Improving Financial Time Series Prediction Through Output Classification by a Neural Network Ensemble

Felipe Giacomel[1], Renata Galante[1], and Adriano Pereira[2]

[1] Federal University of Rio Grande do Sul (UFRGS) - Department of Computer Science
{fsgiacomel,galante}@inf.ufrgs.br
[2] Federal University of Minas Gerais (UFMG) - Department of Computer Science
adrianoc@dcc.ufmg.br

Abstract. One topic of great interest in the literature is time series prediction. This kind of prediction, however, does not have to provide the exact future values every time: in some cases, knowing only time series future tendency is enough for an application. In this paper, we propose a neural network ensemble that receives as input the last values from a time series and returns not its future values, but a prediction that indicates whether the next value will raise or fall down. We perform exhaustive experiments to analyze our method by using time series extracted from the North American stock market, and evaluate the hit rate and amount of profit that could be obtained by performing the operations recommended by the system. Evaluation results show capital increases up to 56%.

Keywords: artificial neural networks, classification, prediction, stock markets, time series

1 Introduction

Multilayer perceptron neural networks outperform other techniques when it comes to financial time series prediction [2], hence they are among the most used techniques to predict future values and tendencies of stock markets [1]. Nevertheless, while methodologies based on single artificial neural networks (ANNs) have been largely used for financial time series prediction, neural network ensembles are still little used in this area. [4] showed that simple ensembles can perform better generalizations than a singular ANN, and [12] that a subset of all possible ANNs can achieve better results than a single ANN.

Although ANNs can perform useful predictions of future values in financial time series, studies that focus on this subject generally also focus on minimizing the ANN error and lack a performance test, which would show how much capital would be gained if the operations recommended by the method would be followed. In fact, these studies do not recommend how to operate at all: they show the predicted values for a stock with a certain margin of error, but do not tell us what to do with these results. In practical terms, this kind of prediction is not enough for use in real life. Finally, methods that try to predict the exact value may not behave properly in periods of high volatility [6], so is not safe to trust them all the time. Auxiliary inputs should be added to these ANNs to help them dealing with these periods of uncertainty.

In this paper we propose the resolution of the movement prediction in financial time series by developing a neural network ensemble that classifies a set of inputs into an actual operation order. It receives as input the time series past values and returns a prediction that indicates what is going to happen to the time series in the next period: if it will raise or fall. With this result, one can be told exactly what to do: to buy or to sell a stock. We present as results the hit rate of the method, the variation of capital that one investor could have obtained if he had followed the recommended operations, and a comparison of this variation to what would have been achieved using both the buy-and-hold and naive approaches. To validate our method, we test it in a set of North American stock market time series with a daily granularity. Results in this datasets are promising, with hit rates around 60% and a capital gain up to 56% in 166 days.

The remainder of this paper is structured as follows: Section 2 gives a background of concepts and review related works. Section 3 introduces our proposed method. Section 4 presents the results obtained with the neural network ensemble. Finally, conclusions and future work are presented in Section 5.

2 Related Work

For stock movement prediction, [10] proposes an ANN that predicts the stock movement five days in the future and an algorithm that simulates buy and sell operations based on these predictions. The performance of the method is measured by the mean error of the network and by the capital gain when executing the recommended operations. [8] makes a very interesting study, where the focus is the prediction of the market directional movement for different combinations of inputs, and not the minimization of the error in predicted values. The best results are obtained when external time series (like the exchange rate of foreign currencies) are added to the ANN inputs. Also [9] proposes a ANN where the output tells if a stock is going to raise or fall in the next minutes, enabling the implementation of a high frequency trading system, which executes several trades on the same day. These studies have satisfactory results, but our premise is that they could be improved by a neural network ensemble.

In the ensembles subject, [3] uses a Flexible Neural Tree ensemble to predict the next values of three big stock markets and obtains results with a very low error. [11] combines market news and older prices into a set of Support Vector Machines to predict market movements, and shows that both the prediction performance and profitability of the system increase. Recently, [7] made a comparison among three types of ensembles (mean, median and mode) and a single ANN, with better results obtained by the ensembles. Although these studies improved time series prediction with ensembles, they generally do not try to predict the movements of the time series and, when they do that, the transformation of the predictions in actual trades is not their main focus.

We identified a gap in this area of research, since these two kinds of studies could complement each other. If we improved a classification system by adding it to an ensemble, a higher number of correct classifications would be made and, therefore, a higher capital gain would be achieved. Thus, this study proposes to fill this gap by implementing a neural network ensemble to classify time series output values and use them in a real trade system, showing the results of a set of simulated trades based on the ensemble's recommendations.

3 Proposed Ensemble

We propose a prediction method that ensembles two artificial ANNs to predict the same thing — financial time series movements. The proposed ensemble is viewed by the user as a black box that receives as inputs the last opening, maximum, minimum and closing values for a stock and classifies this set of inputs into one of three possible values, which refer to the next unit of time: *raise*, *fall* or *do not know*. With this information, one trader can decide how she/he will operate in the next time period.

Internally, the ensemble is composed by two ANNs, hence called "neural networks A and B", and a merge module that combines their outputs. Both ANNs are trained to classify their inputs into one of two possible outputs: *raise* or *fall*. Figure 1 shows how the ANNs and the merge module are organized to provide a final classification. The ensemble's inputs x_1, x_2, x_n are passed directly to both A and B ANNs. These ANNs, using an activation function *f*, are trained to return different kinds of classifications in their outputs y_1 and y_2. Finally, the merge module is responsible to combine the y_1 and y_2 classifications and give a trade advice *z*, which has to be one of the ensemble's possible outputs.

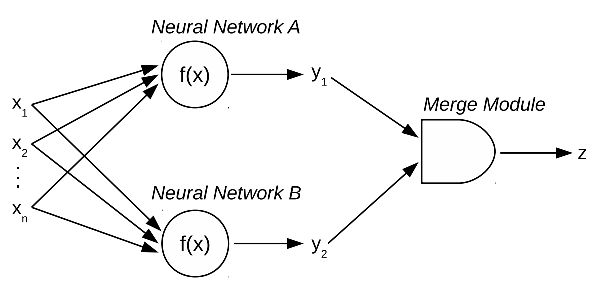

Fig. 1: Architecture of the proposed ensemble

Both A and B ANNs can have similar or different architectures (i.e., number of hidden layers and nodes, training algorithm, etc.), but must receive the same inputs and differ in the kind of output each one returns. These outputs have similar meanings, but are obtained in different ways that vary according to the type of training the ANN received. ANN A is trained to predict the expected opening and closing values for the next time period. With these values, we are able to predict if one stock will raise or fall during a specific time period. If the opening prediction is lower than the closing prediction, we interpret that the ANN is predicting a raise in the prices during the time period. The opposite comparison is made to interpret a prediction of a fall. ANN B, additionally, is trained to predict if a stock is going to raise or fall, no matter how much, returning 1 in the first output node and 0 in the second one when a raise in the prices is predicted for the next time period, or the opposite when a fall is predicted. Due to the nature of ANNs, a decimal value between zero and one is usually the output for both outputs, and never the exact values 1 and 0. What we do is consider the node with the highest value as the winner. The merge module basically works as a logical AND, since both ANNs must agree in the prediction in order to allow the ensemble to give a proper classification. When both networks predict a raise (or a fall) in the prices, the ensemble's recommends to operate according to these predictions. On the other hand, if they disagree, the ensemble gives no advice at all. In this case, we interpret this as an uncertainty period on the market, where any prediction attempt is more likely to be a bet. A stock trader should not perform any operation in this moment.

For the implementation of the networks, we used the Encog Framework [5], where we implemented two feed forward multilayer perceptrons. We chose to use as a training algorithm the resilient propagation, which differs from backpropagation as it does not use fixed values for learning rate and momentum. In fact, in resilient propagation the update on weights and bias are determined by the sign from the derivative of the weight change on each step of the training step, so these values do not have to be initially defined by the user.

In our tests, the generalization of the ANNs did not improve when we added exogenous time series, like foreign currencies and other stock markets indexes to the input dataset. Additionally, adding financial indicators as inputs also did not impact in the ensemble's performance. This happened because the inherent nature of ANNs makes the use of these indicators unnecessary: during its training period, a ANN already learns the relation between the entries, so a redundancy is created when we input the ANN with technical indicators that are, generally, also an interpretation of the past values of one stock.

4 Evaluation

In order to evaluate our method, we first defined a methodology. Then, a group of 9 time series was assembled into one dataset. Finally, we defined the metrics used to analyze the results. The method will be considered satisfactory if it can give profit to a trader who acts according to its outputs. The higher the profit, the better the method is. Also, it is desirable that the hit rate of the method surpasses 50%, otherwise it would not be as good as a coin toss.

4.1 Experimental Setup

To test the capability of our method to generate profit, we adopted the following methodology: if the ensemble predicts a raise in the next time period, we perform a buy operation at the very beginning of this period and a sell operation at the end of this same period; Conversely, if the ensemble predicts a fall in the next time period, we perform a sell operation at the very beginning of this period and a buy operation at the end of this same period. Finally, If the ensemble cannot predict what is going to happen, we do not execute any operation and wait for the next time period.

For the data used in the experiment, we have chosen daily time series from the North American stock market. Table 1 contains the complete list of the used time series and a comparative of the number of periods that a raise or fall occurred in each time series, showing that the data is balanced in all time series. We chose the North American stock market because it is the biggest in the world, and therefore, less willing to be influenced by external factors. We collected daily data for these time series from 19/05/2008 to 01/14/2015, totaling 1,677 days of data for each time series. All sets were divided into two: one for training and one for validation of the ANNs, containing, respectively, 90% and 10% of the available data. We collected opening, maximum, minimum and closing prices of each day of data. All these entries used a time window of five days, totaling 20 inputs for each network. Before entering data to the ANN, we normalized the entry dataset using the classic linear interpolation method.

Table 1: Time series used in the ensemble's validation and data distribution

Symbol	**Name**	**Raises/Falls**	**%**
S&P 500	Standard & Poor's 500	925/752	55/45
AA	Alcoa Inc	858/819	51/49
BAC	Bank of America Corp	852/825	50/50
C	Citigroup Inc	843/834	50/50
F	Ford Motor Co	878/799	52/48
FCX	Freeport-McMoRan Inc	843/834	50/50
GE	General Electric Co	868/809	51/49
JPM	JPMorgan Chase and Co	850/827	50/50
SWN	Southwestern Energy Co	823/854	49/51

4.2 Experiments and Results

For the experiments, we implemented each ANN on the ensemble with three layers: an input with 20 entry nodes, a hidden with six nodes, and an output with two nodes. Although both networks return an output that means different things, both of them need this amount of outputs to be interpreted by the ensemble. We trained all the networks for a maximum of 50,000 iterations unless a minimum error was reached or the network stopped converging. We measured our method with two metrics: the relation of the number of hits and errors that are obtained with its recommendations, and the capital gain that would be obtained if one trader would strictly follow all of its recommendations. This last one is also compared to the capital gain of two common strategies: the *Buy and Hold*, which buys the stock at the beginning of the time period and sells it in the end of the same period, and the *Trivial Strategy*, which assumes that if a stock raised (or fell) in time period t, it will also raise (or fall) in $t+1$

Classification Analysis Here, we measure the amount of days the ensemble gives a prediction (i.e., an output different from *do not know*) and the distribution of raise and fall predictions. After running the ensemble in each time series, we got the results shown in Table 2. Although all time series had the same amount of inputs, the number of recommendations given by each one were not any close to each other. AA and GE time series, for instance, had above 75% of days with recommendations. BAC and F, on the other hand, had recommendations only in less than 20% of the days, which means that most of the times the ensemble's internal ANNs did not agree with the classification results. For time series with low prediction rates, it is preferable that the ensemble does not give a prediction at all than a wrong one. A surprising behavior that is shown in this table is that, even with the parity between raises and falls shown in Table 1, this parity does not reflects in the balance of predictions made by the ensemble.

Table 2: Summary of the classifications made by the proposed ensemble

Symbol	# Days Test Period	Total Advices	% Days W/ Advices	# Raise Prediction	# Fall Prediction
S&P 500	166	39	23%	34	5
AA	166	128	77%	13	115
BAC	166	26	15%	25	1
C	166	101	60%	101	0
F	166	32	19%	31	1
FCX	166	89	53%	33	56
GE	166	127	76%	10	117
JPM	166	106	63%	93	13
SWN	166	78	46%	24	54

Performance Analysis Here we measure the performance of the ensemble under real conditions. If it can achieve profit and have a good hit rate when confronted with real financial time series, the method can be used by traders in their operations. We simulated an initial capital of $100,000.00 and compared it to the final capital we would have by always investing 100% of our money according to the ensemble's recommendations. Moreover, we compare our results with the financial return obtained by buy-and-hold and trivial strategies. Among all results, that can be seen in table 3, the best ones were obtained by S&P 500 time series, which had the highest hit rate and outperformed both buy-and-hold and trivial strategies. It had the lowest capital gain among all time series, but we credit this to the fact that, as we have seen in Table 2, it made a very small number of predictions. Eight out of the nine used time series had a good hit rate (above 56%), and all of them profited. If we had used the buy-and-hold strategy in all time series, we would have profited in five of them, giving this strategy a success rate of 55%. The trivial strategy had a better performance: it profited in six out of nine time series, with a final success rate of 66%. Our ensemble, finally, profited in all time series, with a success rate of 100%. Also, it outperformed both comparative approaches in four time series, one of the comparative approaches in four time series, and none of the comparative approaches in only one time series (even thought it generated profit). Our ensemble has showed itself not as a tool that always gets the best profits, but instead gives good and consistent results. Investors in general prefer techniques that give them small and constant profits than a roller coaster of yields that can generate higher profits but can also lose capital in a high frequency.

Table 3: Results for time series prediction using our ensemble and other methods

Symbol	**Initial $**	**Final $ Buy and Hold**	**Final $ Trivial**	**Final $ Ensemble**	**Ensemble Hit Rate**
S&P 500	100,000.00	100,953.44	103,204.82	106,767.06	64.71%
AA	100,000.00	118,454.61	92,596.88	131,195.15	50.78%
BAC	100,000.00	120,390.31	119,740.16	115,855.32	69.23%
C	100,000.00	116,265.58	105,381.62	109,491.78	56.44%
F	100,000.00	99,168.27	116,384.78	111,210.01	56.25%
FCX	100,000.00	68,604.99	98,364.03	106,495.49	56.18%
GE	100,000.00	96,193.38	103,902.88	112,967.79	58.27%
JPM	100,000.00	116,862.75	89,417.12	113,833.18	61.32%
SWN	100,000.00	57,965.17	159,294.49	156,903.98	60.26%

5 Conclusion and Future Work

In this paper we proposed an ensemble of neural networks to predict the stock market movements. To accomplish that, we turned the prediction problem into a classification problem. We tested our method in the North American stock market with a daily granularity, measuring the market movements hit rate and the amount of capital that our method could profit when compared to both buy-and-hold and trivial approaches, with final results showing very satisfactory indexes.

Our method could be used in the creation of new strategies for algorithmic trading in stock markets, or to perform stock portfolio management, changing stocks according to model trends prediction. Finally, we foresee several possibilities for future work. We plan to test this method by changing one or both of the neural networks to another type of machine learning techniques (like SVM or RBF neural networks). Also, we want to add a technical indicator to the ensemble, which could vote for the classification results. However, since in our tests increasing the number of items in the ensemble made the number of advices decrease, we would have to focus on the increment of advices given by each item in the ensemble before adding new elements to it.

References

1. Atsalakis, G.S., Valavanis, K.P.: Surveying stock market forecasting techniques–part ii: Soft computing methods. Expert Systems with Applications 36(3), 5932–5941 (2009)
2. Cao, Q., Leggio, K.B., Schniederjans, M.J.: A comparison between fama and french's model and artificial neural networks in predicting the chinese stock market. Computers & Operations Research 32(10), 2499–2512 (2005)
3. Chen, Y., Yang, B., Abraham, A.: Flexible neural trees ensemble for stock index modeling. Neurocomputing 70(4), 697–703 (2007)
4. Hansen, L.K., Liisberg, C., Salamon, P.: Ensemble methods for handwritten digit recognition. In: Neural Networks for Signal Processing [1992] II., Proceedings of the 1992 IEEE-SP Workshop. pp. 333–342. IEEE (1992)
5. Heaton, J., Reasearch, H.: Encog java and dotnet neural network framework. Heaton Research, Inc., Retrieved on July 20, 2010 (2010)
6. Kara, Y., Acar Boyacioglu, M., Baykan, Ö.K.: Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. Expert systems with Applications 38(5), 5311–5319 (2011)
7. Kourentzes, N., Barrow, D.K., Crone, S.F.: Neural network ensemble operators for time series forecasting. Expert Systems with Applications 41(9), 4235–4244 (2014)
8. OConnor, N., Madden, M.G.: A neural network approach to predicting stock exchange movements using external factors. Knowledge-Based Systems 19(5), 371–378 (2006)
9. Silva, E., Castilho, D., Pereira, A., Brandao, H.: A neural network based approach to support the market making strategies in high-frequency trading. In: Neural Networks (IJCNN), 2014 International Joint Conference on. pp. 845–852. IEEE (2014)
10. Taylor, B., Kim, M., Choi, A.: Automated stock trading algorithm using neural networks. In: Proceedings of the 2nd International Conference on Intelligent Technologies and Engineering Systems (ICITES2013). pp. 849–857. Springer (2014)
11. Zhai, Y., Hsu, A., Halgamuge, S.K.: Combining news and technical indicators in daily stock price trends prediction. In: Advances in Neural Networks–ISNN 2007, pp. 1087–1096. Springer (2007)
12. Zhou, Z.H., Wu, J., Tang, W.: Ensembling neural networks: many could be better than all. Artificial intelligence 137(1), 239–263 (2002)