Some Issues About the Generalization of Neural Networks for Time Series Prediction

Wen Wang[1,2], Pieter H.A.J.M. Van Gelder[2], and J.K. Vrijling[2]

[1] Faculty of Water Resources and Environment, Hohai University, Nanjing, 210098, China
[2] Faculty of Civil Engineering & Geosciences, Section of Hydraulic Engineering, Delft University of Technology. P.O.Box 5048, 2600 GA Delft, Netherlands

Abstract. Some issues about the generalization of ANN training are investigated through experiments with several synthetic time series and real world time series. One commonly accepted view is that when the ratio of the training sample size to the number of weights is larger than 30, the overfitting will not occur. However, it is found that even with the ratio higher than 30, overfitting still exists. In cross-validated early stopping, the ratio of cross-validation data size to training data size has no significant impact on the testing error. For stationary time series, 10% may be a practical choice. Both Bayesian regularization method and the cross-validated early stopping method are helpful when the ratio of training sample size to the number of weights is less than 20. However, the performance of early stopping is highly variable. Bayesian method outperforms the early stopping method in most cases, and in some cases even outperforms no-stop training when the training data set is large.

1 Introduction

ANNs are prone to either underfitting or overfitting (Sarle, 2002). A network that is not sufficiently complex can fail to detect fully the signal in a complicated data set, leading to underfitting. A network that is too complex may fit the noise, not just the signal, leading to overfitting, which may result in predictions far beyond the range of the training data. Therefore, one critical issue in constructing a neural network is generalization, namely, the capacity of an ANN to make predictions for cases that are unseen in the training set. Two commonly used techniques for generalization are cross-validated early stopping (e.g., Amari et al., 1997; Prechelt, 1998) and the regularization (or weight decay) technique (e.g., Mackay, 1991; Neal, 1996).

In cross-validated early stopping, the available data are usually split into two subsets: training and cross validation (referred to as CV hereafter) sets. The training set is used for updating the network weights and biases. The CV set is used to monitor the error variation during the training process. When the validation error increases for a specified number of iterations, the training is stopped.

Large weights can cause excessive variance of the output (Geman et al., 1992). A traditional way of dealing with the negative effect of large weights is regularization. The idea of regularization is to make the network response smoother through modification in the objective function by adding a penalty term that consists of the mean square of all network coefficients. Mackay (1991) proposed a technique, called

W. Duch et al. (Eds.): ICANN 2005, LNCS 3697, pp. 559 – 564, 2005.

Bayesian regularization, which automatically sets the optimal performance function to achieve the best generalization based on Bayesian inference techniques.

In this paper, we will discuss three issues about the generalization of networks: (1) How many data are demanded to avoid overfitting; (2) How to split the training samples in cross-validated early stopping; (3) Which generalization technique is better for time series prediction, Bayessian regularization or cross-validated early stopping?

2 Experiments and Result Analyses

2.1 Data

Seven data sets are used in this study, including three synthetic data sets and seven observed data sets. Three synthetic time series are as following: (1) Henon map (Henon, 1976) chaotic series; (2) The discretized chaotic Mackey-Glass flow series (Mackey and Glass, 1977); (3) A stochastic time series generated with an ANN model with a structure 5-3-1. 2% Gaussian noises are added to the two synthetic chaotic time series. The four observed real-world time series include: (1) The monthly sunspot number series (1749.1 ~ 2004.12); (2) The yearly sunspot number series (1700 to 2004); (3) Monthly Southern Oscillation index (SOI) series (1933.1 ~ 2004.12); (4) and (5) daily and monthly streamflow series of the Rhine River at Lobith, the Netherlands (1901.1 ~ 1996.12); (6) and (7) daily and monthly streamflow series of the Danube River at Achleiten, Austria (1901.1 ~ 1990.12).

De Oliveira et al. (2000) suggest to use m:2m:m:1 structure to model chaotic series. Follow their suggestion, we use 6:12:6:1 for Henon series as well as the discretized Mackey-Glass series. ANNs of 2-4-1 (Foresee and Hagan, 1997) and 18-6-1 (Conway, 1998) are used for yearly and monthly sunspot series. With trial and error procedure, the chosen ANN structure is 4-3-1 for the SOI series and the two monthly flow series, 23-12-1 for daily flow of Danube, and 16-8-1 for daily flow of Rhine.

The ANNs are constructed with Matlab Neural network toolbox. In all ANNs, tansig transfer function is used in the hidden layer. To avoid of the problem of sensitivity to initial weights, simple ensemble technique is applied. That is, for each network, we run 10 times with different initial weights, then choose five ones, which have best training performance, and take the average of the outputs of the five networks.

2.2 How Many Data Are Demanded to Avoid Overfitting?

Amari et al. (1997) show that, when the ratio (referred to as R hereafter) of the training sample size to the number of weights is larger than 30, no overtraining is observed. This view is accepted by many researchers as a guideline for training ANNs (e.g., Sarle, 2002).

Is there such a clear cut-off value of R? We make experiments for three synthetic series with different values of R ranging from 5 to 50. To avoid the possible impact of nonstationarity, real world data are not applied here. We use the last 1000 points of each synthetic series as the test data, while the training data vary according to the value of R. Networks are trained with Levenberg-Marquardt backpropagation algorithm and the training epoch is 1000. The variations in root mean squared error (RMSE) of training data and test data with different values of R are plotted in Fig. 1.