Convert to tradable strategies
4. Current NN

1. PURPOSE

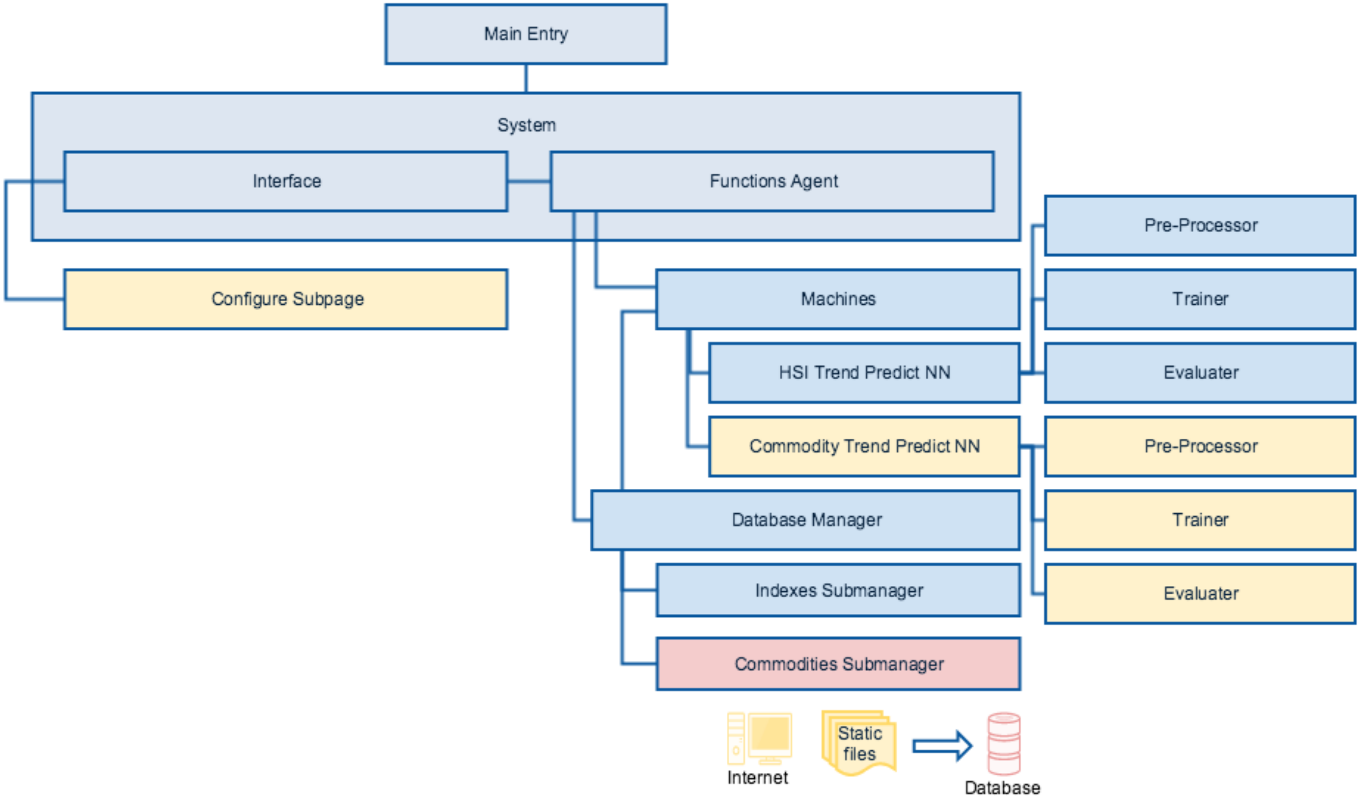
- **Learn from historical trading information to recommend trading strategies for a specific market situation:**
  - Input:



1) real time data: the real time price, volume, time period (divide whole daily trading time in to segments per half-hours);
2) fundamental data: daily open, high, low, close, adjusted close and volume;
3) technical values: volume trend, short-term MA and long-term MA;
4) time information: step follow [9] to initially generate new features contain time information;
5) environment data: HSBC index, foreign exchange rates (HKD-USD, HKD-CNY exchange rates) and their trends, commodities price (gold, oil, coal) and their trends.

- **Meanwhile training out well performance NN and using it to predict price trend.**

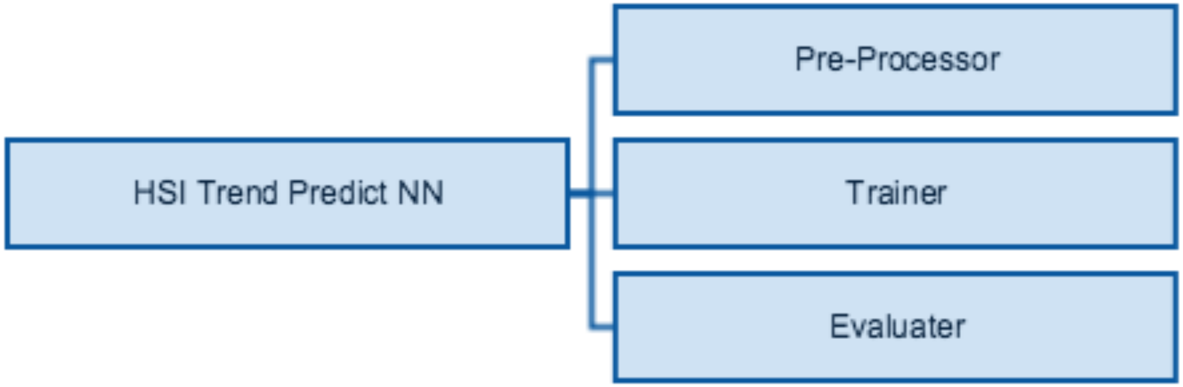
2. SYSTEM STRUCTURE



- degree red : develop now;
- degree yellow : planned to develop next step;
- degree blue : developed.

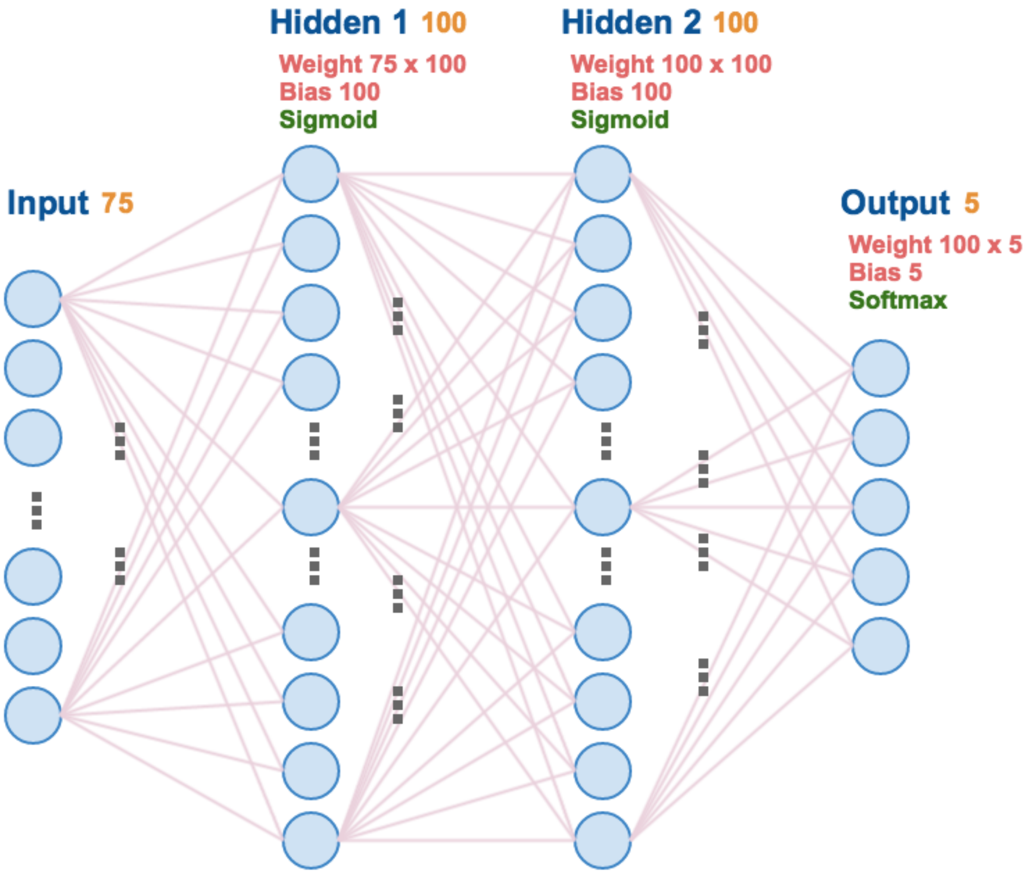
3. CONVERT TO TRADABLE STRATEGIES

- **Separate the part of well performance to do trend predict.**



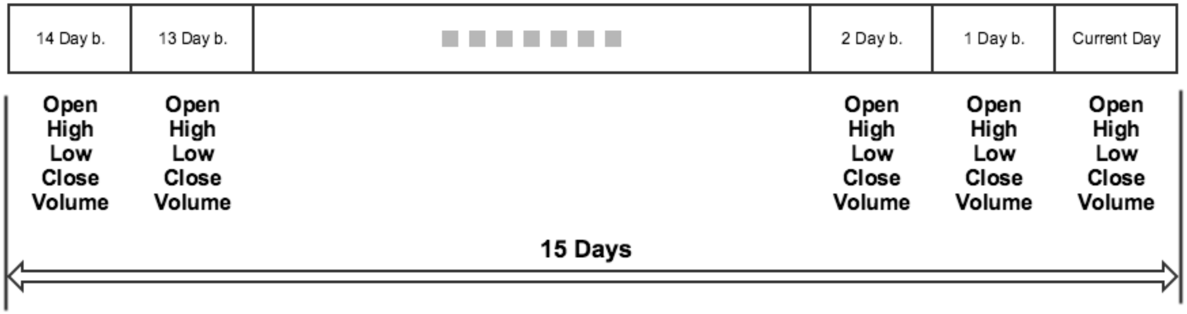
4. CURRENT NN

- HSI daily trend predict NN.
- Structure:

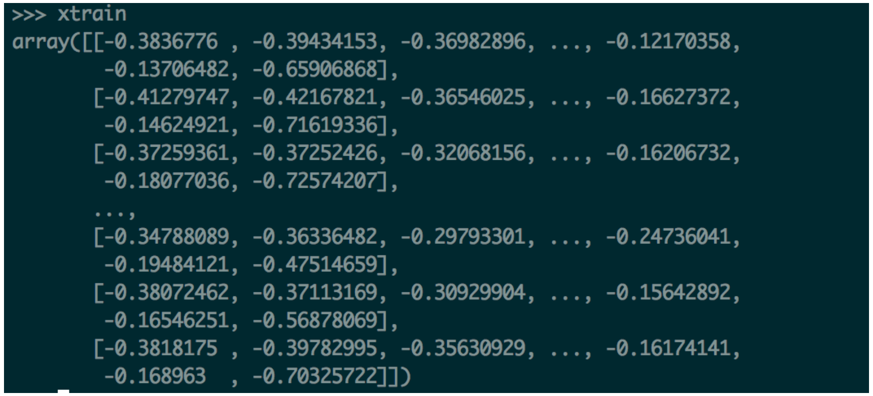


4. CURRENT NN

- **Input:**
  - 15 days' OHLCV data (1 year).



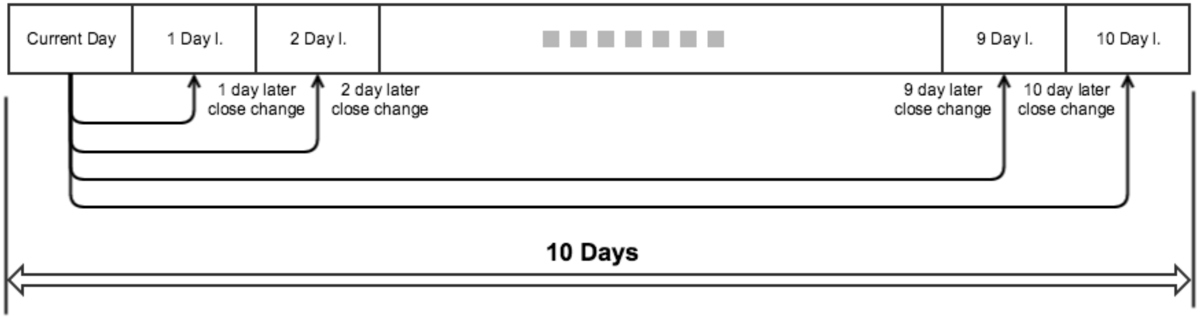
  - Preprocess the data, making of data in the range of [-1,1].



```
>>> xtrain
array([[-0.3836776 , -0.39434153, -0.36982896, ..., -0.12170358,
        -0.13706482, -0.65906868],
       [-0.41279747, -0.42167821, -0.36546025, ..., -0.16627372,
        -0.14624921, -0.71619336],
       [-0.37259361, -0.37252426, -0.32068156, ..., -0.16206732,
        -0.18077036, -0.72574207],
       ...,
       [-0.34788089, -0.36336482, -0.29793301, ..., -0.24736041,
        -0.19484121, -0.47514659],
       [-0.38072462, -0.37113169, -0.30929904, ..., -0.15642892,
        -0.16546251, -0.56878069],
       [-0.3818175 , -0.39782995, -0.35630929, ..., -0.16174141,
        -0.168963  , -0.70325722]])
```

4. CURRENT NN

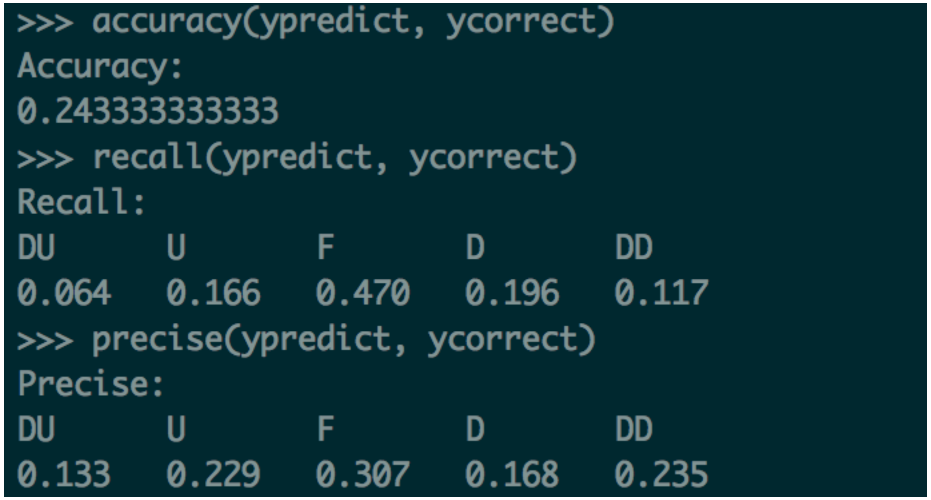
- **Output(labels):**
  - 10 days’ close price change rate.



  - Preprocess the data, dividing the change rate into 5 classes, namely DU-dramatic up, U-up, F-fluctuate, D-down and DD-dramatic down.
  - E.g. [1,0,0,0,0] means DU; [0,0,1,0,0] means F.

4. CURRENT NN

- **Result:**
  - Use accuracy, recall and precise to evaluate the NN.



```
>>> accuracy(ypredict, ycorrect)
Accuracy:
0.243333333333
>>> recall(ypredict, ycorrect)
Recall:
DU      U       F       D       DD
0.064   0.166   0.470   0.196   0.117
>>> precise(ypredict, ycorrect)
Precise:
DU      U       F       D       DD
0.133   0.229   0.307   0.168   0.235
```

  - That means the current NN performs bad.

4. CURRENT NN

- **Improvement:**
  - Try to use longer training data (e.g. 3 years); shorter training data (e.g. 5 months);
  - Try to use different label mechanism (e.g. 2/3 classes);
  - Try to use different Input data (e.g. remove OHL price and rest close price and volume only);
  - Try to add some new features (e.g. SMA indicators);
  - Try add more hidden layers/neurons.

- **Most Interested:**
  - Try another NN, current NN is a simple forward propagation NN, try implement RNN and CNN.

THE END

THANKS