Indirect Training of Grey-Box Models: Application to a Bioprocess

Francisco Cruz, Gonzalo Acuña, Francisco Cubillos,
Vicente Moreno, Danilo Bassi

Facultad de Ingeniería, Universidad de Santiago de Chile, USACH
Av. Libertador Bernandor O'Higgins 3363, Santiago, Chile
francisco928@gmail.com, gacuna@usach.cl

Abstract. Grey-box neural models mix differential equations, which act as white boxes, and neural networks, used as black boxes. The purpose of the present work is to show the training of a grey-box model by means of indirect backpropagation and Levenberg-Marquardt in Matlab®, extending the black box neural model in order to fit the discretized equations of the phenomenological model. The obtained grey-box model is tested as an estimator of a state variable of a biotechnological batch fermentation process on solid substrate, with good results.

1 Introduction

The determination of relevant variables or parameters to improve a complex process is a demanding and difficult task. This gives rise to the need to estimate the variables that cannot be measured directly, and this in turn requires a software sensor to estimate those variables that cannot be measured on line [1].

An additional problem is the one consisting of a model that has parameters that vary in time, because a strategy must be applied to identify such parameters on line and in real time [2]. A methodology that is used in these cases, especially in the field of chemical and biotechnological processes, is that of the so-called grey-box models [3]. These are models that include a limited phenomenological model which is complemented with parameters obtained by means of neural networks.

The learning or training strategies used so far for grey-box neural models assume the existence of data for the parameters obtained by the neural model [4], but most of the time this is not possible. This paper proposes a training process that does not use learning data for the neural network part, instead backpropagating through the phenomenological model the error at its output, as will be detailed below. The creation of the proposed model, the training and the simulations were all carried out using the Matlab development tool.

2 Grey-Box Models

Grey-box neural models are used for systems in which there is some *a priori* knowledge, i.e., some physical laws are known, but some parameters must be determined from the observed data.

Acuña et al. [5] distinguish between two methods of training. The first one corresponds to direct training (Fig. 1(a)), which uses the error originated at the output of the neural network for the correct determination of their weights. The second method is indirect training (Fig. 1(b)), which uses the error originated at the model's output for the purpose of learning by the neural network. Indirect training can be carried out in two ways, one by minimizing an objective function by means of a nonlinear optimization technique, and the other is by backpropagating the output error over the weights of the neural network taking into account the discretized equations of the phenomenological model.

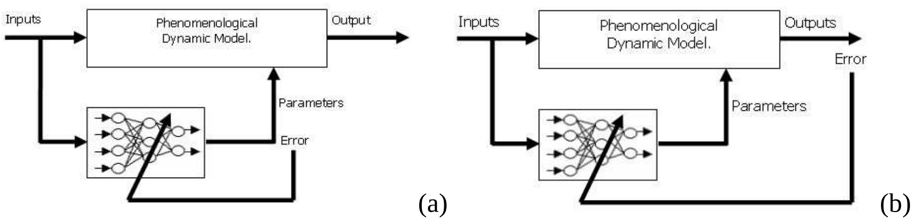


Fig. 1. (a) Grey-box model with direct training. (b) Grey-box model with indirect training.

In this paper the second indirect training method is used, calculating the error at the output of the phenomenological model, and backpropagating it from there to the model's neural part or black-box part.

The backpropagation process considers a network with m inputs and p outputs, n neurons in an intermediate layer, and d data for training. The computed gradients, depending on the the activation and the transfer function used, are shown in Table 1 and Table 2 respectively, where w^k_{ij} is the weight of the connection from neuron i to neuron j in layer k, A^k_i is the activation value of neuron i of layer k, and Z^k_i is the transfer value of neuron i from layer k (output of neuron i).

Table 1. Gradients depending on the activation functions used in the neurons.

f	Sum	Product
$\frac{\partial A_c^{k+1}}{\partial Z_j^k}$	w_{jc}^{k+1}	$(\prod_{q=1} w_{jc}^{k+1}) \cdot (\prod_{\substack{q=1\\q\neq j}} Z_q^k)$
$\frac{\partial A_j^k}{\partial w_{ij}^k}$	Z_j^k	$(\prod_{\substack{q=1\\q\neq j}} w_{qj}^k) \cdot (\prod_{q=1} Z_q^{k-1})$

Table 2. Gradients depending on the transfer functions used in the neurons.

g	sigmoid	tanh	inverse	identity
$\dfrac{\partial Z_j^k}{\partial A_j^k}$	$Z_j^k \cdot (1 - Z_j^k)$	$1 + \left(Z_j^k\right)^2$	$-Z_j^k \cdot Z_j^k$	1

3 Biotechnological Process

In this paper a grey-box neural model is proposed for the simulation of a batch fermentation bioprocess on a solid substrate, corresponding to the production of gibberellic acid from the philamentous fungus *Gibberella fujikuroi*.

A simplified model describes the evolution of the main variables [6]. This phenomenological model based on mass conservation laws considers 8 state variables: active biomass (X), measured biomass (X_{measu}), urea (U), intermediate nitrogen (N_I), starch (S), gibberellic acid (GA_3), carbon dioxide (CO_2) and oxygen (O_2). Only the last two variables can be measured directly on line. The model's equations discretized by Euler's method and considering discrete time t and t+1 are the following:

$$X_{measu(t+1)} = X_{measu(t)} + \left(\mu \cdot X_{(t)}\right) \cdot \Delta t \qquad (1)$$

$$X_{(t+1)} = X_{(t)} + \left(\mu \cdot X_{(t)} - k_d \cdot X_{(t)}\right) \cdot \Delta t \qquad (2)$$

$$U_{(t+1)} = U_{(t)} + \left(-k\right) \cdot \Delta t \qquad (3)$$

$$N_{I(t+1)} = \begin{cases} N_{I(t)} + \left(0,47 \cdot k - \mu \cdot \left(\dfrac{X_{(t)}}{Y_{X/N_I}}\right)\right) \cdot \Delta t, \; si\; U \geq 0 \\ N_{I(t)} + \left(-\mu \cdot \left(\dfrac{X_{(t)}}{Y_{X/N_I}}\right)\right) \cdot \Delta t, U(t) = 0, \; si\; U < 0 \end{cases} \qquad (4)$$

$$S_{(t+1)} = S_{(t)} + \left(-\frac{\mu \cdot X_{(t)}}{Y_{X/S}} - m_s \cdot X_{(t)}\right) \cdot \Delta t \qquad (5)$$

$$GA_{3(t+1)} = GA_{3(t)} + \left(\beta \cdot X_{(t)} - k_p \cdot GA_{3(t)}\right) \cdot \Delta t \qquad (6)$$

$$CO_{2(t+1)} = CO_{2(t)} + \left(\mu \cdot \left(\frac{X_{(t)}}{Y_{X/CO_2}}\right) + m_{CO_2} \cdot X_{(t)}\right) \cdot \Delta t \qquad (7)$$

$$O_{2(t+1)} = O_{2(t)} + \left(\mu \cdot \left(\frac{X_{(t)}}{Y_{X/O_2}} \right) + m_{O_2} \cdot X_{(t)} \right) \cdot \Delta t \quad (8)$$

The measured outputs are the following:

$$y_1 = CO_{2(t+1)} \quad (9)$$

$$y_2 = O_{2(t+1)} \quad (10)$$

On the other hand, the parameters that are difficult to obtain and that will be estimated by the model's neural part are μ and β, corresponding to the specific growth rate and specific production rate of gibberellic acid, respectively. The remaining parameters were identified on the basis of specific practices and experimental conditions. Their values under controlled water temperature and activity conditions (T=25 ºC, Aw=0.992) can be found in [6].

4 Proposed Solution

The proposed solution is a grey-box neural model whose phenomenological part can be described jointly with its black-box part, by means of an extended neural network containing both, the discretized equations of the phenomenological model and the time-varying parameters modeled by the black-box part (Fig. 2). This hybrid neural network has the capacity to fix weights in the training phase, so that it can act as a grey-box model. The weights in Fig. 2 that have a fixed value correspond to the model's phenomenological part. The weights for which no value is given correspond to the model's neural part. These weights were initially assigned pseudo-random values obtained by the initialization method of Nguyen & Widrow [7].

In Fig. 2 it is seen that one of the weights corresponding to the white-box or phenomenological part is graphed as a dotted line. This line represents the switching phenomena that is seen in the fourth state variable (N_I) in the mathematical model, i.e., if the urea (U) is greater than or equal to zero, this weight has the indicated value, otherwise, if urea (U) is less than zero, this weight has a value of zero.

Therefore, the multilayer perceptron, inserted in the model, estimates the values of the two parameters that are difficult to obtain, and in turn they are mixed with the phenomenological part of the model, in that way obtaining its output.

For the black-box neural part the hyperbolic tangent was used as transfer functions in the intermediate layer and the identity function in the output layer, while for the phenomenological part the identity function was used as transfer function.

The activation function most currently used was the sum of the inputs, except for the two neurons immediatelly after the output of the black-box neural part, for which a product was used as activation function in order to follow the discretized phenomenological equations.

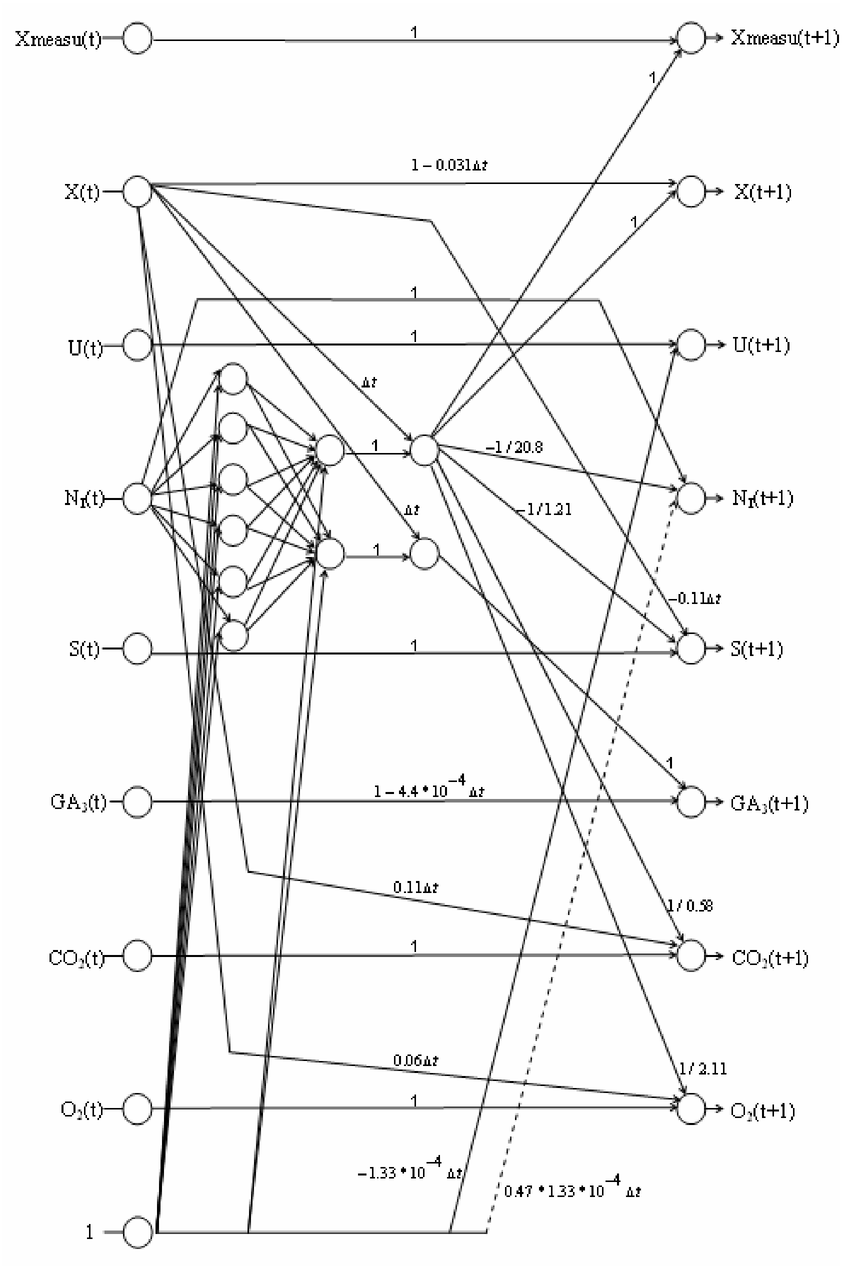


Fig. 2. Grey-box model for the solid substrate fermentation process. Fixed weights represent the discretized phenomenological model. The black-box part that models the unknown time-varying parameters μ and β has variable weights. The dotted line represents a switch on the model of the state variable (N_I)

The training algorithm used corresponds to backpropagation with a Levenberg-Marquardt optimization method. As it was already stated, the algorithm has the capacity to modify only the weights that are indicated, therefore leaving a group of fixed weights which, represent the model's phenomenological part in the training phase.

For the validation of the proposed grey-box neural model, quality indexs such as IA (Index of Agreement), RMS (Root Mean Square) and RSD (Relative Standard Deviation) are calculated, and the values considered acceptable for these indexs are IA>0.9, RMS<0.1 and RSD<0.1. The quality indexs equations are the following:

$$IA = 1 - \frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n}\left(|o_i'| + |p_i'|\right)^2} \quad RMS = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n} o_i^2}} \quad RSD = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{N}} \tag{11}$$

where o_i and p_i are the observed and predicted values, respectively, at time i, and N is the total number of data. Then, $p_i' = p_i - o_m$ and $o_i' = o_i - o_m$, where o_m is the mean value of the observations.

5 Simulation and results

For the simulation, tests were carried out for data with 5% error and an erroneous initial biomass value. The simulation was made using one thousand examples. The initial conditions under normal operation were the following:

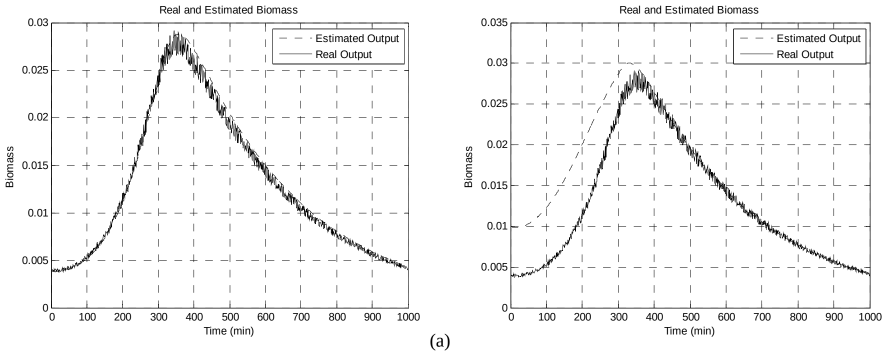
$$X(0) = \begin{bmatrix} 0 & 0.0040 & 0.0040 & 0.5 * 10^{-4} & 0.0040 & 0 & 0 & 0 \end{bmatrix}$$

Case 1: Simulation with 5% noise

The first case evaluated corresponds to the simulation with 5% noise in the input data. Fig. 3(a) shows the real and estimated biomass. In this case the quality indexs obtained were the following: IA=0.99, RMS=0.5E-1 and RSD=0.8E-3.

Case 2: Simulation with noise in the data and erroneous initial condition in the biomass

The second case evaluated corresponds to the simulation with 5% noise in the data in addition to an erroneous initial biomass condition with an error of 250%, to verify the convergence of the model acting as a software sensor of biomass. Fig. 3(b) shows the model's response, where the real and the estimated biomass are seen. The quality indexs were IA=0.93, RMS=0.24 and RSD=0.41E-2. In this case the RMS index does not fulfill the acceptable condition (RMS<0.1), but this is due to the nature of the introduced error, because the model takes about 300 iterations to fit the estimated curve to the real value. This simulation, however, shows the convergence of the software sensor based on the grey-box model, so it is considered acceptable.



(a) (b)

Fig. 3. (a) Real and estimated biomass for case 1 (b) Real and estimated biomass for case 2

6 Conclusions

Grey-box neural models are a real alternative for modeling real world processes. They have advantages over black-box models because they are supported by the *a priori* knowledge available on the process.

The model proposed in this paper combines the phenomenological equations of the process with a multilayer perceptron neural network that estimates the unknown time-varying parameters within a extended neural network for carrying out the backpropagation process. The use of fixed and variable weights and new activation functions were needed in order to fit the discretized phenomenological model thus slightly changing the standard backpropagation method usually found in available neural network softwares.

Good results when using the trained model as a software sensor for estimating biomass concentration in a biotechnological process were shown.

Acknowledgements

The authors gratefully acknowledge the partial financial support of this work by Fondecyt under Project 1040208.

References

1. James, S., Legge, R., Budman, H.: Comparative study of black-box and hybrid estimation methods in fed-batch fermentation. Journal of Process Control 12 (2000) 113-121.
2. Hensen, R., Angelis, G., van de Molengraft, M., de Jager, A., Kok, J.: Grey-box modeling of friction: An experimental case-study. European Journal of Control 6 (2000) 258-267.
3. Chen, L., Bernard, O., Bastin, G., Angelov, P.: Hybrid modelling of biotechnological processes using neural networks. Control Engineering Practice 8 (2000) 821-827.
4. Aguiar, H., Filho, R.: Neural network and hybrid model: a discussion about different modeling techniques to predict pulping degree with industrial data. Chemical Engineering Science 56 (2001) 565-570.
5. Acuña, G., Cubillos, F., Thibault, J., Latrille, E.: Comparison of methods for training grey-box neural network models, Computers and Chemicals Engineering Supplement (1999) 561-564.
6. Gelmi, C., Pérez-Correa, R., Agosin, E.: Modelling Gibberella fujikuroi growth and GA3 production in solid-state fermentation. Process Biochemistry 37 (2002) 1033-1040.
7. Nguyen, D., Widrow, B.: Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights. Proceedings IEEE IJCNN, San Diego (1990) 21-26.