TIME SERIES PREDICTION VIA NEURAL NETWORK INVERSION

Lian Yan and David J. Miller

Department of Electrical Engineering
The Pennsylvania State University
University Park, Pa. 16802

ABSTRACT

In this work, we propose neural network inversion of a backward predictor as a technique for multi-step prediction of dynamic time series. It may be difficult to train a large network to capture the correlation that exists in some dynamic time series represented by small data sets. The new approach combines an estimate obtained from a forward predictor with an estimate obtained by inverting a backward predictor to more efficiently capture the correlation and to achieve more accurate predictions. Inversion allows us to make causal use of prediction backward in time. Also a new regularization method is developed to make neural network inversion less ill-posed. Experimental results on two benchmark time series demonstrate the new approach's significant improvement over standard forward prediction, given comparable complexity.

1. INTRODUCTION

The goal of time series prediction can be stated succinctly as follows: given a sequence up to time N, $x(1), x(2), ..., x(N)$, find the continuation $x(N+1), x(N+2), ...$ The time series may arise from sampling a continuous time system, and may be either stochastic or deterministic in origin. Statistical methods and neural networks are two major approaches to time series prediction. A statistical method is model driven and parametric. A neural network is data driven and less parametric. Neural networks have been applied to many practical problems and have demonstrated good results, e.g. [3],[10]. They have also shown better performance than statistical methods for some time series problems, especially for long-term prediction [1].

When a neural network is used as a standard (forward) predictor, the estimate is the output of the neural network $NN(\cdot)$:

$$X_o = NN(X_i) \tag{1}$$

Here, $X_i = (x_N, x_{N-1}, ..., x_{N-T+1})$ is the input of the network, and $X_o = (x_{N+1}, x_{N+2}, ..., x_{N+L})$ is the prediction. This network predicts L steps in the future based on T steps in the past. The work in this paper is only for $L > 1$, which is the multi-step (vector) prediction problem. Multi-step prediction is required for many applications, such as extended weather forecast, economic trend estimation, and predictive vector quantization [2]. Predicting farther in the future rather than just the immediate next step makes multi-step prediction a more complex problem than single step prediction.

Sometimes the correlation between L future points and T past points is difficult to learn well, especially for dynamic time series with limited available data. Then (1) may only partially capture the correlation. The correlation that is learned from training in the *positive* time direction (learning a forward predictor) is referred to as *forward correlation*. However, we also can learn the correlation from training in the *negative* time direction [8], thus obtaining what we refer to as *backward correlation*. When the forward correlation is not exactly the same as the backward correlation, their combination will capture more of the correlation than either one alone. We will see, though the backward correlation is non-causal in form, a neural network inversion technique can make the overall system causal and practically useful.

In the next section we will consider neural network inversion. Prediction via neural network inversion will be described in section 3. Then we will show some experimental results on two benchmark time series in section 4.

2. NEURAL NETWORK INVERSION

The problem of training a feedforward neural network is to determine a number of adjustable parameters or connection weights on the basis of a training set. A trained feedforward neural network can be regarded as a nonlinear mapping from the input space to the output space. Once a network has been trained on a training set, all the weights are fixed. Thus the mapping from the input space to the output space is determined. This mapping is referred to as the forward mapping:

$$Y = F(W; X) \tag{2}$$

Here, X is the input vector, Y is the output vector, and W denotes the fixed weights of the trained network. In general, the forward mapping is a many-to-one mapping, because each of the desired outputs usually corresponds to several different possible inputs.

A neural network can be trained via supervised learning, which entails an optimization problem that can be tackled by a gradient descent (backpropagation) procedure [3]. The cost function is the mean square error between the specified (object) output (O_s) and the actual output of the network, as follows:

$$C = E(F(W; X) - O_s)^2 \tag{3}$$

This work was supported in part by the NSF under Career Award IRI-9624870.

The weights are adjusted by a gradient descent method in the following way:

$$W^{i+1} = W^i - \eta \frac{\partial C}{\partial W} \tag{4}$$

Here, η is the learning rate.

In contrast, after a neural network is trained, the problem of inversion is to find inputs X, which yield a given output Y [5], [9]. So inverting a neural network finds the mapping from the output space to the input space, which is referred to as the inverse mapping:

$$F^{-1} = Y \rightarrow X \tag{5}$$

The inverse mapping is usually a one-to-many mapping, so the inverse problem is an ill-posed problem in the sense that it has no unique solution and is highly sensitive to initialization [4], [5].

The inversion problem can also be formulated as minimization of the mean square error between the specified output and the actual output of the network. Now the network has already been trained, and W is fixed, so the actual output depends on the input X rather than W. The cost function for inversion is the same as (3), but we adjust the input X by gradient descent as follows:

$$X^{i+1} = X^i - \eta \frac{\partial C}{\partial X} \tag{6}$$

Neural network inversion has been applied to various problems, such as analyzing and improving generalization performance of trained networks [5], [9], improving and expanding a training set [5], [9], adaptive control [3], classification [4], and speech recognition [6]. In the next section, we develop an algorithm for time series prediction based on neural network inversion.

3. PREDICTION VIA INVERSION

As stated in (1), standard prediction methods only try to learn the correlation in a time series in the positive time direction, which tells us how the time series behaves in the future based on the past sequence. However, for a given training set, we also can learn the correlation in the negative time direction, which tells us what behavior in the past can cause the observed "future" sequence. This kind of information is helpful to grasp the correlation in a time series when it is difficult to learn the correlation only from the positive time direction training. Especially for a highly dynamic series with limited available data, the difficulty to learn the complex correlation function will be more serious. Thus, from the positive time direction training, we learn the *forward correlation*, which only partially represents the correlation. Now, in addition to the forward correlation, the *backward correlation* is learned from the negative time direction, which also partially represents the correlation. Generally, for a nonlinear dynamic time series, the information included in the forward correlation is different from that in the backward correlation. When we combine estimates of both correlations, more knowledge of the correlation can be used for prediction. Consequently, predictions based on bidirectional correlation may be expected to be more accurate, which will be seen in section 4.

For a T_f-step prediction problem, a forward prediction neural network $NN_f(\cdot)$ can be learned from the positive time direction:

$$\tilde{X}_o^f = NN_f(X_i^f) \tag{7}$$

Here, $\tilde{X}_o^f = (\tilde{x}_{N+1}, \tilde{x}_{N+2}, ..., \tilde{x}_{N+T_f})$ is the output of the network and $X_i^f = (x_N, x_{N-1}, ..., x_{N-L_f+1})$ is the input of the network. This network estimates the T_f steps in the future according to the L_f steps in the past.

In addition to the forward prediction (7), which represents the *forward correlation*, a backward prediction network $NN_b(\cdot)$ is also trained from the negative time direction to obtain the mapping, which represents the *backward correlation*:

$$\tilde{X}_o^b = NN_b(X_i^b) \tag{8}$$

Here, $\tilde{X}_o^b = (\tilde{x}_N, \tilde{x}_{N-1}, ..., \tilde{x}_{N-T_b+1})$ is the output of the network and $X_i^b = (x_{N+1}, x_{N+2}, ..., x_{N+L_b})$ is the input of the network. This network estimates the T_b steps in the past according to the L_b steps in the future. Note this network is non-causal in form. After training, it has to be *inverted* during use to estimate future steps based on past steps, and make the overall system (including the forward and backward prediction networks) causal. After the inversion, we get the following mapping:

$$\tilde{X}_i^b = NN_b^{-1}(X_o^b) \tag{9}$$

Here, $X_o^b = (x_N, x_{N-1}, ..., x_{N-T_b+1})$, and $\tilde{X}_i^b$ is an estimate of X_i^b. Generally, we set $L_b = T_f$, which means that we obtain an estimate $\tilde{X}_i^b$ for the next T_f steps via inversion. While the output vector length T_f of the forward prediction network is fixed for a specific problem, the output vector length T_b of the backward prediction network can be chosen freely. Thus, compared to T_f, a smaller T_b can be selected, which may allow one to achieve greater prediction accuracy. It is known that predicting a closer point in time is generally easier than predicting a further one, especially for a dynamic time series, because the correlation among close points in time is stronger. Accordingly, we can predict a shorter vector more accurately than a longer one. In terms of training, a simpler network with less output units is also easier to train. For a time series with limited available data, a smaller network can be trained well with desired generalization ability[1], while a large size network may be overfitted from training. Thus, the backward prediction network is generally more reliable than the forward one. That is another advantage of the new approach.

We have noted that neural network inversion is an ill-posed problem, which highly depends on initialization and has no unique solution, so $\tilde{X}_i^b$ may be unreliable. However, we have also obtained the forward correlation. Thus we suggest to incorporate knowledge of the forward correlation to regularize the inversion. In particular, we propose the following regularized inversion cost function:

$$C_r = E(F(W;X) - O_s)^2 + \lambda E(X - \tilde{X}_o^f)^2 \tag{10}$$

[1] On the other hand, T_b should not be made too small, because that can result in insufficient constraints for inversion and make the inversion more ill-posed. So an optimal T_b exists, which can be estimated heuristically or via a validation set.

Here, $F(\cdot) = NN_b(\cdot)$, and $O_s = X_o^b$. We refer to $E(F(W;X) - O_s)^2$ as the inversion term. By minimizing the cost (10), we obtain the final estimation of X_i^b, which is the vector we want to predict:

$$\tilde{X} = \phi(\tilde{X}_o^f, X_o^b), \tag{11}$$

where $\phi(\cdot)$ is a nonlinear function (implicitly) determined by the inversion. Thus, our regularized inverse combines the estimates based on the forward correlation and the backward correlation to obtain the final estimation.

Our regularized inversion is implemented via a continuation method. The λ in (10) is similar to the "temperature" in the deterministic annealing (DA) method [7]. At the beginning, λ is set to be large, which actually puts a constraint on the initialization of X. Then λ is gradually decreased. For each λ, (10) is minimized over X via gradient descent and an estimate of X is obtained, which is further used as the initialization for the next reduced λ. This approach can avoid some local minima and make the inversion less sensitive to initialization. With λ decreasing, the minimization puts more and more weight on the inversion term, with the regularization constraint only partially satisfied. The terminated λ determines the weight ratio between the backward and forward correlation contributions to the prediction. We can choose λ either heuristically or by a validation set. Here we only select λ heuristically according to the prior knowledge of the time series' noise level. When a time series has little noise, the inversion is more reliable. Then, with a small terminated λ, the prediction can count more on the backward correlation. When a time series is very noisy, a larger terminated λ can be chosen to impose a greater constraint on the agreement with forward correlation during inversion.

4. EXPERIMENTAL RESULTS

In our experiments, we used a three-layer Multilayer Perceptron (MLP) architecture for time series modeling. The activation function was chosen to be the sigmoid function in the hidden units, and output neurons were set to be linear.

4.1. Laser series from the Santa Fe competition

The laser time series consists of readings from an 8-bit A/D converter measuring the intensity of a far-infrared laser. It was used in the Santa Fe competition [10]. The laser can be approximately described by the Lorenz equations, a set of three coupled nonlinear differential equations. The time series is dynamic in origin, but with little noise. We used the first 500 points as the training set and the second 500 points as the test set. The first 1,000 points scaled to the interval 0 to 1 are depicted in Figure 1. We picked the task as predicting the next $T_f = 10$ points. The averaged mean square error (AMSE) was used as the cost metric. AMSE is defined as

$$AMSE = \frac{1}{M}\sum_{n=1}^{M}\frac{1}{T_f}\sum_{i=1}^{T_f}(x_{n+i} - \tilde{x}_{n+i})^2 \tag{12}$$

Here, M is the number of multi-step predictions. We also computed the AMSE normalized by the variance of the source, which we refer to as NMSE. We trained a three-layer MLP forward prediction network with (10/6/10) units for (output/hidden/input) layers, which predicted 10 points ahead based on the past 10 points. The AMSE of the test set was 0.0034 (NMSE: 0.1245) for this single forward prediction network. The backward prediction network was designed as an MLP network with (6/6/10) units, which estimated the past 6 points based on 10 "future" observed points. Because this data set is relatively clean, we chose the terminated λ as (small as) 0.01. The AMSE after the inversion reached 0.0004 (NMSE: 0.0146), which is only 10% of the AMSE of the single forward prediction network. Though this comparison demonstrates that incorporation of backward correlation improves prediction accuracy significantly, it is not fair in terms of complexity. In order to compare fairly, we trained another two MLP forward prediction networks. Each had the same complexity, in terms of number of parameters to train, as the two networks together in the new approach. The first one had (10/12/10) units, for which the AMSE was 0.0016 (NMSE: 0.0586). The other one had (10/9/15) units. Thus, it accessed more points in the past. The AMSE of this network was 0.0021 (NMSE: 0.0769). We can see that the inversion approach still achieves large improvement over the two forward prediction networks (with the same complexity). The prediction error (AMSE) is shown in Figure 2.

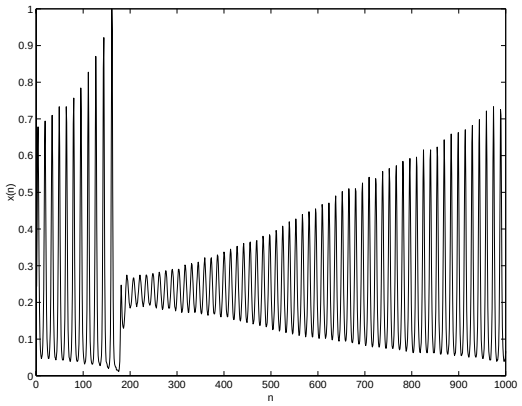

Figure 1: Laser series

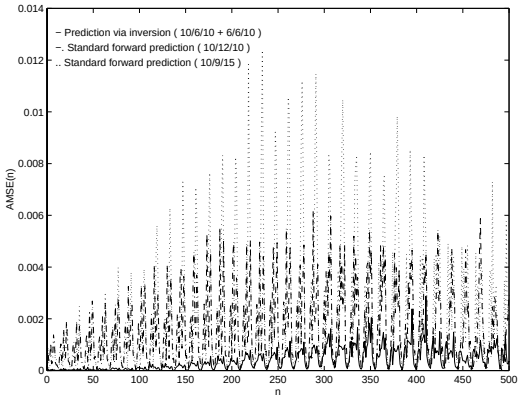

Figure 2: AMSE of the laser series

4.2. Sunspot Series

Sunspots were first observed around 1610, shortly after the invention of the telescope. They are dark blotches on the sun and are often larger in diameter than the earth. The time series is the yearly averages recorded from 1700 to 1988. While the underlying mechanism for sunspot appearances is not exactly known yet, it is known that the series is dynamic and noisy. It has served as a benchmark for time series research. Figure 3 shows the sunspot series scaled to the interval 0 to 1.

For this time series we predicted the next 6 points in the experiment. An MLP network with (6/4/6) units was trained as the forward prediction network, and an MLP network with (3/4/6) units was used as the backward prediction network. The first 200 points were the training set and the other 89 points were the test set. The AMSE of our inversion approach was 0.0258 (NMSE: 0.601), with the single forward prediction network's AMSE being 0.0289 (NMSE: 0.674). The terminated λ was chosen to be 0.1 because of the higher noise level. Again, two other MLP networks with the same complexity as the overall inversion system were trained. One of them had (6/8/6) units, and the other one had (6/6/10) units. The AMSEs for those two networks were 0.0379 (NMSE: 0.88) and 0.1225 (NMSE $>$ 1.0), respectively. Here, we note that the training of the latter failed completely in terms of generalization ability. From this dynamic and small size data set, we can see the difficulty to train a (relatively) large size network, which is supposed to learn the correlation more efficiently, *without* overfitting. The prediction error is depicted in Figure 4. Although the improvement for this noisy series is not as large as for the clean laser series, it is still significant (more than 10%).

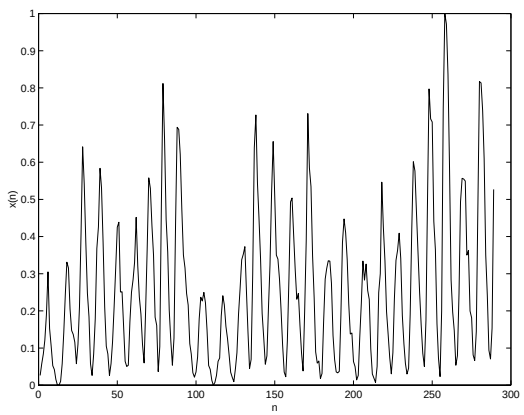

Figure 3: Sunspot series

5. CONCLUSION

For some dynamic and small size data sets, it may be difficult to train a large network to capture the correlations between the samples. Our separate training in the positive and negative time directions is more effective. Each captures some component of the correlation. Then, prediction via inversion combines the *forward correlation* with the *backward correlation* to learn more knowledge about the correlation in a time series, so it can achieve significant

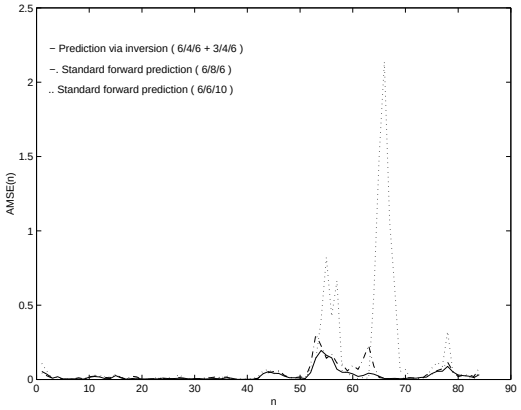

Figure 4: AMSE of the sunspot series

improvement for multi-step prediction problems. The novel contributions of this work include use of bidirectional training and neural network inversion for time series prediction, and a new regularization method is developed for neural network inversion.

6. REFERENCES

[1] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka. Forecasting the behavior of multivariate time series using neural networks. *Neural Networks*, 5:961–970, 1992.

[2] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell, 1992.

[3] S. Haykin. *Neural networks: a comprehensive foundation*. Macmillan, Englewood Cliffs, 1994.

[4] J. Hwang and J. J. Choi. Query-based learning applied to partially trained multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2:131–136, 1991.

[5] J. Kindermann and A. Linden. Inversion of neural networks by gradient descent. *Parallel Computing*, 14:277–286, 1990.

[6] S. Moon and J. Hwang. Robust speech recognition based on joint model and feature space optimization of hidden Markov models. *IEEE Transactions on Neural Networks*, 8:194–204, 1997.

[7] K. Rose, E. Gurewitz, and G. C. Fox. Vector quantization by deterministic annealing. *IEEE Transactions on Information Theory*, 38:1249–1257, 1992.

[8] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681, 1997.

[9] S. Thrun and A. Linden. Inversion in time. In *Proceedings of EURASIP Workshop on Neural Networks*, pages 131–140, 1990.

[10] A. S. Weigend and N. A. Gershenfeld. *Time series prediction: Forecasting the future and understanding the past*. Addison-Wesley, Reading, 1993.