FRE-GY 7773

Yuqing Wang yw3637

Instructor: Jifei Wang & Perry Ken

May 17, 2019

Final Project

Momentum Strategies on Stock Trading

1. Introduction

1.1 Main Purpose

The goal of the project is to develop a momentum trading strategy based on the given features. We divided the whole time period from Feb 2008 to July 2017 into many small trading intervals including 20 days each. At the beginning day of each trading interval, we used model to estimate the probability of each stock with which the stock could beat the market. Based on that probability, we long top 10% of the stocks with the highest probability and short the worst 10% of the stock with the lowest probability. Then we used following formula to calculate the return of our portfolio for each interval and the return for the whole year. Sharp ratio would also be reported in the project.

1.2 Features

Cumulative sum of past 20 trading days return ("ret raw") is used as the basic features. Ret raw norm is generated by normalization of the return from different stock of the same day. Based on ret raw norm, ret raw norm of the 20 previous days and of the 13 previous month is used as feature as well. Besides, an indicator stands for whether the date is January is used in the model as well.

1.3 Model and probability

Logistic regression is used as baseline model for the project. Random forest and Recurrent Neural Network are also used in the project to see whether it would behave better than baseline model. We used these three models to generate the probability we

need for trading. Although our goal is to generate largest out of sample return, the accuracy of predicting whether a stock will beat the market is used as the criterial to select models. We use the data from 2008 to 2012 as our training set and data in 2013 as validation set. The model who can give an accurate prediction of the stock in 2013 based on data from 2008 to 2012 will be employed to generate the probability we use in momentum trading.

2. Baseline Model

Logistic regression is chosen as baseline model. Except that "lbfgs" is chosen as the solver parameter, default parameters are used in our baseline model.

2.1 In Sample Training

Data from 2008 to 2012 is used as training data to train logistic regression model which is used to calculate baseline accuracy, cross entropy, return and sharp ratio for each year. In sample accuracy is 51.76% and cross entropy is 0.6924. The return and sharp ratio are shown in the following chart:

Year	**2008**	**2009**	**2010**	**2011**	**2012**	**Average**
Return	-0.0567	0.3612	0.4093	0.1341	0.0930	0.1882
Sharpe	-0.6923	3.6762	3.2246	2.1182	1.4294	1.9512

2.2 Out of Sample Testing

First of all, one point has to be clarified before discussion of out of sample data. When it comes to out of sample accuracy and cross entropy, model is trained with data from 2008 to 2012 as mentioned above and test with data in 2013. But when it comes to out of sample return and sharp ratio, as back-testing method is employed, data from five year ahead each year from 2013 to 2017 is used to train model and data of that year is used as test data. So, the test return and sharp ratio should be two sequences of 5 numbers each with result from 2013 to 2017.

Out of sample accuracy is 49.48% and out of sample cross entropy is 0.6940. The return and sharp ratio are shown in the following chart:

Year	**2013**	**2014**	**2015**	**2016**	**2017**	**Average**
Return	0.11485	0.09826	0.09775	-0.0384	0.01564	0.0576218
Sharpe	1.68092	1.81847	2.22334	-1.3674	1.58753	1.1885758

2.3 Short Conclusion

Baseline model gives us a normal result in sample accuracy is a little larger than 50% and out of sample accuracy is a little smaller 50%. In sample cross entropy is a little smaller than out of sample cross entropy, indicating overfitting does not exist and this model can generate a small return both in sample and out of sample.

3. Random Forest Model

Random Forest is chosen as the traditional model to estimate the probability. One of the most important reason is that Random Forest could run at fast speed which is convenient for us to change the parameter.

N estimators: the number of trees in the forest with a default value equaling to 10.

Max depth: this is a parameter to certify the depth of the tree: If max_depth is too small, the model could not catch enough information and if max_depth is too large, overfitting problem will arise.

Min sample split: The minimum number of samples required to split an internal node with a default value of 2. Only when there are one or two sample left in a node, splitting will stop.

These are the few important parameters to determine in the project. Besides, we set n jobs to -1 for fast computing and give default value to other parameters such as bootstrap.

We use a naïve method to determine the parameter one by one: when we want to estimate one parameter, we assume others to be constant. This is not the best way, but it is efficient when the computing ability is considered.

3.1 Random Forest with Default Parameter

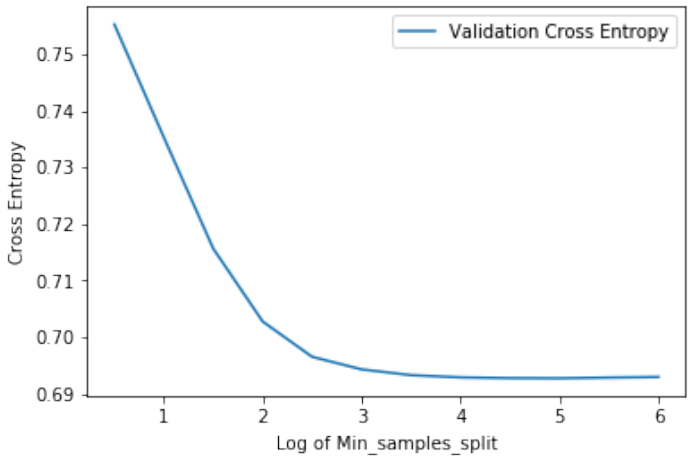
To get a set of good parameters of random forest, the performance of default parameter of random forest should be considered. The same method is employed in random forest with baseline logistic model except set splitting. As we want to use data in 2013 as test set, we should change training set to data from 2008 to 2011 and use data from 2012 as validation set. In sample accuracy is 98.69% in contrast with

50.07% out of sample error. In sample cross entropy is 0.2116 while out of sample cross entropy equals to 0.7611.

Conclusion could draw that there exists an obvious overfitting in default random forest model: In sample accuracy is nearly 100% which is not reasonable while out of sample accuracy is only 50%. And in sample cross entropy beat that of out of sample by a huge amount. So, the direction of parameter adjustment is to make random forest less complicated to avoid overfitting.

3.2 Min Sample Split

To avoid overfitting, min sample split should be larger while max depth should be smaller. There is no obvious answer which parameter should be adjust first. Min sample split is chosen first simply because default number 2 is obviously too small. The graph below shows how in sample and out of sample change with min sample split.



Firstly, it is important to clarify that cross entropy is the indicator on which we select model other than accuracy because only the stocks with highest probability to rise or to fall will be used in momentum trading strategy. A conclusion could be drawn that out of sample entropy decrease before 4.0 and remains almost the same after 4.0. So, 10000 is chosen as Min Sample Split parameter.

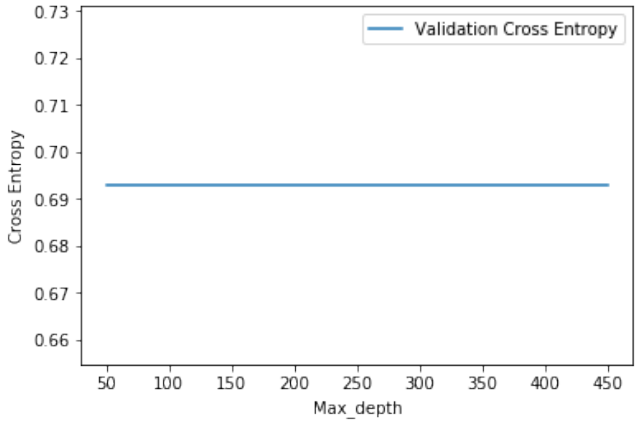
Min Sample Split (10^)	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6
In Sample Accuracy	0.99	0.97	0.90	0.79	0.69	0.62	0.57	0.55	0.53	0.52	0.52	0.50
Out of Sample Accuracy	0.50	0.50	0.50	0.51	0.50	0.51	0.50	0.51	0.51	0.51	0.51	0.50
In Sample Cross Entropy	0.22	0.32	0.47	0.59	0.65	0.67	0.68	0.69	0.69	0.69	0.69	0.69
Out of Cross Entropy	0.76	0.74	0.72	0.70	0.70	0.69	0.69	0.69	0.69	0.69	0.69	0.69

3.3 Max Depth

Based on above result, different Max Depth will be test for random forest model.

Max Depth	50	100	150	200	250	300	350	400	450
In Sample Accuracy	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
Out of Sample Accuracy	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
In Sample Cross Entropy	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
Out of Cross Entropy	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69

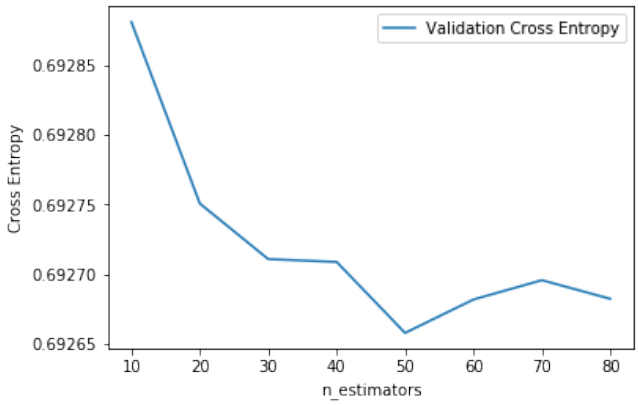
It is surprise that max depth has no effect to validation cross entropy. One of the reasons is there is some relationship between min-sample-split and max depth. The smaller min sample split is, the larger max depth would be. However, the documentation fails to give an explicit explanation how to deal with the conflict between min-sample-split and max depth. Consequently, the default parameter, 10, should be a solid choice.



3.4 N Estimator

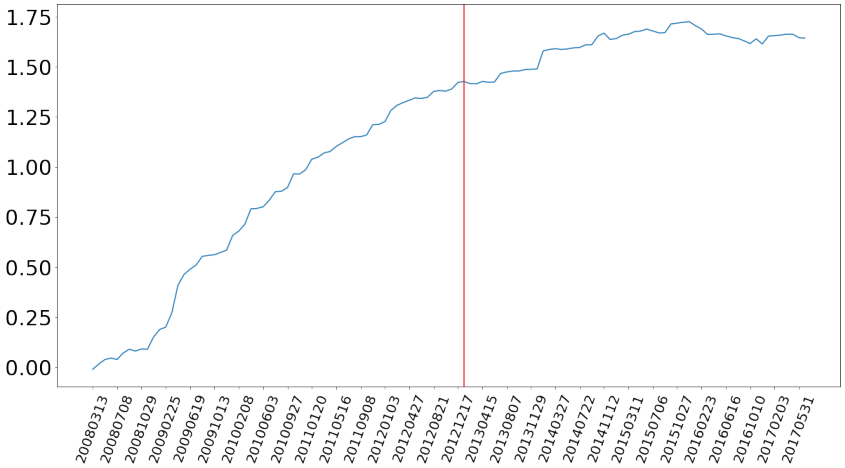
With the increase of N Estimator, validation Cross Entropy will decrease sharply at first, but it will remain almost the same afterward. So, we chose 50 as N-Estimator which gives smallest cross entropy in the experiment.

N Estimator	10	20	30	40	50	60	70	80
In Sample Accuracy	0.54	0.55	0.55	0.55	0.55	0.55	0.55	0.55
Out of Sample Accuracy	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
In Sample Cross Entropy	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
Out of Cross Entropy	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69



3.5 Final Random Forest Model

After setting min sample split, Max Depth and N Estimator, final random forest model could be decided. In sample accuracy equals to 53.95% and out of sample accuracy equals to 50.21%. In sample cross entropy equals to 0.6904 and out of sample cross entropy equals to 0.6933. Final random forest model beat baseline model both in accuracy and cross entropy by a tiny amount.



Our final random forest model beat baseline model in both return and sharp ratio both in sample and out of sample

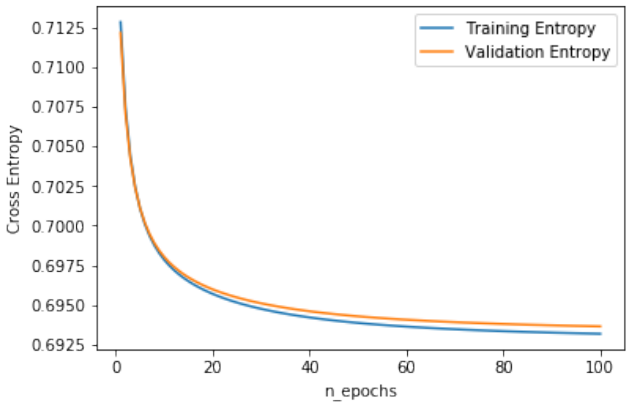
Year	**Return**	**Sharpe**	**Year**	**Return**	**Sharpe**
2008	0.085528	1.633732	*2013*	0.143764	2.307007
2009	0.442348	4.280952	*2014*	0.126868	1.638843
2010	0.423093	4.483873	*2015*	0.1348	3.167101
2011	0.176626	3.760834	*2016*	-0.107631	-2.578657
2012	0.183214	3.713652	*2017*	0.04066	1.762261
Average	0.2621618	3.5746086	*Average*	0.0676922	1.259311

4. Neuron Network Model

For model to predict time series data, recurrent neuron network would always be the best choice. However, it is not reasonable anymore to use recurrent neuron network here because none of the features is current period features; all of them are lagged features. So, employing normal deep learning model will find the hidden relationship between information before and the change of stock return. We have set learning rate to 0.01, layer to 2 and number of neurons per layer is 20.

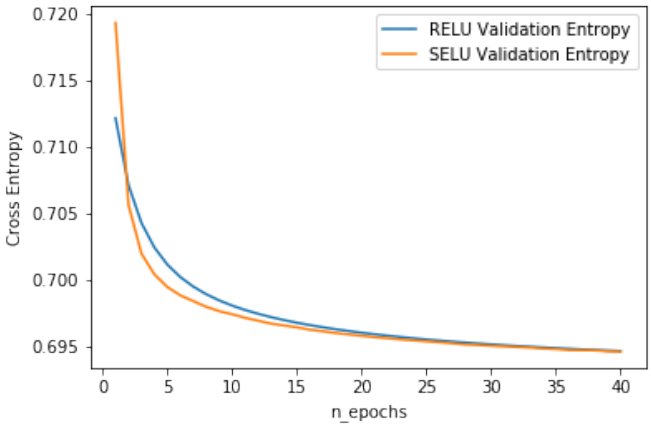
4.1 Number of Epochs

The Number of Epochs of epochs should be decided upon learning rate.



As shown in the graph, cross entropy keeps on decreasing with increment of number of epochs, but it increases more and more slowly. According to the graph, 40 should be a good choice.

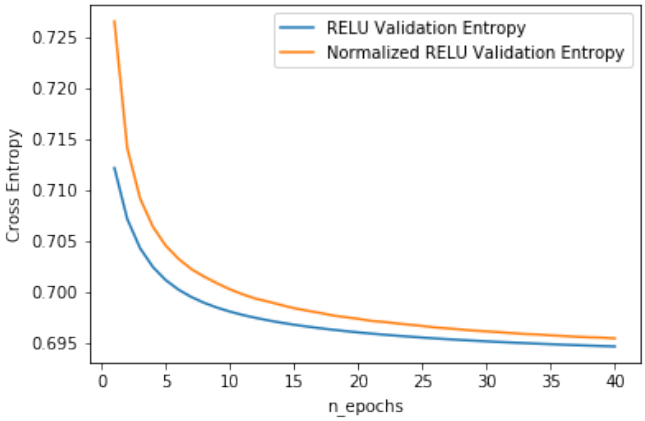
4.2 Activation Function: SELU or RELU



From the graph, it is obvious after training for a while, SELU activation function behaves better than RELU activation function for reason that neurons will never die in SELU activation function model. So SELU activation function is used in deep neuron

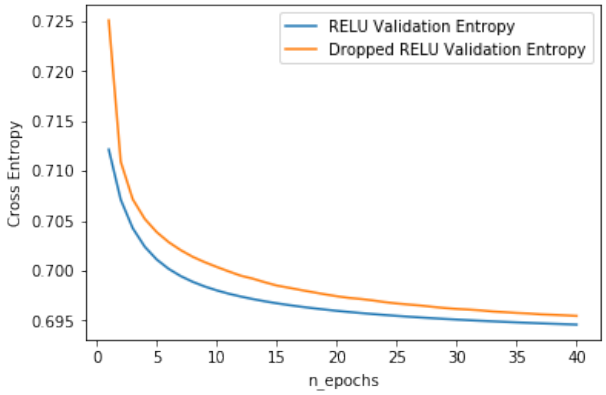
network model in the next parts.

4.3 Batch Normalization



Batch normalization is useless in this situation. Firstly, the input data is lagged normalized return and there are only to layers, so the parameters would still stay friendly even without normalization. Secondly, from Number of Epochs section it could be found that the loss function is a well-defined concave function as the cross-entropy declines in a very stable pattern. So, we simply do not need batch normalization to avoid random resources which will do harm to model.

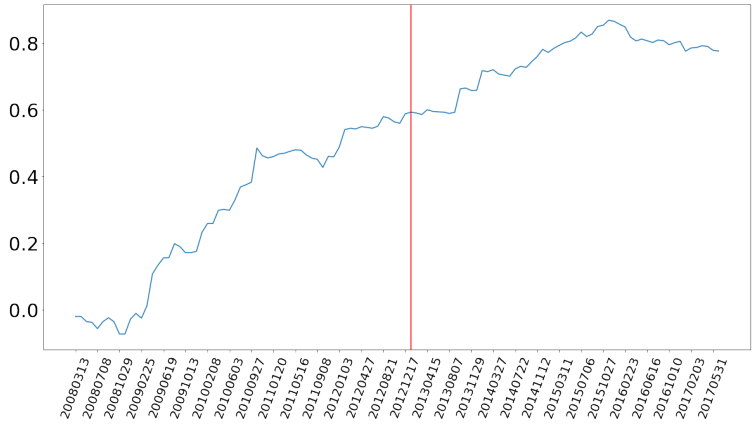
4.4 Dropping Out



Obviously dropping out significantly does harm to our model. This could be preview from the first part where indicated out of sample cross entropy does not differ a lot from in sample entropy, indicating overfitting does not exist. Under this situation, dropping out is just to drop out useful information.

4.5 Final Deep Learning Model

For final deep learning model, we choose SELU as activation function without batch normalization and dropping rate.



Year	Return	Sharpe	Year	Return	Sharpe
2008	-0.028432	-0.528265	*2013*	0.070187	1.219887
2009	0.203136	2.457964	*2014*	0.123115	2.464058
2010	0.280576	2.984843	*2015*	0.084334	2.890076
2011	0.003518	0.096077	*2016*	-0.060262	-2.146598
2012	0.12924	2.421617	*2017*	-0.029014	-1.453523
Average	0.1176076	1.4864472	*Average*	0.037672	0.59478

5. Conclusion

We choose n estimators equals to 50, max depth equals to 10 and min samples split equals to 10000 as parameter for random forest and SELU as activation function without batch normalization and dropping rate for deep neuron network. Among three models including baseline logistic regression model, random forest behaves best will deep neuron network behaves worst. Random forest could get 50.22% out of sample accuracy with 0.6934 out of sample cross entropy and it could create 26% out of sample return over five years.

Reference

1. Hands on Machine Learning with Scikit Learn and Tensorflow, by Aurelien Geron, 2017
2. Documentation of scikit-learn 0.21.1, from website