Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks

Timo Koskela, Mikko Lehtokangas, Jukka Saarinen, and Kimmo Kaski

Tampere University of Technology
Electronics Laboratory
FIN-33101 Tampere, Finland
Emails: timoko@ee.tut.fi, mikkol@ee.tut.fi, jukkas@ee.tut.fi, kaski@ee.tut.fi

ABSTRACT

Multilayer perceptron network (MLP), FIR neural network and Elman neural network were compared in four different time series prediction tasks. Time series include load in an electric network series, fluctuations in a far-infrared laser series, numerically generated series and behaviour of sunspots series. FIR neural network was trained with temporal backpropagation learning algorithm. Results show that the efficiency of the learning algorithm is more important factor than the network model used. Elman network models load in an electric network series better than MLP network and in other prediction tasks it performs similar to MLP network. FIR network performs adequately but not as good as Elman network.

1. Introduction

In this paper we study neural network architectures that are capable of learning *temporal features* in data in time series prediction. The feedforward *multilayer perceptron* (MLP) network is used frequently in time series prediction. MLP network, however, has the major limitation that it can only learn an input - output mapping which is *static* [5]. Thus it can be used to perform a nonlinear prediction of a stationary time series. A time series is said to be stationary when its statistics do not change with time. In many real world problems, however, the *time* when certain feature in the data appears contains important information. More specifically, the interpretation of a feature in data may depend strongly on the earlier features and the time they appeared. A common example of this phenomenon is speech.

A conventional way of modelling stationary time series with MLP networks is presented in Fig. 1. The input vector to the network consists of past samples of the time series as follows: $\mathbf{x} = [x(n-1), x(n-2),..., x(n-p)]^T$. Here parameter p is the *prediction order.* The scalar output $y(n)$ of the MLP network equals *one-step prediction* $y(n) = \hat{x}(n)$. The actual value $x(n)$ of the series represents the desired output. The network tries to model time by giving it a spatial representation. It is not, however, able to deal with time-varying sequences. A better solution is to let time have an effect on the networks response rather than represent time by additional input dimension. This can be achieved when the network has *dynamic* properties such that it will respond to temporal sequences.

2. FIR and Elman Neural Networks

Neural network must contain *memory* in order to process temporal information. There are two basic ways to build memory into the neural networks [5]. The first one is to introduce *time delays* in the network and to adjust their parameters during the learning phase. The second way is using *positive feedback*, that is making the network *recurrent*. To characterize memories in different architectures, two dimensions, *depth* and *resolution* has been proposed. Roughly, depth refers to how far into the past the memory stores information relative to the memory size, and resolution how accurately information concerning the individual elements of the input sequence is preserved [2]. In time series prediction with neural networks the main problems have been deciding prediction order and the structure of the network. These problems remain mostly the same for the architectures studied. In addition, the stability of the model and the learning algorithm must be considered.

In *FIR (Finite impulse response) neural network* each neuron is extended to be able to process temporal features by replacing synapse weights by finite impulse response filters. A general structure of a FIR filter is shown in Fig. 2. A multilayer feedforward network is then built using these neurons as shown in Fig. 3. Networks input layer consists of FIR filters feeding the data into neurons in hidden layer. Network may have one or several hidden layers. Output layer consists of neurons which receive their inputs from previous hidden layer. At each time increment, one new value is fed to input filters, and output neuron produces one scalar value. In effect this structure has the same functional properties as Time Delay Neural Network (TDNN) [6]. However, FIR network is more clearly interpreted as a vectoral and temporal extension of MLP. This interpretation also leads to the temporal backpropagation learning algorithm, which is used to train the network [7].

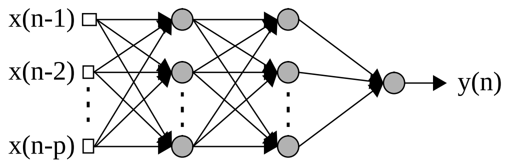

Fig. 1: Multilayer perceptron network used as one-step predictor of a time series.

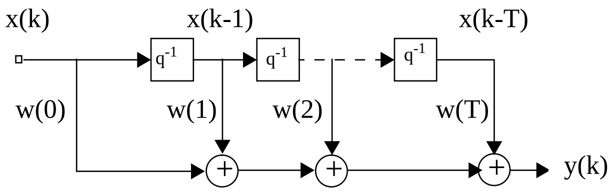

Fig. 2: Finite impulse response filter.

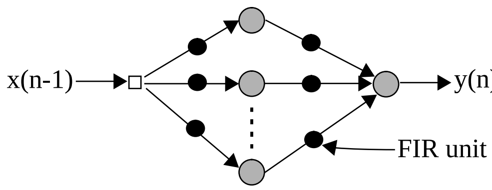

Fig. 3: FIR neural network with one hidden layer, one input and one output.

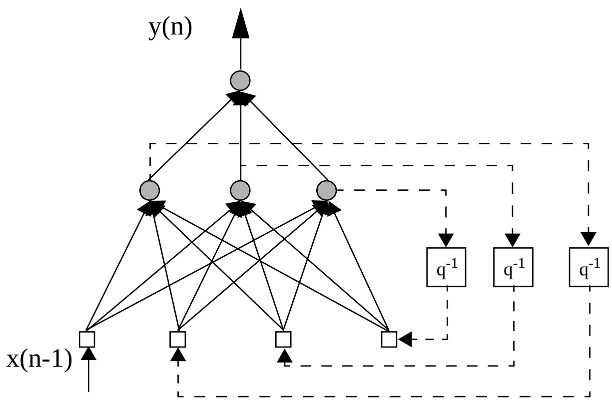

Fig. 4: Elman neural network with one input and one output.

Another training algorithm that is guaranteed to converge to a solution has been proposed in [1]. FIR network is stable and it has a high resolution, low depth memory. In effect the network is unable to learn temporal features that are longer than its filter lengths summed together. Consequently, selection of the lengths of FIR filters is quite critical in achieving good prediction performance.

In *Elman network* positive feedback is used to construct memory in the network as shown in Fig. 4 [4]. The network has input, hidden and output layers. Special units called *context units* save previous output values of hidden layer neurons. Context unit values are then fed back fully connected to hidden layer neurons and thus they serve as additional inputs to the network. Networks output layer values are not fed back to network. The Elman network has a high depth, low resolution memory, since the context units keep exponentially decreasing trace of past hidden neuron output values. The difference to FIR network is that the memory in the network has no rigid limit, and the fact that the information concerning previous data is preserved with better resolution than more distant data in the past.

3. Experiments and Results

The time series that have been modelled are plotted in Figs. 5 and 6. Series 1 represents load in an electric network. Series 2 and 3 were used in Santa Fe Time Series Prediction and Analysis Competition [8]. Series 2 represents fluctuations in a far-infrared laser and series 3 is a numerically generated series. Series 4 represents behaviour of sunspots [9]. All data sets except series 4 were scaled between [-1, 1]. Series 4 was scaled between [0,1] so that we can compare our results with earlier studies [9]. Table 1 shows number of data used to train networks and to test network generalization ability for each time series. Training data starts from the beginning of the series and test data starts from the end of the training data.

Simulations were done with MLP network that has one hidden layer and one nonlinear output neuron, Elman network that has one linear output neuron, and FIR neural network that has one hidden layer and one nonlinear output neuron. Different combinations of prediction order and number of neurons in hidden layer were tried in effort to find the architecture that would model the data most effectively. For FIR networks also the length of FIR filters feeding output neuron was a free parameter. MATLAB neural network toolbox training functions **trainlm** and **trainelm** were used for training MLP and Elman networks, respectively [3]. For FIR networks temporal backpropagation algorithm [7] was implemented with MATLAB.

Table 1: Sizes of training and test sets.

Time Series	1	2	3	4
Training set	1500	1000	3000	221
Test set	500	1000	3000	58

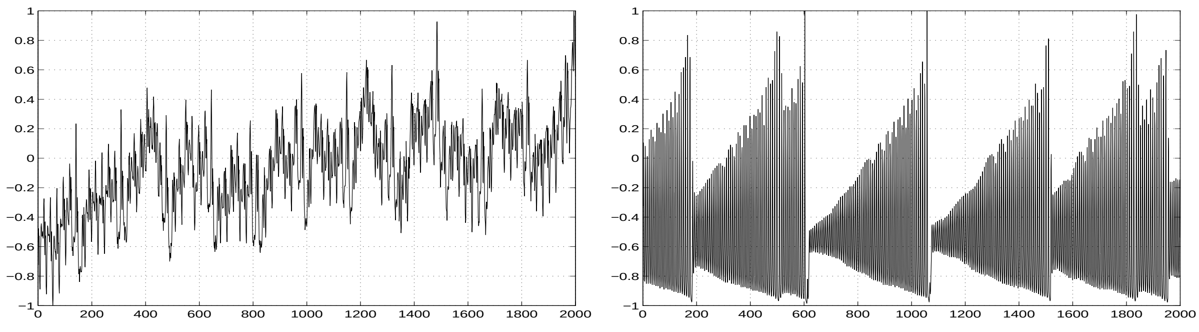

Fig. 5: Load in an electrical net (series 1) and fluctuations in a far-infrared laser (series 2).

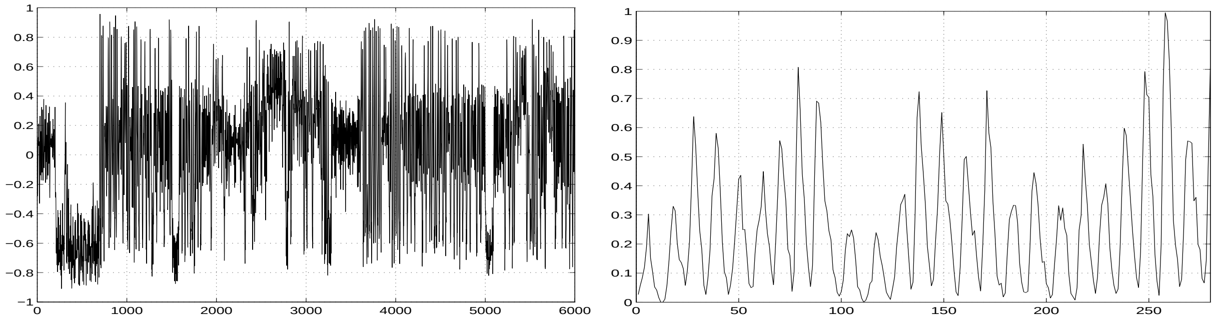

Fig. 6: Numerically generated series (series 3) and behaviour of sunspots (series 4).

Normalized mean square error (NMSE) was used as performance measure. In Eq. (1) σ^2 is the variance of the desired outputs d_i and N is the number of patterns.

$$NMSE = \frac{1}{N\sigma^2} \sum_{i=1}^{N} \left(x_i - d_i\right)^2 \tag{1}$$

Table 2. shows architectures which gave lowest NMSE for test data for each time series and for each neural network model. The number of inputs equals to prediction order p for MLP and Elman networks. For FIR networks the lengths of the FIR filters are shown. For Elman networks number of context units equals number of neurons in hidden layer.

The Elman network performs best in time series 1, which has a low-frequency trend. The network predicts the slope of the trend in the end of the testing data more accurately than MLP network. In other prediction tasks the Elman network performs nearly as good as MLP.

Table 2: Architectures which gave the lowest NMSE for test set
for each time series and neural network model.

Time Series	Network	Number of Inputs	Neurons in hidden layer	NMSE for training set	NMSE for test set
Load in an electrical net (1)	MLP	25	4	0.0152	0.0342
	Elman	25	5	0.0194	0.0249
	FIR	25 - 5	4	0.0416	0.0710
Fluctuations in a far-infra-red laser (2)	MLP	6	8	0.00639	0.00815
	Elman	8	6	0.00971	0.0161
	FIR	3 - 2	6	0.2556	0.2662
Numerically generated series (3)	MLP	5	10	0.0128	0.0152
	Elman	2	6	0.0199	0.0261
	FIR	4 - 2	4	0.0278	0.0305
Behaviour of sunspots (4)	MLP	4	7	0.0866	0.0979
	Elman	10	3	0.1203	0.1181
	FIR	8 - 3	4	0.2933	0.2571

The FIR network was unable to learn adequately series 2 and 4, but it performed quite well on the two other time series. Series 2 is a stationary, noise-free laboratory measurement data and series 4 is also stationary time series which has quite few training patterns. Consequently, the learning algorithm seems to be unable to train the network to model these series. On other two time series FIR network did not perform as good as MLP, which is also due to the learning algorithm. Since the temporal backpropagation algorithm updates network weights at each time increment, it uses an approximation of the gradient. Thus the learning rate must be kept substantially small, which slows the learning process compared with epoch-wise learning algorithms. Also FIR network has one more free parameter and its selection can be crucial to the network performance.

Since all prediction tasks were one-step predictions, MLP network performs well. The learning algorithm used is quite important factor in networks performance. In our present studies we are considering multistep prediction tasks. Preliminary results indicate that in multistep prediction tasks the temporal extension in FIR and Elman neural networks allows these architectures to perform better than MLP network.

4. Conclusions

The results show that the efficiency of the learning algorithm is more important factor than the neural network model used. The learning algorithms used with Elman and FIR neural networks were unable to fully implement the richer structure of these networks. Training of Elman networks was three to ten times slower than for MLP depending on the training data size and the number of network parameters. For FIR network training time was five to twenty times longer than for MLP. Elman network models series 1 best, and its performance in other tasks is similar as MLP networks performance. For FIR networks trained with temporal backpropagation adequate performance was reached for two prediction tasks.

Acknowledgements

This study was financially supported by the *Academy of Finland*.

References

[1] A.D. Back and A.C. Tsoi, “FIR and IIR synapses, a new neural network architecture for time series modeling”, Neural Computation, Vol. 3, pp. 375-385, 1991.

[2] B. de Vries and J. Principe, “The gamma model: A new neural net model for temporal processing”, Neural Networks, Vol. 5, pp. 565-576, 1992.

[3] H. Demuth and M. Beale, Neural Network Toolbox for Use with MATLAB, The MathWorks Inc., April 1993.

[4] J.L. Elman, “Finding structure in time”, Cognitive Science, Vol. 14, pp. 179-211, 1990.

[5] S. Haykin, Neural networks: A comprehensive foundation, Macmillan College Publishing Company, New York, 1994.

[6] K. Lang, A. Waibel and G. Hinton, “A time-delay neural network architecture for isolated word recognition”, Neural Networks, Vol. 3, pp. 23-43, 1990.

[7] E. Wan, Finite impulse response neural networks with applications in time series prediction, Ph.D. Dissertation, Stanford University, November 1993.

[8] A. Weigend and G. Gershenfeld (eds.), Time series prediction: Forecasting the future and understanding the past, Addison-Wesley, Reading, 1994.

[9] A. Weigend, B. Huberman and D. Rumelhart, “Predicting the future: A connectionist approach”, Int. Journal of Neural Systems, Vol. 1, pp. 193-209, 1990.