Prediction of Water Activity Coefficient in TEG-Water System Using Diffusion Neural Network (DNN)

H. Karimi,[a,*] N. Saghatoleslami,[b] and M. R. Rahimi[a]
[a]Chemical Engineering Department, Yasouj University,
Yasouj 75914-353, Iran
[b]Department of Chemical Engineering,
Ferdowsi University of Mashhad, Mashhad, Iran

Original scientific paper
Received: November 26, 2008
Accepted: September 7, 2009

Accurate determination of activity coefficients of water in a binary triethylene glycol (TEG)-water system, is of prime concern in designing the natural gas dehydration process. In this work, a hybrid model (a combination of information diffusion theory and neural network) and a so-called diffusion neural network (DNN) have been developed for the prediction of activity coefficients of a binary TEG-water system. Owing to the insufficient experimental data available in the literature for binary mixtures, and in particular for infinite dilution, we have employed the information diffusion technique as a tool in extrapolating data points from the original data. By means of this technique, a new dataset has been trained and optimized for the DNN model with more nodes in the input and the output layers. The result of this study reveals that DNN model is superior to the conventional neural nets in predicting the activity coefficient of water in the range of temperature (293–387.6 K) and mole fractions with mean absolute error of 0.31 % (MAE = 0.31 %), and high correlation coefficient of 0.999 (r = 0.999). Furthermore, the results of this work using DNN have also been compared with Parrish's correlation. The findings of this work demonstrate that the DNN model exhibits better results over Parrish's correlation in predicting the activity coefficients of water in a binary triethylene glycol-water system with a mean absolute error of 5.03 percent (MAE = 5.03 %).

Key words:
Triethylene glycol, water, diffusion neural network, activity coefficient, gas dehydration

Introduction

The dehydration of natural gas is an important operation in the gas processing industry. A number of liquids possess the ability of absorbing water from the gas stream. Owing to the fact that triethylene glycol (TEG) has a high affinity for water, it is extensively used in gas dehydration operation process. Triethylene glycol has the property of not solidifying in a concentrated solution; is non-corrosive in nature and is easily regenerated. Knowledge of liquid-liquid and vapor-liquid equilibrium data is the pre-requisite for the design of such processes. These experimental data may be utilized as a database to optimize the design of TEG dehydration units or extractive distillation and solvent recovery columns in the separation processes of aromatics. Generally speaking, a good mechanistic or empirical model is required for the design of such processes. In industrial circumstances, it is generally very expensive to develop a comprehensive model for the case. However, it is customary to re-establish an experimental design procedure to derive a simple empirical model for it, where it can provide maximum information with the lowest number of experimental runs. However, the conventional approach (i.e. EOS) for modeling of the vapour-liquid equilibrium requires an iterative method that may sometimes pose some problem for real time control of the plant operation. In such cases, other more robust and faster alternative methods would be of interest. Artificial neural networks have received extensive attention during the last two decades. It is well known that they can solve many practical problems such as pattern recognition,[1] function approximation,[2] system identification,[3] time series forecasting. Nowadays, the artificial neural networks (ANNs) method has found extensive application in the field of thermodynamics and transport properties such as estimation of VLE,[4–10] viscosity,[11–12] density,[13] simultaneous determination of concentration,[14] compressibility factor[13] and thermal conductivity.[15]

Neural information processing models principally assume that the data are compatible and the learning data for training a neural network are sufficient. For cases where the data are insufficient, it is impossible to recognize a nonlinear system. In other words there must exist a non-negligible error

[*]Corresponding author: hakar@mail.yu.ac.ir; Tel/fax: + 98-741-2221711

between the real function and the estimated function from a trained network.[16] Moreover, using basic neural nets with small data points (experimental points), it is difficult to guarantee that a good predictive model will be obtained in complete experimental domain. In more cases only good predictions are achieved in regions in the vicinity of experimental points contained in the training dataset.[17]

In this paper, we utilized a hybrid model based upon the technique of information diffusion and neural network to reduce the error of estimated function with respect to a small experimental dataset. The model was evaluated by analysis of the results obtained from test data, and the characteristic performance of proposed model was compared with conventional neural networks and the Parrish *et al.* correlation.[18]

Conventional neural networks

Neural networks provides a non-linear function mapping of a set of input variables into the corresponding network output, without having to specify the actual mathematics form of the relation between the input and output variables.

The multilayer percepetron (MLP) and radial base function (RBF) are two types of feedforward neural network that have been used commonly for the approximate functions.[19]

Each network is discussed in detail in the literature, only a few prominent features of each are given here to indicate the general nature of the networks and some of their main differences.

Both types of network comprise of a set of processing units arranged in layers. The two networks, however, differ markedly. For example, the RBF contains a single hidden layer of units that employ a considerably different transformation function to that used in the MLP which may also have more than one hidden layer. With both the MLP and RBF, the training data are input directly to the network and used to tune it into an accurate training algorithm. For example, with the MLP, each training data contributes directly to the determination of the error in the network's iterative training stage. However, the two networks partition feature space in a very different manner. The MLP divides the entire feature space with hyperplanes, while the RBF defines hyperspheres, and so the two networks may give different allocations and vary in sensitivity to properties of the data used. The derivation of the outputs, however, differs considerably between the two types of network. For example, with the MLP, assuming the use of the popular sigmoidal activation function, the activation level of the s_{th} unit is determined from:

$$O_s = 1/(1 + \exp(-\lambda \, \mathrm{net}_s)) \tag{1}$$

where net_s and λ are the net input to the unit and gain parameter respectively. Alternatively, with the RBF, the output of the network is derived from:

$$O_s = \sum_{j=1}^{M} w_{sj} \phi_j + w_{so} \tag{2}$$

where M is the number of basis functions, w_{sj} a weighted connection between the basis function and output layer, and ϕ_j the nonlinear function of unit j, which is typically a Gaussian of the form:

$$\phi_j = \exp\left[\frac{(P - W_j)(P - W_j)}{2\sigma_j^2}\right] \tag{3}$$

In which P is the input vector; W_j is the weight vector associated with the hidden unit j and σ_j is the parameter specifying the radial basis function's width.

Diffusion neural network theory

In the real world, the experimental data are strongly scattered and incompatible (contradictory). However, neural information processing models largely assume that the learning data for training a neural network are compatible and sufficient.[20] If the data are incompatible then a neural network would not converge in view of the fact that the adjustments of weights and bias are unable to identify where they should switch due to the uncertainty brought forth by the contradictory patterns. That is, when input patterns are the same but output patterns are very different, the directions of change in weights and bias are uncertain when we use these datasets as patterns to train a neural network. Hence, to estimate the relationship, we need to handle the fuzziness and roughness of the dataset. To do this, an information diffusion model was introduced to process the dataset before we use them in the neural networks.

It was shown that a combination of information diffusion method and neural-network can be adopted in cases where incompatible data exist.[20–21] In circumstances where the learning data are insufficient, in other words when the sample size n is small, we say that the sample consists of incomplete information with a lot of gaps in the domains of the variables, it is impossible to recognize a nonlinear system (i.e., there must exist a non-negligible error between the real function and the estimated function from a trained network). Thus, the theory of information diffusion can be employed to extrapolate more data to partly fill in the gaps in small datasets. Therefore, we would be able to obtain a better result by employing a neural network technique to recognize a non-linear function. The hybrid model incorporates the developed model and hence a corresponding neural network is derived, which is called a diffusion-neural network.[16] The diffusion-neural network can be considered a primary model based upon the theory of information diffusion in order to improve accuracy of artificial neural networks with respect to a small experimental dataset.

The principle of information diffusion

Using information diffusion is based upon the theory of fuzzy set[22] that a dataset can be changed into a fuzzy subset which naturally fills up the information gaps caused by incomplete data. The integration of the relationships thus produced can change the contradictory patterns into more compatible ones which can smoothly and quickly train the neural network to obtain the desired relationship.

This technique guarantees the existence of reasonable diffusion functions to improve the non-diffusion estimates when the given dataset are incomplete. However, the rule does not provide any indication on how to find the diffusion functions.

Let $X = \{s_1, s_2, \dots s_n\}$ be a given dataset with an input of p and output of t (i.e., $s_i = (p_i, t_i)$), where it can be employed to estimate a relationship for R on the universe U. If and only if X is incomplete, there must exist a diffusion function $\mu(s_i, u)$ and a corresponding operator η that leads to diffusion estimate $\widetilde{R}(\eta, D(X))$ so that it is closer to the real R than any non-diffusion estimate. According to FS:

$$\widetilde{R}(\eta, D(X)) = \{\eta(\mu(s_i, u)| s_i \in X, u \in U\} \tag{4}$$

The suggested normal diffusion function[16] μ is given:

$$\mu(s,u) = \frac{1}{h\sqrt{2\pi}} \exp\left[-\frac{(s-u)^2}{2h^2}\right] \tag{5}$$

where h is diffusion coefficient calculated from following suggested formula:

$$h = \begin{cases} 0.6841(b-a) & \text{for } n = 5 \\ 0.5404(b-a) & \text{for } n = 6 \\ 0.44821(b-a) & \text{for } n = 7 \\ 0.3839(b-a) & \text{for } n = 8 \\ 2.6851(b-a)/(n-1) & \text{for } n \geq 9 \end{cases} \tag{6}$$

$$b = \max_{1 \leq i \leq n} \{s_i\}, \qquad a = \min_{1 \leq i \leq n} \{s_i\}$$

Deriving data by information diffusion

The simplest model which is based upon the principle of information diffusion for deriving data points is proposed by Huang[23] (1998). This model can be employed to derive 10–50 points through a pseudo-random generator controlled by possibility density function (PDF).

In practice, developing the above model for a two-dimensional case in deriving data points from $s_i = (p_i, t_i)$ is not suitable for training a network. The reason is that the deriving data may be incompatible and hence the neural network would not converge. Therefore, the more derivative data points there are, the more difficult the training. Therefore, to avoid the convergence problem, Huang and Moraga, 2004 proposed a combined model based on the membership function of a fuzzy set and the possibility function to derive data points from given data. According to this model, the normal diffusion eq. (5) derives the following possibility distribution:

$$\pi(u) = \frac{1}{h\sqrt{2\pi}} \exp\left[-\frac{(s-u)^2}{2h^2}\right] \tag{7}$$

The nearer the u is to s, the larger the possibility to be equal to s. The largest possibility is $1/h\sqrt{2\pi}$. The corresponding normalized distribution is:

$$\pi(u) = \exp\left[-\frac{(s-u)^2}{2h^2}\right] \tag{8}$$

where the largest possibility is 1.

In this technique and in order to study the relationship between input-output data, a linear correlation coefficient r which measures the strength of the relationship between the paired p and t values in a dataset, is defined. In other words r provides some information to assign the distance from given pattern to derived patterns. In the case where $r = 1$, the estimation function must be a line. In other words, for a given dataset X with $r = 1$ it is unnecessary to derive any new data points. Therefore, when $r = 1$ it assigns a value of 0 to the distance between a given point s and its derivative point u. On the other hand and in the case where $r \neq 1$, it assigns $\psi(r)$ (is mapping to represent the change in r) to a distance between a given point s and its derivative point u. Thus, we can avoid the convergence problem. The r (i.e. $r = r_{pt}$) can be computed by:

$$r_{pr} = \frac{l_{pt}}{\sqrt{l_{pp} l_n}} \tag{9}$$

where

$$l_{pp} = \sum_{i=1}^{n} (p_i - \bar{p})^2 \qquad l_{tt} = \sum_{i=1}^{n} (t_i - \bar{t})^2$$

$$l_{pt} = \sum_{i=1}^{n} (p_i - \bar{p})(t_i - \bar{t})$$

To find u from s as a derivative point the eq. (8) is rearranged as follow:

$$\pi(u) = \exp\left[-\frac{(s-u)^2}{2h^2}\right] = \Psi(r) \qquad (10)$$

where r is characterized by:

$$r = 0.9 + m \cdot 10^{-2} \qquad m = 1,2,3,\ldots 9 \qquad (11)$$

and $\Psi(r)$ is defined as:

$$\begin{aligned} &\psi : r \mapsto \text{poss} \qquad \psi(1) \mapsto 1, \\ &\psi(r) \mapsto \underbrace{0.9\ldots9}_{m9s} \qquad \forall r \in \{0.91, 0.92, \ldots, 0.99\} \end{aligned} \qquad (12)$$

The correlation of u in terms of (p_i, t_i) is obtained by solving eq. (10):

$$u = s \pm \sqrt{-2h^2 \ln \psi(r)} \qquad (13)$$

Hence, from p_i we can derive three points as follows:

$$\begin{aligned} p_i' &= p_i + \sqrt{-2h_p^2 \ln \psi(r)} && \text{with poss} = \psi(r) \\ p_i & && \text{with poss} = 1 \quad (14) \\ p_i'' &= p_i + \sqrt{-2h_p^2 \ln \psi(r)} && \text{with poss} = \psi(r) \end{aligned}$$

Moreover from t_i we have

$$\begin{aligned} t_i' &= t_i + \sqrt{-2h_t^2 \ln \psi(r)} && \text{with poss} = \psi(r) \\ t_i & && \text{with poss} = 1 \quad (15) \\ t_i'' &= t_i + \sqrt{-2h_t^2 \ln \psi(r)} && \text{with poss} = \psi(r) \end{aligned}$$

Note that from one pair (p_i, t_i) we could obtain three pairs as follows:

$$\begin{aligned} &(p_i', t_i') && \text{with poss} = \psi(r) \\ &(p_i, t_i) && \text{with poss} = 1 \\ &(p_i'', t_i'') && \text{with poss} = \psi(r) \end{aligned}$$

The derivative data points can be employed to train a neural network with more input-output data. The typical architecture of a net is shown in Fig. 1. It is also pointed out that besides the original input-output variables; their possibilities also are counted as variables. In the input layer, one node is for the variable p and the other for its possibility {poss p}. In output layer, one node is for the variable t and the other for its possibility {poss t}. We can consider p_0 as (p_0, 1) being the input of the trained DNN. However, if the output of the DNN is (t_0 ,d) and d is near 1, we say that the estimated value is t_0. In this way, we can obtain a new estimated function from a given dataset of X.

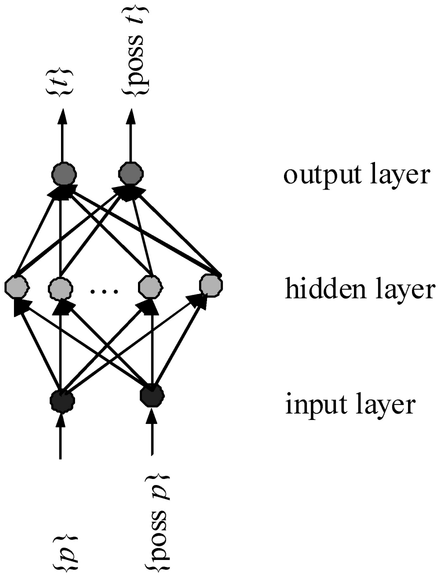


Fig. 1 – *Architecture of diffusion neural network (DNN). {p} and {poss p}, are as input data. {t] and {poss t} are as output data.*

Methodology

Experimental data

To carry out this task, the first step for this model was the development of an experimental database to train the network and hence to evaluate its ability for generalization. Numerous studies have been conducted in the field of the equilibrium of water in the TEG-water in finite concentration and infinite dilution regions system.[18,24–27] The equilibrium data comprises the mole fraction of TEG and water, temperature and activity coefficient. Herskowitz and Gottlieb,[25] (1984) measured the activity coefficients of water in TEG at two temperatures, 297.60 and 332.60 K, respectively. The lowest mole fraction of water for which activities were measured was 0.1938 and 0.2961 at 297.60 and 332.60 K, respectively.

It is worth noting that the infinite dilution region was not considered in their measurements. Parrish *et al.*[18] (1986) measured activity coefficients at infinite dilution zone for TEG-water binary mixture as a function of temperature in the range of

300–378 K. Bestani and Shing[26] (1989) also measured activity coefficients of water in TEG at infinite dilution in the range 323–383 K. Their data had a deviation of 13–17 % from those reported by Parrish *et al.*[18] (1986). Their findings were based upon the Bestani extrapolation at higher temperature (i.e., 477.15 K), indicating that the activity coefficient was above one. It is founded that their results inadequately defined the ability of TEG as a dehydrating agent in practice[28] (GPA, 1994). Therefore, Bestani data was not utilized in this work. However, as the number of experimental data points available from literature is limited, 54 data points in finite concentration were taken from both Herskowitz and Gottlieb,[25] 1984, Parrish,[18] 1986 and Campbell,[29] 1994 and 9 data points in infinite dilution region of water-TEG system.

Artificial neural network modeling

A hybrid model that was employed in this work embraces both the information diffusion technique and neural network, so-called diffusion neural network (DNN). The modeling was carried out by utilizing the conventional neural network toolbox developed in MATLAB version 7. Diffusion neural network had an input layer of four neurons corresponding to TEG mole fraction (x), temperature (T) and their possibilities (poss x, poss T), plus a variable number of neurons in the hidden layer and two neurons in the output layer corresponding to the water activity coefficient (γ) and its possibility (poss γ), as is shown in Fig. 2. The input and output layer nodes had a linear transfer function while only the hidden layer nodes had sigmoid transfer function for the DNN model. The network was trained according to Levenberg-Marquardt algorithm available in the neural network toolbox of MATLAB.

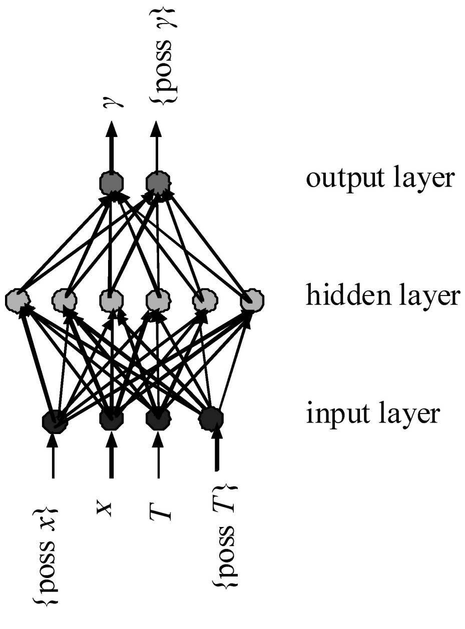


Fig. 2 – *Diffusion neural network (DNN) used in the study where x, T, poss x and poss T are liquid mole fraction TEG, temperature, possibility x and possibility T respectively, as input data. γ and poss γ are activity coefficients of water and possibility γ, respectively as output data.*

Parrish correlation

We studied the Parrish model as a typical representative of empirical correlation. Parrish *et al.*, 1986 proposed an empirical hyperbolic equation to predict the activity coefficients of TEG and water over a range of temperatures and mole fractions, as exhibited in eqs. (16) and (17):

$$\ln\gamma_1 = \frac{B^2 \ln[\cosh(\tau)]}{A} - \frac{x_2 B(\tanh(\tau))}{x_1} - Cx_2^2 \quad (16)$$

$$\ln\gamma_2 = B(\tanh(\tau) - 1] - Cx_1^2 \quad (17)$$

where $\tau = \dfrac{Ax_2}{Bx_1}$

In the above equations indices 1 and 2 represents TEG and water components, respectively and A, B, and C are temperature-dependent parameters and are defined as follows:

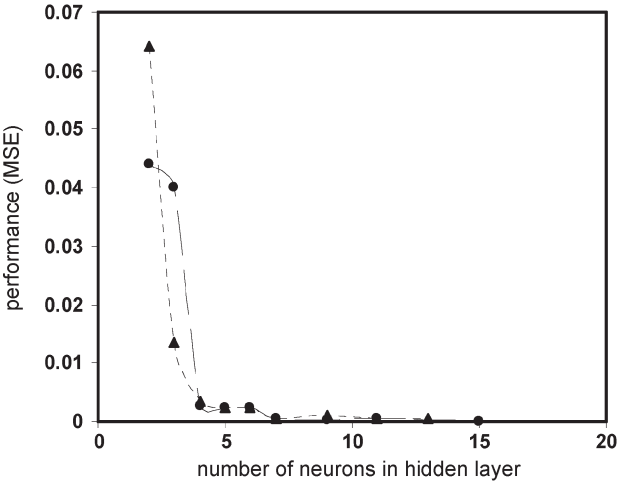
$$\begin{aligned} A &= \exp(-12.792 + 0.03293\ T) \\ B &= \exp(0.77377 - 0.000695\ T) \quad (18) \\ C &= 0.88874 - 0.001915\ T \end{aligned}$$

where T is the temperature in K.

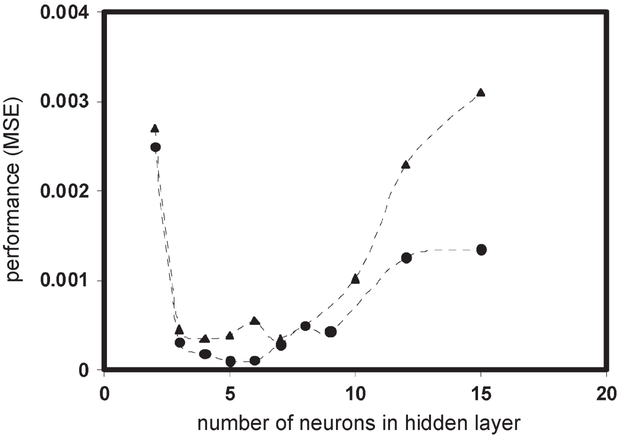
Results and discussion

Implementing the random selection method, 80 % of all the data were assigned to the training set of models, while the rest of the data were utilized to test the model. The raw input data was required to be pre-processed to convert them into a suitable structure. Therefore, in the first step all data points were scaled to the range of [–1, 1]. For the training step, the number of neurons in the hidden layer had an important role in the net optimization. Therefore, in order to optimize the net, achieve generalization of the model and avoid over-fitting, we started with 2 neurons in the hidden layer and gradually increased the number of neurons until no significant improvement in performance of the net was observed. For this study, the mean square error (MSE) was chosen as a measure of the performance of the nets.

At first, a conventional neural network (CNN) such as multilayer percepetron (MLP) and radial base function (RBF) neural networks were trained and optimized. For the case of the multilayer percepetron: to determine the optimal net size, the training was carried out with various numbers of neurons (1–10 neurons) in hidden layer. The MSE (mean square error) on network training performance is plotted vs. the number of hidden neurons and the results were demonstrated in Fig. 3. As it can be seen, the network with four neurons in hidden layer is not sufficiently powerful for given learning and generalization, while the performance of a network with 6 neurons or higher will be improved. But as shown in Fig. 4, further increases in the number of neurons lead to the increase in the error in the trained network while the test data were fed to the network (i.e. the network is over-fitting). Therefore, to avoid over-fitting the six neurons were selected for net hidden layer for subsequent studying. Also the optimal condition was found for RBF net with respect to the set of RBF widths, and the output layer weights. In the present study, the RBF was trained with various values of width and cluster. A width 5 was observed to give higher accuracy.



F i g . 3 – *Variation of net. Performance (MSE) vs. number of neurons in hidden layer based on training data: (●) Diffusion neural network (DNN), (▲) Multilayer percepetron (MLP).*



F i g . 4 – *Variation of net. Performance (MSE) vs. number of neurons in hidden layer as test data fed to trained network. (●) Diffusion neural network (DNN), (▲) Multilayer percepetron (MLP).*

In these stages, only the original dataset was utilized for the training and optimization of nets. Therefore, the input variables were mole fraction of TEG, temperature and the output was the activity coefficient of water. The results of nets are shown in Table 1. Data analysis based on statistical parameters such as relative error (RE), was calculated. The analysis of results showed that the capability of MLP model to predict the activity coefficient on both finite-concentration zone and low concentration zone (infinite dilution) is superior to RBF model. Moreover, it seems that the capability of both models to predict the activity coefficient in low concentration zone (infinite dilution) of binary mixture, less than finite-concentration zone, the number of available training data in this zone is insufficient. As a result, it is difficult to guarantee that a good predictive model will be obtained in complete experimental domain. Therefore, to improve the performance of neural networks, we employed diffusion neural network (DNN) model, previously described to learn from a small dataset. Using information diffusion technique eqs. (9–15), we may derive more data from their original data especially in infinite dilution of binary mixture. Based on statistical analysis of derivative input-output data and subsequent network results, we suggested the following equations for determining h and $\psi(r)$:

$$h = 1.10(a - b) \quad \text{for } n = 9 \text{ (data point in infinite region)} \tag{19}$$

$$\psi(r) = r \tag{20}$$

To do this, r was calculated by eqs. (9–12), $\psi(r)$ is determined by eq. (20) and pairs (p'_i, t'_i) and (p''_i, t''_i), $i = 1, 2, 3, \ldots, 9$ are calculated by using eq. (14) and eq. (15). p'_i, p''_i are corresponding to derivative temperature variable as input, and t'_i, t''_i are corresponding to derivative activity coefficient variable as output of nets. In Table 2 the original and their derivative data are presented for infinite dilution.

In the infinite dilution, the TEG mole fraction is assumed as unity. Therefore, in derived data the TEG mole fraction is not considered. In this way the new data set for infinite dilution increase to $3n$ (i.e. $n = 9$). It must be considered in this case the input variables are mole fraction (assumed as unity), temperature and their possibilities and output layer are activity coefficient and its possibility.

The derivative data together with original data can be used to train the networks (i.e. DNNs) with four nodes in the input layer and two nodes in the output layer. As shown in Fig. 2, the topology

Table 1 – *Results obtained for the test data modeled with CNN for water (2)-TEG (1) binary system, with respective errors*

Finite-concentration mixture								
T/K	X(1)	Exp (γ_2)	CNN				Parrish	R.E %
			MLP	R.E %	RBF	R.E %		
332.6	0.4562	0.8571	0.8468	1.1998	0.8539	0.3779	0.8911	−3.9654
332.6	0.4970	0.8504	0.8238	3.1314	0.8407	1.1372	0.8687	−2.1495
332.6	0.2429	0.9276	0.9249	0.2912	0.9291	−0.1601	0.9811	−5.7667
297.6	0.6731	0.6760	0.6870	−1.6205	0.6832	−1.0600	0.6742	0.2599
297.6	0.3057	0.8264	0.8295	−0.3720	0.8305	−0.4964	0.8220	0.5295
297.6	0.2187	0.8784	0.8768	0.1784	0.8808	−0.2724	0.8765	0.2179
297.6	0.1157	0.9507	0.9511	−0.0400	0.9492	0.16200	0.9649	−1.4970
323	0.9100	0.8050	0.8192	−1.7652	0.8054	−0.0498	0.6428	20.1537
323	0.9201	0.7981	0.8230	−3.1237	0.7988	−0.0880	0.6387	19.9762
323	0.9401	0.7824	0.8146	−4.1136	0.7834	−0.1336	0.6306	19.4027
323	0.9501	0.7728	0.8005	−3.5888	0.7734	−0.0726	0.6266	18.9223
332.6	0.5107	0.8215	0.8159	0.6775	0.8367	−1.8526	0.8611	−4.8236
332.6	0.0454	0.9980	0.9829	1.5088	0.9980	−0.0002	0.9995	−0.1484
M.A.E*				**0.45**		**1.66**		**7.52**

*Mean Absolute Error

Table 1 – *(continued)*

Infinite-dilution mixture							
T/K	Exp (γ_2)	MLP	R.E %	RBF	R.E %	Parrish	R.E %
311.71	0.5750	0.5661	1.5478	0.5699	0.885	0.5826	−1.3232
300.43	0.5510	0.5352	2.8675	−0.1389	125.2059	0.5587	−1.4017
343.43	0.6240	0.6190	0.8013	0.6462	−3.5511	0.6503	−4.2121
M.A.E			**1.73**		**49.38**		**2.31**

Table 2 – *Original and derivative data points by information diffusion technique in infinite dilution*

Original data points		Derivative data points			
$(T, \gamma, \psi(r) = 1)$		$(T', \gamma', \psi(r) = 0.999)$		$(T'', \gamma'', \psi(r) = 0.999)$	
313	0.5786	312.88	0.5784	313.12	0.5788
300.43	0.551	300.31	0.5508	300.55	0.5512
311.71	0.575	311.59	0.5748	311.83	0.5752
323.26	0.59	323.14	0.5898	323.38	0.5902
333.76	0.617	333.64	0.6168	333.88	0.6172
343.43	0.624	343.31	0.6238	343.55	0.6242
355.93	0.636	355.81	0.6358	356.05	0.6362
364.93	0.669	364.81	0.6688	365.05	0.6692
378.32	0.692	378.2	0.6918	378.44	0.6922

4-K-2 was chosen for networks. Therefore, the training was implemented based on a new dataset. The training performance of diffusion neural net based on multilayer percepetron (DNN-MLP) is illustrated in Figs. 3 and 4. In this case the optimal number of neurons in hidden layer was 6. The characteristics of the optimized DNN-MLP are summarized in Table 3.

Moreover, the diffusion neural net based on radial base function (DNN-RBF) was trained on a new dataset. In this case, the optimal condition was found with respect to various values of width and cluster. A width 5 was observed to give higher accuracy.

Data analysis based on the statistical parameters such as relative error (RE), and correlation coefficient (r) between the networks prediction and

Table 3 – *Optimized parameters employed for the construction of DNN-MLP model*

Parameters							
Number of nods in layers						Transfer function	
input	hidden	output	learning rate	momentum	No. of epoch	hidden layer	output layer
4	6	2	0.50	0.80	100	sigmoid	linear

Table 4 – *Results obtained for test data modeled with DNNs and Parrish et al. nonlinear correlation for water(2)-TEG(1) binary system, with respective errors*

Finite-concentration mixture								
T/K	*X*(1)	Exp. (γ_2)	DNN–MLP	R.E %	DNN-RBF	R.E %	Parrish	R.E %
323	0.9100	0.8050	0.8056	−0.0769	0.8071	−0.2657	0.6428	20.1537
332.6	0.4562	0.8571	0.8629	−0.6729	0.8437	1.5634	0.8911	−3.9654
323	0.9501	0.7728	0.7729	−0.0080	0.7831	−1.3299	0.6266	18.9223
332.6	0.0454	0.9980	0.9979	0.0106	0.9957	0.2277	0.9995	−0.1484
297.6	0.1026	0.9563	0.9567	−0.0398	0.9606	−0.4501	0.9756	−2.0141
297.6	0.2505	0.8608	0.8609	−0.0092	0.8561	0.5428	0.8545	0.7372
332.6	0.2429	0.9276	0.9244	0.3457	0.9276	−0.0017	0.9811	−5.7667
332.6	0.4970	0.8504	0.8492	0.1356	0.8227	3.2582	0.8687	−2.1495
332.6	0.0808	0.9893	0.9964	−0.7198	1.0028	−1.3610	0.9984	−0.9155
297.6	0.4967	0.7371	0.7365	0.0816	0.7446	−1.0127	0.7390	−0.2630
297.6	0.8062	0.6604	0.6644	−0.6045	0.6689	−1.2880	0.6255	5.2838
313	0.9201	0.7621	0.7623	−0.0280	0.7724	−1.3493	0.6164	19.1238
332.6	0.5107	0.8215	0.8444	−2.7922	0.8159	0.6760	0.8611	−4.8236
M.A.E*				**0.42**		**1.02**		**6.48**

*Mean Absolute Error

Table 4 – *(continued)*

Infinite-dilution mixture							
T/K	Exp. (γ_2)	DNN-MLP	R.E %	DNN-RBF	R.E %	Parrish	R.E %
300.31	0.5508	0.5501	0.1299	0.5492	0.2864	0.55847	−1.3925
311.83	0.5752	0.5768	−0.2850	0.5811	−1.0293	0.582863	−1.3322
323.38	0.5902	0.5905	−0.0504	0.5971	−1.1734	0.607418	−2.9172
378.44	0.6922	0.6917	0.0781	0.6910	0.1710	0.725958	−4.8769
313	0.5786	0.5778	0.1361	0.5820	−0.5824	0.585346	−1.1659
333.76	0.6170	0.6170	0.0052	0.6169	0.0117	0.62957	−2.0373
355.93	0.6360	0.6363	−0.0395	0.6363	−0.0526	0.677173	−6.4738
311.71	0.5750	0.5767	−0.3013	0.5810	−1.0505	0.582608	−1.3232
M.A.E			**0.12**		**0.54**		**2.68**

experimental results were carried out in Table 4. The prediction performance of activity coefficients in the entire rang of temperature and mole fraction was illustrated in Fig. 5. It is immediately obvious that the DNN-MLP model possesses a high ability to predict the both infinite dilution activity coefficient and finite concentration activity coefficient in the range of temperature with a mean absolute error

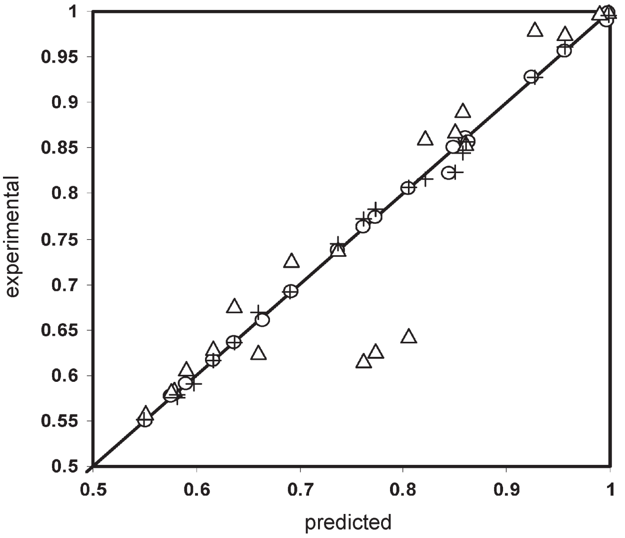


Fig. 5 – *Modeling ability of the optimized DNNs and a comparative study between the experimental and predicted activity coefficients of water in entire range of temperature and composition (○) DNN-MLP modeling (r = 0.999, MAE = 0.31 %); (+) DNN-RBF modeling (r = 0.997, MAE = 0.84 %); (∆) Parrish correlation (r = 0.919, MAE = 5.03 %)*

of 0.42 % and 0.12 % respectively. The results demonstrated good agreement between the predicted and the experimental values of activity coefficient (r = 0.999). The standard deviation in relative errors was 0.61 %. This value showed that the dispersion around the average value was small. According to Tables 1 and 4, $MAE_{DNN\text{-}MLP} < MAE_{MLP}$ means the DNN-MLP estimate is closest to the real function than the estimate from the conventional MLP network. In this case the advantage of DNN-MLP is defined by:

$$Ad = \frac{MAE_{MLP} - MAE_{DNN-MLP}}{MAE_{MLP}} \cdot 100 \quad (21)$$

For the finite concentration zone we have MAE_{MLP} = 0.0045 and $MAE_{DNN\text{-}MLP}$ = 0.0042. Therefore, using formula 21, Ad = 6.6 %, it means that DNN-MLP can reduce the error by 6.6 %. Similarity for infinite dilution zone MAE_{MLP} = 0.0173 and $MAE_{DNN\text{-}MLP}$ = 0.0012 and thus Ad = 93 %. Due to increased training data especially in infinite dilution zone by diffusion technique the advantage of diffusion neural net in infinite dilution zone is better than the finite concentration zone.

On the other hand, the DNN-RBF model, although it provided a satisfactory correlation in prediction, was less accurate than the DNN-MLP model. This is established by higher mean absolute error in prediction of activity coefficient in both infinite dilution activity coefficient and finite concentration with 1.02 % and 0.54 % respectively.

The results obtained using DNNs model was compared with Parrish's model in Table 4. Parrish's model systematically overpredicts the data with a positive average error of 2.68 % and 6.48 % for infinite dilution and finite concentration activity coefficient respectively. The standard deviation in relative errors was 6.29 %. This value showed the dispersion from the average value was higher.

On the other hand, the Parrish correlation, although it provided a quite satisfactory correlation in prediction, was less accurate than the DNNs model. Another important observation can be made from Fig. 6.

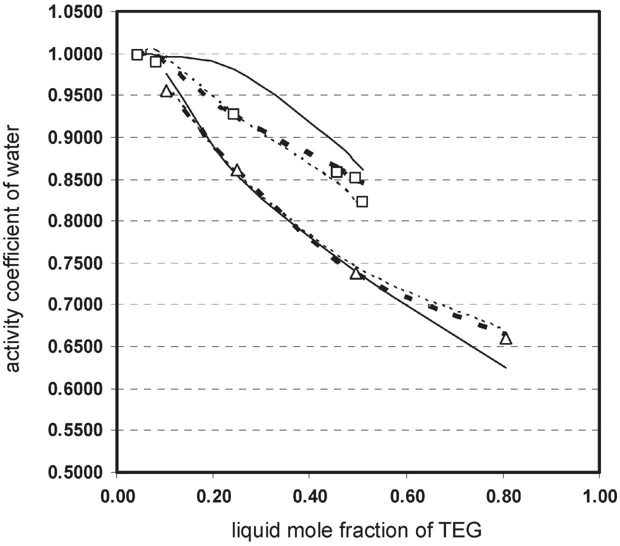


Fig. 6 – *Activity coefficient of water vs. liquid mole fraction TEG: (∆) T = 297.6 K, (□) T = 332.6 K; (– – –) DNN-MLP modeling; (...) DNN-RBF modeling; (–) Parrish correlation*

The DNNs model predicts the activity coefficients accurately in low and high temperature. But Parrish's correlation predicts higher activity coefficients at higher temperature. This difference in the predicted trend can have significant consequences in the extrapolation of activity coefficients to high temperatures.

Conclusion

A hybrid model integrating the deriving model and a corresponding neural network called a diffusion-neural (DNN) was developed for the prediction of activity coefficients of water in water-TEG mixture. DNN can be considered a primary model based on the principle of information diffusion to improve accuracy of conventional neural networks (CNN) with respect to small dataset. This paper shows how the principle of information diffusion can be used to derive more data to partly fill the gaps in a small dataset (i.e. in infinite-dilution) and thus obtain a better result when employing a neural network to recognize a non-linear function. DNN was trained according to the Levenberg-Marquardt algorithm. Optimization of network size, to achieve generalization of the model and avoid over-fitting, was done according to minimizing mean square error (MSE) of network based on training dataset. The DNNs model specially DNN-MLP exhibits a suitable prediction of the activity coefficients by relatively low mean absolute error (MAE = 0.31 %) and high correlation coefficient (r = 0.999) in the entire range of concentration and temperature. The percent error in predicting the activity coefficients was found to be lower than Parrish's correlation (MAE = 5.03 %) and correlation coefficient (r = 0.919).

Nomenclature

A, B, C	– temperature-dependent parameters eq. (18)
ANN	– artificial neural network
CNN	– conventional neural network
DNN	– diffusion neural network
FS	– fuzzy set
h	– diffusion coefficient in eqs. (5), (6)
MAE	– mean absolute error
MLP	– multilayer percepetron
MSE	– mean square error
NLR	– non-linear regression
P	– input vector of network
PDF	– possibility density function
r	– correlation coefficient
RBF	– radial base function
RE	– relative error
T	– temperature, K
TEG	– triethylene glycol
u	– derivative point
W_j	– weight vector of unit (neuron) j
x	– mole fraction

Greek letters

γ	– activity coefficient
μ	– diffusion function
π	– normalized possibility distribution
$\psi(r)$	– mapping function
λ	– gain parameter in eq. (1)
σ_j	– radial basis function's width. eq. (6)
η	– an operator defined in eq. (3)

References

1. *Ripley, B. D.*, Pattern recognition and neural networks, Cambridge University Press, 1996.
2. *Van der Smagt, P. P.*, Neural networks **7** (1994) 1.
3. *Lu, S., Basar, T.*, IEEE Transactions of Neural Network **9** (1998) 407.
4. *Petersen, R., Fredenslund, A., Rasmussen, P.*, Comput. Chem. Eng. **18** (1994), s63.
5. *Guimaraes, P. R. B., McGreavy, C.*, Comput. Chem. Eng. **19** (1991) 741.
6. *Sharma, V., Singhal, D., Ghosh, R., Dwivedi, A.*, Comput. Chem. Eng. **23** (1999) 385.
7. *Ganguly, S.*, Comput. Chem. Eng. **27** (2003) 1445.
8. *Urata, S., Takada, A., Murata, J., Hiaki, T., Sekiya, A.*, Fluid Phase Equilib. **199** (2002) 63.
9. *Mohanty, S.*, Fluid Phase Equilib. **235** (2005) 92.
10. *Mohanty, S.*, Int. J. Refrigeration **29** (2006) 243.
11. *Rai, P., Majumdar, G. C., DasGupta, S., De, S.*, J. of Food Eng. **68** (2005) 527.
12. *Elsharkwy, A. M., Gharbi, R. B. C.,* Advaces in Eng. Software **32** (2001) 215.
13. *Laugier, S., Richon, D.*, Fluid Phase Equilib. **210** (2003) 247.
14. *Karimi, H., Ghaedi, M.*, Annali di Chimica **96** (2006) 657.
15. *Shyam, S. S., Oon-Doo, B., Michele, M.*, J. of Food Eng. **52** (2002) 299.
16. *Huang, C. F., Moraga, C.*, Intrna. J. Aprox. Reason. **35** (2004) 137.
17. *Lanouette, R., Thibault, J., Valade, J. L.*, Computer and Chemical Engineering **23** (1999) 1167.
18. *Parrish, W. R., Won, K. W., Baltatu, M. E.*, Phase behavior of the triethylen glycol-water system and dehydration/regeneration design for extremely low dew point requirements, in: Proceeding of the 65th Annual GPA Convention, San Antonio, TX, March 10–12, 1986.
19. *Van der Smagt, P. P.*, Neural Networks **7** (1) (1994) 1.
20. *Huang, C. F., Ruan, D.*, Information diffusion principle and application in fuzzy neuron, in: *Ruan, D.* (Ed.), Fuzzy Logic Foundations and Industrial Applications, Kluwer Academic Publishers, Boston, 165, 1996.
21. *Huang, C. F., Lueng, Y.*, Fuzzy Sets and Systems **107** (1999) 131.
22. *Huang, C. F., Shi, Y.*, Towards efficient fuzzy information processing—using the principle of information diffusion. physics – Verlag, Heidelberg, 2002.
23. *Huang, C. F.*, Deriving samples from incomplete data, in: Proceedings of FUZZ-IEEE'98, Anchorage, USA, 645, 1998.
24. *Rosman, A.*, Soc. Pet. Eng. J., 1973, 293.
25. *Herskowitz, M., Gottlieb, M.*, J. Chem. Eng. Data **29** (1984) 173.
26. *Bestani, B., Shing, K. S.*, Fluid Phase Equilib. **50** (1989) 209.
27. *Twu, C. H., Tassone, V., Sim, W. D., Watanasiri, S.*, Fluid phase Equlib. **228** (2005) 213.
28. GPA, Editorial review board. Recent developments in gas dehydration and hydrate inhibition, in: Proceedings of the Laurance Reid conference, Norman, Oklahoma, 1994.
29. *Campbell, J. M.*, Gas Conditioning and Processing, Campbell Petroleum series, 2, 1994.

gaps in a small dataset (i.e. in infinite-dilution) and thus obtain a better result when employing a neural network to recognize a non-linear function. DNN was trained according to the Levenberg-Marquardt algorithm. Optimization of network size, to achieve generalization of the model and avoid over-fitting, was done according to minimizing mean square error (MSE) of network based on training dataset. The DNNs model specially DNN-MLP exhibits a suitable prediction of the activity coefficients by relatively low mean absolute error (MAE = 0.31 %) and high correlation coefficient ($r = 0.999$) in the entire range of concentration and temperature. The percent error in predicting the activity coefficients was found to be lower than Parrish's correlation (MAE = 5.03 %) and correlation coefficient ($r = 0.919$).

Nomenclature

A, B, C	– temperature-dependent parameters eq. (18)
ANN	– artificial neural network
CNN	– conventional neural network
DNN	– diffusion neural network
FS	– fuzzy set
h	– diffusion coefficient in eqs. (5), (6)
MAE	– mean absolute error
MLP	– multilayer percepetron
MSE	– mean square error
NLR	– non-linear regression
P	– input vector of network
PDF	– possibility density function
r	– correlation coefficient
RBF	– radial base function
RE	– relative error
T	– temperature, K
TEG	– triethylene glycol
u	– derivative point
W_j	– weight vector of unit (neuron) j
x	– mole fraction

Greek letters

γ	– activity coefficient
μ	– diffusion function
π	– normalized possibility distribution
$\psi(r)$	– mapping function
λ	– gain parameter in eq. (1)
σ_j	– radial basis function's width. eq. (6)
η	– an operator defined in eq. (3)

References

1. *Ripley, B. D.*, Pattern recognition and neural networks, Cambridge University Press, 1996.
2. *Van der Smagt, P. P.*, Neural networks **7** (1994) 1.
3. *Lu, S., Basar, T.*, IEEE Transactions of Neural Network **9** (1998) 407.
4. *Petersen, R., Fredenslund, A., Rasmussen, P.*, Comput. Chem. Eng. **18** (1994), s63.
5. *Guimaraes, P. R. B., McGreavy, C.*, Comput. Chem. Eng. **19** (1991) 741.
6. *Sharma, V., Singhal, D., Ghosh, R., Dwivedi, A.*, Comput. Chem. Eng. **23** (1999) 385.
7. *Ganguly, S.*, Comput. Chem. Eng. **27** (2003) 1445.
8. *Urata, S., Takada, A., Murata, J., Hiaki, T., Sekiya, A.*, Fluid Phase Equilib. **199** (2002) 63.
9. *Mohanty, S.*, Fluid Phase Equilib. **235** (2005) 92.
10. *Mohanty, S.*, Int. J. Refrigeration **29** (2006) 243.
11. *Rai, P., Majumdar, G. C., DasGupta, S., De, S.*, J. of Food Eng. **68** (2005) 527.
12. *Elsharkwy, A. M., Gharbi, R. B. C.,* Advaces in Eng. Software **32** (2001) 215.
13. *Laugier, S., Richon, D.*, Fluid Phase Equilib. **210** (2003) 247.
14. *Karimi, H., Ghaedi, M.*, Annali di Chimica **96** (2006) 657.
15. *Shyam, S. S., Oon-Doo, B., Michele, M.*, J. of Food Eng. **52** (2002) 299.
16. *Huang, C. F., Moraga, C.*, Intrna. J. Aprox. Reason. **35** (2004) 137.
17. *Lanouette, R., Thibault, J., Valade, J. L.*, Computer and Chemical Engineering **23** (1999) 1167.
18. *Parrish, W. R., Won, K. W., Baltatu, M. E.*, Phase behavior of the triethylen glycol-water system and dehydration/regeneration design for extremely low dew point requirements, in: Proceeding of the 65th Annual GPA Convention, San Antonio, TX, March 10–12, 1986.
19. *Van der Smagt, P. P.*, Neural Networks **7** (1) (1994) 1.
20. *Huang, C. F., Ruan, D.*, Information diffusion principle and application in fuzzy neuron, in: *Ruan, D.* (Ed.), Fuzzy Logic Foundations and Industrial Applications, Kluwer Academic Publishers, Boston, 165, 1996.
21. *Huang, C. F., Lueng, Y.*, Fuzzy Sets and Systems **107** (1999) 131.
22. *Huang, C. F., Shi, Y.*, Towards efficient fuzzy information processing—using the principle of information diffusion. physics – Verlag, Heidelberg, 2002.
23. *Huang, C. F.*, Deriving samples from incomplete data, in: Proceedings of FUZZ-IEEE'98, Anchorage, USA, 645, 1998.
24. *Rosman, A.*, Soc. Pet. Eng. J., 1973, 293.
25. *Herskowitz, M., Gottlieb, M.*, J. Chem. Eng. Data **29** (1984) 173.
26. *Bestani, B., Shing, K. S.*, Fluid Phase Equilib. **50** (1989) 209.
27. *Twu, C. H., Tassone, V., Sim, W. D., Watanasiri, S.*, Fluid phase Equlib. **228** (2005) 213.
28. GPA, Editorial review board. Recent developments in gas dehydration and hydrate inhibition, in: Proceedings of the Laurance Reid conference, Norman, Oklahoma, 1994.
29. *Campbell, J. M.*, Gas Conditioning and Processing, Campbell Petroleum series, 2, 1994.