

Time Delay Dynamic Fuzzy Networks for Time Series Prediction

Yusuf Oysal

Anadolu University, Computer Engineering Department, Eskisehir, Turkey
yoysal@anadolu.edu.tr

Abstract. This paper proposes a Time Delay Dynamic Fuzzy Network (TDDFN) that can be used for tracking and prediction of chaotic time series. TDDFN considered here has unconstrained connectivity and dynamical elements in its fuzzy processing units with time delay state feedbacks. The minimization of a quadratic performance index is considered for trajectory tracking applications. Gradient with respect to model parameters are calculated based on adjoint sensitivity analysis. The computational complexity is significantly less than direct method, but it requires a backward integration capability. For updating model parameters, Broyden-Fletcher-Golfarb-Shanno (BFGS) algorithm that is one of the approximate second order algorithms is used. The TDDFN network is able to predict the Mackey-Glass chaotic time series and gives good results for the nonlinear system identification.

1 Introduction

Some of the nonlinear dynamical systems produce chaotic time series outputs that are highly depend on initial conditions. If the initial condition does not specified within a suitable precision range, it is very difficult to predict the long time future behavior of these time series. But the short time behavior can be exactly encapsulated. There are various types of evolutionary systems and neural networks with time delays to solve time series prediction in short time. For example in [1], an adaptive-network-based fuzzy inference system (ANFIS) was used to identify nonlinear components on-line in a control system to predict a chaotic time series. In another study, a genetic fuzzy predictor ensemble (GFPE) was proposed for the accurate prediction of the future in the chaotic or nonstationary time series [2]. Moreover, an evolutionary system, i.e., EPNet was used to produce very compact artificial neural networks (ANNs) to predict the Mackey-Glass time series prediction with generalization ability in comparison with some other algorithms [3] and in [4] a hybrid approach to fuzzy supervised learning was applied through software called GEFREX for approximation problems, classification problems, and time series prediction.

Another approach for time series prediction is to provide dynamical neural network structures. The typical workaround is the usage of a large parallel input vector consisting of a number of states or past samples of process data. This “tapped delay line”

approach has proven successful for chaotic time series prediction ([5],[6]), but it has the drawback of the curse of dimensionality: the number of parameters in the units increases exponentially and parameters can get larger values.

This work focuses on modeling and prediction of nonlinear systems with time delay dynamic fuzzy networks (TDDFNs) to overcome these drawbacks. TDDFNs are continuous-time recurrent neural networks that contain dynamical elements such as integrators in their fuzzy processing units and time delayed feedbacks. Successful control and modeling applications of DFNs without time delay elements can be found in [7] and [8]. In this study an approximate second order gradient algorithm based on adjoint theory [9]-[12] which is faster than the direct method is used for training the TDDFNs to obtain the appropriate parameters. Given a desired trajectory, a nonlinear optimization problem is solved to determine appropriate values for network parameters.

2 Time Delay Dynamic Fuzzy Network Architecture

The dynamic fuzzy network considered here represents the biological neuron that is constrained to be feedforward with dynamic elements in its fuzzy processing units, and with time delay state feedbacks. The processing unit is called “feuron” (stands for fuzzy neuron) [7],[8]. It represents a biological neuron that fires when its inputs are significantly excited through a lag dynamics (i.e. Hopfield dynamics).

The feuron’s activation model which resembles the receptive field units found in the visual cortex, in parts of the cerebral cortex and in outer parts of the brain is a standard fuzzy system with Gaussian membership functions, singleton fuzzifier, product inference engine and a center average defuzzifier [13].

The activation function of the i^{th} feuron can be expressed as:

$$\phi_i(x_i) = \frac{\sum_{k=1}^{R_i} b_{ik} \mu_k(x_i)}{\sum_{k=1}^{R_i} \mu_k(x_i)} = \frac{\sum_{k=1}^{R_i} b_{ik} \exp\left(-\frac{1}{2}\left(\frac{x_i - c_{ik}}{\sigma_{ik}}\right)^2\right)}{\sum_{k=1}^{R_i} \exp\left(-\frac{1}{2}\left(\frac{x_i - c_{ik}}{\sigma_{ik}}\right)^2\right)} \tag{1}$$

where c_{ik} is the center and σ_{ik} is the spread of the k^{th} receptive field unit of the its i^{th} feuron.

The membership functions of the feuron are assumed to be normal and orthogonal with the boundary membership functions (the lower and upper membership functions of the universe of discourse) represented by hard constraints, i.e., it is assumed that membership value is equal one at out of range.

An example of the computational model for TDDFNs with two-feuron and two-inputs/two-outputs that is used in this study is shown in Fig.1.

The general computational model that we have used for TDDFN is summarized in the following equations:

$$z_i = \sum_{j=1}^n q_{ij} y_j, \quad i=1,2,\dots,M \tag{2}$$

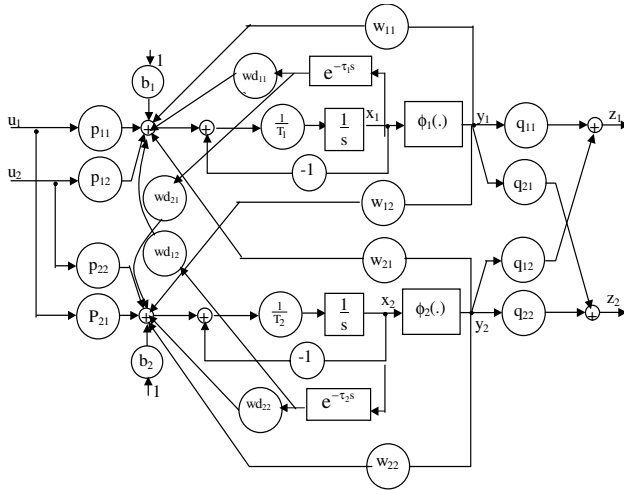


Fig. 1. The state diagram of a TDDFN with two-neurons two-inputs/two-outputs

$$y_i = \phi_i(x_i, \pi_i) \tag{3}$$

$$\dot{x}_i = f_i(x_i, p) = \frac{1}{T_i} [-x_i + \sum_{j=1}^n w_{dij} x_j(t - \tau_j) + \sum_{j=1}^n w_{ij} y_j + \sum_{j=1}^L p_{ij} u_j + r_i]; x_i(0) = x_{i0}, i = 1..n \tag{4}$$

where w , p , q , r are the interconnection parameters of a TDDFN with n units and L input signals, T is the time constants and π is the parameter sets (centers c , spreads σ , output centers b) corresponding to fuzzy SISO activation functions of the neurons.

In general, the units have dynamics associated with them, as indicated, and they receive inputs from themselves and delayed-themselves and all other units. The output of a unit y_i is a standard fuzzy system $\phi(x_i, \pi_i)$ of a state variable x_i associated with the unit. The output of the network is a linear weighted sum of the unit outputs. Weights p_{ij} are associated with the connections from input signals j to units i , w_{ij} with interunit connections from j to i , w_{dij} with delay-interunit connections from j to i and q_{ij} is the output connection weights from j^{th} neuron to i^{th} output. T_i is the dynamic constant, r_i is the bias (or polarization) term and τ_i is the time-delay of i^{th} neuron.

3 Illustrative Examples of Some Dynamical Behaviors of TDDFN

This model (TDDFN) can be used to approximate many of the behaviors of nonlinear dynamical systems with time delay. In this section, examples are given in which TDDFN converges to a point attractor, a periodic attractor (limit cycle). For this aim, given a set of parameters, initial conditions, and input trajectories, the set of equations (2), (3) and (4) can be numerically integrated from $t=0$ to some desired final time t_f . This will produce trajectories overtime for the state variables x_i ($i=1..n$). We have used Adams-Bashforth predictor method, extended with trapezoidal corrector in some

cases. The integration step size has to be commensurate with the temporal scale of dynamics, determined by the time constants T_i . In our work, we have specified a lower bound on T_i and have used a fixed integration time step of some fraction (e.g., 1/10) of this bound.

As a first example, a TDDFN is modeled as a point attractor system by a training algorithm whose details will be given in the next section. The interconnection parameters of the TDDFN given in Fig. 1 after training are found to be:

$$w = \begin{bmatrix} 0 & 2 \\ -1 & 0 \end{bmatrix}, wd = \begin{bmatrix} -0.5 & 0 \\ 0 & -.5 \end{bmatrix}, r = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

And the data for time constants, input centers, spreads and output centers with three membership functions in each feuron respectively are:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, c = \begin{bmatrix} -0.791 & -1.356 & 0.087 \\ 0.996 & -0.974 & 0.785 \end{bmatrix},$$

$$\sigma = \begin{bmatrix} 1.412 & 0.770 & 1.283 \\ 0.837 & 1.116 & 0.875 \end{bmatrix}, b = \begin{bmatrix} 0.542 & -0.764 & 1.211 \\ 0.898 & 0.176 & 1.176 \end{bmatrix}$$

In this case initial conditions are chosen as $x(0) = [1 \ 1]^T$. Fig. 2a shows the example of zero-input state space trajectories for two-feuron network with time delay (19 seconds) that converges to a point attractor.

As a second example, a TDDFN is modeled as a periodic attractor system. In this case the interconnection parameters of the TDDFN are calculated as:

$$w = \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}, wd = \begin{bmatrix} -0.85 & 0 \\ 0 & -0.85 \end{bmatrix}, r = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

The data for time constants, input centers, spreads and output centers with three membership functions in each feuron respectively are:

$$T = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, c = \begin{bmatrix} -4.174 & 1.418 & -2.987 \\ -0.470 & -0.177 & 0.374 \end{bmatrix},$$

$$\sigma = \begin{bmatrix} 1.988 & 2.296 & 2.167 \\ 1.015 & 1.041 & 0.877 \end{bmatrix}, b = \begin{bmatrix} 0.001 & 1.000 & -0.007 \\ -0.551 & 1.152 & 1.206 \end{bmatrix}$$

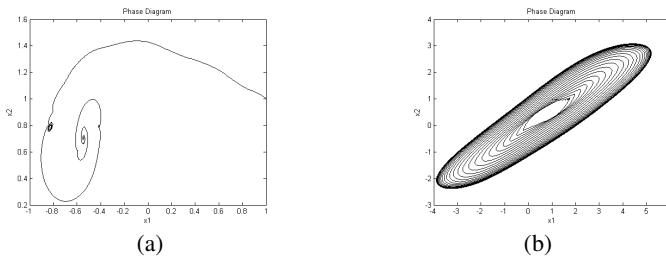


Fig. 2. State space trajectory of TDDFN a) Point attractor, b) periodic attractor

In this application trajectory tracking performance is excellent with initial conditions $x(0) = [1 \ 1]^T$. Fig. 2b shows the example of zero-input state space trajectories for two-feuron network with time delay (19 seconds).

4 Training of TDDFN Based on Adjoint Sensitivity Analysis

In this section, we consider a particular application of the TDDFN: trajectory tracking for process modeling. That is, we desire to configure the network so that its output and input trajectories have been specified, but all the parameters w, w_d, r, τ and π, T are adjustable. This is done by minimizing the cost functional. We associate a performance index of the network for this task as given by

$$J = \frac{1}{2} \int_0^{t_f} [z(t) - z^d(t)]^T [z(t) - z^d(t)] dt \tag{5}$$

where $z^d(t)$ and $z(t)$ are actual and modeled process responses respectively.

Our focus in this paper has been on gradient-based algorithms for solving this computational problem. We require the computation of gradients or sensitivities of the performance index with respect to the various parameters of the TDDFN:

$$\left[\frac{\partial J}{\partial w}, \frac{\partial J}{\partial w_d}, \frac{\partial J}{\partial \tau}, \frac{\partial J}{\partial T}, \frac{\partial J}{\partial r}, \frac{\partial J}{\partial c}, \frac{\partial J}{\partial \sigma}, \frac{\partial J}{\partial b} \right] \tag{6}$$

In this study above gradients are calculated by ‘‘adjoint’’ method which is based on the use of calculus of variations [9]-[12], [14]. In this method, a set of dynamical systems is defined with adjoint state variables λ_i :

$$-\dot{\lambda}_i = -\frac{\lambda_i}{T_i} + \frac{1}{T_i} \sum_j w_{ij} x'_j(t - \tau_j) \lambda_j - \frac{1}{T_i} \sum_j w_{ij} y'_j \lambda_j + e_i(t) \sum_j q_{ij} y'_j; \lambda_i(t_f) = 0 \tag{7}$$

where $y'_j = \frac{\partial \phi_j(x_j)}{\partial x_j}$ and can be computed by partial differentiation of (3):

$$\frac{\partial \phi_j(x_j)}{\partial x_j} = \frac{\sum_{k=1}^{R_j} (\phi_j - b_{jk}) \exp\left(-\frac{1}{2} \left(\frac{x_j - c_{jk}}{\sigma_{jk}}\right)^2\right) \left(\frac{x_j - c_{jk}}{\sigma_{jk}^2}\right)}{\sum_{i=1}^{R_j} \exp\left(-\frac{1}{2} \left(\frac{x_j - c_{jk}}{\sigma_{jk}}\right)^2\right)} \tag{8}$$

The size of the adjoint vector is thus n and is independent of the number of DFN parameters. The computation of sensitivities using the adjoint method requires the solution of n differential equations. This is a significant savings for real-time applications. Then, the cost gradients with respect to TDDFN parameters are given by the following quadratures;

$$g = \frac{\partial J}{\partial p} = \int_0^{t_f} \left(\frac{\partial f}{\partial p}\right)^T \lambda dt \tag{9}$$

The cost gradients as in [7], [8], [12] can be easily computed. We assume that at each iterations, gradients of the performance index with respect to all TDDFN parameters are computed. Once g is computed several of gradient-based algorithms can be used to update parameter values of the TDDFN. Here for updating model parameters, Broyden-Fletcher-Golfarb-Shanno (BFGS) algorithm [15] that is one of the approximate second order algorithms is used.

5 Mackey-Glass Time Series Prediction with TDDFNs

This section deals with a complex problem of approximating a nonlinear dynamical time series using TDDFN. Here, a benchmark chaotic time series first investigated by Mackey and Glass [16] which is a widely investigated problem in the fuzzy-neural literature [1], [2] is considered. The time series is generated by the following differential equation:

$$\frac{dx}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{10}$$

As τ in this equation varies, the system can exhibit either fixed point, limit cycle or chaotic behavior. For $\tau = 17$ the systems response is chaotic and we attempt the problem of approximating time series function of (10) for this value. The fifth order Runge-Kutta method was used to obtain simulation data with the following initial conditions $x(0)=1.2$ and $x(t-\tau)=0$ for $0 \leq t < \tau$.

A TDDFN with one feuron was used in the simulations. Time delay of the feuron was taken to be the fixed value as the same as the Mackey-Glass, $\tau=17$. The other weights are adjusted as presented previously. The first 100 data points were used to train the TDDFN. The prediction performance of the TDDFN was tested after 200th data points. Fig. 3 shows the result of the test with 200 training points. As seen in Fig. 4, the neural network prediction capability is excellent. Table 1 compares the performance of TDDFN with various classical models, neural networks, and fuzzy neural networks. The comparison is based on normalized root mean square error (NRMSE), which is defined as the RMSE divided by the standard deviation of the target series.

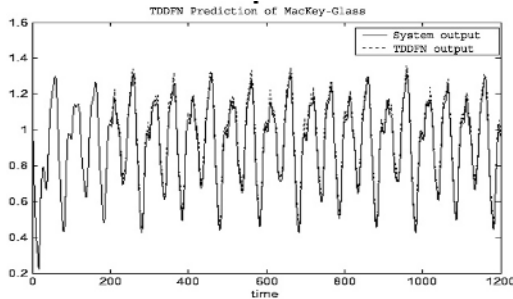


Fig. 3. TDDFN prediction result (solid-line: Mackey-Glass, dotted-line: TDDFN output)

Note from the Table 1 that both ANFIS [1] and GEFREX [4] outperform all other model in terms of NRMSE. However, ANFIS has the drawback that it has less interpretability in terms of learned information, and the implementation of the GEFREX is difficult. Excluding ANFIS, GEFREX and EPNet [3], TDDFN performs the best with an NRMSE of 0.024. In comparison to other models, the proposed TDDFN model is much less complex with easy implementation, and less number of parameters to be calculated. These mean that TDDFN can be easily used for real-time applications.

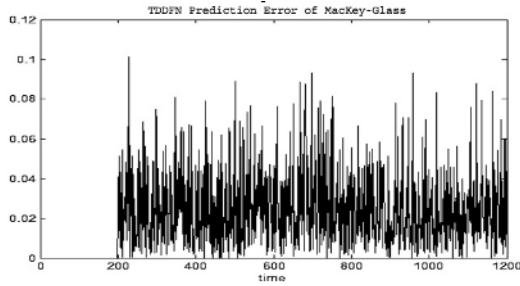


Fig. 4. TDDFN prediction error

Table 1. Comparison of TDDFN with other models for Mackey-Glass time series prediction problem (*: Results adapted from [2])

Method	NRMSE
GEFREX [4]	0.0061
ANFIS [1]	0.0074
EPNet [3]	0.02
TDDFN	0.025
GFPE*	0.026
6 th order polynomial*	0.04
Cascade Correlation-NN Model*	0.06
Auto Regressive Model*	0.19
Linear Predictive Model*	0.55

6 Conclusion

The work reported here has concentrated on laying the theoretical and analytic foundations for training of TDDFN. The TDDFN presented in this paper was successfully applied to time series prediction. They are to be used in real-time applications with process modeling and advanced control.

The significance of this work is that efficient computational algorithms have been developed for parameter identification and training of fully nonlinear dynamical systems with time delay. The gradients can be computed by adjoint sensitivity analysis methods. Another time-delay approach for neural networks should be investigated to improve the prediction capability.

References

1. Jang, J.S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Trans. Syst. Man, Cybern.*, Vol. 23 (1993) 51-63
2. Kim, D., Kim, C.: Forecasting Time Series with Genetic Fuzzy Predictor Ensemble. *IEEE Trans. Fuzzy Systems*, Vol. 5 (1997) 523-535
3. Yao, X., Lin, Y.: A New Evolutionary System for Evolving Artificial Neural Networks. *IEEE Trans. Neural Networks*, Vol. 8 (1997) 694-713
4. Russo, M.: Genetic Fuzzy Learning. *IEEE Trans. Evolutionary Computation*, Vol. 4 (2000) 259-273
5. Alex, A.: Dynamic Recurrent Neural Networks Towards Prediction and Modeling of Dynamical Systems. *Neurocomputing*, Vol. 28 (1999) 207-232
6. Ryad, Z., Daniel, R., and Nouredine, Z.: Recurrent Radial Basis Function Network for Time series Prediction, Vol. 16 (2003) 453-463
7. Oysal, Y., Becerikli, Y. and Konar, A.F.: Generalized Modeling Principles of a Nonlinear System with a Dynamic Fuzzy Network. *Computers & Chemical Engineering*, Vol. 27 (2003) 1657-1664
8. Becerikli, Y., Oysal, Y., Konar, A.F.: Trajectory Priming with Dynamic Fuzzy Networks in Nonlinear Optimal Control. *IEEE Trans on Neural Networks*, Vol.15 (2004) 383-394
9. Pearlmitter, B.: Learning State Space Trajectories in Recurrent Neural Networks. *Neural Computation*, Vol. 1 (1989) 263-269
10. Barhen, J., Toomarian, N., and Gulati, S.: Adjoint Operator Algorithms for Faster Learning in Dynamical Neural Networks. In *Advances in Neural Information Processing Systems 2*, D.S. Touretzky (Ed.) San Mateo, Ca.: Morgan Kaufmann, (1990)
11. Leistriz, L., Galicki, M., Witte, H., and Kochs, E.: Training Trajectories by Continuous Recurrent Multilayer Networks. *IEEE Trans. on Neural Networks*, Vol. 13(2) (2002)
12. Becerikli, Y., Konar, A.F. and Samad, T.: Intelligent Optimal Control with Dynamic Neural Networks. *Neural Networks*, Vol.16(2) (2003) 251-259
13. Passino, K. M., & Yurkovich, S. : *Fuzzy control*, Menlopark, Cal.: Addison-Wesley (1998)
14. Bryson, A.E. and Ho, Y.C.: *Applied Optimal Control*. Hemisphere Publishing Corporation (1975)
15. Edgar, T. F., & Himmelblau, D. M.: *Optimization of Chemical Processes*. McGraw-Hill (1988)
16. Mackey, M., and Glass, L.: Oscillation and Chaos in Physiological Control Systems. *Science*, Vol. 197 (1977) 287-289