Investment Society, QF department, Machine Learning -- Report

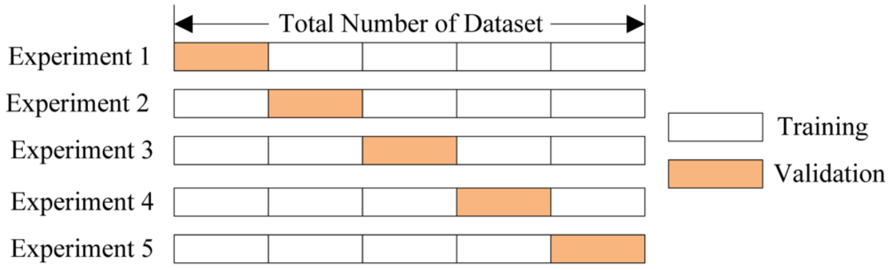
Author: Dong Shaocong, deputy head, machine learning team

Introduction

Utilizing machine learning, or predictive models in general, can give people a statistical understanding of the future trends, which can be very useful for financial analysts in general, and quantitative traders in particular. In this semester, we implemented the machine learning model – multi-layer perceptron with the aid of *scikit-learn* machine learning framework to examine the effectiveness of its usage in NIKKEI index value prediction.

In finance, most data are presented in a time series format, which means the order of the data points are important and cannot be ignored during analysis. In time series machine learning, there are several pitfalls, which we want to avoid:

1. *Data snooping* bias. Data snooping occurs where testing data has been used before the test phase is conducted. To avoid that, we order the data chronologically and combine to form training and testing data separately.
2. *Cross validation*. For validation purposes, we need to have valid training and validation set for each round of validation. As time series data has an intrinsic order, we can only give later time values as validation data as compared to the training set.
3. *Window size* parameter. In time series prediction, walk-forward analysis should be used. A window size of 5 has been chosen so that the training features can capture enough past information to get a better prediction. We shouldn't choose a window size that is too big to result in over fitting, which is one of the most frequent issues in time series forecasts. A more demonstrative graph is given below.



Data collection

We collect the core financial data from quandl platform including NIKKEI index, japan's monthly inflation, annual unemployment rate and annual population, and annual participating number of labors. These are the features we believe will help predict the index more precisely.

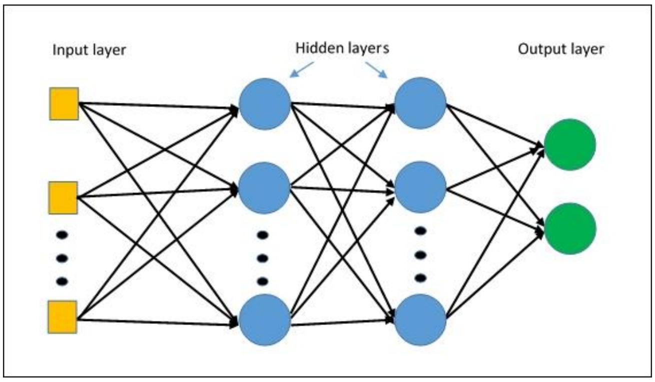
Data processing

As the data given are annual, or monthly or daily (as given by NIKKEI index), we append the different data to different date in NIKKEI index by their closest month's inflation data and the closest year's unemployment rate, total population and the number of labors. There is no need to scale the data as the data for each feature are of the same unit or scale. To avoid data snooping, we will refrain ourselves from doing any normalization.

Comparing against the data we have, we choose a time horizon of January 2010 to December 2016. Combining all the data into one big feature matrix and one column of labels, we split the 1712 data points into training set and testing set. For a rough proportion of 1/10 of testing data, we arranged the most recent 200 data points as the testing set, whereas the rest are training set.

Modelling & Prediction

Neural network is one of the most popular model in machine learning. In this analysis, we chose one of the most basic form of feedforward neural network with gradient descent optimization – multi-layer perceptron regressor. *scikit*-learn framework is used.



From the left, we can get an idea of how multi-layer perceptron (MLP) works. Input has been feed into the network with different weights and continue into the hidden layers and finally output to the output layer, which is the NIKKEI index value in our case. The weights are the main parameters the model "learns" from the data.

We implement this model in Jupyter notebook, after validation, we get a R^2 score of 0.69, which is quite close to 1, which is much better than a constant model (always output the expected value of the labels, and of score 0).

Limitation & Further Work

In this analysis, we have several limitations which are addressed here:

1. The data points are not enough. We should have more data points to get a better approximation.
2. The features are not enough, there are more data available by industry, by gender and by age groups. More features will add more predictive power.
3. More rounds of cross validation should be used. And the model default parameters should be tuned as well.

REFERENCES

1. Stack Overflow (2017) *How to implement work forward* https://stackoverflow.com/questions/45459102/how-to-implement-walk-forward
2. Multi-layer perceptron Regressor (2017) http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html
3. Multi-layer perceptron (2017) https://www.safaribooksonline.com/library/view/getting-started-with/9781786468574/ch04s04.html