Neural, Parallel, and Scientific Computations 20 (2012) 17-22

APPLICATIONS OF BOOTSTRAP METHOD IN NEURAL NETWORKS

REZA HABIBI AND HAMED HABIBI

Department of Statistics, Central Bank of Iran
Department of Mechanics, Tehran University, Tehran, Iran

ABSTRACT. This paper is concerned with bootstrap bias correction and forming a bootstrap confidence interval for outputs of a given trained neural network. We apply our method to the problem of symmetry detection for eight bits code as well as in time series prediction. A real data set is also considered. Conclusions are given.

Keywords: Bias correction; Bootstrap; Confidence interval; Neural network; Symmetry detection; Time series prediction; Variance estimation

1. Introduction

Neural networks (hereafter NNs) have been popular in analyzing nonlinear problems occurring in many fields such as biology, finance, engineering, mathematics and physics. McCulloch and Pitts (1943) first studied NNs in the context of mathematical biophysics. In fact, they are extensions of mathematical models designed for biological nervous systems. Some excellent references in this area are Hertz *et al.* (1991) and Masters (1993[a], 1993[b]). NNs have also many applications in statistics. They are used frequently in regression analysis, time series prediction and for change point detection (see Ripley (1997)).

NNs decompose the original data set to two partitions, namely training and test sets. A special NN is trained using the first data set. In the second stage, the performance of above mentioned network is evaluated by applying it to test set and an error (variance) is computed. Dupret and Koda (2000) showed that the error calculated in this way don't show the actual error and it may be misleading. They proposed the bootstrap solution. In their approach, the NN is trained B (enough large) times using the bootstrap re-sample sets of train set. Consequently, we derive B errors. The mean of errors presents an accurate estimate of actual error.

The bootstrap method proposed by Efron (1979) is a computer intensive statistical techniques. It belongs to the class of re-sampling based approaches. Indeed, it is a sampling with replacement from a given sample. Valid bootstraps are often accessible

Received August 9, 2011

practical solutions to statistical inference problems. In computational statistics setting, the bootstrap is used often for estimation of standard deviation, bias correction and constructing empirical confidence intervals. As follows, we review Dupret and Koda's work briefly.

Let $x_i = (\mathbf{I}_i, T_i)$, $i = 1, \ldots, n$ be $i-th$ element of the training set $\mathbf{x} = \{x_1, \ldots, x_n\}$at which $\mathbf{I}_i$ is vector input and the target value is T_i. The NN is a function like h which produces the actual output $O = h_x(\mathbf{I})$. For a suitable distance d, the prediction error in the NN is

$$E_F(d(T, h_x(\mathbf{I}))),$$

where F denotes the distribution of $(T, h_x(\mathbf{I}))$ and a Monte Carlo estimator of error is

$$\widehat{e}_x = (1/n) \sum_{i=1}^{n} d(T_i, h_x(\mathbf{I}_i)).$$

Dupret and Koda (2000) suggested to train the specified NN (B times) with bootstrap sample $\mathbf{x}^*$ and to derive B errors $\widehat{e}_{x^{*b}}$, $b = 1, \ldots, B$ and finally they advised to calculate

$$(1/B) \sum_{b=1}^{B} \widehat{e}_{x^{*b}}.$$

Their approach motivates us to survey how the bootstrap method can be applied to improve the performance of a given NN. This paper is organized as follows. In section 2, we consider the bias correction and constructing confidence intervals for NN using bootstrap technique. It is shown that in this way, the accuracy is improved. We apply our method to problems of symmetry detection for eight bits code and time series prediction. A real data set is studied in section 3. Conclusions are given in section 4.

2. **Bootstrap NN**

In this section, we study bootstrap bias correction and bootstrap confidence intervals for NN problems. Estimating the bootstrap variance is studied by Dupret and Koda (2000) which we reviewed it in introduction. Both methods are described and three examples are given. First, we consider the bias correction.

2.1 Bias correction. Suppose that h_x is a trained NN. The main question is if h_x is unbiased or it is biased and therefore the outputs O's are biased. In practice, h_x may be biased and it is reasonable to correct the nasty bias. Therefore, we search for a network like h_x^* such that its bias is zero. The first selection is

$$h_x^* = h_x - bias(h_x).$$

Next, suppose that we train our network by bootstrap train sets x^{*b}, $b = 1, \ldots, B$. The bootstrap estimation for bias(h_x) is $\overline{h}_{x^*} - h_x$, where $\overline{h}_{x^*} = (1/B) \sum_{b=1}^{B} h_{x^{*b}}$.

Therefore, the bootstrap estimate of h_x^* is

$$\widehat{h}_x^* = 2h_x - \overline{h}_{x^*},$$

see Gentle (2002) for more description.

2.2 Confidence interval. Here, we study the interval estimation for prediction of a NN using the bootstrap method. There many kinds of bootstrap confidence intervals. Here, we only consider two types of them, first the bootstrap-t confidence interval and second bootstrap percentile interval.

(a) Bootstrap-t intervals. The confidence interval for the mean of a normal population, when the population variance is unknown, works very well, even in cases that the underlying distribution is not normal. Following this idea, a $100(1-\alpha)$ percent single confidence interval for *i-th* output $O_i = h_x(\mathbf{I}_i)$ is given by

$$(O_i - \widehat{t}_{1-(\alpha/2)}\sqrt{\widehat{\text{var}}(O_i)}\ ,\ O_i - \widehat{t}_{\alpha/2}\sqrt{\widehat{\text{var}}(O_i)}),$$

where $\widehat{t_\alpha}$ is estimated by α-th quantile of

$$\frac{O_i^* - O_i}{\sqrt{\widehat{\text{var}}(O_i^*)}},$$

at which $\widehat{\text{var}}(O_i^*) = \frac{1}{B-1}\sum_{b=1}^{B}(O_i^{*b} - \overline{O}_i^*)^2$ with $\overline{O}_i^* = \frac{1}{B}\sum_{b=1}^{B} O_i^{*b}$. To construct the simultaneous confidence intervals, for example, for (O_i, O_j), $i \neq j$ we should consider the bootstrap distribution of $(\frac{O_i^*-O_i}{\sqrt{\widehat{\text{var}}(O_i^*)}}, \frac{O_j^*-O_j}{\sqrt{\widehat{\text{var}}(O_j^*)}})$. In this way, we derive the joint (multivariate) quantiles $\widehat{t}_\alpha^i$ and $\widehat{t}_\alpha^j$.

(b) Bootstrap percentile intervals. The $100(1-\alpha)$ percent single bootstrap percentile confidence interval is given by

$$(t^*_{(\alpha/2)}, t^*_{1-(\alpha/2)}),$$

where $t^*_{(\alpha/2)}$ is the $[B\alpha]$-th order statistic of O_i^{*b}, $b = 1, 2, \ldots, B$. The simultaneous confidence intervals are constructed similarly.

Example 1. Here, we study the simulated data, taken form Shariat-panahi (2011). There are 130 vector of observations with length 6. This data set is broken to 100 train data and 30 observations is kept for test set. We trained feed-forward NN with three hidden layers of 10 neurons as varying parameters, using back-propagation. The transfer function is Tansic for three layers and it is Purelin for the last layer. The number of bootstrap replication is $B = 1000$. The prediction error is 17.515 while its bootstrap estimate is 10.244. The prediction for 10-th observation is $O_{10} = 25.65$ and its bias corrected estimate is 23.015 target value $T_{10} = 22.054$. It is seen that the bias corrected estimate is too accurate. Then, we construct a bootstrap 0.95% for T_{10}. One can see that the bootstrap t-confidence interval is $(20.45, 25.32)$. This interval has length 4.87 which is small relative to target value $T_{10} = 22.054$. The bootstrap percentile interval is $(19.35, 26.12)$.

Example 2: symmetry detection. Here, following Dupret and Koda (2000), we apply the above mentioned two bootstrap facilities to the problem of symmetry detection for six bits code. To describe more, the input vector is $\mathbf{I} = (\phi_1, \ldots, \phi_8)^T$ and each ϕ_i takes values in $\{0, 1\}$. Therefore, there are $2^8 = 256$ inputs. An input is symmetric if $\phi_i = \phi_{9-i}$, $i = 1, 2, 3, 4$. There are 16 symmetric inputs and 240 asymmetric inputs. There is one output which is 1 if $\mathbf{I}$ is symmetric and 0 otherwise. Similar to Example 1, we again use the feet-forward with 3 hidden layers with 10 neurons. The transfer functions are Tansic and Purelin. The number of bootstrap replications is 1000. We bootstrap symmetric and asymmetric codes, separately. Following Dupret and Koda (2000), we considered a set the symmetric proportions (which its true value is $\frac{240}{256} = 0.9375$) moving form $\frac{16}{256}$ to $\frac{250}{256}$. We found that the best result relates to case of the true value 0.9375. Therefore, here, we only focus on this case. Consider 20-*th* observation. Its 8-bits code is asymmetric and therefore the $T_{20} = 0$. The proportion of zeros for x_{20} in 1000 bootstrap replications is 0.94 which shows that the network works well. Note that this proportion in original data set is $\frac{240}{256} = 0.9375$. The network also works well for other x's but we don't present them, here. Using the bootstrap method, we derive 1000 re-samples for each x_i (with actual output zero), $i = 1, 2, \ldots, 240$. We found that the mean of proportion of zeros is 0.9268 with standard deviation 0.00325. A 95% confidence interval for the actual proportion of zeros is $(0.8938, 0.9598)$.

Example 3: Time series prediction. In time series setting, we are interested to predict future value of x_{n+1} based on $x_1, \ldots, x_n$. A suitable integer p is chosen and the prediction of x_{n+1} is

$$\widehat{x}_{n+1} = g(x_n, \ldots, x_{n-p}).$$

Touretzky and Laskowski (2006) described how the NN's can be applied to predict a time series. As follows, we give a schematic structure of inputs and outputs. The lengths of inputs and output sequences are i and o, respectively. The train set involves p sequences of length i. Here, we study the prediction of monthly returns for AT&T taken form Hipel and Mcleod (1994). There are 84 returns for time period January 1961-December 1967. Again, we select feet-forward network with three hidden layers with 10 neurons in each layers. We let $p = 5$ and $i = o = 40$. The transfer functions are Tansic and Purelin. The MATLAB software proposes a 5×40 matrix of errors. The maximum of errors is 1.2×10^{-2} which shows that the network works very well. In this problem, we focus on 5-*th* to 40-*th* (by step 5) elements of 3-*th* output. The number of bootstrap again is 1000. The 0.90% bootstrap percentile confidence intervals for these elements are (−0.0586, 0.0688), (−0.0574, 0.0765), (−0.102, 0.053), (−0.101, 0.0326), (−0.0571, 0.0372), (−0.111, 0.185), (−0.168, 0.125), and (−0.0494, 0.0391), respectively. One can see that how much the length of intervals are small and they are close to actual values.

Training set

x_1	x_2	$\cdots$	x_i
x_2	x_3	$\cdots$	x_{i+1}
x_3	x_4	$\cdots$	x_{i+2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_p	x_{p+1}	$\cdots$	x_{p+i-1}

Targets

x_{i+1}	x_{i+2}	$\cdots$	x_{i+o}
x_{i+2}	x_{i+3}	$\cdots$	x_{i+o+1}
x_{i+3}	x_{i+4}	$\cdots$	x_{i+o+2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{p+i}	x_{p+i+1}	$\cdots$	$x_{p+i+o-1}$

3. Real data set: Change point detection

When analyzing (or prediction) a time series using NN technique, a key assumption is the stability of actual NN over time. A common method to assess this stability is to derive a NN over the rolling window. If the actual NN doesn't differ over time then errors of predictions using rolling NN's are very close to zero. To compute prediction errors, we use the method of example 3. The maximum absolute of these errors is a good criterion. To compute the confidence interval for the mentioned maximum, we need its probability distribution. This distribution is completely unknown, and we use the bootstrap technique.

Hansen (1992) fitted a discrete time AR(1) model for annual U.S. output growth rates. He showed that this model has remained stable over time period 1889- 1987. Similar to example 2, we fit a feet-forward NN with 3 hidden layers with 10 neurons. The transfer functions are Tansic. Here, we want to check if this network remains unchanged over time. We let the length of rolling window be 24. One can see that the bootstrap confidence interval is $(0.04, 0.08)$, where its length is too small and this shows that our NN is enough for whole observed time series.

4. Conclusions

In this paper, we applied the bootstrap method for bias correction in NN setting. We guessed that the output of a trained NN may be biased, since the unbiased condition is not considered for training of a NN. Methods are proposed and by some examples, it is seen that, using the bootstrap method, the accuracy of NN output is improved. It is also generally believed that the interval estimate (output) is very better than point estimation. Therefore, we use the bootstrap method for forming the confidence interval. We also applied these methods for change point detection in a real time series. It is seen, using bootstrap method and NN technique simultaneously, we can monitor time series.

ACKNOWLEDGMENTS

The author is grateful to the referee for several suggestions for improving the paper.

REFERENCES

[1] Dupret, G. and Koda, M. (2000). Bootstrap Training for Neural Network Learning. *Proceedings 4th Workshop on Stochastic Numerics*, 27–35.

[2] Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics* **7**, 1–26.

[3] Gentle, J. E. (2002). *Elements of computational statistics.* Springer.

[4] Hansen, B. E. (1992). Testing for parameter instability in linear models. *Journal of Policy Modeling* **14**, 517–533.

[5] Hertz, J., Krogh, A. and Palmer, R. G. (1991). *Intorduction to the theory of neural computation.* Addison-Wesley Co.

[6] Hipel, K.W. and McLeod, A.I. (1994). *Time Series Modelling of Water Resources and Environmental Systems.* Elsevier.

[7] Masters, T. (1993[a]). *Advanced algorithms for neural networks.* Wiley.

[8] Masters, T. (1993[b]). *Practical neural network recipes in C++.* Academic press.

[9] McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **7**, 115–133.

[10] Ripley, B. D. (1997). Can statistical theory help us use neural networks better? *Proceedings of Interface* **97**. *The 29th Symposium on the Interface: Computing Science and Statistics*, 1–6.

[11] Shariat-panahi, M. (2011). Artificial inteligence and expert system.

[12] Touretzky, D. and Laskowski, K. (2006). Neural networks for time series prediction. *Lecture notes.*