

Neural networks modeling of hollow fiber membrane processes

A. Shahsavand*, M. Pourafshari Chenar

Chemical Engineering Department, Ferdowsi University of Mashhad, P.O. Box 91775-1111, Mashhad, Islamic Republic of Iran

Received 24 January 2007; received in revised form 28 February 2007; accepted 3 March 2007

Available online 15 March 2007

Abstract

Generalization performances of two different neural networks are compared with a conventional software for prediction of hollow fiber permeances and the corresponding separation factors. Two experimental data sets were used as the training data for separation of carbon dioxide from methane. Both Radial Basis Function (RBF) and Multi-layer Perceptrone (MLP) networks provide superior performances compared to the conventional software (Table Curve or TC). It is also shown that the RBF networks provide better predictions than MLP networks because of their powerful noise filtering capabilities. For RBF networks, both appropriate choices of isotropic spread and the corresponding optimal level of regularization are crucial for proper reconstruction of the true underlying hyper-surface from a set of noisy data set. The in-house algorithms are used for training both MLP and RBF networks. It is clearly illustrated that the computation of optimal isotropic spread is crucial for proper performance of the RBF network. The use of Leave One Out Cross-validation (LOOCV) criterion was also essential for appropriate estimation of optimum level of regularization.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Neural networks; MLP; RBF; Regularization network; Membrane; Hollow fiber

1. Introduction

Modeling of a process covers a broad spectrum. At one extreme lies theoretical (or *parametric*) models based on fundamental knowledge of the process. At the other end lies empirical (or *non-parametric*) models which do not rely on the fundamental principles governing the process. Feed-forward neural networks are well-known examples of the latter group, which may successfully recover the true underlying multidimensional hyper-surface hidden in bunch of noisy measured data for various chemical engineering processes.

Although, various networks such as Multi-layer Perceptrone (MLP), Radial Basis Function (RBF), ADALINE¹ or ANFIS² networks have been extensively employed for input–output mapping purposes, however, the first two networks (MLP and RBF) are usually more popular [1]. The MLP networks are well suited for multiple inputs multiple outputs (MIMO) systems and RBF or Regularization networks (RN) have a solid theoretical back-

ground and are equipped with influential noise filtering facilities [2,3].

Application of neural networks in chemical engineering was started with the pioneering work of Himmelblau and Hoskins [4] which employed a Zone Node Layer (ZNL) artificial neural network approach for the fault diagnosis of a process consisting of three CSTR's in series. In 1989 Venkatasubramanian and Chan [5] used a binary-input network to diagnose faults in a fluidized catalytic cracking (FCC) process and Watanabe et al. [6] presented a network architecture to estimate the degree of failure for a system with three measurements and five faults. Neural networks have been extensively used in the previous decade to solve various chemical engineering problems. A concise review of such applications is presented elsewhere [7].

Modeling of membrane processes via neural networks has been received immense attention in the past few years. These separations techniques are used in many purification applications especially in gas industries (e.g. air separation and hydrogen recovery). Membrane processes are attractive because of (a) lower energy consumption (better separation economy), (b) possible continuous operation, (c) variety of membranes for different applications and (d) simple scale up possibility of the process [8]. Membranes can also be used in combination with traditional separation techniques to take advantage of both tech-

* Corresponding author. Tel.: +98 511 7626234; fax: +98 511 8816840.

E-mail address: shahsavand@um.ac.ir (A. Shahsavand).

¹ Adaptive Linear Neuron (or Adaptive Linear Element).

² Adaptive Network Based Fuzzy Inference System.

nologies where in some occasions neither process could achieve a proper result individually. For example, in natural gas sweetening, the bulk of CO₂ and H₂S can be removed by proper membranes while the final purification to pipeline quality can be achieved by an amine absorption process [9].

Niemi et al. [10] simulated a reverse osmosis membrane process by using the Levenberg-Marquardt optimization routine of MATLAB toolbox to train a relatively simple MLP network. The network architecture (number of hidden layers, number of neurons, etc.) was not reported in the article. They concluded that “use of neural network reduces the computing time of membrane simulation”. Lee et al. [11] employed a simple gradient descent learning algorithm for training a two layered feed-forward MLP network with tangent hyperbolic activation functions. The network was successfully used as an empirical tool to model a polymer electrolyte membrane fuel cell (PEFC).

Wang et al. [12] used a simple un-regularized RBF network to model a membrane process for hydrogen recovery from refining gases. They reported that “there are almost no discrepancies between the (network) predictions and the (practical) verifications”. Curcio et al. [13] successfully used a three input-single output feed-forward MLP network to control flux decay in an experimental ultra-filtration (UF) process performed under pulsating condition. Like most of previous works, MATLAB routines were employed to construct and train the neural network. Sahoo and Ray [14] used genetic algorithm to optimize the architecture and parameters of both back-propagation (BP) and Radial Basis Function (RBF) neural networks for a borrowed set of experimental data on flux decline in cross-flow membranes. MATLAB toolbox subroutines (“newff” and “newrb”) were used to model the data. Although, the main objective of the RBF network is its regularization potential for filtering the noise, however, this issue was not received proper attention in this article. In a similar work, Chen and Kim [15] used an un-regularized RBF network to predict long-term permeate flux decline in a cross-flow membrane for filtration of colloidal suspension. Once again, MATLAB routines were employed to construct and train the network. The root-mean-square-error (RMSE) merit function was also minimized for selection of the optimal linear weights.

As it was mentioned above, most of the previous works used MATLAB routines to construct and train various neural networks. In-house software was prepared in this work for training both MLP and RBF networks based on learning algorithms presented in the next section. Furthermore, the noise filtering capability of the RBF networks was utilized by resorting to the Leave One Out Cross-validation (LOOCV) criterion for automatic selection of optimal regularization level.

2. Theoretical aspects

As illustrated earlier by Bakshi and Utojo [16], all feed-forward neural networks (such as MLP and RBF) can be represented by the following equation:

$$\hat{y}(x, \alpha, \beta) = \sum_{j=1}^M \alpha_j \phi_j(x, \beta_j) \quad (1)$$

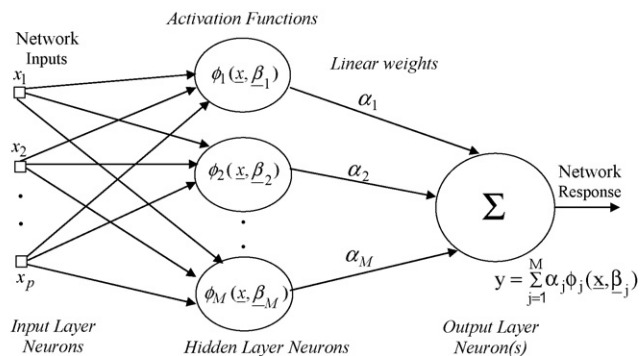


Fig. 1. Schematic representation of a feed-forward neural network.

where $\phi_j(\cdot)$ can be chosen as any arbitrary non-linear function. The model is always linear with respect to α_j s but may be non-linear with respect to the β_j s. Fig. 1 represents a feed-forward neural network with a single hidden layer for a Multiple Input Single Output (MISO) system.

The individual basis functions, $\phi_j(\cdot)$ s may be chosen to have either a *local* (e.g. RBF) or a *global* (e.g. MLP) response. The response of a *local* basis function (e.g. a Gaussian) is restricted to a limited range about some central point, while, the response of a *global* basis function (e.g. a line) is distributed over the entire input-space and affects the overall response at every point. The weighted sum of all basis functions must reconstruct the true underlying response.

For the case of multiple independent variables, the non-linear parameters β_j s are used to transform the input vector into a scalar argument for the basis function $\phi_j(\cdot)$. This is usually achieved in one of two ways:

- Projection based input transformation.* The parameters β_j s project the inputs on a hyper-surface. This projection may be either linear (e.g. MLP networks) or non-linear as illustrated in Fig. 2a.
- Kernel based input transformation.* The parameters β_j s are used to define a scalar norm measure (usually Euclidean) of the input vector with respect to a fixed point (center) as depicted in Fig. 2b. Radial Basis Function (RBF) networks are well-known examples of such transformation.

Once the input transformation is specified, it remains to choose the functional form of each basis function with respect to its scalar argument. Table 1 presents a number of basis functions commonly employed in feed-forward neural network applications.

With a specified input transformation of $z_j(x, \beta)$ and a specified functional form for $\phi_j(\cdot)$, the objective is to find a set of optimal (*best-fit*) linear α^* and non-linear β^* parameters which minimize a suitably defined merit function for a given set of observations.

Both projection and kernel based neural networks have been shown to provide reliable approximation properties for the reproduction of multivariate non-linear functions. Wang et al. [17] showed that a projection based feed-forward neural network

Download English Version:

<https://daneshyari.com/en/article/638592>

Download Persian Version:

<https://daneshyari.com/article/638592>

[Daneshyari.com](https://daneshyari.com)