Available online at www.sciencedirect.com
SCIENCE DIRECT®
International Journal of Heat and Mass Transfer 48 (2005) 2483–2492
International Journal of HEAT and MASS TRANSFER
www.elsevier.com/locate/ijhmt

Predictions of heat transfer coefficients of supercritical carbon dioxide using the overlapped type of local neural network

Junghui Chen *, Kuan-Po Wang [a], Ming-Tsai Liang [b]

[a] Department of Chemical Engineering, Chung-Yuan Christian University, Chung-Li, Taiwan 320, Republic of China
[b] Department of Chemical Engineering, I-Shou University, Kaohsiung County, Taiwan 840, Republic of China

Received 19 January 2003; received in revised form 29 October 2004
Available online 16 March 2005

Abstract

An overlapped type of local neural network is proposed to improve accuracy of the heat transfer coefficient estimation of the supercritical carbon dioxide. The idea of this work is to use the network to estimate the heat transfer coefficient for which there is no accurate correlation model due to the complexity of the thermo-physical properties involved around the critical region. Unlike the global approximation network (e.g. backpropagation network) and the local approximation network (e.g. the radial basis function network), the proposed network allows us to match the quick changes in the near-critical local region where the rate of heat transfer is significantly increased and to construct the global smooth perspective far away from that local region. Based on the experimental data for carbon dioxide flowing inside a heated tube at the supercritical condition, the proposed network significantly outperformed some the conventional correlation method and the traditional network models.

Keywords: Supercritical flow; Carbon dioxide; Neural network; Heat transfer correlations

1. Introduction

Supercritical carbon dioxide is characterized by high solvent power, high diffusivity, low viscosity, low surface tension, and adjustable physical properties by pressure and temperature. It is widely and maturely applied to natural products extraction due to its non-toxicity, non-residual, and non-combustibility. In an extraction plant, liquefied CO_2 from storage tank is pressurized and pre-heated under the supercritical condition to extract active compounds from the extractor which is pre-loaded with raw materials and depressurized to precipitate the dissolved compounds in a separator. In the separator, the depressurized CO_2 need to be heated to prevent CO_2 liquefaction and to reduce the viscosity of the precipitated solutes. The depressurized gaseous CO_2 is liquefied again and recovered to a storage tanks[1–4]. Obviously the extraction plant contains a pre-heater operated at high pressure, a heater operated at low pressure, and a liquefier operated at low pressure. Usually the area needed for the heater and the liquefier is much bigger than that of the pre-heater. Recent research on the heat transfer of supercritical CO_2 has been focused on the development of a transcritical cycle for a conditioner [5]. Since the heat transfer coefficients to supercritical CO_2 can significantly reach high value, it is an ideal substituted refrigerant for automobile air conditioners.

* Corresponding author. Fax: +886 3 365 4199.
E-mail address: jason@wavenet.cycu.edu.tw (J. Chen).

0017-9310/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijheatmasstransfer.2004.12.040

Nomenclature

a	linear coefficients vector of MRBFN
a_0	constant of MRBFN
$\mathbf{c}_i$	center of the neuron i
$\bar{c}_p$	the averaged isobaric capacity
Ec	Eckert number
K	number of training patterns
$\dot{m}$	mass flow rate
NPT	number of points
Nu	Nusselt number
Pr	Prandtl number
$\dot{q}$	heat flux
$\mathbf{q}_j$	orthogonal column i extracted by the factorization of regressor matrix $\mathbf{R}$
$\mathbf{r}_i$	regressor vector i, of regressor matrix $\mathbf{R}$
R	regressor matrix of MRBFN
Re	Reynolds number
S	number of neurons
$[SSR]_j$	variation of the measured quality in the direction $\mathbf{q}_i$
T	temperature
w_i	coefficient of the neuron i of RBFN
$\mathbf{x}$	input vector, $\in R^n$
y	output pattern, $\in R$
$\hat{y}$	predicted output, $\in R$

Greek symbols

α	heat transfer coefficient
ρ	density
φ_i	radial basis function i
σ_i^2	is variance of the radial basis function i
Γ	matrix from QR decomposition
∇	gradient operator
$\mathbf{\Omega}$	unknown parameter vector, which consists of the center and variance of each neuron of MRBFN

Subscripts

b	at bulk
r	reduced
w	at wall

In order to design a highly efficient heat transfer system, the estimation of convective heat transfer for supercritical carbon dioxide is strongly needed. So far many empirical equations are proposed to calculate the heat transfer coefficient. When the fluid is close to its critical point, the modification of the empirical equation is also proposed [5,6]. Traditionally, dimensional analysis is often used to reduce the number of variables to the internal and external Nusselt numbers as functions of the corresponding Reynolds and Prandtl numbers. Then such correlations could be estimated from experimental measurements. In practice, property variations, especially the variation of liquid viscosity, make correlation obtained highly dependent on fluid temperatures. Procedures that take this variation into account become very complex and potentially lose generality. The heat transfer system is complex due to the physical phenomena presented in the transfer of heat and the larger number of variables involved in its operation. This increases the difficulty of solving the governing equation based on a first-principal approach. Most of the models that are available rely on assumptions and simplifications that disagree at the real conditions of operation.

The neural network technique offer an alterative approach to the problem information compression for the heat exchanger. It is a method that is often used for predicting the response of a physical system that cannot be easily modeled mathematically. Neural networks have demonstrated the strong capability of learning non-linear and complex relationships between process variables without any prior knowledge of system behaviors. Since the highly complex behavior of heat transfer systems in the near critical region is presently ahead of the theoretical method from a fundamental physical standpoint, the network is derived from the data presented instead of the exact form of the analytical function on which the model should be built. By training the net to reduce the difference between the neural network output and the actual experimental values, each neural network represents a non-linear or complex behavior for the output that it learned. The neural network is ideally suited for the heat transfer process problem mainly because of the derivation from the data presented instead of the exact form of the heat transfer on which the model should be built. Many researchers have focused on the neural network approach to heat transfer modeling. In recent years, the number of applications of neural networks to heat transfer process has increased dramatically. Research shows that neural network models exhibit superior predictive abilities over traditional statistical methods and require less experimental training data. Literature that deals with the use of neural networks to the heat transfer problems includes: the prediction of the heat transfer coefficient [7], Nusselt number [8,9], heat transfer rates [10,11], the simulation of a liquid-saturated steam heat exchanger [12] and so on.

Basically, a neural-network approach can save time and money. It can learn and extract the process behavior from the past operating information. It can be used as a model for process optimization and design. In the past research, the backpropagation neural network (BPN) [13] is often used. However, a large amount of data

points which should cover the whole design region are needed for training a healthy neural network model. It is not applicable to process design since only the data located at some local design regions is critical to building the neural network model. The structure of the locally tuned and overlapping receptive field has been widely applied in the region of the cerebral cortex, the visual cortex, etc. Drawing on knowledge in biological receptive fields, the radial basis function network (RBFN) [14] with the property of the local function is proposed to eliminate unnecessary and extrapolation errors. The non-linear mapping is used to transfer the inputs into the intermediate outputs covered at some local regions. Due to the local structure, the curse problem, which refers to the exponential increase in the number of hidden neurons with the increase of the input space dimensions, still exists.

In this paper, a neural network with the overlap structure, the modified RBFN (MRBFN), is proposed to improve the modeling the heat transfer coefficient for supercritical CO_2 in the heat exchanger system. It has not only the valid smooth approximation in the design space but also the fine and variation approximation in several local neighborhoods. It provides a systematic modeling procedure based on the input–output data with the desired accuracy. The comparisons with the conventional correlation method and the model based on BPN and RBFN are also made.

2. Experimental data

Experiments from Olson and Allen [15] were conducted with the counterflow heat exchanger. Six sets of experiments were obtained from a series of tests of the fixed carbon dioxide pressure at the different values. Within each set, carbon dioxide flow rate varied, as was the amount of the heat transferred from the water to the carbon dioxide. The mass flow rate, inlet temperature, outlet temperature, inlet pressure and the differential pressure for both carbon dioxide and water sides were measured. A total of 1115 data were collected to use as the training and validation sets.

3. Neural networks: Global and local approximation

Assume that a set of training patterns is given, that is, a set of input and desired output pairs

$$T = \{(\mathbf{x}_k, y_k) : \mathbf{x}_k \in R^n, y_k \in R, k = 1, 2, \ldots, K\} \quad (1)$$

The data vectors $\mathbf{x}_k$ are referred to as input patterns and y_k output patterns. The goal of training the network is to adjust the parameters of the network to minimize the objective function which is usually the square error between the predicted output of the network and the desired output. When it comes to train the net, the back-propagation (gradient descent) algorithm is often used. Once trained, each neural network represents a non-linear or complex function for the output that it learned.

Despite all the mathematical theorems that have been advanced to support BPN as an approximation scheme, there are some problems in the practical considerations: (1) The learning convergence is not fast enough. Due to the global activation function residing on each network neuron, the neuron in the network typically influences the output over a large range of input values. (2) The hyperplane activation functions (such as the tangent function and the logistic function) are often used in BPN. When they have an activation very close to zero or to one, the weight adjustments would be close to zero. In this situation, the learning algorithm of BPN becomes paralyzed. (3) The BPN has larger extrapolation error. Because BPN is done without considering the availability or density of training data in a given range of inputs, it may lead to large extrapolation error without warning. (4) The design procedures or guidelines on how to build the structure of ANN are poorly developed. They mostly rely on some kinds of trial and error procedures.

RBFN is an alternate model to BPN for process identification. It refers to a neural network using the radial basis function rather than the hyperplane function. The radial basis function is a local and multidimensional function. It employs a distance term to divide an input space with localized input field. RBFN shown in Fig. 1 consists of three layers: an input layer, a hidden layer and an output layer. It can be mathematically defined as:

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{S} w_i \varphi_i(\|\mathbf{x} - \mathbf{c}_i\|) \quad (2)$$

where S is the number of neurons, φ is the basis function, $\varphi_i = \exp[-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{2\sigma_i^2}]$, $\mathbf{c}_i$ serves as the center of the basis function i, σ_i^2 is variance of the basis function i, $\hat{y}$ is the predicted output, and w_i is referred to the radial weighting coefficients.

Due to the local nature of the activation function, RBFN minimizes the problems of long training time and slow convergence. It still suffers from some shortcomings. (1) The local nature of RBF may result in

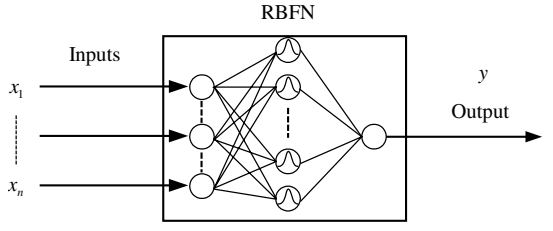

Fig. 1. RBFN network.

the exponential increase in the number of hidden neurons with the increase of the input space dimensions [16]. (2) The operation regions of any two RBFs may not overlap. As a result, if a test input falls outside of the convex sets, the response of RBF would be close to zero, yielding incorrect output. To lessen this problem, some researchers have tried to use regularization theory [17] whose regularization term can compromise between the error of approximation and the degree of smoothness of an unknown process.

4. The overlapped type of local neural networks

As we discussed in the previous section, the structures of the neural network can be classified as global and local approximations. The global approximation (like BPN) is valid in the whole region defined by side constraints, but it cannot easily match the occasionally quick changes in some of the regions as in the near-critical region where the rate of heat transfer is significantly increased than that far away from the region. On the other hand, local approximation (like RBFN) is valid in the vicinity of the points at which they are generated. They can help reducing the complexity of the problem at some local region. Due to their local characteristics, they lack a global perspective and convergence to a smooth model. In this paper, a modified RBFN (MRBFN) will be developed to improve the ability of the prediction of RBFN. Based on the data distribution, the proposed network can properly patch the essential features in different locations so that a systematic neural network design procedure can be conducted.

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{S} w_i \varphi_i(\|\mathbf{x} - \mathbf{c}_i\|) + \mathbf{a}^{\mathrm{T}}\mathbf{x} + a_0 \tag{3}$$

Eq. (3) is a modified version of the traditional RBFN. It consists of two parts—a traditional RBFN to identify the non-linear characteristics of the process; and a linear model to capture linear characteristics of the batch process. Thus, the vector $\mathbf{a}$ contains the linear coefficients and a_0 is a constant. Three steps for training based on the previously developed work are used here [18]. The training procedures are briefly outlined as follows:

(1) *Selecting neuron candidates*. Neuron candidates are selected based on the distribution density of the input training data. The distribution of the input training data can be viewed at different levels of resolutions. A neuron is assigned to the center location of each resolution cell that covers the training data. For brief explanation, the resolution cells in the two-dimensional space are illustrated in Fig. 2. The "Level" axis represents different levels of resolution. The binary partitions are selected for efficient computation and easy explanation. As shown in Fig. 2, Level 1 stretches the entire operating

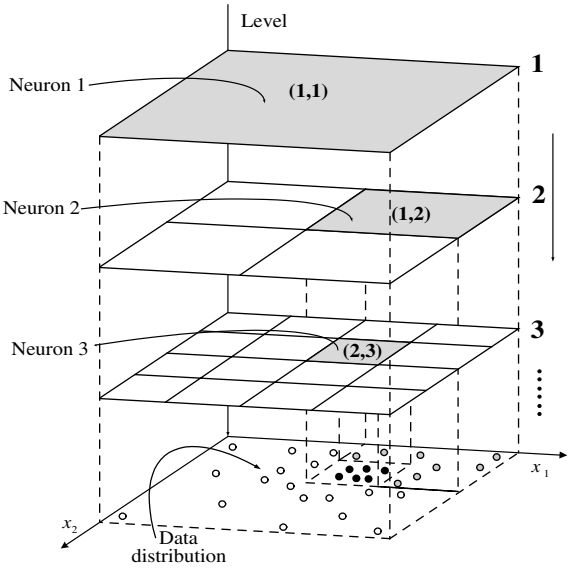

Fig. 2. The data distribution for radial base functions at different levels of resolutions, where the shaded regions represent the possible neuron candidates.

space; thus, it creates a coarse resolution to support all data; i.e. Neuron 1 would cover cell $(1,1)$. Contrarily, the resolution becomes finer by going downward through the levels. Therefore, only a limited number of data are covered in the finer resolution of the cell. The input data (gray points and black points) distributed at Cell $(1,2)$ of Level 2 would be covered by Neuron 2, and the input data (black points) at Cell $(2,3)$ of Level 3, by Neuron 3. Based on the desired data density, the active neurons are selected as the candidates. As the candidates are found, the corresponding center ($\mathbf{c}_i$) and the width (σ_i) values are determined by the location of the active cell.

(2) *Purifying neurons*. Not all of the neuron candidates have a strong impact on the desired output. They should be refined using the output training data. The classical Gram–Schmidt algorithm (CGS) [19,20] is used as a purifying scheme to retain the most contributive neurons and remove the least contributive ones. To identify RBFN model, substituting each data pair into Eq. (3) yields a set of K equations. These equations can be written in a concise matrix form as

$$\mathbf{y} = \mathbf{R}\boldsymbol{\theta} + \mathbf{e} \tag{4}$$

where

$$\begin{aligned}
\mathbf{J} &= [\,y_1 \quad y_2 \quad \cdots \quad y_K\,]^{\mathrm{T}} \\
\mathbf{e} &= [\,e_1 \quad e_2 \quad \cdots \quad e_K\,]^{\mathrm{T}} \\
\boldsymbol{\theta} &= [\,w_1 \quad w_2 \quad \cdots \quad w_S \quad a_1 \quad \cdots \quad a_N \quad a_0\,]^{\mathrm{T}} \\
\mathbf{R} &= [\,\mathbf{r}_1 \quad \cdots \quad \mathbf{r}_S \quad \mathbf{r}_{S+1} \quad \cdots \quad \mathbf{r}_{S+N} \quad \mathbf{r}_{S+N+1}\,]^{\mathrm{T}}
\end{aligned} \tag{5}$$

and

$$\mathbf{r}_i = \begin{cases} \begin{bmatrix} \varphi_i(\|\mathbf{x}_1 - \mathbf{c}_i\|) & \varphi_i(\|\mathbf{x}_2 - \mathbf{c}_i\|) & \cdots & \varphi_i(\|\mathbf{x}_K - \mathbf{c}_i\|) \end{bmatrix}^{\mathrm{T}} \\ \quad 1 \leqslant i \leqslant S \\ \begin{bmatrix} x_{i-(S+1),1} & x_{i-(S+1),2} & \cdots & x_{i-(S+1),K} \end{bmatrix}^{\mathrm{T}} \\ \quad S+1 \leqslant i \leqslant S+N \\ \mathbf{1} \quad i = S+N+1 \end{cases} \tag{6}$$

In Eq. (6), $x_{i,k}$ is the input variable i of the pattern k, $\mathbf{1}$ is a column vector consisted of K 1's. The variable e represents the residuals or errors of the approximation problem defined by the training data. The goal of the approximation is to minimize the sum of square errors.

$$\sum_{k=1}^{K} e_p^2 = \mathbf{e}^{\mathrm{T}}\mathbf{e} \tag{7}$$

Generally the columns of $[\mathbf{r}_1 \cdots \mathbf{r}_S\ \mathbf{r}_{S+1} \cdots \mathbf{r}_{S+N}\ \mathbf{r}_{S+N+1}]$ are not mutually orthogonal. The regressor matrix $\mathbf{R}$ can be factored into a product using an unnormalized QR decomposition, $\mathbf{R} = \mathbf{Q}\boldsymbol{\Gamma}$,

$$\mathbf{y} = \mathbf{Q}\mathbf{g} + \mathbf{e} \tag{8}$$

and the vector $\mathbf{g} = \boldsymbol{\Gamma}\boldsymbol{\theta}$ can be determined by the least squares method

$$\mathbf{g} = (\mathbf{Q}^{\mathrm{T}}\mathbf{Q})^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{y} \tag{9}$$

where $\mathbf{Q}=[\mathbf{q}_1\ \mathbf{q}_2 \cdots \mathbf{q}_{S+N+1}]$. Because the columns of $\mathbf{Q}$ are orthogonal, Eq. (9) becomes

$$g_i = \frac{\mathbf{q}_i^{\mathrm{T}}\mathbf{y}}{(\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_i)} \quad i = 1, 2, \ldots, S+N+1 \tag{10}$$

and $\mathbf{g} = [g_1\ g_2 \cdots g_{S+N+1}]$. Under the assumption that $\mathbf{e}$ is white noise, Eq. (8) yields

$$\mathbf{y}^{\mathrm{T}}\mathbf{y} = \sum_{j=1}^{S+N+1} g_j^2\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j + \mathbf{e}^{\mathrm{T}}\mathbf{e} \tag{11}$$

Define $[SSR]_j = g_j^2\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j$. This term, sum-square-regression, is used to explain the variation of $\mathbf{Y}$ expanded in direction $\mathbf{q}_i$, which corresponds to one column of the matrix $\mathbf{R}$. The most important neuron or linear term that maximizes $[SSR]_j$ should be selected and the corresponding weight coefficient of each regressor ($\mathbf{r}_j$) is also computed (w_i, or a_i). Therefore, by projecting the output training data onto the neurons, the projections may be dense on some neuron candidates and sparse on the others. Those neurons responding to a few or no data from the output space viewpoint are considered redundant and they should be removed.

(3) *Optimizing the neural parameters*. The refined network can be made to be even closer to the system model using a gradient search algorithm for optimization. Newton's method is used to minimize the objective function:

$$M(\boldsymbol{\Omega}_q) = \sum_{k=1}^{K}(\hat{y}(\mathbf{x}_k) - y_k)^2 \tag{12}$$

and

$$\boldsymbol{\Omega}_{q+1} = \boldsymbol{\Omega}_q - \lambda[\mathbf{H}(\boldsymbol{\Omega}_q)]^{-1}\nabla M(\boldsymbol{\Omega}_q) \tag{13}$$

where $\boldsymbol{\Omega}_q = [\{\mathbf{h}_i\}, \{\sigma_i\}]$, $[H(\boldsymbol{\Omega}_q)]^{-1}$ is the inverse of Hessian matrix of $M(\boldsymbol{\Omega}_q)$; $\nabla M(\boldsymbol{\Omega}_q)$ contains the gradient vectors, $\partial M/\partial \mathbf{z}_i$, $\partial M/\partial \sigma_i$, and λ is the step size.

The MRBFN modeling procedures of the heat transfer coefficients of supercritical CO_2 are summarized as follows:

(I) Select the physical variables ($\mathbf{x}$) that significantly affect the heat transfer coefficients (y). The fundamental physical equations and the transport property equations in the conventional heat transfer correlation can give the possible selected variables. This part is further explained in the next section.

(II) Initiate the neural network structure, including
  (1) Selecting the candidate neurons based on the binary partition method to find out the locations of the neurons that cover the minimum required data density.
  (2) Purifying the redundant neurons that have no significant contribution to the desired values of the heat transfer coefficients. The removing procedures using the Gram–Schmidit orthogonalization method are shown as follows:

Step 1. $k = 1, 1 \leqslant i \leqslant S+N$

$$\mathbf{q}_1^{(i)} = \mathbf{r}_\mathbf{i}$$

$$g_1^{(i)} = \frac{(\mathbf{q}_1^{(i)})^{\mathrm{T}}\mathbf{J}}{(\mathbf{q}_1^{(i)})^{\mathrm{T}}\mathbf{q}_1^{(i)}}$$

$$[SSR]_1^{(i)} = (g_1^{(i)})^2(\mathbf{q}_1^{(i)})^{\mathrm{T}}\mathbf{q}_1^{(i)}$$

select the first important vector $\mathbf{q}_1 = \mathbf{q}_1^{(i_1)}$ such that $i_1 = \max_i[SSR]_1^i$ and $SSR_1 = [SSR]_1^{(i_1)}$.

Step 2. $k = k + 1, 1 \leqslant i \leqslant S+N$ and $i \neq i_1, \ldots, i \neq i_{k-1}$

$$\alpha_{jk}^{(i)} = \frac{\mathbf{g}_j^{\mathrm{T}}\mathbf{r}_i}{\mathbf{g}_j^{\mathrm{T}}\mathbf{g}_j}$$

$$\mathbf{q}_k^{(i)} = \boldsymbol{\phi}_i - \sum_{j=1}^{k-1}\alpha_{jk}^{(i)}\mathbf{q}_j$$

$$g_k^{(i)} = \frac{(\mathbf{q}_1^{(i)})^{\mathrm{T}}\mathbf{J}}{(\mathbf{q}_1^{(i)})^{\mathrm{T}}\mathbf{q}_1^{(i)}}$$

$$[SSR]_k^{(i)} = (g_1^{(i)})^2(\mathbf{q}_k^{(i)})^{\mathrm{T}}\mathbf{q}_k^{(i)}$$

select the kth important vector $\mathbf{q}_k = \mathbf{q}_k^{(i_k)}$ such that $i_k = \max_i [SSR]_k^i$ and $SSR_k = [SSR]_1^{(i_k)}$. The iterative procedure can be terminated when

$$1 - \frac{\sum_{i=1}^{k} SSE_i}{(\mathbf{y} - \bar{y})^{\mathrm{T}}(\mathbf{y} - \bar{y})}$$

is below the desired performance, where $\bar{y} = \frac{1}{K}\sum_{i=1}^{K} y_i$.

Step 3. The coefficients θ can be directly computed after getting the desired neurons ($\mathbf{\Phi}_{\mathrm{pur}} = [\mathbf{r}_{i_1}\ \mathbf{r}_{i_2}\ \cdots\ \mathbf{r}_{i_{K_s}}\ \mathbf{r}_{S+N+1}]$),

$$\boldsymbol{\theta} = \mathbf{\Phi}_{\mathrm{pur}}^{+}\mathbf{y}$$

where K_s is the number of the selected neurons and $\mathbf{\Phi}_{\mathrm{pur}}^{+}$ is the pseudoinverse of matrix $\mathbf{\Phi}_{\mathrm{pur}}$.

(III) Refine the initial neural network. The Gauss–Newton method (Eq. (12)) is used here. Due to the good initialization of the neural network, it is much more efficient than the widely used back-propagation procedure.

5. Result and discussion

In this study, the conventional method and three different neural network modeling approaches (BPN, RBFN and MRBFN) are applied. In the network training, four different types of input and output are used [9] which are shown in Table 1, where Re, Pr, Ec, Nu, α, P_r, T_r, $\dot{m}$, $\dot{q}$ and $\frac{\bar{c}_p}{c_{p,b}}$ are Reynolds number, Prandtl number, Eckert number, Nusselt number, heat transfer coefficient, reduced pressure, reduced temperature, mass flow, heat flux, the average heat capacity of CO_2 between the bulk (T_b) and the wall (T_w) temperature, respectively.

The data from Olson and Allen [15] are randomly selected from 250 points of 1115 experimental data points. The validity range of the model is covered by the whole experimental data. To have a systematical comparison, three validation indices [9] are used

$$\begin{aligned}(\Delta\%)_i &= \frac{x_i^{\mathrm{exp}} - x_i^{\mathrm{calc}}}{x_i^{\mathrm{calc}}} \times 100\\ \mathrm{ADD}\% &= \frac{1}{\mathrm{NPT}}\sum_{i=1}^{\mathrm{NPT}} |\ \Delta\%|_i\\ \mathrm{Bias}\% &= \frac{1}{\mathrm{NPT}}\sum_{i=1}^{\mathrm{NPT}} (\Delta\%)_i, \quad \mathrm{NPT} = 1115\end{aligned} \tag{14}$$

Table 1
Four different types of input and output variables that are used to train the neural network models

Types	Inputs ($\mathbf{x}$)	Output (y)
(i)	Re, Pr and Ec	Nu
(ii)	P_r, T_r, $\dot{m}$ and $\dot{q}$	α
(iii)	Re, Pr, ρ_w/ρ_b and $\bar{c}_p/c_{p,b}$	Nu
(iv)	P_r, T_r, $\dot{m}$ and T_w/T_b	α

Table 2
Comparison among three different neural networks and the correlation method under different input and output structures: (a) Type I; (b) Type II; (c) Type III; and (d) Type IV

	KPPG	BPN (7 neurons)	RBFN (10 neurons)	RBFN (25 neurons)	MRBFN (10 neurons)
(a)					
ADD (%)	3.50	1.55	6.0775	4.2964	1.8024
Bias (%)	3.04	0.08	0.2083	0.1552	−0.2180
Max (%)	15.86	−12.07	32.9908	32.6051	21.2581
	KPPG	BPN (6 neurons)	RBFN (10 neurons)	RBFN (20 neurons)	MRBFN (10 neurons)
(b)					
ADD (%)	3.50	1.72	3.7926	3.6797	0.7922
Bias (%)	3.04	0.61	0.1340	0.3426	0.0353
Max (%)	15.86	13.01	36.1438	33.9764	7.0535
	KPPG	BPN (6 neurons)	RBFN (10 neurons)	RBFN (22 neurons)	MRBFN (10 neurons)
(c)					
ADD (%)	3.50	2.98	5.5539	2.5300	0.8026
Bias (%)	3.03	0.29	−0.1461	0.0380	−0.0491
Max (%)	15.85	12.46	20.5924	18.7107	5.5366
	KPPG	BPN (6 nodes)	RBFN (10 nodes)	RBFN (25 nodes)	MRBFN (10 nodes)
(d)					
ADD (%)	3.50	1.19	2.6648	2.2084	0.5212
Bias (%)	3.04	0.05	0.0886	0.0158	−0.0172
Max (%)	15.86	−24.81	13.3901	16.1010	2.8552

The comparison with the performances of the different predicted models under the different input and output structures are given in Table 2. KPPG is referred to as the Krasnoshchekov, Protoppov, Pethukov, Gnielinski conventional correlation [15]. The values of KPPG and BPN are directly obtained from Scalabrin and Piazza paper [9]. RBFNs with two different numbers of local networks cannot perform as good as the BPN network in terms of the evaluated indices even if RBFNs are better than the conventional KPPG method. After the network structure with the combination of the different overlapped local networks, the proposed MRBFN surpasses the others inapproximating the heat transfer model. The performance of MRBFN with only 10 neurons is significantly better than that of RBFN.

According to the definition of the heat transfer coefficient,

$$\alpha(T_w - T_\infty) = -k\left(\frac{\partial T}{\partial r}\right)_{\text{at wall}} = \dot{q} \tag{15}$$

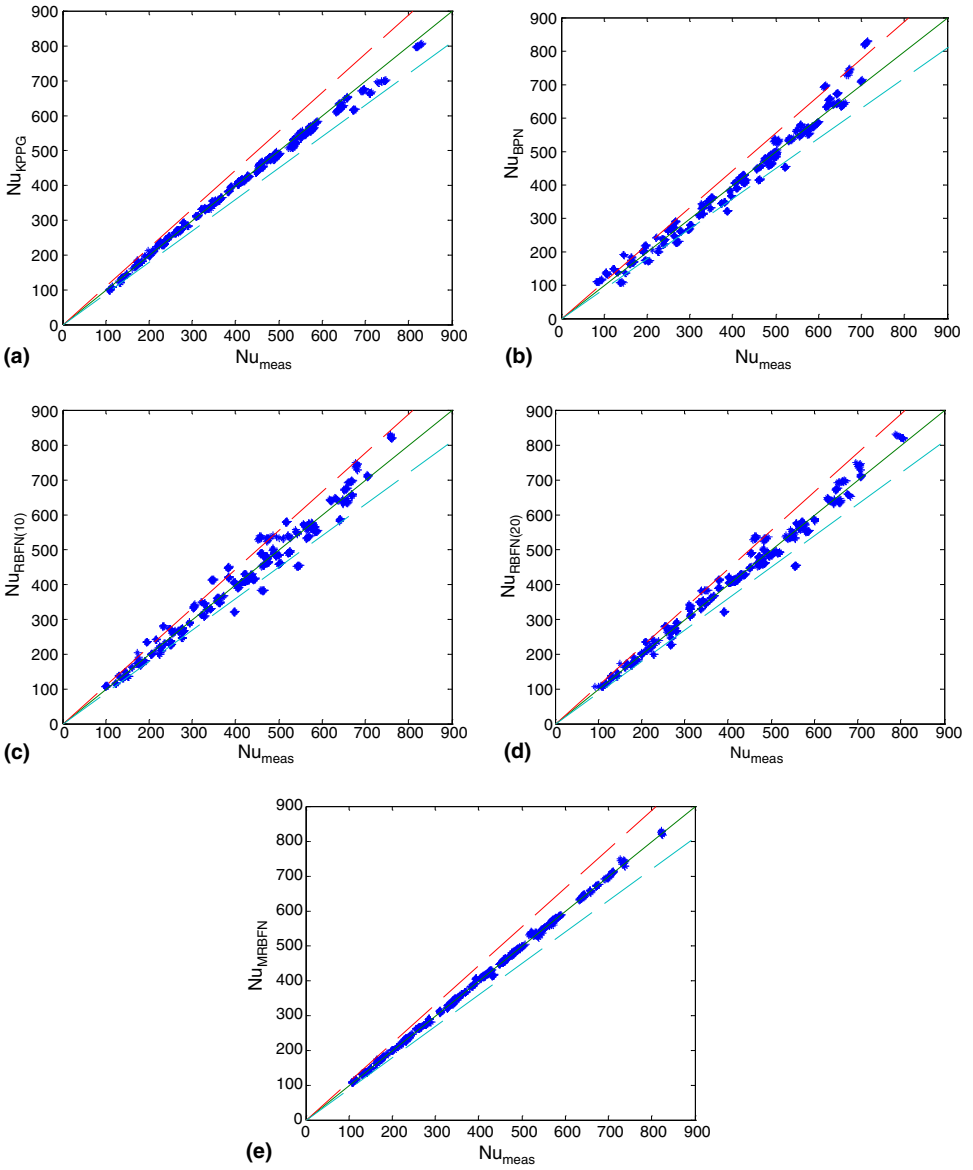

Fig. 3. Comparison of Nusselt number predicted by: (a) KPPG; (b) BPN; (c) BRFN (10 neurons); (d) BRFN (20 neurons), and (e) MRBFN (10 neurons) to the measured value (Nu_{meas}) for all experimental data. ±10% deviation lines are shown for comparison.

the heat transfer coefficient, α, directly depends on the temperature difference, $T_w - T_\infty$, the heat conductivity of CO_2, k, and the temperature gradient, $\left(\frac{\partial T}{\partial r}\right)_{\text{at wall}}$. The thermal conductivity (k) is a function of temperature and pressure. Analogy to the analysis of the velocity gradient, temperature gradient is mainly dominated by the mass flux inside the tube. Therefore, among the four types of input and output structures, Type IV gives the most original and direct information to the estimation of heat transfer coefficient. It has the best-predicted performance.

In the input structure of Type II and IV, the only difference is that the heat flux, $\dot{q}$, is used in Type II and T_w/T_b, in Type IV. Based on Eq. (15), the heat transfer coefficient is directly related to the temperature difference $T_w - T_\infty$ and the heat flux, $\dot{q}$. Since the input structure of Type II does not have anything to do with temperature directly, T_w/T_b in Type IV given as the temperature profile can enhance a better prediction performance.

The input structures of Types I and III based on some dimensionless variables (*Re*, *Pr* or *Ec*) instead of

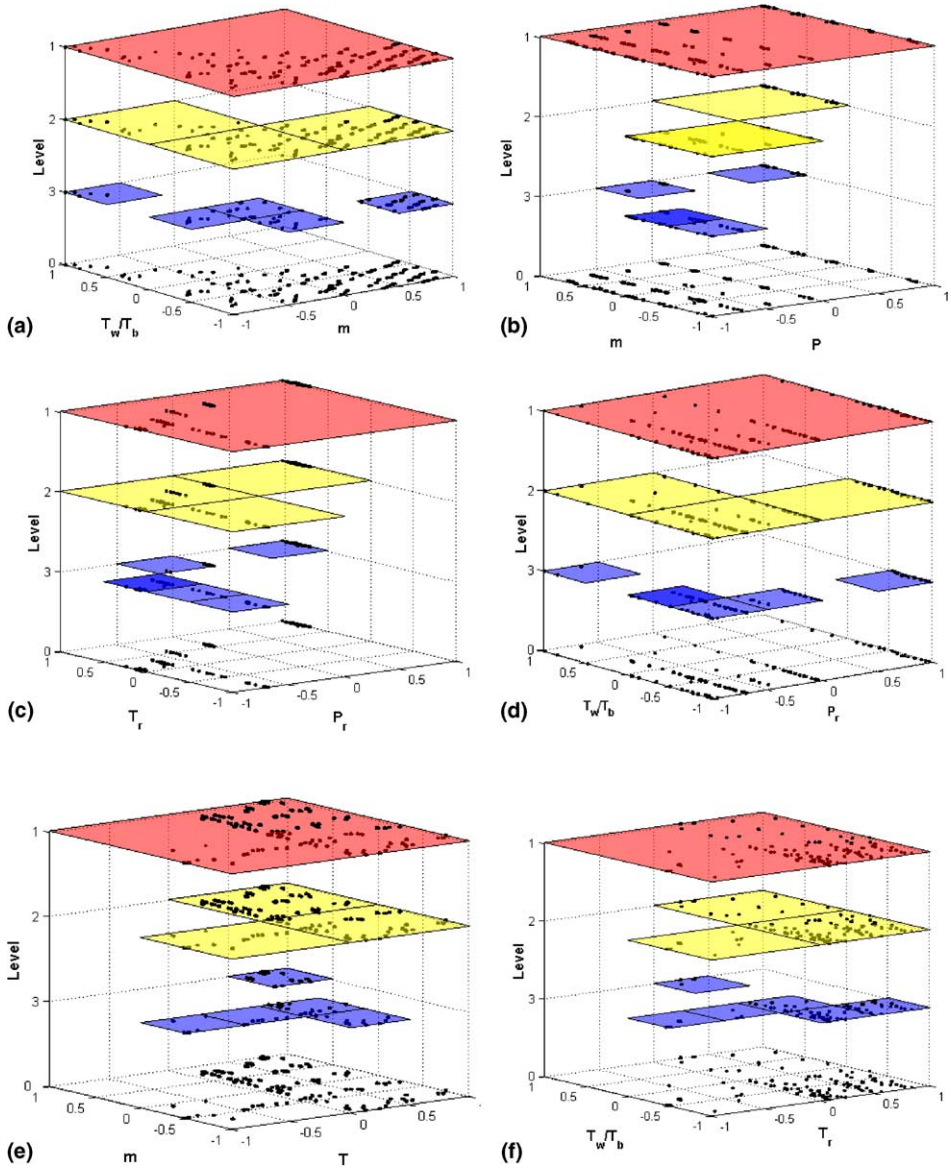

Fig. 4. The selected radial base functions whose cross points are the distribution of the experimental data, and the shaded regions at each level are covered by the selected neurons: (a) $\dot{m}$ and T_w/T_b; (b) P_r and $\dot{m}$; (c) P_r and T_r; (d) T_w/T_b and P_r; (e) $\dot{m}$ and T_r; (f) T_w/T_b and T_r.

the physical variables (like T and P) are not directly correlated to the heat transfer itself. By these input variables, the calculation of the thermo-physical properties may create some uncertainty. Therefore, the prediction performances are a bit worse than the other types of input structures. However, ρ_w/ρ_b and $\bar{c}_p/c_{p,b}$ in type III give more direct thermo-physical information, when MRBFN is used. The prediction of Type III gives a better performance than that of Type I.

In Fig. 3, the representative points for all experimental data, as the predicted Nusselt number, Nu_{pred}, based on the input structure of Type III, versus the measured Nusselt number Nu_{meas} are plotted. Also plotted are a 45° line representing exact agreement between the predictions and the experimental measurements and ±10% deviation lines. It can be noticed that both the accuracy and the precision of the predictions based on MRBFN are remarkable when comparing the estimations of other models. In this study, it demonstrates that a small effort to optimize MRBFN in the training process can enhance modeling accuracy.

Fig. 4 shows the input region (of Type IV) is divided into several local region models whose predictions can be combined to yield a prediction to the whole system. For easy visualization, two variables selected from the four input variables are plotted in two-dimensional space. A total of six plots are needed. In each subplot, the level axis represents different resolutions. The cross points (x) in the plots represent data distribution. The whole region on top has coarse resolution while the lower regions have finer resolutions. The shaded areas are the selected neurons that consist of corresponding data points. From the set of figures, we can see the advantages of building the local model of MRBFN at multiple resolutions. The resolutions of the various local regions in the input space are decided by the distributions of the training data. The regions with finer resolutions that are used to approximate the quick change in the regions are also partially overlapped with the regions lower resolutions which cover larger design space. This explains why MRBFN outperforms the other structures of the neural networks.

6. Conclusions

Due to the complexity of the physics involved in the supercritical condition, the neural network modeling framework based on the structure of the overlapped local neurons applied in this paper is intended to improve the prediction ability of the heat transfer coefficient for supercritical CO_2. While the conventional correlation method is often sufficient for many heat transfer design and optimization problems, the neural network modeling may be of interest when the system is operated with an extremely non-linear and quick-change behavior in the near supercritical region. The alterative neural network design procedure, MRBFN, offers a systematical framework for constructing and training network. For practical use, the overlapped and localization property of the proposed network brings the benefits of the fast convergence and easy training. Compared with the other two existing and popular networks (RBFN and BPN) in this study, the proposed network which includes the features of the overlapped local regions can achieves better estimations. Although the supercritical CO_2 is used in this paper, a proposed model can be easily extended to predict the heat transfer coefficient when any supercritical fluid is applied to a heat exchanger.

Acknowledgement

We are indebted to Dr. Douglas Olson (National Institute of Standards and Technology, MD, U.S.A.) for sharing his experimental data. This work was partly sponsored by National Science Council, R.O.C. and by the Ministry of Economic, R.O.C.

References

[1] T. Clifford, Fundamentals of Supercritical Fluids, Oxford University Press, 1999.
[2] G. Brunner, Gas Extraction, Springer, New York, 1994.
[3] M. Mukhopadhyay, Natural Extracts Using Supercritical Carbon Dioxide, CRC Press, Boca Raton, FL, 2000.
[4] L.B. Koppel, J.M. Smith, Turbulent heat transfer in the critical region, ASME Int. Dev. Heat Transfer 3 (1961) 585–590.
[5] S.S. Pitla, D.M. Robinson, E.A. Groll, S. Ramadhyani, Heat transfer from supercritical carbon dioxide in tube flow: a critical review, HVAC&R Res. 4 (1998) 281–301.
[6] H. Tanaka, N. Nishiwaki, M. Hirata, A. Tsuge, Forced convention heat transfer to fluid near critical point flowing in circular tube, Int. J. Heat Mass Transfer 14 (1971) 739–750.
[7] K. Jambunathan, S.L. Hartle, S. Ashforth-Frost, V.N. Fontama, Evaluating convective heat transfer coefficient using neural networks, Int. J. Heat Mass Transfer 39 (1996) 2329–2332.
[8] J. Thibault, B. Grandjean, A neural network methodology for heat transfer data analysis, Int. J. Heat Mass Transfer 34 (1991) 2063–2070.
[9] G. Scalabrin, L. Piazza, Analysis of forced convention heat transfer to supercritical carbon dioxide inside tubes using neural networks, Int. J. Heat Mass Transfer 46 (2003) 1139–1154.
[10] G. Diaz, M. Sen, K.T. Yang, R.L. McClain, Simulation of heat exchanger performance by artificial neural networks, Int. J. HVAC&R Res. 5 (3) (1999) 195–208.
[11] A. Pacheco-Vega, M. Sen, K.T. Yang, R.L. McClain, Neural network analysis of fin-tube refrigerating heat exchangers with limited experimental data, Int. J. Heat Mass Transfer 44 (2001) 763–770.

[12] S. Bittanti, L. Piroddi, Nonlinear identification and control of a heat exchanger: a neural network approach, J. Frankin Inst. 334B (1) (1997) 135–153.
[13] D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing: Exploration in the Microstructure of Cambridge, Cambridge, MA, 1986.
[14] P. Niyogi, F. Girosi, On the relationship between generalization error, hypothesis complexity and sample complexity for radial basis functions, Neural Computat. 8 (1996) 819–842.
[15] D.A. Olson, D. Allen, Heat transfer in turbulent supercritical carbon dioxide flowing in a heated horizontal tube, NISTIR 6234 Report, NIST, Gaithersburg, USA, 1998.
[16] S. Haykin, Neural Networks a Comprehensive Foundation, Macmillan College Publishing Company, New York, 1994.
[17] T. Poggio, F. Girosi, Network for approximation and learning, in: Proc. IEEE 1990.
[18] J. Chen, D.D. Bruns, WaveARX neural network development for system identification using a systematic design synthesis, Ind. Eng. Chem. Res. 34 (1995) 4420–4435.
[19] T. Proll, N. Karim, Model-prediction pH control using real-time NARX approach, AIChE J. 40 (2) (1994) 269–282.
[20] G. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans Neural Networks 2 (2) (1991) 302–309.