Adsorption 5, 183–192 (1999)

Neural Network Modeling of Adsorption of Binary Vapour Mixtures

M. CARSKY
University of Durban-Westville, Department of Chemical Engineering, Durban, South Africa

D.D. DO
The University of Queensland, Department of Chemical Engineering, Brisbane, Australia

Received February 25, 1997; Revised August 11, 1998; Accepted August 26, 1998

Abstract. Three neural network models were used for prediction of adsorption equilibria of binary vapour mixtures on an activated carbon. The predictions were compared both with published experimental data and calculated values from the Ideal Adsorption Solution (IAS) model. The neural network was trained using both binary and single component experimental adsorption data. Even for a limited number of data points (about 60) the network models were capable of approximating experimental data very precisely.

Keywords: adsorption, activated carbon, neural network, ideal adsorption solution theory

Introduction

Adsorption of organic vapours on activated carbon has been extensively studied and the experimental data can be found in the literature, e.g., (Bansal et al., 1988; Szepesy and Illes, 1963; Hyun and Danner, 1982; Talu and Zwiebel, 1986; Nakahara et al., 1981). However, as such measurements are very time consuming and not easy, the prediction of multicomponent adsorption behaviour is very desirable.

A number of empirical and semi-empirical models are employed for single component adsorption. For example, in Valenzuela and Myers (1989), the Toth equation (for predicting n, the number of moles adsorbed per gram of adsorbent) was used to describe the experimental data of Szepesy and Illes (1963):

$$n = \frac{mP}{(b + P^t)^{1/t}} \tag{1}$$

For low pressure, Eq. (1) becomes

$$n \cong \frac{mP}{b^{1/t}} \tag{2}$$

while for high pressure

$$n \cong m \tag{3}$$

In the case of binary adsorption, both empirical and theoretical models have been developed to predict the adsorption equilibria based on single component adsorption data. Thus, for example the ideal adsorption solution theory of Myers and Prausnitz assumes a standard state such that the spreading pressure of the mixture is the same as the spreading pressures of all pure components, that is:

$$\frac{z}{RT} = \frac{z_j^0}{RT} = \int_0^{P_j^0} \frac{n_j^0}{P_j^0} dP_j^0 \quad \text{for all } j \tag{4}$$

where z is the spreading pressure of the mixture and z_j^0 is the spreading pressure of the pure component j. n_j^0 and P_j^0 are the adsorbed concentration and the pressure of the pure component, respectively. The pure component isotherm can take any appropriate form, such as the Toth equation, described above.

The relationship between the gas phase and the adsorbed phase at equilibrium is:

$$Py_j = P_j^0 x_j \quad \text{for all } j \tag{5}$$

which is an analog to Raoult's law in vapor-liquid equilibrium. Here P_j^0 is the hypothetical pressure of the pure component such that it will give the same spreading pressure as that of the mixture.

The mole fractions in the gas phase and in the adsorbed phase must satisfy, respectively:

$$\sum_{j=1}^{N} y_j = 1 \tag{6a}$$

and

$$\sum_{j=1}^{N} x_j = 1 \tag{6b}$$

For given gas phase conditions (i.e., the total pressure and the gas mole fraction) the spreading pressure z, the hypothetical pressure P_j^0 and the mole fractions of the adsorbed phase x_j can be calculated from Eqs. (4)–(6). Knowing these, the total amount adsorbed (n) can be calculated from:

$$\frac{1}{n} = \sum_{j=1}^{N} \frac{x_j}{n_j^0} \tag{7}$$

where n_j^0 is the adsorbed concentration of the pure component j evaluated at the hypothetical pressure P_j^0, according to the isotherm:

$$n_j^0 = f^0(P_j^0) \tag{8}$$

Finally the total amount adsorbed (n) can be related to the amount of component j adsorbed by:

$$n_j = x_j n \tag{9}$$

While for some binary systems these calculations are quite accurate, substantial deviations are experienced for other systems. It is obvious that the actual process is quite complex and is not yet fully clarified.

Yang et al. (1996) have used a simple neural network model (5-4-1) for the prediction of multisolute adsorption. Their neural network model was effective even for a very limited number of training data sets. They were able to predict both single solute and binary solute adsorption behaviour, even in some very non-ideal systems.

An important feature of neural networks is their ability to approximate arbitrarily complex relationships without detailed knowledge of the underlying process. Such a successful approximation normally requires sufficient and reliable data, the neural model being rather non-parsimonious otherwise. The objective of this present work is to demonstrate the ability of neural network (nn) modeling of ideal binary adsorption systems, as a first step an ongoing study of nn modeling of multicomponent non-ideal adsorption systems. The existence of a relationship between a single component and binary adsorption behaviour was expected and this was used to guide our approach in building up the nn models.

Artificial Neural Networks

The development of the first artificial neural networks was motivated by biological nerve systems which consist of densely connected networks of real neurons. An individual biological neuron has only a limited computation ability but interesting computational properties emerge when several neurons are combined together in various ways. This concept is used in artificial neural nets. An artificial neural network is typically a massively parallel interconnected network of "artificial neurons" (also called Processing Elements, Nodes or Units). The way in which these processing elements are mutually interconnected determines the network architecture. Over 50 different types of network architecture can be found in the literature (Morris et al., 1994; Bulsari, 1995). In this paper we have used the most common type of a neural network architecture, the feedforward network (often called backpropagation network or Multi-Layer Perceptron). In a feedforward network each processing element has several inputs and one output. The inputs are combined and then modified by the activation function and passed to the output of the processing element. A feedforward neural network is made up of layers of processing elements (see Fig. 1). The input layer acts as an input data holder which distributes inputs into the network. The data from the input layer are propagated through the network via the interconnections to processing elements in the first hidden layer where they are combined and modified by activation functions. The signals keep proceeding in this way from layer to layer until they reach the output layer. An example of the structure of a feedforward neural network with one input layer, one hidden and one output layer is illustrated in Fig. 2. This network would be sometimes called a (3-5-1) feedforward network, referring to the numbers of neurons in the input, hidden, and output layers, respectively.

A continuous multivariable function $F(\boldsymbol{x})$ is approximated in neural network by a selected function $f(\boldsymbol{x},\boldsymbol{w})$ for a fixed number of input variables

$$\boldsymbol{x} = (x_0; x_1, \ldots, x_I) \tag{10}$$

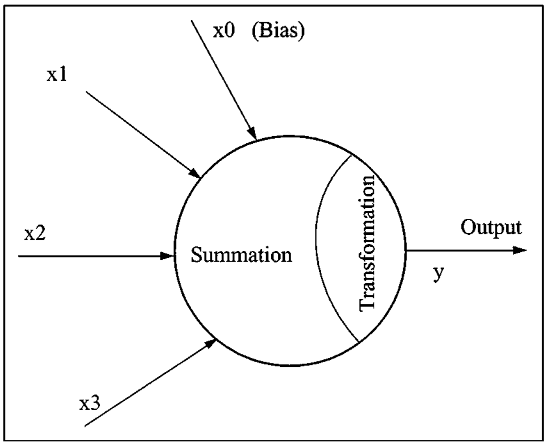

Figure 1. A processing element in the hidden layer of a neural network.

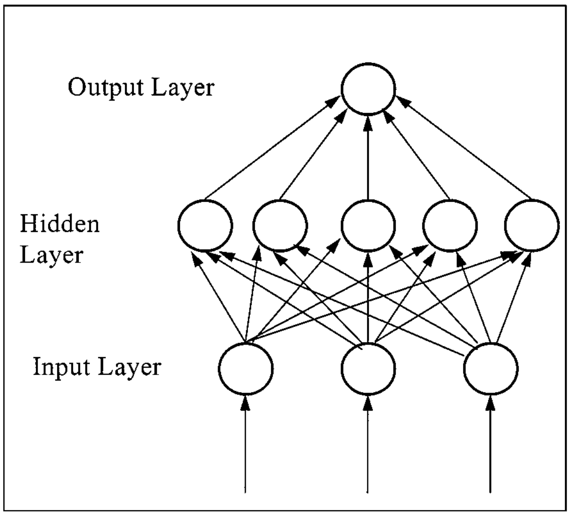

Figure 2. An example of a 3-5-1 feedforward neural network.

and $\boldsymbol{w}$ is an array of weights, defined below. $x_0 = 1$ is the constant input, called "Bias", that is used to simulate thresholding effects in the neuron and which also serves to simplify the mathematics; $x_i, i = 1, \ldots, l$ are neural network inputs, and l is the number of input nodes.

The output from the hidden layer is

$$\boldsymbol{y} = (y_0; y_1, \ldots, y_m) \tag{11}$$

where $y_0 = 1$ is the constant output from the bias neuron, m is the number of processing elements in the hidden layer, and y_j is an output from the jth processing element of the hidden layer.

$$y_j = f\left(\sum_{i=0}^{l} w_{ji}^0 x_i\right), \quad j = 1, \ldots, m \tag{12}$$

w_{ji}^0 is a weight associated with a connection between the ith processing element in the input layer and the jth processing element in the hidden layer. For bias the weight w_{j0}^0 is taken as equal to 1.

The formula for the output layer of the neural network is similar to that of Eq. (12), only the signal from the bias neuron does not exist:

$$\boldsymbol{z} = (z_1, z_2, \ldots, z_n) \tag{13}$$

where

$$z_k = f\left(\sum_{j=0}^{m} w_{kj}^1 y_j\right), \quad k = 1, \ldots, n \tag{14}$$

n is the number of output neurons, and w_{kj}^1 is weight associated with a connection between the jth processing element in the hidden layer and the kth processing element in the output layer. For bias the weight w_{k0}^1 is again taken as equal to 1.

The activation function $f(.)$ is mostly chosen to be a sigmoidal function $f(x) = 1/(1 + e^{-x})$ or a symmetric sigmoidal function $f(x) = \tanh(x)$. (Interestingly, sigmoidal nonlinearity was observed also in human neuron behaviour (Holden, 1976)). Within a neural network, these functions can represent nonlinear relationships. The activation function of processing elements in the input layer can either be omitted or made linear (Carsky and Hajek, 1995), but other transformations of input data are also possible (NeuralWare, 1995).

Neural Network Training

The purpose of developing a neural model is to devise a network (set of formulae) that captures the essential relationships in the data. These formulae are then applied to new sets of inputs to produce corresponding outputs. This is called generalisation, but before generalisation is possible a training set of data is needed in order to fit the model parameters. A number of papers have shown that a feedforward network can uniformly approximate any non-linear function. The problem of neural network learning, in effect, reduces to determining the set of parameters of function the $f(\boldsymbol{x},\boldsymbol{w})$

$$\boldsymbol{w} = \left(w_{ij}^{0}, w_{kj}^{1}\right), \tag{15}$$

(the terms on the right hand side being the weights of interconnections between processing elements in the input and hidden layer, and between the hidden and output layer, respectively), that provides the best approximation to multivariable function F, when trained on an "example" data set (Morris et al., 1994).

There are also many methods based on a technique called gradient back propagation. A multilayer neural network trained with the back-propagation algorithm can be viewed as a tool for performing a nonlinear input-output mapping of a general nature. The input-output relationship of such a network, in effect, defines a mapping from a one-dimensional Euclidean input space to a n-dimensional Euclidean output space (Carsky and Hajek, 1995).

Generally the neural network may have more than a single hidden layer, as discussed above. It has been proved, however (Cybenko, 1989), that an artificial neural network with a single hidden layer is sufficient to uniformly approximate any continuous function with arbitrary accuracy. This universal approximation theorem is an existence theorem and it does not say anything about how many processing elements should be used in the hidden layer and whether a single hidden layer is the optimum in terms of learning time and ease of implementation.

Neural network training leads to finding values of connection weights that minimise differences between network outputs $\boldsymbol{z} = (z_1, \ldots, z_n)$ and the target values $\boldsymbol{d} = (d_1, \ldots, d_n)$ specified by a teacher. An objective function is specified which is a measure of how closely the outputs of the network match the target values from the training set of data. The global error E is minimised by modifying the weights $\boldsymbol{w}$ by propagating the gradient of the objective function with a momentum term back through the network to improve the objective function:

$$\left(\Delta w_{ji}^{0}\right)_t = -\lambda \frac{\partial E}{\partial w_{ji}^{0}}\bigg|_{t-1} + m\left(\Delta w_{ji}^{0}\right)_{t-1} \tag{16a}$$

for a hidden layer

$$\left(\Delta w_{kj}^{1}\right)_t = -\lambda \frac{\partial E}{\partial w_{kj}^{1}}\bigg|_{t-1} + m\left(\Delta w_{kj}^{1}\right)_{t-1} \tag{16b}$$

for output layer,

where λ is a learning coefficient, m is a momentum coefficient, and t is an iteration counter. The momentum term acts as a low-pass filter on the incremental weights terms and usually increases the speed of learning.

The partial derivatives $\partial E/\partial w_{ji}^{0}\, \partial E/\partial w_{kj}^{1}$ are evaluated locally by back propagating the global error E backwards from the output to the input layer and thus attributing to each processing element a certain responsibility for the global error (Rumelhart and McClelland, 1986).

In network training the experimental data is split into training and validation data sets. A back-propagation learning algorithm should lead to a good fit to the training samples and, simultaneously, to a network that has a good generalisation capability. A network is said to generalise well when the input-output relationship, found by the network, is correct for input/output patterns of validation data which were never used in training the network. Network weights are updated in the process of fitting the training data and the network is tested on validation data after presenting the whole batch of training samples. All weights remain constant during testing, of course. The training termination criterion is evaluated on validation data examples that are not used in training. It is necessary to avoid overtraining (memorising the training samples) to achieve good generalisation.

Training is said to be *stable* if the input-output relationship represented by a network does not depend on the initial settings of the connection weights and on the way of splitting the experimental data into training and testing sets. It is therefore recommended to repeat training several times under different conditions.

Neural Network for Binary Adsorption

A neural network for adsorption of vapour binary mixtures on activated carbon should be able to predict the adsorption isotherms for the systems for which no experimental data are available. The neural network models the relationship between single component adsorption data (independent variables network inputs) and binary adsorption isotherms (a dependent variable, network output).

A neural network package "*Predict*" was used for this simulation (NeuralWare, 1995). Predict uses one hidden layer only. Otherwise, the network architecture is not fully determined in advance. The number of nodes in the output layer is given by the number of model output variables. For a hidden layer, *Predict* uses a method called “Cascade Learning” (Fahlmann and Lebiere, 1988) to determine a suitable number of hidden nodes. A chosen number of processing elements (usually 1–3) is added at a time. Construction is stopped when performance on an independent test set shows no further improvement. The symmetric sigmoidal function Tanh is used as activation function for hidden layer nodes while the sigmoid function is used for output nodes. The number of input nodes may correspond to the number of input variables chosen. Or, alternatively, *Predict* can employ its genetic algorithm to analyse input variables and to find the most advantageous set of them for neural network modeling. It is possible to find variables that have the most significant effect on model outputs, and to disregard input variables that show small or negligible effect on the process under consideration. The input variables can be chosen either manually, or by the program. They also can be transformed (e.g., linear, logarithmic, inverse function transformation) into forms suitable for network training. (Tukey, 1977; Chatterjee and Price, 1991).

The adsorption systems used for simulation were the following vapour mixtures: Butane-Propane, Ethane-Ethylene, Ethane-Methane, Ethylene-Carbon Dioxide, Ethylene-Methane, Propane-Ethane, Propylene-Ethylene and Propylene-Propane on activated carbon (type: Nuxit-AL, cylinders 1.5 mm in diameter, 3–4 mm long), at 20°C. The experimental data for these systems was taken from Szepesy and Illes (1963). A training set of experimental data was used to train the neural network as discussed above. Simultaneously as a part of training, and to avoid overfitting 30% of the training data was randomly chosen (test set) and was left aside to check the ability of the neural net model to generalise. As an objective function for neural net training, the root mean square error was used:

$$E = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\frac{1}{n}\sum_{k=1}^{n}\left(d_k^{(r)} - z_k^{(r)}\right)^2} \qquad (17)$$

As the objective function for testing the generalisation ability of a neural network, the correlation coefficient defined by Eq. (18) was used:

$$C = \frac{\sum_{r=1}^{R}\sum_{k=1}^{n}\left(d_k^{(r)} - \bar{d}_k\right)\left(z_k^{(r)} - \bar{z}_k\right)}{\sqrt{\left\{\sum_{r=1}^{R}\sum_{k=1}^{n}\left(d_k^{(r)} - \bar{d}_k\right)^2\right\}\left\{\sum_{r=1}^{R}\sum_{k=1}^{n}\left(z_k^{(r)} - \bar{z}_k\right)^2\right\}}} \qquad (18)$$

where r is record index across the test set, ($r = 1, \ldots, R$), n number of output nodes, $z_k^{(r)}$ is network output for test record r at output node k, $d_k^{(r)}$ is target output for test record r at output node k, $\bar{z}_k$ and $\bar{d}_k$ are the average values of $z_k^{(r)}$ and $d_k^{(r)}$ values across the test set.

Simulation Results

To provide as large a training set as possible binary systems with more experimental data points (i.e., Butane-Propane, Ethane-Ethylene, Ethane-Methane, Ethylene-Methane, Propane-Ethane and Propylene-Propane) were used for the neural network learning. As discussed above 30% of randomly chosen training data was put aside as a test set. After the training was completed process (including a test for a generalisation ability) the neural network could be used for prediction of the adsorption isotherm of the two remaining binary system with a smaller number of experimental data points, namely Ethylene-Carbon Dioxide and Propylene-Ethylene.

To start the training the following variables were chosen as inputs:

- Pressure,
- Vapour phase concentration,
- mmols of pure components (1 and 2) adsorbed per 1 g of adsorbent (for a single component adsorption),
- Constants of the Toth equation (both components),
- Critical pressure (both components),
- Critical temperature (both components),
- Critical volume (both components),
- Molar mass (both components).

The outputs were the total number of mmols of a binary mixture adsorbed on 1 g of adsorbent, and the concentration of components in the adsorbed mixture.

The variable selection procedure of *Predict* gradually rejected pressure, critical pressures, temperatures, volumes, and molar masses of both components as input variables. Afterwards, three neural models were tested with fixed inputs using variables not rejected previously by the program’s variable selection procedure:

1. Model A: y_1 (transformation: inverse function
   b_1, m_1, b_2, m_2 (all linear transformation)
2. Model B: y_1 (logarithmic transformation)
   n_1, n_2 (linear transformation)

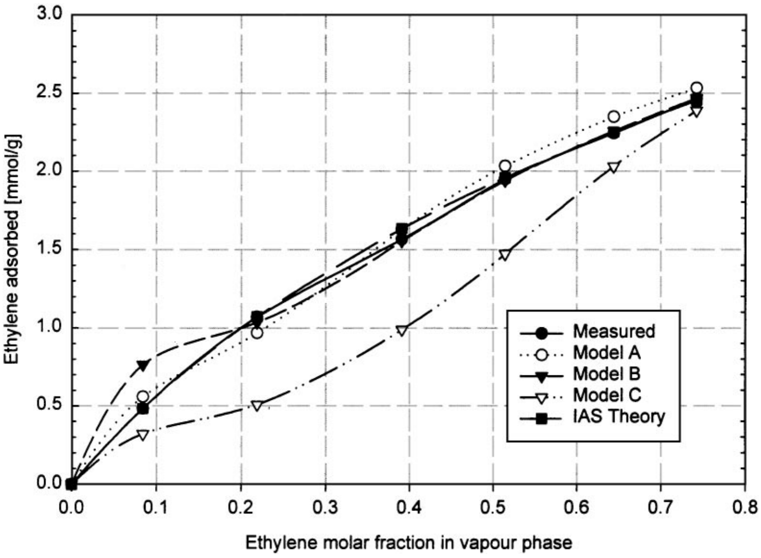

Figure 3. Adsorption of ethylene on activated carbon from ethylene-carbon dioxide vapour mixture at 293 K.

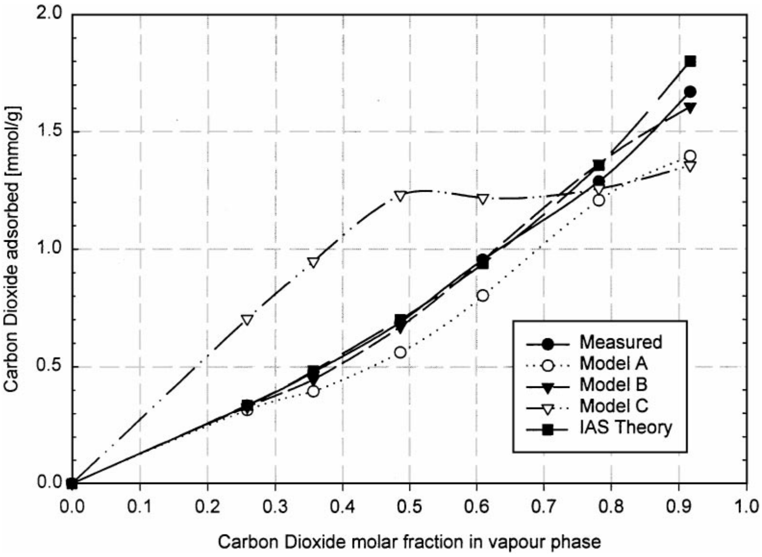

Figure 4. Adsorption of carbon dioxide on activated carbon from ethylene-carbon dioxide vapour mixture at 293 K.

3. Model C: y_1 (transformation: inverse function)
$b_1, t_1, m_1, b_2, t_2, m_2$ (all linear transformation)

where y_1 is molar fraction of component 1 in a binary vapour mixture, b_i, t_i, and m_i are constants of Toth equation (for component i), and n_i is number of mmols of component i adsorbed per 1 g of adsorbent.

The neural network is capable of fitting the experimental data fairly well as can be seen from Figs. 3–8. Splitting the training data as discussed above prevented the network from "memorising" experimental points and forced it to find a reasonable compromise between data fitting and generalisation. A comparison of the three neural network models and the ideal adsorption solution theory with the experimental data in terms of the root mean square errors and the average relative errors is given in Tables 1 and 2. It may be clearly seen from this comparison that neural network models provide prediction with the errors of the order of

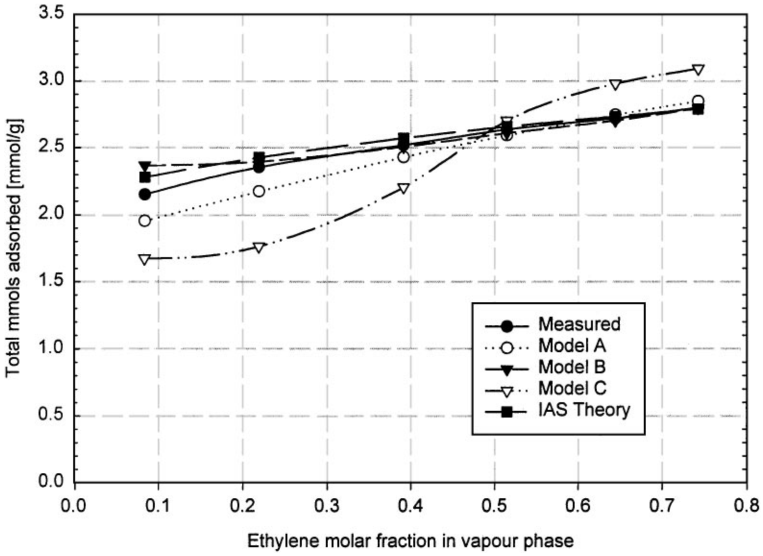

Figure 5. Adsorption of ethylene-carbon dioxide vapour mixture on activated carbon at 293 K.

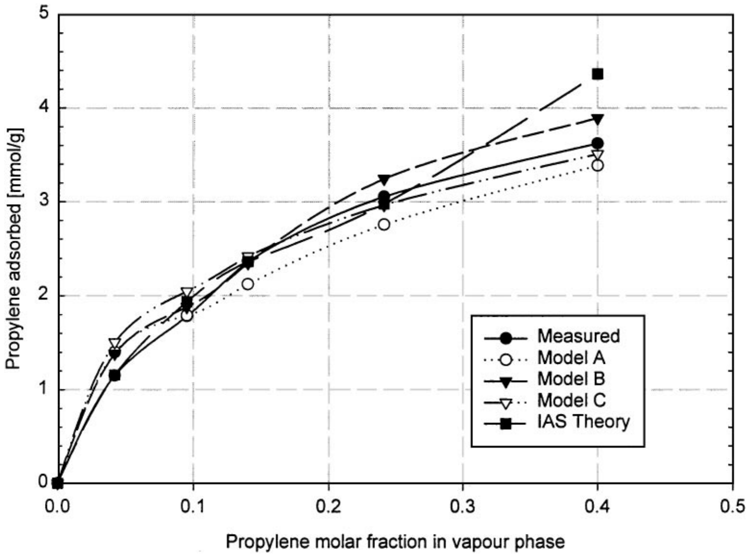

Figure 6. Adsorption of propylene on activated carbon from propylene-ethylene vapour mixture at 293 K.

magnitude as the ISA theory. If we compare predictions from among the three neural network models then the models A (5 inputs) and B (3 inputs) provided the best agreement between the predicted and measured binary adsorption isotherms. Model C with 7 inputs was created from the model A by inclusion of the constant t of Toth equation for both components. Model C, however, is in most cases worse in predicting the binary adsorption behaviour than the simpler models A and B. The input variable t in the model C was almost constant, varying between 0.34 and 0.48 only (with the exception of CO_2 when $t = 0.63$). As such t will not contribute to any improvement in model predictions. Also, with seven inputs this model is quite complicated for the relatively limited number of modeling data used in model building (ca. 60).

The ideal adsorption solution theory provided values very close to those measured experimentally. This due

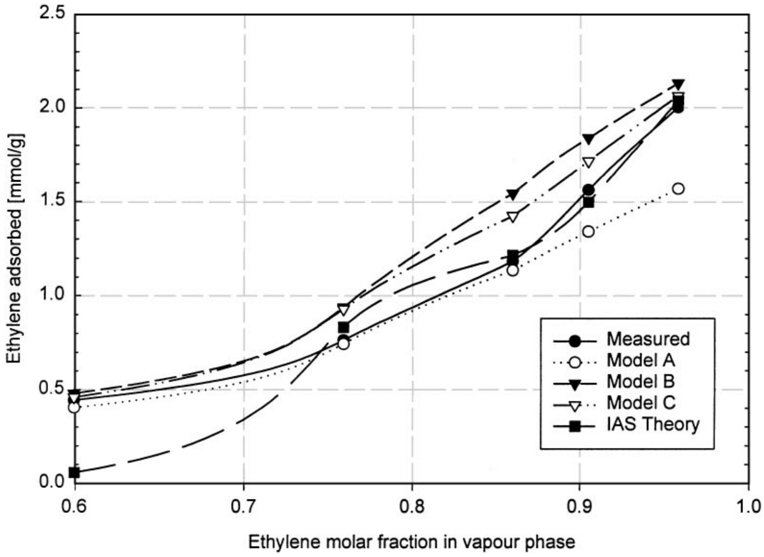

Figure 7. Adsorption of ethylene on activated carbon from propylene-ethylene vapour mixture at 293 K.

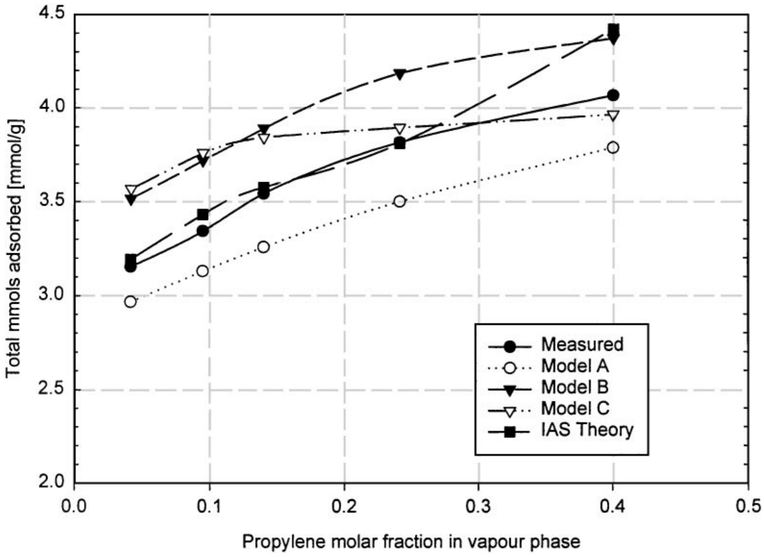

Figure 8. Adsorption of propylene-ethylene vapour mixture on activated carbon at 293 K.

to the fact that the systems with activated carbon as an adsorbent, together with the binary vapour mixtures can be considered as ideal in the sense of the ideal adsorption solution theory.

The important feature of neural networks is their ability to approximate arbitrarily complex relationships without detailed knowledge of the underlying process, all that is needed is sufficient and reliable data, of course. The use of nn modeling is thus not limited to "ideal systems" (in the sense of the Ideal Adsorption Solution model). The IAS theory, on the contrary, is often failing for other than ideal systems. But equally it must be clearly stated that a neural network does not contain any information about the mechanism that governs the simulated process, so it cannot provide any reasonable extrapolation beyond the range of training data. Moreover, a sigmoidal shape of activation function limits the outputs of processing elements, thus

Table 1. Root mean square errors of neural network models and ISA theory.

	Root mean square error in prediction (mmol/g adsorbed)					
	System ethylene (1)–CO_2 (2)			System Propylene (1)–Ethylene (2)		
Model	Comp. 1	Comp. 2	Mixture	Comp. 1	Comp. 2	Mixture
nn A	0.0851	0.1459	0.1188	0.2290	0.2205	0.2600
nn B	0.1162	0.0432	0.0921	0.1864	0.2266	0.3532
nn C	0.3988	0.3706	0.3724	0.2087	0.1508	0.3006
IAS theory	0.0289	0.0613	0.0658	0.3400	0.1789	0.1652

Root mean square error $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{i,\text{ model}} - x_{i,\text{ experimental}})^2}$.

Table 2. Average relative errors of neural network models and ISA theory.

	Average relative error in prediction (%)					
	System ethylene (1)–CO_2 (2)			System propylene (1)–Ethylene (2)		
Model	Comp. 1	Comp. 2	Mixture	Comp. 1	Comp. 2	Mixture
nn A	5.2	16.2	4.7	9.6	18.5	7.3
nn B	7.1	4.8	3.6	7.8	19	9.9
nn C	24.5	41.1	14.7	8.7	12.7	8.4
IAS theory	1.8	6.8	2.6	14.2	15	4.6

Average relative error $= \frac{\text{Root mean square error}}{\text{Mean experimental value}} \times 100$.

keeping output variables within the intervals into which they were scaled.

Conclusions

A neural network with 1 input layer, 1 hidden layer and 1 output layer was capable of “learning” the relationship between single component and binary adsorption data very well, even for the relatively limited number of data used in model building (ca. 60). The network was trained with the goal to achieve a reasonable compromise between data fitting and generalisation. Models A and B provided the best agreement between the predicted and measured binary adsorption isotherms. Model C has shown itself in most cases to be worse in prediction the binary adsorption behaviour than the simpler models A and B.

Because the systems chosen (with an activated carbon as an adsorbent) can be regarded as ideal in the sense of the ideal adsorption theory, the predictions by ISA theory were close to those measured experimentally. The neural network models A and B provided predictions of comparable precision to the IAS theory.

The important feature of neural networks is their ability to approximate arbitrarily complex relationships without detailed knowledge of the underlying process. Unlike the Ideal Adsorption Solution model the use of nn modeling is not limited to “ideal systems”. On the contrary IAS theory often fails for other than ideal systems.

The neural network does not contain any information about the mechanism that governs the simulated process and any extrapolation beyond the range of training data must be taken with caution. On the other hand the neural network may identify which variables may be significant for the process.

In conclusion the paper has demonstrated that nn modeling can be successfully applied to ideal binary adsorption data and that the predictions are virtually as good as the IAS theory. The work should be regarded as a first step only in the effort of nn modeling of multicomponent non-ideal adsorption systems.

Nomenclature

b, m, t	constants of Toth equation (Eq. (1))	—
C	correlation coefficient defined in Eq. (18)	—
$\boldsymbol{d}$	array of target values	—
E	mean root square error defined in Eq. (17)	—
m	momentum coefficient in Eq. (16a) and (16b)	—
n	number of moles adsorbed per 1 g of adsorbent	mol/kg, or mmol/g
n_j^0	number of moles of the pure component j adsorbed per 1 g of adsorbent	mol/kg, or mmol/g
P	pressure	kPa
P_j^0	pressure of the pure component j	kPa
R	universal gas constant	8.3145 KJ/ (kmol K)
T	temperature	K
$\boldsymbol{w}$	array of weights of neural network interconnections	—
$\boldsymbol{x}$	array of inputs to neural network	—
x_0, y_0	bias	—
x_i, x_j	mole fraction of the adsorbed component i, j, respectively	—
$\boldsymbol{y}$	array of hidden layer outputs	—
y_i	molar fraction of component i in binary vapour mixture	—
$\boldsymbol{z}$	array of neural network outputs	—
z	spreading pressure of the mixture	kPa
z_j^0	spreading pressure of the pure component j	kPa

Greek Letters

λ	learning coefficient in Eqs. (16a) and (16b)

References

Bansal, R.C., J.B. Donnet, and F. Stoeckli, *Active Carbon*, Marcel Dekker, New York, 1988.

Bulsari, A.B., *Neural Network for Chemical Engineers*, Elsevier, Amsterdam, 1995.

Carsky, M. and M. Hajek, “Extraction of Alginates from Kelp. Physical versus Neural Network Modelling,” *South African Journal of Chemical Engineering*, **7**, 12–24 (1995).

Chatterjee, S. and B. Price, *Regression Analysis by Example*, John Wiley & Sons, 1991.

Cybenko, G., “Approximation by Superpositions of a Sigmoidal Function,” *Mathematics of Control, Signals, and Systems*, **2**, 303–314 (1989).

Fahlmann, S.E. and C. Lebiere, “The Cascade-Correlation Learning Architecture,” *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann, 1988.

Holden, A.V., “Models of the Stochastic Activity of Neurons,” *Lecture Notes in Biomathematics*, Vol. 12, Springer-Verlag, 1976.

Hyun, S.H. and R.P. Danner, *J. Chem. Eng. Data*, **27**, 196 (1982).

Morris, A.J., G.A. Montague, and M.J. Willis, “Artificial Neural Networks: Studies in Process Modelling and Control,” *Trans. I. Chem. Eng.*, **72**(A), 3–19 (1994).

Nakahara, T., M. Hirata, and J. Komatsu, *J. Chem. Eng. Data*, **26**, 161 (1981).

NeuralWare. Inc., *NeuralWorks Predict, Complete Solution for Neural Data Modelling*, Pittsburgh, 1995.

Rumelhart, D.E. and J.L. McClelland, “Parallel Distributed Processing: Explorations in the Microstructure of Cognition,” Vol. 1, MIT Press, Cambridge, MA, 1986.

Szepesy, L. and V. Illes, *Acta Chim. Hung.*, **35**, 245 (1963).

Talu, O. and I. Zwiebel, *AIChE Journal*, **32**, 1263 (1986).

Tukey, J.W., *Exploratory Data Analysis*, Addison-Wesley, 1977.

Valenzuela, D.P. and A.L. Myers, *Adsorption Equilibrium Data Handbook*, Prentice Hall, 1989.

Yang, M., J. Hubble, A.D. Lockett, and R.R. Rathbone, “A Neural Network Model for Prediction of Binary Adsorption Using Single Solute and Limited Binary Solute Adsorption Data,” *Separation Science and Technology*, **31**, 1259–1265 (1996).