International Journal of Industrial Chemistry (IJIC)
a SpringerOpen Journal

RESEARCH Open Access

Estimation of vapor pressures, compressed liquid, and supercritical densities for sulfur dioxide using artificial neural networks

Abdolreza Moghadassi*, Mahmoodreza Nikkholgh, Sayedmohsen Hosseini and Fahime Parvizian

Abstract

Background: Artificial neural networks (ANNs) as a solution for semi-structural or non-structural problems have widespread applications in engineering and science with acceptable results. In this research, the ability of multilayer perceptron artificial neural networks based on back-propagation algorithm was investigated to estimate sulfur dioxide densities.

Results: The best network configuration for this case was determined as a three-layer network including 15, 10, and 1 neurons in its layers, respectively, using Levenberg-Marquardt training algorithm. The uncertainties in the presented network for prediction of unseen data including *PρT* and saturated liquid densities are less than 0.5% and 1%, respectively. Another network for estimation of vapor pressure has trained with uncertainty less than 0.67%. Comparisons among the artificial neural network predictions, several equations of state, and experimental data sets show that the ANN results are in good agreement with the experimental data better than the equations of states.

Conclusion: Artificial neural network can be a successful tool to represent thermophysical properties effectively, if developed efficiently.

Keywords: Neural network, Multilayer perceptron, Sulfur dioxide, Density, Equation of state

Background

In order to study chemical systems, thermophysical properties of pure compounds and mixtures under specified conditions are required. It is not always possible to find reliable experimental values of these properties for interesting compounds nor is it practical to measure the properties as the need arises. Particularly in process simulation, reliable and accurate property estimation methods play an important role in the solution of various simulation problems [1].

Sulfur dioxide (SO_2) is a colorless gas or liquid with a pungent odor. It is produced from the burning of fossil fuels (coal and oil) and the smelting of mineral ores (aluminum, copper, zinc, lead, and iron) that contain sulfur. It is one of the major primary pollutants in the atmosphere [2]. It is the most important industrial-based sulfur product, especially used in the manufacture of sulfuric acid. Apart from this main use, sulfur dioxide is used in different applications, from refining raw materials to preserving food, paper and pulp production, sulfoxidation and sulfochlorination, disinfectants, and water and waste treatment [3,4].

Since the 1940s, artificial neural networks (ANNs) have been used in various applications in engineering and science. ANNs are generally the software systems that imitate the neural networks of the human brain. The ANNs show graceful degradation, they can easily form models for complex problems. Especially in the development of solutions for semi-structural or non-structural problems, ANN models can have very acceptable results. Moreover, they can be cheaper, faster, and more adaptable than traditional methods [5]. One of the major advantages of ANN is efficient handling of highly nonlinear relationships in the data, even when the exact nature of such relationship is unknown.

Recently, ANN has been used to predict some pure substances and petroleum fraction's properties [6], activity coefficients of isobaric binary systems [7], dew point

* Correspondence: A_Moghadassi@yahoo.com
Department of Chemical Engineering, Faculty of Engineering, Arak University, Arak 38156-8-8349, Iran

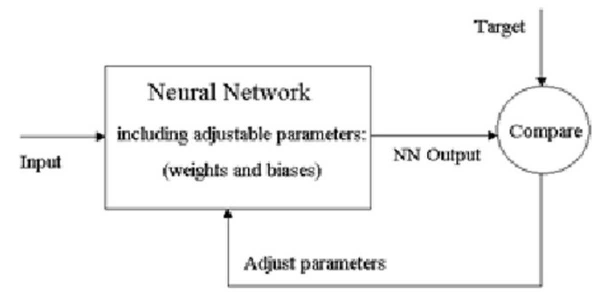

Figure 1 Generalized neural network training algorithm.

pressure [8], vapor-liquid equilibrium data [9], thermodynamic properties of refrigerants [10,11], and activity coefficient ratio of electrolytes in amino acid solutions [12], etc.

The aim of this work is to construct artificial neural networks that can predict SO_2 densities (in the sub- and supercritical conditions) and vapor pressures in large ranges of temperatures and pressures. It was shown that multilayer perceptron networks can approximate any continuous function that sufficient connection weights are used. Consequently, multilayer perceptrons can be used in all of the models. The neural network toolbox of MATLAB7.0, as a popular numerical computation and visualization software, has been used for training and testing of multilayer networks. Finally, estimations of the ANN have been compared with the experimental data and results of several equations of state.

Methods

Artificial neural network

ANN can be considered as a black box consisting of a series of complicated equations for the calculation of outputs based on a given series of input values. It is able to develop a model relating the output of network to existing actual data used as inputs. One of the major advantages of ANN is efficient handling of highly nonlinear relationships in the data when the exact nature of such relationship is unknown [5].

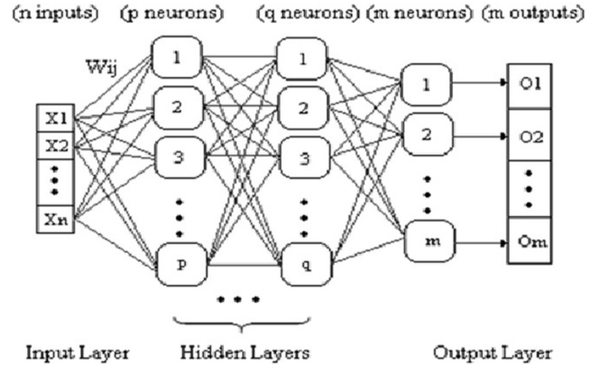

Figure 2 The feed-forward neural network architecture.

Table 1 Comparison among networks containing one and two hidden layers

Number of hidden layers	Neuron number in each layer	Train MSE	Test MSE
1	10-1	44.793	66.928
1	15-1	0.9877	3.2875
1	20-1	0.5267	0.9468
1	25-1	1.6491	143.25
2	5-5-1	0.0529	0.2960
2	10-5-1	0.0036	0.0748
2	5-10-1	0.0174	0.1246

Commonly, neural networks are adjusted (or trained) so that a particular input leads to a specific target output (Figure 1). The network adjusts based on a comparison of the output and the target, until the network output matches the target. Typically, many input/target pairs are used in supervised learning to train a network [13].

The most popular ANN is the feed-forward multilayer network that uses back-propagation learning algorithm as shown in Figure 2.

Feed-forward neural network usually has one or more hidden layers and an output layer, which enable the network to model nonlinear and complex functions. Scaled data are introduced into the input layer of the network then propagated from the input layer to the hidden layer and finally to the output layer.

A parameter W_{ij} (known as weight) is associated with each connection between two cells. Thus, each cell in the upper layer receives weighted inputs from each node in the layer below and then processes these collective inputs before the unit sends a signal to other layers.

At first, each neuron in the hidden or output layer acts as a summing junction that combines and modifies the inputs from the previous layer using the following equation [14]:

$$A_i = b_j + \sum_{j=1}^{n} X_j W_{ij} \tag{1}$$

$$Y_j = S(A_j), \tag{2}$$

where A_j is the net input to node j in the hidden or output layer; X_j, the inputs to node j (or outputs of the previous layer); W_{ij}, the weights representing the strength of the

Table 2 Performance parameters used for the learning process

Parameter	Value
Maximum iteration	1,500
Minimum gradient	1e−9
mu, mu_dec, mu_inc, mu_max	0.001, 0.1, 10, 1e10

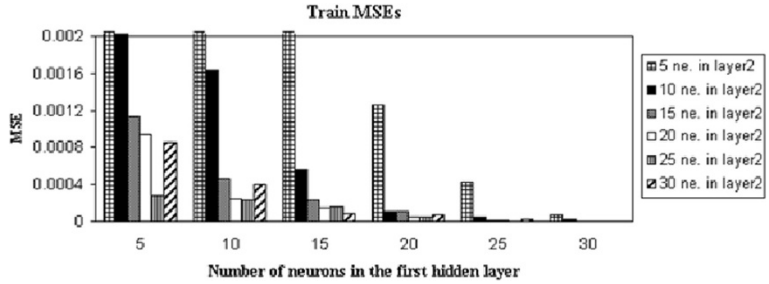

Figure 3 **Comparison among produced training MSEs using different numbers of neurons.**

connection between the i_{th} node and j_{th} node; *i*, the number of nodes; and b_j, the bias associated with node *j*. Each neuron consists of a transfer function expressing internal activation level. Generally, the transfer functions for function approximation (regression) are sigmoidal function, hyperbolic tangent, and linear function, in which sigmoidal function is widely used for nonlinear relationship. Y_j (the output of node *j*) is also an element of the inputs to the nodes in the next layer [5].

There are many various back-propagation algorithms. The simplest implementation of back-propagation learning updates the network weights and biases in the direction of performance function decreasing. An iteration of this algorithm is as follows [14]:

$$x_{k+1} = x_k - a_k g_k, \tag{3}$$

where x_k is the vector of current weights and biases; g_k, the current gradient; and α_k, the learning rate.

There are two different ways for this gradient descent algorithm: incremental mode and batch mode. The gradient is computed in the incremental mode, and the weights are updated after each input is applied to the network. In the batch mode, all of the inputs are applied to the network before the weights are updated. The objective is to find the value of the weight that minimizes differences between the actual output and the predicted output in the output layer in order to minimize the mean squared errors (MSEs), the average squared error between the network predicted outputs, and the target outputs.

The Levenberg-Marquardt algorithm is one of the best training rules that designed to approach second-order network training speed [15]. This algorithm trains 10 to 100 times faster than the usual gradient descent back-propagation method and update as follows:

$$x_{k+1} = x_k - \left[J^T J + \mu I\right]^{-1} J^T e, \tag{4}$$

where *J* is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases; *e*, the vector of network errors; and *I*, always the ones square matrix that is the same size as the $J^T J$. The Jacobian matrix can be computed through a standard back-propagation technique. The scalar μ decreases after each successful step (reduction in the performance function) and increases only when a tentative step would increase the performance function. In this way, the performance function will reduce at each iteration of the algorithm [13].

In the learning process, several variables influence on the ANN training. These variables are the number of iterations, learning rate, the momentum coefficient, the number of hidden layers, and the number of hidden neurons. To find the best amount of variables and parameters, all of those must be varied and the best combination chosen [13].

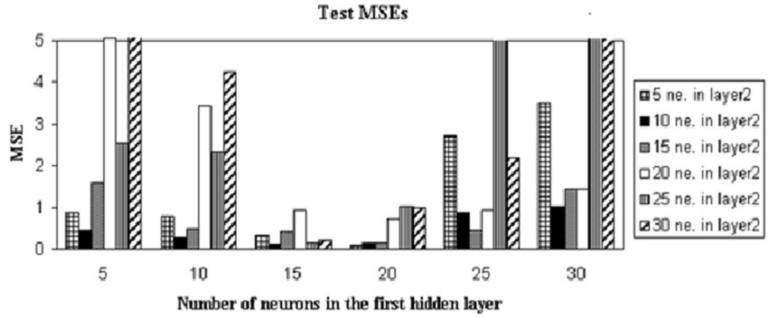

Figure 4 **Comparison among produced testing MSEs using different numbers of neurons.**

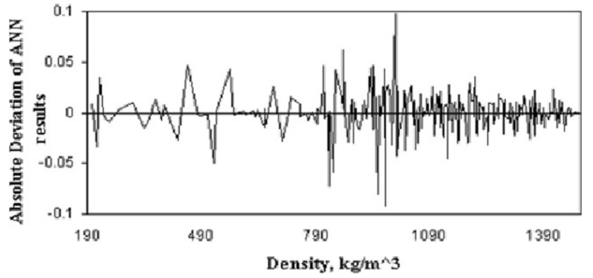

Figure 5 Absolute deviation of ANN results in contrast with the experimental data in the training step.

Results and discussion

Development of ANN models

At the following experimental section, it is mentioned that we have 363 overall data points. More than 80% of the data set has been used to train each ANN, and the rest have been used to evaluate their accuracy and trend stability.

After determining the number of input variables by statistical analyses, the most appropriate architecture for the network should be determined. In this stage, two networks should be considered for training and testing. The number of layers, the optimum number of neurons per layer, and the transfer function(s) in the hidden layer(s) are obtained by trial and error. As a rule of thumb, the number of adjustable parameters should be equal or smaller than the number of available training data [5]. Numbers of adjustable parameters are related to neuron numbers directly.

Therefore, several feed-forward neural networks with different architectures were tried to finally arrive at a three-layer network, including two hidden layers. Summarily, comparison between two network architectures is given in Table 1.

As it is considered, the performances of two hidden layer neural networks are much better than the one-layer networks, even when a two-layer network has fewer neurons. It shows less computational requirements and of course better operation. Therefore, a feed-forward network with two hidden layers is used, in which temperature and pressure are input variables and density is the output variable.

In all networks, linear transfer functions have been used in the outer layers. Experiments were done using different transfer functions, which are located in either layers of the network. However, it has been proven that utilizing hyperbolic tangent sigmoid (tansig) in the first layer and logarithmic sigmoid (logsig) in the second layer will produce better results.

The input and output data are normalized between −1 and 1, then the Levenberg-Marquardt back-propagation algorithm that represents a simplified version of Newton's method is applied as the training algorithm in this study. The MSE as an excellent criterion for evaluating the performance of the neural network is used. Furthermore, the network has been trained with Levenberg-Marquardt algorithm in MATLAB environment. This algorithm appears to be the fastest method for training moderate-sized feed-forward neural networks (up to several hundred weights) [13]. The performance parameters used in the training step of the networks are given in Table 2.

The parameter mu is the initial value for μ. This value is multiplied by mu_dec whenever the performance function is reduced in a step. It is multiplied by mu_inc whenever a step would increase the performance function. If mu becomes larger than mu_max, the algorithm is stopped. In addition, training stops when the maximum number of iteration (repetitions) is reached or the performance gradient falls below the minimum gradient.

The optimum performance for the network is obtained iteratively by changing the number of neurons in the hidden layers. On the other hand, we replaced neuron numbers from 5 to 30 in the first and second hidden layers and checked the testing error for each structure. The network with the least testing and training error and the most convergence rate has been selected. The results of networks using different numbers of neurons are presented below. Comparisons between produced MSEs to determine the best number of neurons in each hidden layer, from the

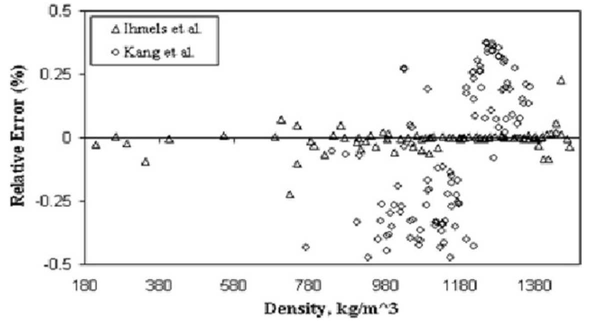

Figure 6 Deviation of ANN results in comparison with the unseen experimental densities.

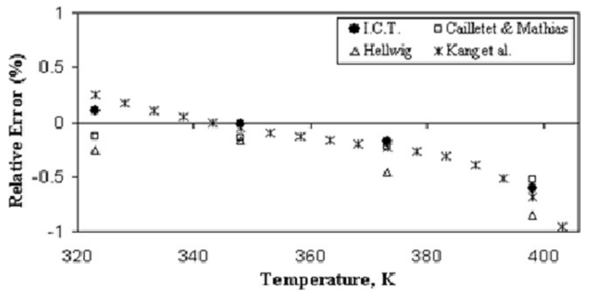

Figure 7 Deviation of ANN results in comparison with the unseen experimental saturated liquid densities.

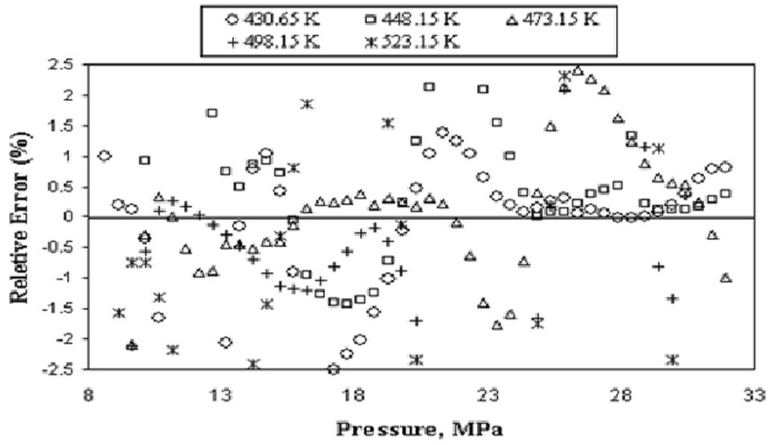

Figure 8 **Deviation of ANN results in comparison with the unseen experimental compressibility factors.**

training and testing data, are shown in Figures 3 and 4, respectively.

If there are a few neurons in the hidden layers, the performance of the network is not satisfactory. However, if there are too many, convergence is very slow and may be compromised by local minima. The optimal number of hidden neurons is determined empirically, as the minimal number of neurons for which the prediction performance is sufficient without leading to over fitting or an unreasonably long computational time.

Network training for at least five times indicates that using 15 and 10 neurons in order in the first and second layers has better results. Using this structure, the train and test MSEs are 0.000561 and 0.1085, respectively. Figure 5 shows absolute error fluctuations of data in the training step. It is defined as follows:

$$\text{Absolute deviation} = \left(\rho^{\text{exp}} - \rho^{\text{ANN}}\right). \tag{5}$$

The results show that the ANN can predict densities much close to the experimentally measured ones.

In addition, the ANN results are compared with some other unseen experimental data sets. Percent deviations of predicted densities using ANN in comparison with the testing data measured by Ihmels et al. [3] and finally other experimental data set reported by Kang et al. [16] are presented. Results are shown in Figure 6.

Table 3 Deviation of ANN results in comparison with the unseen experimental compressibility factors

Temperature (K)	Pressure range (MPa)	*Z*-factor range	AAD (%)
430.65	8.612 to 31.917	0.199 to 0.531	0.6955
448.15	10.132 to 31.917	0.347 to 0.536	0.7526
473.15	9.625 to 31.917	0.393 to 0.615	0.7569
498.15	9.626 to 30.397	0.494 to 0.712	0.7804
523.15	9.119 to 29.891	0.580 to 0.786	1.3952

Also, for saturated liquid densities, percents of deviations between predicted data by ANN and collected data in some other references [16-18] are calculated. Results are presented in Figure 7.

As another test, compressibility factors of sulfur dioxide are calculated using the network outputs (densities) and compared with the experimental values reported by Kang et al. [16]. Results are shown in Figure 8. Then, they are presented quantitatively in Table 3; the average absolute deviation (AAD) is as follows:

$$\text{AAD} = \frac{1}{n}\sum_{n}\left|\frac{\rho_{\text{exp}} - \rho_{\text{calc}}}{\rho_{\text{exp}}}\right|. \tag{6}$$

Comparisons indicate the percentage of absolute deviations in *PρT*, saturated liquid density, and compressibility factors which are lower than 0.5%, 1%, and 2.5%, respectively. These results show the ability of presented network obviously.

Table 4 Comparison among deviations of different EoS results for SO_2 densities

EoS	AAD (%)
Soave-Redlich-Kwong - 1972 [20]	11.64037
Peng-Robinson - 1975 [21]	7.66589
Heyen - 1983 [22]	6.15123
Modified PT by Valderrama and Cisternas - 1986 [23]	3.15492
Duan-Hu - 2003 [24]	3.63235
Dashtizadeh et al. - 2006 [25]	7.81983
Pazuki et al. - 2007 [26]	4.06891
This work	0.02893

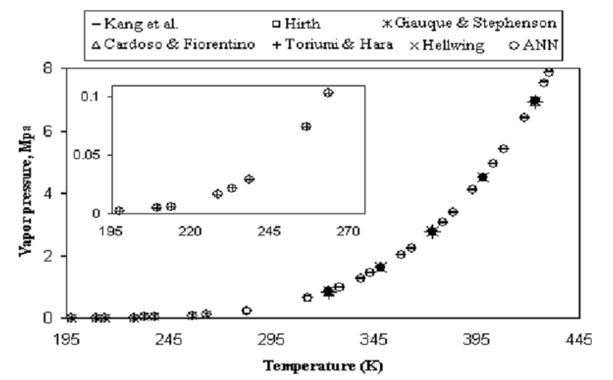

Figure 9 Vapor pressures of sulfur dioxide.

Validation of ANN results in contrast with the equations of state

Equations of state play an important role in chemical engineering design, and they have assumed an expanding role in the study of thermophysical properties. The accurate and simple equations of states (EOSs) are widely used for theoretical and practical studies in chemical process design, petroleum industry, reservoir fluids, etc. Among various types of EOSs, the cubic EOSs are simple, flexible to handle, and reliable (according to its accuracy) in different practical applications [19]. Consequently, in this work, the cubic equations of state are used as a chosen role to calculate sulfur dioxide densities in comparison with the ANN results. Some of the cubic equations of state used were as follows: (a) Soave-Redlish -Kwong equation [20], (b) Peng-Robinson equation [21], (c) Heyen equation [22], (d) the modified Patel and Teja equation reported by Valderrama and Cisternas [23], (e) Duan-Hu equation [24], (f) equation of state reported by Dashtizadeh et al. [25], and (g) the new equation of state reported by Pazuki et al. [26].

The first step in our analyses of the equations involves the calculation of SO_2 density at large ranges of temperatures and pressures. The densities are calculated at temperatures from 273 to 523 K and pressures up to 35 MPa (363 data points), for which densities between 194 and 1,485 kg/m^3 were covered. Then, the differences between calculated densities and experimental values have been determined. The results of these comparisons are listed in Table 4. As it is shown, equations of state lead to densities with an uncertainty (AAD) more than 3%.

As it is clear, the ANN predictions are very closer to the experimental values than the EoSs results. These results prove the ability of the network clearly. Note that ANNs can act similar to EoSs. When each equation of state considers as a nonlinear function between some inputs and outputs, it may be presented in shape of a network that takes the EoS inputs and gives back outputs accurately. Then, the main point is network training or detection of adjustable parameters. As it is clear, artificial neural networks have more generality than usual EoSs. In addition, they trained for any specific system and present high accuracy answers.

Prediction of vapor pressure

Vapor pressure sulfur dioxide is studied as another thermophysical property. Existing equations often do not have enough accuracy, e.g., Lee-Kesler and Antoine and their modified forms. Therefore, a simple network for predicting vapor pressure has been trained as a function of temperature to obtain accurate results.

After comparisons of several networks with different structures (different neurons and layers), an optimal network consisting of two layers (one hidden layer and one output layer) is selected, in which the temperature and vapor pressure are the input and output variables, respectively. In addition, numbers of neurons in the first and second layers are three and one neurons, respectively. In the first layer, the hyperbolic tangent sigmoid function was used as the neuron transfer function. In addition, the linear transfer function was applied for the output layer neuron. This network is trained using the experimental data (14 data points) reported by Kang et al. [16], Hirth [27], and Giauque and Stephenson [28] which are measured between 200.803 K (2.202 kPa) up to near the critical point. This network has been trained using the Levenberg-Marquardt back-propagation algorithm by MATLAB environment, like the presented network for densities. Then, the network results have been

Table 5 Comparison between ANN and the equation presented in *Perry's Handbook* for calculation of SO_2 vapor pressure

Data set	Temperature range (K)	Pressure range (MPa)	AAD[a] (%)	AAD[b] (%)	Reference
Kang	323.15 to 430.65	0.859 to 7.883	0.158	2.111	[16]
Hirth	283.15 to 423.15	0.229 to 6.973	0.503	2.463	[28]
Giauque and Stephenson	197.64 to 263.48	0.002 to 0.103	0.623	1.417	[29]
Cardoso and Fiorentino	323.15 to 423.15	0.845 to 6.931	0.479	2.679	[16]
Toriumi and Hara	323.15 to 423.15	0.846 to 6.910	0.670	2.590	[16]
Hellwig	323.15 to 423.15	0.856 to 6.966	0.271	1.993	[19]

[a]Deviation of ANN results in contrast with the experimental data set; [b]deviation of Perry's equation in contrast with the experimental data set.

compared with several sets of unseen experimental temperature-vapor pressure data. These comparisons are shown in Figure 9.

The references of experimental data used in this figure, their temperature and pressure ranges, and deviations of ANN results from the experimental data, in addition, deviation of another equation for estimation of vapor pressure of SO_2 by *Perry's Chemical Engineers' Handbook* [29] are presented in Table 5. As it is clear, deviations between ANN results and experimental data are lower than 0.67% that it is very better than 2.68% for Perry's equation. These results show the capability of the presented network obviously.

Experimental

For sulfur dioxide density, several thermophysical data are found in the literature. In this study, detailed data reported by Ihmels et al. [3] have been used to train the network. Densities have been measured with a computer-controlled high-temperature, high-pressure vibrating tube densimeter system in the sub- and supercritical states at temperatures from 273 to 523 K and pressures up to 35 MPa (363 overall data points), for which densities between 194 and 1,485 kg/m^3 were covered. The uncertainty in density measurement was estimated to be no greater than 0.1% in the liquid and compressed supercritical states, but near the critical temperature and pressure, the uncertainty increases to 1%.

Conclusion

The ability of artificial neural networks based on back-propagation algorithm to predict densities of sulfur dioxide have been investigated. To predict densities, several feed-forward neural networks with different architectures have been used. A feed-forward network with two hidden layers is used, in which temperature and pressure were input variables and density was the output variable. In addition, it has been proven that utilizing hyperbolic tangent sigmoid in the first layer and logarithmic sigmoid in the second layer will produce better results. The Levenberg-Marquardt algorithm has been applied as the training rule. Training a network for at least ten times indicates that using 15 and 10 neurons in the first and second layers, respectively has better results. Network predictions in comparison with several equations of state have proven that the ANN results are more accurate than predictions of EoSs. This has been shown with comparisons among several experimental set of $P\rho T$ data, saturated liquid densities, and compressibility factors in contrast with the ANN predictions, too. These results show the capability of the presented network obviously.

For predicting vapor pressures, a feed-forward network with two layers is used in which, temperature and vapor pressure are its input and output variables, respectively. Also, comparisons have shown good agreement between the experimental data and predicted results. All of these results prove that artificial neural networks can be a successful tool to represent complex nonlinear systems effectively (e.g., prediction of thermophysical properties), if developed efficiently.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

The main idea of this work was suggested by AM. He supervised the whole process of the manuscript preparation and also drafted the manuscript. He was the corresponding author of the manuscript and did essential coordination. MN was an MS student who collected data sets from the literature and also developed the neural network models and tested them. SH was the first advisor and has written the discussion of the results and cooperated in writing other explanations in the whole manuscript. FP was the second advisor who assisted in programming and developing the ANN models and also collaborated in the writing of some sections. All authors read and approved the final manuscript.

Acknowledgment

The authors gratefully acknowledge Arak University for the financial support during this research.

Received: 20 July 2011 Accepted: 30 March 2012
Published: 18 February 2013

References

1. Jayashanker Prasad Y, Sunil Bhagwat S (2002) Chem Eng Technol 25:11
2. Alkezweeny AJ (1995) Atmosfera. 8:91
3. Ihmels EC, Lemmon EW, Gmehling J (2003) Fluid Phase Equilibria 207:111
4. Nikkholgh MR (2011) Estimation of PVTx data of gas mixture using Neural Networks. Arak University, M.S. Thesis
5. Menhaj M (2002) Introduction of artificial neural network. Amirkabir University, Tehran
6. Bozorgmehry RB, Abdolahi F, Moosavian MA (2005) Fluid Phase Equilibria 231:188
7. Biglin M (2004) J Serb Chem Soc 69:669
8. Zambrano G (2002) Development of neural network models for the prediction of dew point pressure of retrograde gases and saturated oil viscosity of black oil systems. M.S. Thesis, Texas A&M University
9. Ganguly S (2003) Comp Chem Eng 27:1445
10. Sozen A, Arcakilioglu E, Ozalp M (2005) Applied Thermal Engineering 25:1808
11. Chouai A, Laugier S, Richon D (2002) Fluid Phase Equilibria 199:53
12. Dehghani MR, Modarress H, Bakhshi A (2006) Fluid Phase Equilibria 244:153
13. Demuth H, Beale M (2002) Neural network toolbox user's guide for use with MATLAB. Mathworks, Inc., Natick, MA
14. Hagan MT, Demuth HB, Beal M (1996) Neural network design. PWS, Boston
15. Hagan MT, Menhaj M (1994) IEEE Transactions on Neural Networks 5:989
16. Kang TL, Hirth LJ, Kobe KA, McKetta JJ (1961) J Chem Eng Data 6:220
17. Kohnstamm P (1928) International critical tables, vol 111. McGraw-Hill, New York
18. Hellwig LR (1955) Pressure-volume-temperature properties of sulfur dioxide. University of Texas, Ph.D. Dissertation
19. Farrokh-Niae AH, Moddarress H, Mohsen-Nia M (2007) J Chem Thermodynamics 40:84
20. Soave GS (1972) Chem Eng Sci 27:1197–1203
21. Peng DY, Robinson DB (1976) Ind Eng Chem Fundam 15:59
22. Heyen G (1981) A cubic equation of state with extended range of application. Proceedings of the 2nd world congress of chemical engineering, Montreal, Canada, pp 4–9. October, pp 41–46
23. Valderrama JO, Cisternas LA (1986) Fluid Phase Equilibria 29:431
24. Duan Z, Hu J (2004) Geochimica et Cosmochimica Acta 68:2997
25. Dashtizadeh A, Pazuki GR, Taghikhani V, Ghotbi C (2006) Fluid Phase Equilibria 242:19

26. Pazuki GR, Nikookar M, Omidkhah MR (2007) Fluid Phase Equilibria 254:42
27. Hirth LJ (1958) Gas compressibility of nitrous oxide and of sulfur dioxide by the Burnett method. University of Texas, Ph.D. Dissertation
28. Giauque WF, Stephenson CC (1938) J Am Chem Soc 60:1389
29. Perry RH, Green DW, Maloney JO (1999) Perry's chemical engineers' handbook, 7th edition. McGraw-Hill, New York, pp 2–54

doi:10.1186/2228-5547-4-14
Cite this article as: Moghadassi *et al.*: Estimation of vapor pressures, compressed liquid, and supercritical densities for sulfur dioxide using artificial neural networks. *International Journal of Industrial Chemistry* 2013 **4**:14.