Estimation of the Heat Transfer Coefficient in a Liquid–Solid Fluidized Bed Using an Artificial Neural Network

J. Manickaraj and N. Balasubramanian

Department of Chemical Engineering, A. C. College of Technology, Anna University–Chennai, Chennai 600 025, India

Received 19 March 2007; accepted 5 June 2007

Abstract
A non-iterative procedure has been developed using an artificial neural network (ANN) for estimating the fluid–particle heat transfer coefficient, \(h_{fp}\), in a liquid–solid fluidized system. It is assumed that in a liquid–solid system, the liquid temperature is time dependent, and the input parameters and output parameters for the ANN are considered on a linear scale. The output configuration yields an optimal ANN model with 10 neurons in each of the three hidden layers. This configuration is capable of predicting the value of \(Bi\) in the range of 0.1–10 with an error of less than 3%. The heat transfer coefficient estimated using the ANN has been compared with the data reported in the literature and found to match satisfactorily.

© Koninklijke Brill NV, Leiden and Society of Powder Technology, Japan, 2008

Keywords
Artificial neural network, modeling, heat transfer coefficient, liquid–solid fluidized bed

Nomenclature

\[ A_i, B, C_i, D \] coefficient of the tridiagonal matrix

\[ Bi \] Biot number \((h_{fp}r/k)\)

\[ Fo \] dimensionless time, Fourier number \((\alpha t/r^2)\)

\[ h_{fp} \] heat transfer coefficient \((\text{w/m}^2\text{°C})\)

\[ k \] thermal conductivity \((\text{w/m}°\text{C})\)

\[ m \] slope

\[ N \] surface condition

* To whom correspondence should be addressed. E-mail: nbsbala@annauniv.edu

© Koninklijke Brill NV, Leiden and Society of Powder Technology, Japan, 2008
1. Introduction

Fluidized beds find wide applications in chemical process industries as they provide a large interfacial area, high degree of mixing and temperature uniformity. In particular, liquid–solid fluidized beds are increasingly used in chemical processes such as fermentation, biological wastewater treatment, flue gas desulfurization, ore reduction, etc. Although fluidization can be achieved either by liquid or gas as the fluidizing medium, gas fluidized beds have gained more importance in the scientific community due to their more numerous applications. The emergence of biosciences in the recent past and the adaptability of liquid fluidized beds for various applications has meant the solid–liquid fluidized beds gaining greater attention among scientists and researchers [1].

The presence of the particles in liquid–solid fluidized beds enhances the heat transfer, because the movement of the particles leads to an increased turbulence in the system. The turbulence affects the thermal boundary layer and reduces the heat transfer resistance, resulting in an improved heat transfer rate. The heat transfer plays an important role in solid–liquid fluidized beds. In liquid–solid fluidized beds, the heat transfer rate is mainly investigated using water as liquid and glass particles as solid. The effects of the process parameters on the heat transfer coefficient have been discussed. Quantitative information on the heat transfer coefficient between surface and bed is essential for the rational design and online process monitoring of liquid–solid fluidized bed systems [2–4].

The heat transfer coefficient, $h_{fp}$, plays an important role in the design and optimization of the processes involved in liquid–solid reactive systems. Experimental determination of the fluid–particle heat transfer coefficient requires measurements of the temperature inside the particle at a specified location and solving the inverse heat conduction problem (IHCP) resulting in the temperature history at a specified location for an assumed $h_{fp}$. Several algorithms based on finite difference and finite element methods have been developed for solving the IHCP [5]. In recent years, artificial neural networks (ANNs) have gained wide popularity in many disciplines of engineering sciences due to their ability to learn and generalize relationships in complex datasets. The ANN is a versatile tool, and has been used for various
applications in chemical processes such as prediction and estimation of process parameters, modeling of chemical process, fault detection and diagnosis, process control and optimization, etc. [6–8].

An ANN is a collection of interconnecting computational elements called neurons. The neurons are connected together by lines of communication called connections. Inputs coming to the neuron are associated with a weight indicating its strength. In the neuron, the values of the input, $X_i$, are multiplied by the corresponding weights, $w_i$, all products summed together, and a bias, $b$, is added to the sum (Fig. 1). This sum is transformed through a transfer function, $F$, to produce a single output, $Y$, which may be passed on to other neurons. Typical transfer functions employed in building ANN applications include a linear threshold transfer function, step function, sigmoid function, hyperbolic tangent and others. A typical function of a neuron can be expressed mathematically as:

$$Y = F\left(\sum_{i=1}^{N} w_i X_i + b\right),$$

where $Y$ is the output and $X$ is the input variable. The interconnection of neurons in an ANN has a great impact on the operation and performance of the network. The most frequently used typology is the back-propagation algorithm. Standard back-propagation is a gradient descent algorithm in which the network weights are moved along the negative of the gradient of the performance function. The term back-propagation refers the manner in which the gradient is computed for nonlinear multilayer networks. The trained back-propagation network gives reasonable output for the given inputs that are used [12]. The objective of the present investigation is to develop a noniterative procedure for estimating the heat transfer coefficient in a liquid–solid system. The required data have been generated using a finite difference method.

![Figure 1. Model of a single neuron.](image-url)
2. ANN Model

In general, there are three steps involved in developing an ANN model: data generation required for training, training and testing of the ANN model, and evaluation of the ANN configuration. In the present case, the heat conduction equation for a spherical geometry can be written as

\[
\frac{\partial T}{\partial t} = \alpha_p \left( \frac{\partial^2 T}{\partial r^2} + \frac{2}{r} \frac{\partial T}{\partial r} \right). \tag{2}
\]

The initial and boundary conditions are:

for \( t = 0 \); \( T = T_i \) for all \( r \)

\[
\frac{\partial T}{\partial r} = 0 \quad \text{at} \quad r = 0, \ t \geq 0
\]

and

\[
K_p \frac{\partial T}{\partial r} = h_{fp}(T_f - T_N) \quad \text{(convective boundary condition).} \tag{3}
\]

The schematic of the spherical system with boundary conditions is shown in Fig. 2. The thermo-physical properties of the fluid and particles, and \( h_{fp} \) are assumed to be constant. The fluid temperature is considered to be time dependent and can be given as:

\[
\left( \frac{T_f - T_i}{T_i - T_i} \right) = 10^{-S_f \Phi_0}. \tag{4}
\]

Since the circumferential variations in the temperature are ignored in the present study, the problem becomes essentially a one-dimensional conduction heat transfer with the boundary conditions. The above equations have been solved using the Crank–Nicholson scheme. The spatial derivatives have been approximated using a central difference scheme and the solutions are generated using a uniformly spaced grid in the radial direction. The following discretized algebraic equations are developed by substituting the finite difference approximations of derivatives to the parabolic equation for the temperature distribution at each node as variable.

The first and second differential equations based on the space derivatives can be written as:

\[
\left( \frac{\partial T}{\partial r} \right)_{n+1} = \left[ \frac{T_{i+1}^{n+1} - T_{i-1}^{n+1} + T_{i+1}^{n} - T_{i-1}^{n}}{4 \Delta r} \right], \tag{5}
\]

\[
\left( \frac{\partial^2 T}{\partial r^2} \right)_{n+1} = \left[ \frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1} + T_{i+1}^{n} - 2T_i^{n} + T_{i-1}^{n}}{2 \Delta r^2} \right]. \tag{6}
\]

Similarly, the first and second differential equations based on time derivatives can be written as

\[
\left( \frac{\partial T}{\partial t} \right)_{n+1} = \left[ \frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1} + T_{i+1}^{n} - 2T_i^{n} + T_{i-1}^{n}}{2 \Delta r^2} \right]. \tag{7}
\]
Equation (8) is used in the convective boundary conditions. The reduced form of the discretized equation at all internal nodes in the finite difference formulation can be given as:

\[ A_i T_{i-1}^{n+1} + B T_i^{n+1} + C_i T_{i+1}^{n+1} = (-A_i)T_{i-1}^n + D T_i^n + (-C_i)T_{i+1}^n, \]  

(9)

where

\[ A_i = \frac{1}{2\Delta r^2} - \frac{1}{2(i-1)\Delta r^2}; \quad B = \frac{-1}{2 \alpha \Delta t}; \quad C_i = \frac{1}{2\Delta r^2} + \frac{1}{2(i-1)\Delta r^2}; \quad D = \frac{1}{2 \alpha \Delta t}. \]

The nodal equations for node \( i = 1 \) can be given as

\[ B T_1^{n+1} + \frac{1}{\Delta r^2} T_1^{n+1} = D T_1^n + \frac{-1}{\Delta r^2} T_2^n. \]

(10)

When the node \( i = N \), it can be obtained by applying the backward difference for convective boundary condition, i.e.:

\[ .T_{N-1} + \left( 1 + \frac{h_{fp} \Delta r}{k} \right) T_N = \frac{h_{fp} \Delta r}{k} T_1. \]

(11)

The set of equations will result in the tridiagonal matrix and can be solved by the TDMA algorithm with time variable boundary conditions. The famous Thomas algorithm has been used to solve this TDMA. The finite difference program for solving TDMA has been executed using MATLAB. The transient particle central point temperature has been generated for wide of range of Biot number and fluid heating rates. The computational grid detail is shown in Fig. 2.

The Biot number varied from 0.1 to 10 and the heating rate of fluid \( (S_f) \) from 2 to 14. The variation of fluid temperature has been considered as linear on a semi-logarithmic scale — a similar situation encountered in the heating of low-viscosity
liquid and food particle mixtures such as vegetables in brine or fruit in sugar syrup. The finite difference program run for 281 different combinations has resulted in 41 different $Bi$ numbers and seven heat transfer rates ($S_f$) to obtain the particle center point temperature profiles. Since the nondimensional particle center temperature varied linearly with the Fourier number on a semi-log scale, the temperature profile can be characterized by the slope ($m$) and the intercept ($C$) for the entire 281 cases and used in the training of the ANN.

2.1. Training of the ANN

The data used to train the ANN can be selected using factorial design. Experience has shown that training the ANN with random points lowers the error of the predictions with the training and testing. The term ‘training points’ refers to the dataset used to train the ANN, while ‘testing points’ refers to the dataset used to test the prediction capabilities of the ANN. Several ANN models have been trained and tested using the dataset of 281 cases. The feed-forward network structure used in the present study is given in Fig. 3.

It can be seen from Fig. 3 that the training employs an input layer of three neurons, corresponding to each of the input parameters $[S_f, m, C]$ and the output layer consists of a neuron representing the output parameter $Bi$. The number of neurons in the input and output layers corresponds to the number of input and output variables, respectively. The number of hidden layers and neurons within each hidden layer can be varied, depending on the complexity and the size of the dataset. In the present investigation, the number of neurons in each hidden layers varied from 2 to 10 with an increments of 2, resulting in a total of 15 networks for one, two and three hidden layers.

In the present study, the dataset is divided into three sets while developing the ANN model, consisting of 91 datasets for training, 98 datasets for validation, and 49 datasets for simulation and validation. The training dataset was used for learning
the ANN and the validation dataset was used to test the generalization of the ANN. The simulation dataset was used to test the ANN predictability. Table 1 gives the characteristics of the ANN training procedure used implemented in MATLAB. The network is trained to recognize the relationship between the input and output parameters. The network parameters, such as connection weights and bias, are adjusted to minimize the difference between predicted and the desired values.

The supervised training technique of the back-propagation algorithm [12] has been used for training the given dataset, where the network weights and biases are initialized randomly at the beginning of the training. The training function ‘TRAINLM’ in the ANN toolbox is used to train the network. Figure 4 gives the training style in MATLAB for the error performance versus epochs. Both the back-propagation algorithm and hyperbolic-tangent sigmoidal transfer function have

Table 1.
Characteristics of the neural network training procedure implemented in MATLAB

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental data</td>
<td>Pattern: dimensionless slope of the fluid heating (Sf) rate, slope (m) and intercept (C) of the dimensionless center point temperature, targets: Biot number (Bi)</td>
</tr>
<tr>
<td>Pattern for training control</td>
<td></td>
</tr>
<tr>
<td>Maximum number of iterations (epochs)</td>
<td>10000</td>
</tr>
<tr>
<td>Error goal</td>
<td>10^{-10}</td>
</tr>
</tbody>
</table>

Figure 4. Training style in MATLAB for the error performance versus epochs.
been used for all the hidden layers. To overcome the potential problem of over-
training/or memorization while employing the back-propagation algorithm [12], the
option of saving the best configuration has been selected in the present inves-
tigation, resulting in networks with the best test results being saved for the selected
number of training/testing cycles of 10 000.

2.2. Optimal Configuration

The optimal configuration has been selected from the ANN configurations based
on the outcome of the following errors equations and the predictive performance of
each configuration has been validated. The error measures used to assess the pre-
dictive performance of the various ANN configurations are now given. The means
absolute error can be given as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \Delta B_A,$$

(12)

where:

$$\Delta B_A = |B_P - B_D|,$$

(13)

the standard deviation of the mean absolute can be written as:

$$\text{STD}_A = \sqrt{\frac{\sum_{i=1}^{n} (\Delta B_A - \bar{\Delta B}_A)^2}{n - 1}},$$

(14)

where $n$ refers to the data points. The mean relative error can be written as:

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^{n} \Delta B_R$$

(15)

and $\Delta B_R$ is given as $\Delta B_R = |(B_P - B_D)/B_D|$, where $B_P$ and $B_D$ represent the
predicted output and desired output, respectively. Finally the standard deviation of
the mean relative error can be give as:

$$\text{STD}_R = \sqrt{\frac{\sum_{i=1}^{n} (\Delta B_R - \bar{\Delta B}_R)^2}{n - 1}}.$$  

(16)

Further, the coefficient of determinant $R^2$ between the output predicted using ANN
and the desired output validates the performance.

3. Results and Discussion

The performance was evaluated with the input dataset after the given ANN config-
uration had been trained, and the analysis was repeated with different combinations
of linear and transformed variables. In the present study, the ANN optimized con-
figuration was selected from the minimized four error measures and the optimized
$R^2$ value. Although the ANN model does not require any prior knowledge of the
relationships between inputs and outputs, it is attempted in the present investigation to improve the performance of the ANN model by transforming the output parameters using a scaling method with the limit of 0 and 1. The data was scaled down before the training for the sake of numerical flexibility and brought back to the original after the training procedure, resulting in improved quality of data fitting. The scaling has been performed using the following equation

\[ y_n = 0.1 + 0.9 \times \left( \frac{y}{y_{\text{max}}} \right) \frac{y_{\text{max}}}{y_{\text{min}}} \], \quad (17) \]

where \( y_n \) is the scaled output variable, \( y \) is the raw output variable, and \( y_{\text{max}} \) and \( y_{\text{min}} \) are the maximum and minimum of the raw output variable.

Table 2 shows the error analysis and the \( R^2 \) values for all configurations for the training dataset, while Table 3 gives the simulation result of error analysis and the \( R^2 \) values. It can be ascertained from Tables 2 and 3 that the values of error measures are reduced with an increase in the number of hidden layers and the neurons. It can be further ascertained that for the given system, hidden layers of three and 10 neurons per hidden layer give the best ANN configuration. The training performance in the ANN toolbox is given in Fig. 4. The parameters used for the training are as given in Table 1.

In the present study, the estimation of the heat transfer coefficient is referred to in terms of a dimensionless number, i.e. the Biot number. Figure 5 shows the simulated Biot number predicted for the range of 7.25–10 along with the desired Biot number. It can be seen from Fig. 5 that the predicted Biot number matches
Table 3.
Statistical analysis of error for the various network configurations for simulating

<table>
<thead>
<tr>
<th>No. of hidden layers</th>
<th>No. of neurons/layer</th>
<th>% MRE</th>
<th>STD_R</th>
<th>MAE</th>
<th>STD_A</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>20.214</td>
<td>0.33357</td>
<td>1.7929</td>
<td>1.036</td>
<td>0.0124</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>15.145</td>
<td>0.28359</td>
<td>1.3662</td>
<td>0.8855</td>
<td>0.1461</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>10.216</td>
<td>0.23651</td>
<td>0.93417</td>
<td>0.75332</td>
<td>0.4603</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>31.616</td>
<td>0.64291</td>
<td>2.8994</td>
<td>1.9491</td>
<td>0.0102</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>6.8253</td>
<td>0.20712</td>
<td>0.6269</td>
<td>0.64989</td>
<td>0.6039</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>12.71</td>
<td>0.22439</td>
<td>1.1584</td>
<td>0.73387</td>
<td>0.797</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.79</td>
<td>0.16684</td>
<td>0.35726</td>
<td>0.52738</td>
<td>0.9639</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2.4902</td>
<td>0.14419</td>
<td>0.23768</td>
<td>0.45509</td>
<td>0.9889</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2.1608</td>
<td>0.13692</td>
<td>0.20683</td>
<td>0.43198</td>
<td>0.9893</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2.262</td>
<td>0.14053</td>
<td>0.21662</td>
<td>0.44289</td>
<td>0.9877</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>13.77</td>
<td>0.22953</td>
<td>1.2522</td>
<td>0.75299</td>
<td>0.7055</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>8.0593</td>
<td>0.20672</td>
<td>0.74554</td>
<td>0.66245</td>
<td>0.9134</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2.1937</td>
<td>0.13788</td>
<td>0.20987</td>
<td>0.43531</td>
<td>0.9877</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2.3654</td>
<td>0.14115</td>
<td>0.22591</td>
<td>0.44573</td>
<td>0.9879</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2.2384</td>
<td>0.1394</td>
<td>0.21428</td>
<td>0.43942</td>
<td>0.9904</td>
</tr>
</tbody>
</table>

Figure 5. Comparison of the predicted with the desired Biot number for the Biot number range 7.25–10 (simulation output).

The network satisfactorily with the desired Biot number within 3% mean relative error and a $R^2$ value of 0.9904. Figure 6 shows the generalized behavior of the ANN for predicting the Biot number in the range of 0.1–10. It can be ascertained from Fig. 6 that the...
Figure 6. Comparison of the predicted with the desired Biot number for the Biot number range 0.1–10 (overall performance).

predicted Biot number matches satisfactorily with the desired values within 3% mean relative error and a $R^2$ value of 0.9987.

4. Conclusion

An ANN model has been developed to estimate the heat transfer coefficient in liquid–solid systems with time-dependent boundary conditions. The proposed ANN is able to predict $\text{Bi}$ satisfactorily with a mean relative error of less than 3% and the model can predict the $\text{Bi}$ (dimensionless heat transfer coefficient) value without employing a time-consuming iterative solution procedure.

References


