



Identification of Dynamical System's Parameters Using Neural Networks

Nataliia Shybytska

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 7, 2021

Identification of Dynamical System's Parameters Using Neural Networks

Nataliia Shybytska

*Kyiv National University of Technologies and Design (KNUVD),
Nemyrovycha-Danchenko Street, 2, Kyiv. 01011, Ukraine*

Abstract

The article shows that neural networks can be effectively used to identify the parameters of dynamic systems. The main attention in the paper is paid to modelling and practical results obtained in the MatLab Neural Network Toolbox environment. The use of the feedforward network and the Elman recurrent network is discussed. The simulation results show that the identification of dynamic systems using neural networks is most effective when the experimental data on the system contains internal redundancy or are incomplete.

Keywords

Neural Networks, Dynamical System's Parameters, Feedforward Network, Elman's Recurrent Network, System of Linear Equations (SLE).

1. Introduction

At present, much attention is paid to the issues of experimental determination (identification) of dynamic systems models [1,2]. At the stage of constructing a mathematical model of any process, it is necessary to select the general structure of the model and the class of equations that are supposed to describe the observed process, i.e., solve the so-called structural identification problem. The choice of the model structure depends on the measurement process during the experiment and the choice of software for processing the results.

After the structure of the model and the class of equations are determined, it is necessary to determine the numerical values of the constants included in the equations of the model. At this stage, the problem of parametric identification arises, which consists in obtaining, from the available experimental data, the numerical values of the parameters of dynamic objects. Due to the fact that not all values can be measured or measurements contain an error, consider the parametric identification problem based on a neural network [3].

2. Methods

Neural network models have internal regularizing properties that allow one to obtain small errors of generalization [4-6]. The regularizing properties of neural networks are especially useful to take into account in situations where the experimental data about the system contain internal redundancy. Redundancy allows to represent a collection of data with a model that contains fewer parameters than there is data. Thus, the neural network model compresses experimental information, eliminating noise components and emphasizing continuous dependences.

The conformity of the custom model to the control object determines the quality of the identification. In this case, the minimization of the quality criterion is achieved by solving a system of linear algebraic equations.

ITTAP'2021: 1nd International Workshop on Information Technologies: Theoretical and Applied Problems, November 16–18, 2021, Ternopil, Ukraine

EMAIL: shybytska.nm@knutd.edu.ua

ORCID: 0000-0001-5607-0081



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

Thus, the regularizing methods are reduced to optimizing the error functional:

$$E[G] = \sum_{\alpha=1}^{N_{\alpha}} \left(G(\bar{x}^{\alpha}) - \bar{y}^{(\alpha)} \right)^2 + \lambda \varphi[G],$$

where φ is a regularizing functional, λ - is a non-negative regularization constant.

To solve the identification problem, it is proposed to use a multilayer perceptron (Fig.1). Control based on a multilayer neural network allows solving poorly formalized control problems for complex dynamic objects in cases where a priori models and algorithms are unknown [7].

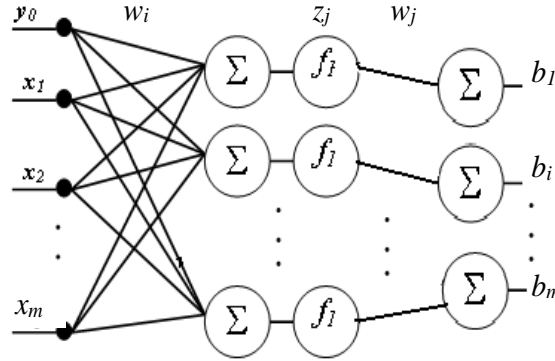


Figure 1: Multilayer neural network

An m -dimensional feature vector $\{x_i, i=1, 2, \dots, m\}$ is fed to the network input.

A fully connected structure is used between the input and hidden layers, as well as between the hidden and output layers, where

m is the dimension of the original data, $m = n + l$,

n is the number of unknown coefficients;

p is the number of neurons on the hidden layer;

x_i is the vector of inputs, $i = 1..m$;

w_i is the weight coefficients between the input and hidden layers, $i = 0, 1, \dots, m$,

w_j is the weight coefficients between hidden and output layers, $k = 0, 1, \dots, l$.

z_j is the value of the output of the j -th neuron of the hidden layer, $j = 1..l$,

$$z_j = f_1 \left(\sum_{i=0}^p w_{ji} x_i \right);$$

b is the value of the output neuron of the network (network output);

$f_1(x)$ is the activation function of neurons of the hidden layer;

$f_2(x)$ is the activation function of the neuron of the output layer.

The sigmoidal activation function $f(x)$:

$$f(x) = \frac{1}{1 + \exp(-x)},$$

with the derivative:

$$f'(x) = f(x)(1 - f(x)),$$

is used as the activation function of the neurons of the hidden layer.

Let's consider the case when the dynamical system is described by the second-order equations with unknown coefficients:

$$b_2 \frac{d^2 y}{dt^2} + b_1 \frac{dy}{dt} + b_0 y = u(t).$$

training function (Fig. 3) was used, which returns weights and biases using the Levenberg–Marquardt algorithm (*LMA*), which based on the method of *Mean Squared Error (MSE)*.

The function learning quality - *mse* from *MatLab* is determined by the root mean square error. In the network adaptation mode, the *learningdm* function of correction of weights and biases was used as a tuning function. Sufficient computational accuracy was obtained using the gradient optimization algorithm with an iterative component and a *purelin* linear activation function.

To estimate the required number of synaptic weights, the formula is applicable:

$$\frac{mN}{1+\log_2 N} \leq L_w \leq m \left(\frac{N}{m} + 1 \right) (n + m + 1) + m,$$

where *n* is the dimension of the input signal, *m* - is the dimension of the output signal, *N* - is the number of training sample elements.

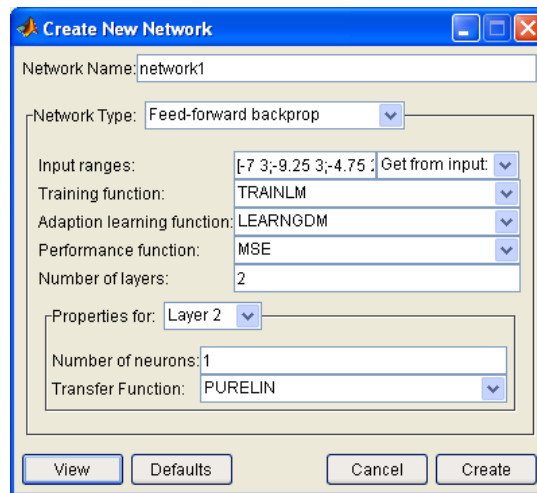


Figure 3: Neural network parameters

After evaluating the required number of weights, it is possible to calculate the number of neurons in the hidden layers. For example, the number of neurons in a two-layer network is:

$$L = \frac{L_w}{n + m}$$

In the process of learning the neural network, based on the proposed methodology, fairly fast convergence of the method and a low level of computational error were obtained (Fig. 4).

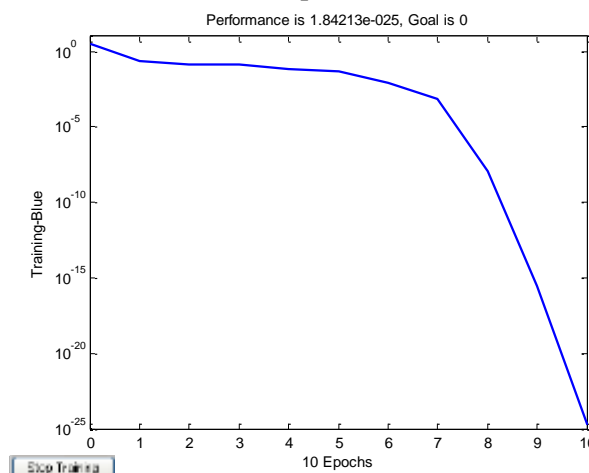


Figure 4: Computational error

The obtained result - the coefficients of the differential equation (roots of the *SLE*) - is displayed in the *Data: network1_outputs* dialog box (Fig. 5).

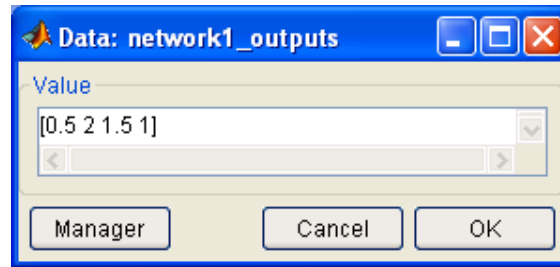


Figure 5: The roots of the SLE

The value of the computational error - in the *Data: network1_errors* window (Fig. 6)

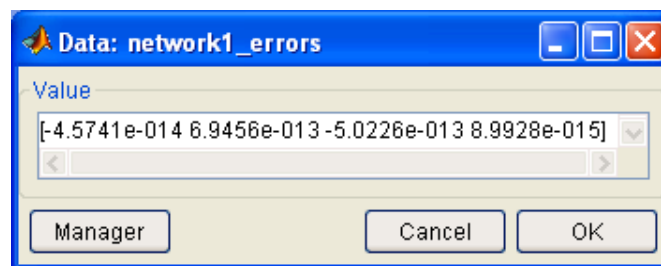


Figure 6: The value of the computational error

To solve this problem, a comparative study of the results was carried out using the models of the *Feedforward network* and the *Elman's recurrent network*, characterized by partial recurrence in the form of feedback between the hidden and the input layer [11]. Elman's network is implemented using *tansig* function. It is advisable to use Elman's networks in control systems of moving objects, for example in the construction of technical vision systems [12].

4. Conclusion

In conclusion, this study confirms a technique for solving *SLEs* using neural networks in the *MatLab Neural Network Toolbox*. A series of model experiments and calculations allow us to make a conclusion that the minimum calculation error and high computational accuracy are achieved for neural networks even with a small number of layers when using a network model with *error back-propagation*. Using *Elman's networks* in control systems of moving objects, for example in the construction of technical vision systems is advisable.

The proposed method for identifying parameters and synthesizing dynamic systems based on a neural network is most effective when the experimental data on the system contains internal redundancy or the problem is formulated incorrectly.

5. References

- [1] Tsyppkin, Ya.Z., Information Theory of Identification. – M.: Nauka, 1995.– 336 p.
- [2] V.Babak, N.Shybytska, A.Isak, Identification of dynamic objects parameters in the technical diagnostics systems. // The Second World Congress "Aviation in the XXI– st century". Kiev, NAU – 2005. p.2.12 – 2.16.
- [3] Narendra, Kumpati and K.Parthasarathy (1990). "Identification and Control of Dynamical System Using Neural Networks". In: vol. NN– 1, 1737 –1738 vol.2.

- [4] Callan R. Basic concepts of neural networks. – M.: Williams Publishing House, 2001. 287 p.
- [5] Osovsky S. Neural networks for information processing / Per. from Polish I.D. Rudinsky. –M.: Finance and Statistics, 2002. – 344 p.
- [6] Khaikin S. Neural networks: a full course, 2nd edition .– M. Williams Publishing House, 2006. — 1104 p.
- [7] VA Terekhov, DV Efimov, I Yu Tyukin Neural network control systems. – M.: IPRZhR, 2002. – 480 c.
- [8] Dinh, Huyen, Shubhendu Bhasin, and W.E. Dixon (2011). “Dynamic Neural Networkbased Robust Identification and Control of a class of Nonlinear Systems”. pp. 5536 –5541
- [9] Dinh, H. T., Bhasin, S., Kim, D., and Dixon, W. E., 2012, “Dynamic Neural Network-Based Global Output Feedback Tracking Control for Uncertain Second-Order Nonlinear Systems,” American Control Conference, Montréal, Canada, June 27–29, pp. 6418–6423.
- [10] Cen, Z., J. Wei, and R. Jiang (2011). “A Grey– Box Neural Network based identification model for nonlinear dynamic systems”. In: The Fourth International Workshop on Advanced Computational Intelligence, pp. 300–307.
- [11] Pham, D and Dervis Karaboga (1999). “Training Elman and Jordan networks for system identification using genetic algorithms”. In: Artificial Intelligence in Engineering 13, pp. 107–117.
- [12] V.Kvasnikov, N.Shybytska, A.Borkovskiy, Adjustment of technical sight system with the use of reference points reflection from mirror surface. // The Second World Congress "AVIATION IN THE XXI-st CENTURY". Kiev, NAU – 2005. p.2.43 – 2.46.