OBTAINING NEURAL NETWORKS BASED STATE SPACE MODELS USING TIME-LAGGED NEURONS

F. N. Chowdhury, K. N. Rao, V G. Siddhanti

University of Louisiana at Lafayette, Lafayette, USA

Keywords

Time-lagged neural networks, state space models, real-time training

Abstract

A new method of obtaining discrete-time state space models from experimental data is presented in this paper. Time-Lagged Recurrent Neural Networks (TLRN) are used for this purpose. Unlike the conventional multiple layer perceptron (MLP), the TLRN topology includes memory elements, which allow it to represent patterns that occur as functions of time over multiple samples. State space models of dynamic systems can be directly mapped onto these TLRN structures. Simulation results from preliminary studies indicate that this tool could be developed into a very promising technique in the field of dynamic system modeling and identification. With minor changes, the method can be applied to both linear and nonlinear systems.

1 Introduction

In this paper, a new method of obtaining discrete-time state space models from experimental data is presented. To obtain the state space model, we use a special kind of neural network: Time-Lagged Recurrent Neural Networks (TLRN). TLRN is a neural network topology that includes memory elements, which allow it to identify patterns that occur over multiple samples. Typically, neural networks based system identification has focused on input-output models [1, 2]. In this paper, we draw attention to the fact that by changing the basic model of the neuron from a memoryless unit to one with one-step-back memory, we can map state space models directly onto the neural network. The main motivation for this line of research is briefly described below.

Practically all the work in the field of nonlinear control system design is concentrated on systems having a state space representation [3, 4]. It is interesting to note that despite the large volume of work going on in both nonlinear system identification and (state-space based) controller design fields, there is serious gap between the two. On one hand, researchers working on controller design assume that either the right model is available, or the plant is identifiable in the "suitable" model structure. On the other hand, neural networks researchers in nonlinear system identification have developed sophisticated methods of accurately estimating NN-based NARMA (Nonlinear AutoRegressive Moving Average) models, but have not paid much attention to the fact that these models are not realizable in the classical state space form, and consequently, not useful in controller design using existing state space theory.

Our goal is to develop new types of neural networks that can directly generate state space models. This paper reports ongoing work on this topic, and presents some results; however, it has become evident during our studies so far that the existing training algorithms need to be modified in order to successfully train these state space networks - so our current work includes new architechtures and new training algorithms.

2 Neural Network Models for Dynamic System Identification

Nonlinear system identification is an active research field, and the most successful identification techniques are currently based on recursive input-output (i/o) models. These are usually NARMA-type models, given by:

$$y_t = \Psi(y_{t-1}, \ldots, y_{t-n}, u_{t-1}, \ldots, u_{t-n}), \tag{1}$$

that can be estimated by various techniques. Function Ψ is often chosen to be a feed-forward neural network with $2n$ inputs and one output since neural networks have good function approximation abilities. The linear counterpart of these input-output models is the ARMA or ARMAX, which are very well known in the literature.

It has been shown previously [6] that the typical NN-based NARMA-type model does not lend itself to the classical state space realization. A few authors have suggested a new subclass of NN-based models that can be easily realized in the classical state-space form [6] - this new subclass is called the "Additive NARMA" (ANARMA) model. They have proposed new ways of NN training to estimate the parameters of the ANARMA model [6, 5]. But the approximation ability of the special structure corresponding to this subclass of NN-based models has not yet been established theoretically and remains the topic of future research. While work is going on in that direction, it is worth exploring the possibility of direct state-space modeling from experimental data. The main contribution of this paper is that it presents an alternative to the conventional input-output models, and it uses the power of time-lagged neurons to achieve a direct mapping of the state space onto the neural network.

3 Description and Training of TLRN

The Time-Lagged Recurrent neural Network is a special kind of neural network, first proposed by Werbos [2]. This neural network topology includes memory elements, and self-feedback of neurons. The output of the $j-th$ time-lagged neuron in a network at time step k is described by:

$$Y_j(k) = \sigma(\Sigma_{i=1}^{n} W_{ij} U_i(k-1) + \Sigma_{i=n+1}^{n+1+m} W_{ij} Y_{i-n}(k-1)), \tag{2}$$

where σ is the activation function, W_{ij} is the weight, $u_i(k)$ is the $i-th$ external input to the network, n is the number of external inputs, and m is the number of neurons in the network.

In the NN architecture proposed in this paper, only the hidden-layer neurons are time-lagged; the outer layer neurons are conventional memoryless linear units. In this section, we show the mapping of state models onto a TLRN. The discrete-time linear state space model is given by the equation

$$x(k+1) = Fx(k) + Gu(k) \tag{3}$$
$$y(k) = Hx(k).$$

The corresponding nonlinear state model used in this work is given by:

$$x(k+1) = f\{x(k), u(k)\} \tag{4}$$
$$y(k) = g\{x(k)\}.$$

In the above equations, F, G, H constitute the linear state space model, and the functions f, g represent the underlying nonlinear equations in the nonlinear state model.

Re-arranging Eq. 3, we get:

$$x(k+1) = Wq(k), \tag{5}$$

where $W = [F, G]$ is the weight matrix for the hidden layer, and $q(k) = [x(k), u(k)]$ is the input to the NN. At every time-step k, the vector $q(k)$ can be computed from current data. The output of the hidden layer is $x(k)$, which is fed back, with a one-step delay, to the input layer. It is thus clear that training of such models can only be accomplished in an incremental fashion: batch-mode training is not applicable.

4 Illustration of the Method for State-Space Modeling with Input/Output Data

A real time simulation model of the F-14 aircraft is taken from MATLAB/SIMULINK to illustrate the neural network based black-Box modeling approach. F-14 is an aircraft model built in simulink that takes the pilot signals as input and produces two outputs: the angle of Attack and The Gravitational Force. In our simulations, a Gaussian random sequence is used as the pilot stick input. This stick input and the Angle of attack are used in the neural network for training. The angle of attack is defined as the angle between the plane of the wing (airfoil chord) and the direction of motion (free stream velocity). The nomenclature and the differential equations involved are:

- α = Angle Of Attack
- q= Pitch Rate
- W_{gust} = Vertical gust (rad/sec)
- Q_{gust} = Rotary gust (rad/sec)
- d = Elevator Deflection (rad)
- nz = Gravitational Force (n)
- W = Vertical Velocity (ft/sec)

Elevator Command

$$E(k) = ((u(k)\frac{1}{TS.s+1}) - (q(k-1)K_q\frac{s+w1}{s+w2} + \alpha(k-1)K_a\frac{1}{Ta1.s+1}))(K_f + \frac{K_i}{s}) \quad (6)$$

Vertical Velocity

$$W(k) = (d(k)Z_d - W_{gust} + q(k-1)U_0)\frac{1}{s-zw} \quad (7)$$

Pitch Rate

$$q(k) = (d(k)M_d - q_{gust} + W(k-1)M_w)\frac{1}{s-M_q} \quad (8)$$

Gravitational Force

$$nz = (s.q(k)22.8 + U_0q(k) - W(k)s)/g \quad (9)$$

Angle of Attack

$$\alpha = q(k)/U_0 \quad (10)$$

The angle of Attack is also influenced by the Drydem wind gust model block. The gust

is produced by a white noise block in simulink. We used the angle of attack as the output of the system.

Using the input-output data from the F-14 model, we train a TLRN with conventional backpropagation algorithm. In this version of the paper, the results of a linear model-fitting are presented. The result is shown in Fig. 1. It is clear that the training is not entirely successful: numerical instabilities tend to occur due to the past time-dependencies in the algorithm. We trained the network for 200 data points, and then the weights were frozen. Current work in concentrating on refining the training method and exploring other training techniques [7, 8]. Training of the nonlinear version of the modeling approach is also underway, and a comparison with the conventional NARMA/ARMA modeling is being done.

5 conclusion

In this paper, we used a new approach to dynamic system modeling using time-lagged neural networks. State space modeling using input-output data from the MATLAB F-14 model is shown. This is work in progress, and we expect to have more results by the time of the conference.

References

[1] K. S. Narendra, “Adaptive Control of Dynamical Systems Using Neural Networks”, in *Handbook of Intelligent Control*, Ed. D. White D. and Sofge, Van Nostrand Reinhold, pp. 141-183, New York, 1992

[2] P. J. Werbos, T. McAvoy and T. Su, “Neural Networks, System Identification, and Control in the Chemical Process Industries”, in *Handbook of Intelligent Control*, Ed. D. White and D. Sofge, Van Nostrand Reinhold, pp. 283-356, New York, 1992

[3] Isidori, A. Nonlinear Control Systems, 1989, Berlin, 2nd Ed.

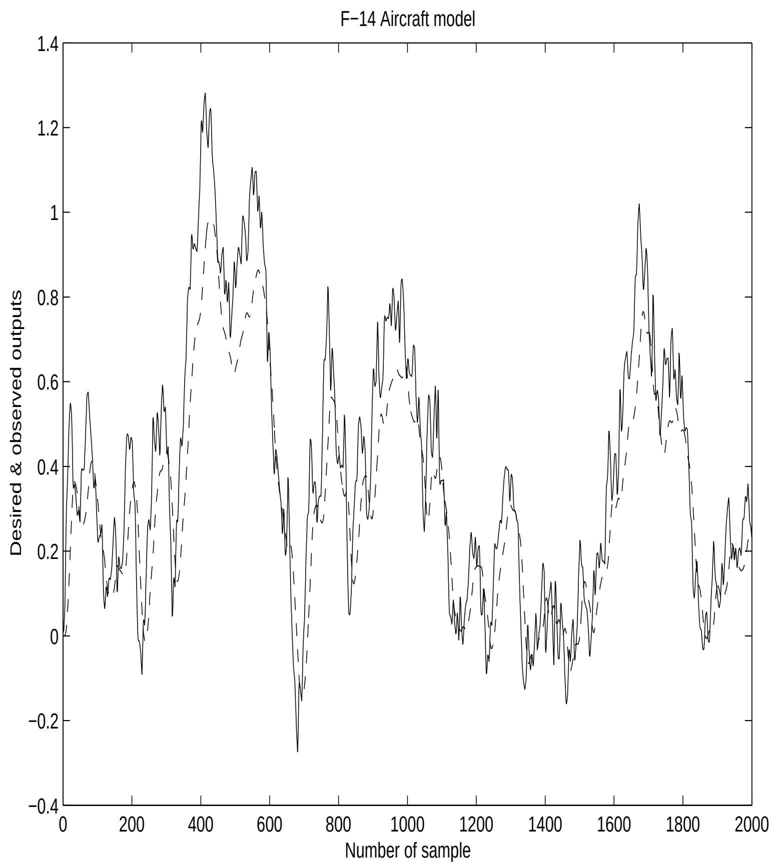

Figure 1: Actual plant output and TLRN-state-space-model output for the F-14 model

[4] Nijmeijer, H., van der Schaft, A., J. Nonlinear Dynamical Control Systems. 1990, Springer Verlag, New York.

[5] Chowdhury, F. N., "A Novel Method for On-line Training of Dynamic Neural Networks", *Proc. IEEE Conf. Control Applications*, Mexico City, Mexico, Sept. 2001

[6] Chowdhury, F. N., Ü. Kotta and S. Nõmm, "On Realizability of Neural-networks-based Input-output Models", *Proc. 3rd Int. Conf. on Differential Equations and Applications*, St. Petersburg, Russia, June 2000

[7] Rivals, I. and L. Personnaz, "Black-box Modeling with State-space Neural Networks", in *Neural Adaptive Control Technology*, World Scientific, 1995.

[8] Bone, R. and M. Crucianu, "An Algorithm for the Addition of Time Delay Connections to the Recurrent Neural Networks", *Proc. European Symposium on Artificl Neural Netwotks*, April 2000