***Feature Extraction Using Restricted Boltzmann Machine for Stock Price Prediction***

Group 9: Jing Zhou, Lulu Deng, Linna Hu

* **The project objectives**

Restricted Boltzmann Machine (RBM) as an unsupervised Artificial Neural Network (ANN) model for machine learning algorithm that are applied to extract highly discriminative low-dimensional features and improve forecasting accuracy of the regression models.

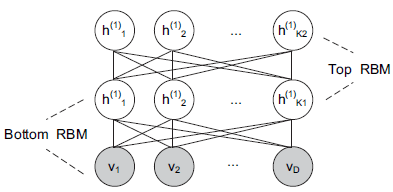
The main contribution of this paper is that it presents a combined approach to predict stock price, which utilize the RBM and its deep form DBN as the feature extractors and the SVM as the regression model.

1. **Background**

**Using RBM as a Feature Extractor**

can be viewed as the extracted features of v (since hidden units model correlation of dataset). Extracted features can be input for training another RBM to capture higher order abstract features; repeatable as many times as needed to build deeper network. DBN Training:

1. Train the bottom RBM with input v.

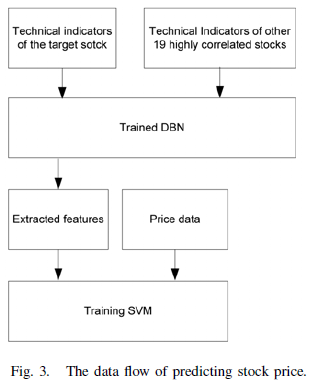
2. hidden units of bottom RBM become deterministic, which output with input v and serve as the visible units for the top RBM.

3. Repeat 1 and 2 for as many layers as desired (more = higher accuracy). Backpropogation (weights, biases) then used to improve extracting performance.

* **Techniques and Tools**

1. **Experiment**
   1. **Steps of the Experiment**

Utilize the RBM and its deep form Deep Belief Network as the feature extractor, and the Support Vector Machine as the regression model.

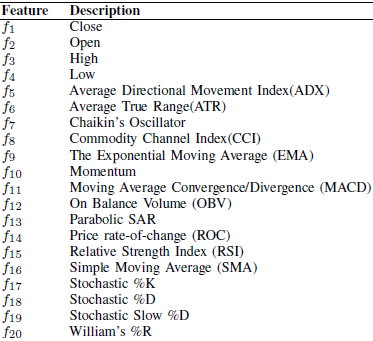
We use the SVM for regression with the input data consisting of *f1~f20* of S&P 500. (SVM+1)

We use the SVM for regression with the input data consisting of f1~f4 of S&P 500 index and f5~f20 of the 20 stocks. (SVM+20)

We train the DBN using *f5~f20* of the 20 stocks, and then use the extracted features from DBN and *f1~f4* of S&P 500 as the input of SVM. (SVM+DBN)

* 1. **Initial Data Set**

The set of data we used was historical price information downloaded from Yahoo Finance as Excel csv files. Besides the S&P information fundamental to our prediction of its own price, the nineteen other stocks we chose were mostly from the information technology sector. These were: ACN, ADP, AKAM, AMAT, CRM, CSCO, CTSH, CTXS, EBAY, FIS, GOOGL, HP, INTU, MA, NFLX, NTAP, T, VZ, WU for the time period ranging from 11/16/2006 to 11/14/2017. This immediately covered features f1-f4.

* 1. ** Input Features: 20 Technical Indicators**

After gathering the initial dataset, excel functions were utilized to produce formulas that calculated the indicators for every stock. Below is a brief summary.

* 1. **Using SVM for Regression**

Support Vector Machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis:

*SVC: find a hyperplane from which the distances of nearest sample points tend to be largest*

*SVR: find a hyperplane from which the distances of furthest sample points tend to be smallest*

* 1. **Framework of our Python Replication**

We decided to use the Tensorflow library in Python, an extremely popular one used in Machine Learning.

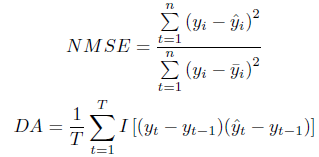
There are two parts to our code.

A) The basic SVM prediction is in the **prediction.py** file.

We import train\_test\_split from sklearn.cross\_validation to split the dataset into training set and test set, and then, from sklearn.svm we import SVR and do the the regression.

B) The Gaussian-Bernoulli Restricted Boltzmann Machine is the in **gbrbm.py** file.

We import the Tensorflow library and do the training. First we define the rbm function in **rbm.py** containing the fitting, reconstruction and initial conditions, based on which, we define the gbrbm function which contains the training process.

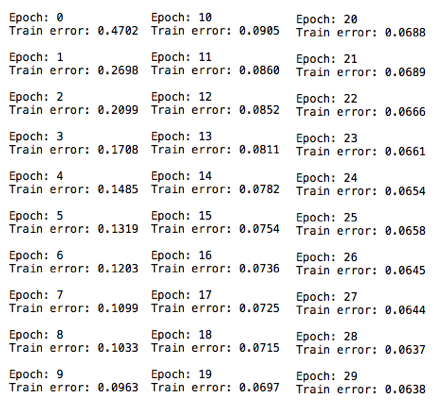
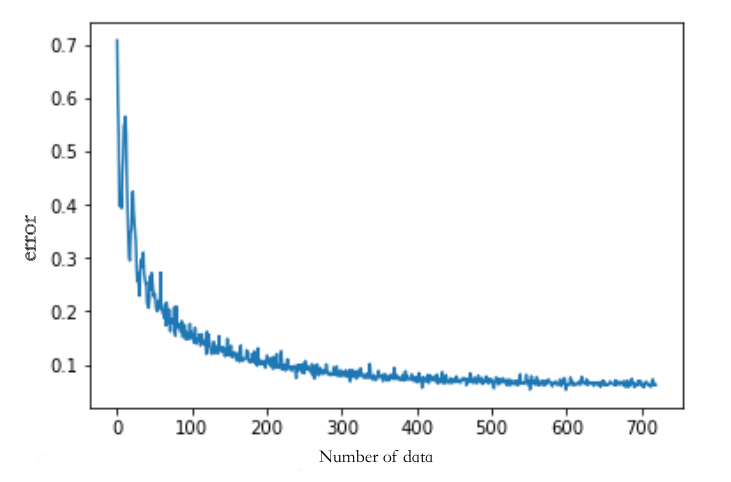
* 1. ** Performance Measurement**

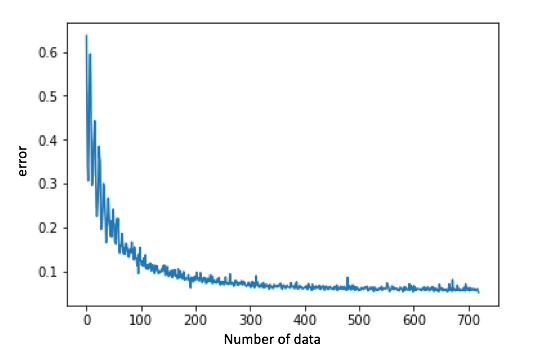
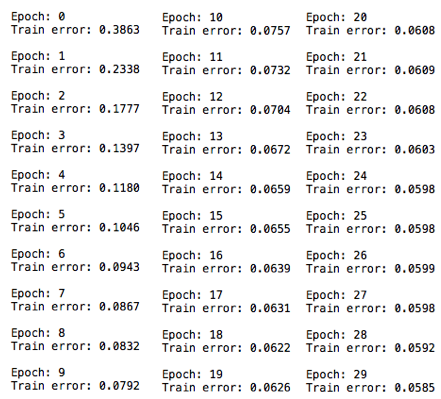
***Normalized Mean Square Error*** - measure of the deviation between the actual and predicted values. The smaller the value of the NMSE, the closer are the predicted values to the actual values.

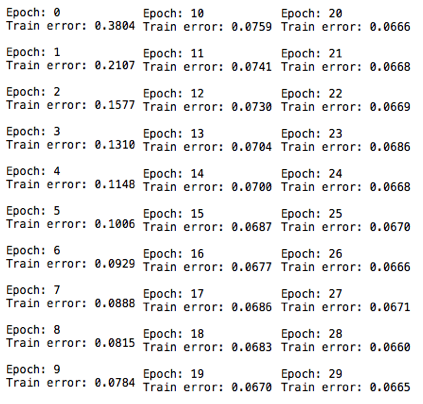
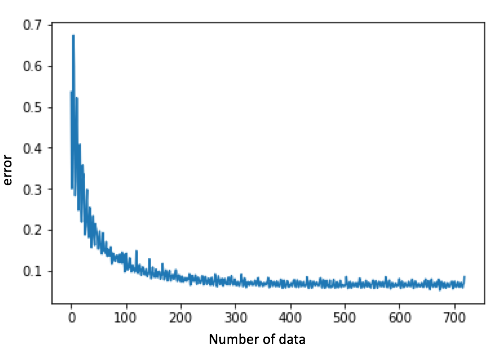
***Directional Accuracy*** - provides an indication of the correctness of predicted direction of stock price.

* **Results and Conclusions**
  1. **Training Output**

Set 320 visible nodes and 2000, 4000 and 8000 hidden nodes respectively to the machine and calculate the err of each step within 30 epochs and the batch size of 90 with the following formula:

****

** **

****

According to the graphs above, the errors converges to a certain value, which means out training is plausible.

* 1. **Results**

|  |  |  |
| --- | --- | --- |
| Methods | NMSE | DA |
| SVM+1 | 0.0094718 | 98.40810% |
| SVM+20 | 0.0158766 | 97.25036% |
| SVM+DBN(2000) | 0.0117525 | 97.39508% |
| SVM+DBN(4000) | 0.0119996 | 97.25036% |
| SVM+DBN(8000) | 0.0120328 | 97.53979% |

* 1. **Conclusion**

Comparing SVM+1 and SVM+20, the generalization performance of SVM is deteriorated when dimension of input is increasing even though more input variables contain more information

SVM+DBN converge to a small NMSE with a higher DA on the test set than SVM+1 and SVM+20, which means using DBN as feature extractor would greatly facilitate regression.

* 1. **Individuals contributions**

All of us were very engaged in this project. The code, presentation slides and the report were the results of our hard working and cooperating. Everyone have done their own job well.

Particularly, Jing came up with the idea for the topic and did SVM regression and conclusion part. Lulu did the background introduction and importing some function packages and calculating part. Linna did the data processing and calculating part. We all did presentation and wrote code for our own parts. Finally, we worked together for this report.

***Reference***

*Paper:*

*“Feature Extraction Using Restricted Boltzmann Machine for Stock Price Prediction”*

*By Xianggao Cai, Su Hu, and Xiaola Lin*

*Indicators:  
http://www.stockcharts.com*

*Code:*

*https://github.com/meownoid/tensorfow-rbm*