Applied Soft Computing 47 (2016) 251–261

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Parametric system identification using neural networks

Tarek A. Tutunji

Mechatronics Engineering Department, Philadelphia University, Jordan

ARTICLE INFO

Article history:
Received 25 November 2013
Received in revised form 4 March 2016
Accepted 8 May 2016
Available online 12 May 2016

Keywords:
Neural networks
Transfer functions
System identification
System response
ARMA models

ABSTRACT

Neural networks are used in many applications such as image recognition, classification, control and system identification. However, the parameters of the identified system are embedded within the neural network architecture and are not identified explicitly. In this paper, a mathematical relationship between the network weights and the transfer function parameters is derived. Furthermore, an easy-to-follow algorithm that can estimate the transfer function models for multi-layer feedforward neural networks is proposed. These estimated models provide an insight into the system dynamics, where information such as time response, frequency response, and pole/zero locations can be calculated and analyzed. In order to validate the suitability and accuracy of the proposed algorithm, four different simulation examples are provided and analyzed for three-layer neural network models.

1. Introduction

Artificial Neural Networks (ANN) are mathematical models that are used to imitate the biological neurons in the brain. They are used as black box models to identify unknown functions by mapping input–output data. Many books have been written on the use of ANN in identification and control applications. Haykin [1], collected and established solid foundations in the theory of ANN. Liu [2] presented various ANN structures and discussed their applications in nonlinear identification and control systems (adaptive and predictive). Norgaard et al. [3] described different approaches to ANN-based identification and control of dynamic systems. Zilouchin with Jamshidi [4] collected and edited several articles that were concerned with the theory and applications of intelligent controllers. Furthermore, Demuth et al. [5] described different ANN control architectures (model predictive control, NARMA-L2 control, and model reference control) that were used in the ANN Toolbox guide for MATLAB.

Isermann and Munchhof [6] published a well-structured and comprehensive book entitled *Identification of Dynamic systems* where they described and compared many system identification methods. In their description of neural networks, they made the following statement *"Their main disadvantage is the fact that for most neural networks, the net parameters can hardly be interpreted in a physical sense, making it difficult to understand the results of the modeling process (Page 19)"*. They reiterated the statement again *"The main disadvantage is the fact that the resulting models cannot be interpreted well as the structure of the neural nets in general does not allow a physical interpretation (Page 534)"*. This paper is concerned with transforming the ANN model into transfer function model and therefore providing an insight into the physical system behavior.

Neural networks have been used extensively in identifying dynamic systems in different publications. Efe and Kaynak [7] studied and compared different ANN structures used in the identification of nonlinear systems. Liu et al. [8] used Volterra polynomial basis function neural networks for on-line identification of nonlinear systems. Gabrijel and Dobnikar [9] used recurrent neural networks for on-line identification and reconstruction. These researchers showed the capability of neural networks to identify systems.

Sahoo et al. [10] used different neural network model structures (polynomial and trigonometric expansions) to identify nonlinear autoregressive models. The functional expansions were used to capture the delayed input–output data which were then multiplied by the network weights and used as inputs to a hyper-tangent activation function. They proposed a robust $H\infty$ filter learning algorithm to update the network weights. They used simulated nonlinear time-varying plants to show that their proposed algorithm provides lower mean-square-error than forgetting factor recursive least squares algorithm, especially when noise is added. However, the converged ANN weights were not compared to the simulated plants' parameters.

Coban [11] proposed a recurrent neural network with added context layer for dynamic system identification. The proposed network architecture was constructed using a general feedforward network, but with added 'special' hidden layer that interacts exclu-

E-mail addresses: ttutunji@yahoo.com, ttutunji@philadelphia.edu.jo

http://dx.doi.org/10.1016/j.asoc.2016.05.012
1568-4946/

sively with the 'original' hidden layer. Dynamic backpropagation was used to train the network and update its weights. The work was validated using linear and nonlinear simulated plants and experimental DC motor and showed that the results of the proposed network provided better performance than the Elman recurrent network. However, the converged network weights were not analyzed nor were they compared to the original plants' parameters.

Deng [12] proposed a series-parallel hybrid structure with two neural networks where one network was used to generate the desired plants' outputs that were used for training the second network. Such hybrid structure can improve the mapping capability of the networks and tested the proposed model on experimental 3D crane system to demonstrate the validity of his work. However, the relationship between the network weights and the crane's transfer function was not studied.

Darus and Al-Khafaji [13] used neural networks for nonparametric identification of flexible plates. They conducted laboratory experiments and validated their work by comparing the model response with the measured data. They also performed correlation tests with the multi-layer perceptron neural, adaptive Elman networks, and adaptive neural fuzzy networks. However, their interest was concerned with nonparametric identification only.

Han et al. [14] investigated an automatic self-organizing neural network that adapted its architecture (number of neurons and topology) during the training process in order to improve the network performance. They provided the pseudo code of the *adaptive connecting and pruning algorithm* and *feedforward computation* used. They performed simulations of nonlinear models to compare their algorithm results with other adaptive networks and showed that the proposed algorithm provided better performance in CPU time, mean-square-error, and average-percentage error. However, they did not elaborate on the relationship between the network weights and the simulated plant's parameters.

Xie et al. [15] developed an identification method using ANN based on Bouc–Wen differential model to identify memory-type nonlinear hysteretic systems. They conducted laboratory experiments to identify the restoring force of wire cable vibration isolation system. They were able to identify the parameters of the Bouc–Wen model, but they did not relate these parameters to the plant's transfer function.

All the previously described publications succeeded in developing ANN architectures, learning algorithms, and mathematical models for system identification applications. However, none of these researchers provided a clear mathematical relationship between the network weights and the identified systems in parametric format.

Another closely-related application that researchers worked on was time-series forecasting. Khashei and Bijari [16] used neural network models for time-series forecasting while Zhang [17] used a hybrid Auto-Regressive-Moving-Average and neural network model for time-series forecasting. Again, these researchers did not provide the mathematical relationship between the network weights and the estimated functions.

From the vast amount of research published in the area of neural network identification, only a few investigated the mathematical relationship between the network weights and the parameters of the identified systems. Fung et al. [18] derived equations for the frequency response and general transfer functions of multi-layer networks in terms of the network weights. They used series expansions and Volterra kernel within the network models to establish their equations. However, their work was general, very mathematically involved, and did not provide a clear path to follow. Chon and Cohen [19] did impressive work in estimating the parameters for linear and nonlinear Auto-Regressive-Moving-Average (ARMA) models using neural network weights. They provided simulation results for several systems and compared the results between the network identification and least square ARMA identification. However, their work was restricted to polynomial activation functions and did not consider the frequency responses of their models. Lopez and Caicedo [20] used multilayer perceptron for parametric identification. They showed explicit equations for the linear activation cases, but they did not provide those equations for the nonlinear activation functions. Instead, they re-structured the error criteria and used Bayesian training to deal with the nonlinearities. In all of the described work, none provided a clear and easy-to-follow algorithm that shows how to relate the network weights to system's parameters.

Chen and Chen [21] discussed a neural-network-based system identification technique to determine the z-transfer function of a building envelope from experimental data. Neural networks were used to determine the Markov parameters of the process and Eigensystem realization algorithm was used to identify a minimal order state space presentation. However, the neurons were assumed to operate in the linear range only. Also, the work studied only the hyperbolic activation function and the simulation results were applied to the specific case of heat conduction through a wall.

Fei et al. [22] proposed a linear recurrent neural network and identified transfer function matrix models for multi-variable systems. Simulation results were provided to show that the proposed method can deliver the transfer function parameters from the neural network weights. However, the active functions of the hidden and output layers were linear and therefore the proposed method can only be applied in the identification of linear systems. They concluded that investigation is required to establish whether similar results can be found when nonlinear activation functions are used.

In previous work, Tutunji [23] presented a method to identify transfer functions for linear models using neural network weights with single layer only. This paper builds on those results and expands the work to include multi-layer neural network and nonlinear models.

The main contribution of the paper is the establishment of a clear relationship between the ANN weights and the transfer function parameters. Therefore, providing better interpretation (in a physical sense) where important information, such as frequency response and pole locations, can be explored. More importantly, a clear and easy-to-follow algorithm is provided that can transform the network results into ARMA models and therefore identify the system's transfer function.

In Section 2, theoretical background for system identification and neural network architecture is provided. Section 3 provides the mathematical derivations for the proposed method and describes the algorithm used. The simulation results are given in Section 4 and the conclusion is provided in Section 5.

2. Theoretical background

This section provides the background theory that is essential to the proposed algorithm and is divided into two parts: system identification and neural network architecture.

2.1. System identification

System identification is the process of using appropriate mathematical models and learning algorithms in order to map experimental data by minimizing an error criterion between the system's desired output and the model output.

Auto-Regressive Moving-Average (ARMA) models are linear regression models that use difference (or differential) equations to relate the model output to present inputs, past inputs and past