A Study on Training Ensembles of Neural Networks - A Case of Stock Price Prediction -

Young-Chan Lee* · Soo-Hwan Kwak**

신경망 학습앙상블에 관한 연구 - 주가예측을 중심으로 -

이영찬* · 곽수환**

Abstract

In this paper, a comparison between different methods to combine predictions from neural networks will be given. These methods are bagging, bumping, and balancing. Those are based on the analysis of the ensemble generalization error into an ambiguity term and a term incorporating generalization performances of individual networks.

Neural Networks and AI machine learning models are prone to overfitting. A strategy to prevent a neural network from overfitting, is to stop training in early stage of the learning process. The complete data set is spilt up into a training set and a validation set. Training is stopped when the error on the validation set starts increasing. The stability of the networks is highly dependent on the division in training and validation set, and also on the random initial weights and the chosen minimization procedure. This causes early stopped networks to be rather unstable: a small change in the data or different initial conditions can produce large changes in the prediction. Therefore, it is advisable to apply the same procedure several times starting from different initial weights. This technique is often referred to as training ensembles of neural networks.

In this paper, we presented a comparison of three statistical methods to prevent overfitting of neural network.

Key words: Neural Network, Generalization Performance, Overfitting, Bagging, Bumping, Balancing

* Institute for Business Research, Sogang University
** Graduate School of Business, Korea University

1. Introduction

Neural networks are mostly good at recognizing complex patterns. A typical network receives for the relations between inputs and the expected outputs. It then searches for the relations between input and output. Once the computational rules have been found, the network is able to produce outputs on any input, but an error of a few percent is normal.

The neural network is usually to estimate the function (which is implicit in the training patterns) as closely as possible. In many cases, however, a number of problems can arise which might prevent the network from learning this function.

- A satisfactory approximation can not be reached by a small network due to the lack of parameters to express the function.
- The network can suffer from overfitting to certain training examples. It will perform poorly on other input patterns. This problem can for example be solved by restricting the number of training cycles and the number of units in the hidden layer(s).
- Optimization techniques, where a minimization of an error- or cost-function is wanted, can become trapped in local minima, instead of propagating to the desired global minimum. Some of the most commonly used learning rules are believed to stabilize the network in local minima.

Many different types of neural networks have been investigated in the last few decades. These networks differ in areas such as used types of units, network topology, applied learning rule, and their behavior. For an overview consult [2]. We want to use feed-forward networks on the stock price prediction problem. The input patterns will contain a number of different stock values to produce as output a future value of stock price.

2. Theoretical Background

In this paper, a comparison between some methods to combine predictions from neural networks will be given. One of these methods is balancing. This method is based on the analysis of the ensemble generalization error into an ambiguity term and a term incorporating generalization performances of individual networks. The method is described in [2].

Neural Networks and AI machine learning models are prone to overfitting[5]. A strategy to prevent a neural network from overfitting, is to stop training in an early stage of the learning process. The complete data set is split up into a training set and a validation set. Training is stopped when the error on the validation set starts increasing. The stability of the networks is highly dependent on the division in training and validation set, and also on the random initial weights and the chosen minimization procedure. This causes early stopped networks to be rather unstable: a small change in the data or different initial conditions can produce large changes

in the prediction. Therefore, it is advisable to apply the same procedure several times starting from different initial weights. This technique is often referred to as training ensembles of neural networks[1][4][6][7][8][9].

1) Bagging

With bagging, the prediction on a newly arriving input vector is the average over all network predictions. Bagging completely disregards the performance of the individual networks on the data used for training and stopping[3].

2) Bumping

Bumping throws away all networks, except the one with the lowest error on the complete data set[3].

3) Balancing

Balancing is an intermediate form of bagging and bumping[3]. Each network receives a weighting factor α_i which depends on the expected performance of the network on new values. This estimation is based on the performance of the network on the training and validation sets. The prediction of all networks on pattern ν is defined as the following weighted average.

$$\tilde{m}^{\nu} \equiv \sum_{i=1}^{n_{run}} \alpha_i \quad \hat{o}_i^{\nu}$$

where $\hat{o}_i$ is the predicted output in i^{th} network, $\tilde{m}$ is the weighted average of the predicted output for all networks, and n_{run} is the number of networks.

The goal is to find the weighting factors α_i, subject to the following constraints

$$\sum_{i=1}^{n_{run}} \alpha_i = 1 \quad \text{and} \quad \alpha_i \geq 0 \quad \forall_i,$$

yielding the smallest possible generalization error

$$E_{test} \equiv \frac{1}{p_{test}} \sum_{\nu=1}^{p_{test}} (\tilde{m}^{\nu} - \hat{t}^{\nu})^2$$

where $\hat{t}$ is the target value, P_{test} is the number of patterns.

We have to find reasonable estimates for these generalization errors based on the network performances on validation data. Once we have obtained these estimates, finding the optimal weighting factors α_i under the constraints is a straightforward quadratic programming problem.

3. Problem Description

In this section a comparison of the methods for combining the neural network outputs will be given. The differences and the results are discussed in the following subsections.

1) Purpose of the analyses

On the data 263 separate neural network analyses were performed. Each of the analyses had to process the output for one weekday in 1996, based on the values of the preceding 10 days. To make a prediction, each model is fitted on 40 directly

preceding weekdays.

For example: to process a prediction for 29/01/96 based on the 10 preceding weekdays, a model is fitted on the values of the 40 preceding weekdays(04/12/95-26/01/96).

Consequently, a prediction of the value on 29/01/96 is processed.

2) The network models

For each analysis 50 neural networks are trained on the appropriate 40 observations. Each network is a 3-layer network, with 10 input units, 3 units in the hidden layer, and 1 output unit. The hidden units have a hyperbolic tangent(tanh) activation function, and the output node has a identity or linear activation function.

Training of the network was performed using the back-propagation algorithm. The back-propagation algorithm was stopped after 500 epochs of training, or if the mean squared error(MSE) on the validation set increased for 4 subsequent steps. The learning rate and momentum parameters were 0.01 and 0.6 respectively.

3) Normalization

To decrease the learning time of the neural networks the data for the analyses have been normalized. The complete data set has been transformed to its Z- value:

$$x = \frac{x - mean}{sd}$$

The resulting data set has an average of 0 and a standard deviation of 1.

4) Percentage of correct direction

For each of the 263 weekdays of 1996, three collective models were made from the 50 separate networks for bagging, bumping, and balancing methods. Note that for each of the 263 days three models were constructed. For each of these 263 predictions the MSE was determined, and the percentage of predictions with the correct direction.

A prediction of the value for weekday t has the correct direction if an increase or decrease was predicted with respect to the predicted value for weekday t-1, which is the same direction as the actual difference.

5) Differences

In this section the differences are discussed and the possible impacts:

- **activation functions** : The feed forward networks(FFNs) used in this paper have tanh activation functions in the hidden layer and identity or linear activation functions in the output unit.
- **learning rate and momentum** : Different values for learning rate and momentum may cause the FFNs to stop training in a different point of 'weight space'. The impact of this difference is difficult to be estimated.
- **stop criterion** : The learning algorithm of the networks in this chapter uses a validation stop criterion. The back-propagation algorithm is stopped when the MSE on the validation set increases in 3 subsequent steps. The

validation data are selected from the training data using the 'bootstrapping' algorithm[3]. Furthermore, the back-propagation algorithm is proceeded for a maximum of 500 epochs. As a result of different stop criteria the networks may terminate in different configurations. The impact on the predictions is expected to be rather small.

- **normalization** : The output values for the FFNs in this chapter are normalized in the same way as the input values (Z-transformation)., while in some cases, the output values are scaled to the domain [0:1]. This is necessary when a sigmoid transfer function in the output node is used. This difference has an effect on the computed MSE for both methods, but it has no impact on the percentages of predictions in the correct direction. It is possible to scale the MSEs back to a scale in which a comparison can be made.

6) Results

The 50 FFNs models that have been trained have to be combined to form a collective model. Using this model predictions can be made based on 10 preceding values. There are three methods to make this combination: bagging, bumping, and balancing:

- **individual** : The average individual generalization error, i.e. the generalization error we will get on average when we decide to perform only one run. It serves as a reference with which the other methods will be compared.
- **bagging** : The average of the predictions of 50 feed-forward networks is taken to produce an ensemble output.
- **bumping** : The best of 50 feed-forward networks is chosen, based on the best predictions made on the data in the modeling phase. This network is taken as the eventual model.
- **balancing** : By way of bootstrapping procedures an estimate of the performance of each of the networks on new data is made. Based on these estimates a weight is assigned to each network, using quadratic programming, which optimizes the weighted average of the estimated predictions. The ensemble output is the weighted average of the 50 networks.

The mean MSE of all 263 times 50 FFNs is 8.69. This value has been scaled back to the scale of the original data.

The results of the bagging, bumping, and balancing methods for combining the predictions of the 50 networks are expressed in the relative decrease of the MSE with respect to the MSE of the 50 separate predictions. Because this value has been computed for each of the 263 selected weekdays, the mean value and the standard deviation are reported.

	Average decrease of the MSE(sd) (n=263)
Bagging	28% (30%)
Bumping	39% (65%)
Balancing	37% (44%)

In the next tables the number and percentages of correct or wrong prediction of increase or decrease of the stock price are given. The rows indicate the actual directions, the columns indicate the according predicted directions. Subsequent observations which remained the same (no increase, no decrease) are given in a separate row.

bagging	decreased predicted	increased predicted
decrease	22 (8%)	79 (30%)
no change	5 (2%)	17 (6%)
increase	45 (17%)	94 (36%)

bumping	decreased predicted	increased predicted
decrease	36 (14%)	65 (25%)
no change	9 (3%)	13 (5%)
increase	47 (18%)	92 (35%)

balancing	decreased predicted	increased predicted
decrease	29 (11%)	72 (27%)
no change	10 (4%)	12 (5%)
increase	45 (17%)	94 (36%)

Next table shows the total number of correct and wrong predicted directions and the number of undefined changes in direction. If a direction is defined wrong when

$$(target(t) - target(t-1)) * (output(t) - output(t-1)) < 0$$

All predictions where target(t) - target(t-1) = 0 are considered 'correct'. The results of this approach are given in the last column.

	correct direction	wrong direction	undefined
bagging	116 (44%)	124 (47%)	22 (8%)
bumping	128 (49%)	112 (43%)	22 (8%)
balancing	123 (47%)	117 (45%)	22 (8%)

4. Conclusion

In the results that we obtain by combining the FFNs, the ensemble of networks perform much better than the separate FFNs if we consider the MSE. In subsection results we have seen that we can decrease the MSE considerably by combining the FFNs.

References

[1] Bartlett, P. L., "For valid generalization, the size of the weights is more important than the size of the network," in Mozer, M.C., Jordan, M.I., and Petsche, T., (eds.) *Advances in Neural Information Processing Systems 9*, Cambrideg, MA: The MIT Press, 1997.

[2] Hertz, J., A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.

[3] Heskes, T., "Balancing between bagging and bumping," In Mozer, M. C., M. I. Jordan, and T. Petsche, (eds.) *Advances in Neural Information Processing Systems 9*, Cambridge, MA: The MIT Press, 1996.

[4] Geman, S., Bienenstock, E. and Doursat, R., "Neural Networks and the Bias/Variance

Dilemma", *Neural Computation*, 4, 1992, pp.1-58.

[5] Lawrence, S., C. L. Giles, and Ah Chung Tsoi, "Lessons in Neural Network Training: Overfitting May be harder than Expected," *Proceedings of the Fourteenth National Conference on Artificial Intelligence,* AAAI-97, AAAI Press, Mento Park, California, 1997, pp.540-545.

[6] Moody, J. E., "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems", *NIPS 4*, 1992, pp.847-854.

[7] Sarle, W. S., "Stopped Training and Other Remedies for Overfitting," *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*, 1995, pp.352-360.

[8] Smith, M., *Neural Networks for Statistical Modeling*, NY: Van Nostrand Reinhold, 1993.

[9] Weigend, A., "On overfitting and the effective number of hidden units," *Proceedings of the 1993 Connectionist Models Summer School*, 1994, pp.335-342.