Paper presented at CHEMECA'06, Auckland, New Zealand from 17-20 September 2006

M Kundu is presently with National Institute of Technology Rourkela

PREDICTION OF EQUILIBRIUM SOLUBILITY OF CO_2 IN AQUEOUS ALKANOLAMINES THROUGH ARTIFICIAL NEURAL NETWORK

R. Rajesh, S. Chattopadhyay, M. Kundu
Chemical Engineering Group
Birla Institute of Technology and Science, Pilani, Rajasthan, India
E-mail: mkundu@bits-pilani.ac.in, madhushreek@yahoo.com

Abstract

The removal of acid gases from gas streams by using suitable solvent like alkanolamine, commonly referred to as gas sweetening, is a technology that has been in use industrially for over half a century. In this work artificial neural network (ANN) has been used to predict the equilibrium solubility of CO_2 over the alkanolamine solvents N-methyldiethanolamine (MDEA) and 2-amino-2-methyl-1-propanol (AMP) instead of using any thermodynamic model. A multilayer feed forward network with back propagation training algorithm has been used here in an effort to predict the VLE data of CO_2-MDEA-H_2O and CO_2-AMP-H_2O system with a comparable accuracy to those predictions based on rigorous thermodynamic model. It has been found that the predictions are within accuracy of ± 5% for 95 % of the data.

1. INTRODUCTION

Removal of acid gas impurities such as carbon dioxide (CO_2), carbonyl sulfide (COS), and hydrogen sulfide (H_2S) from gas streams is a very important operation for natural gas processing, oil refineries, ammonia manufacture, coal gasification, and petrochemical plants. Many commercial gas-treating processes are still designed by experience and heuristics resulting in over design and excessive energy consumption. There is a need for the available data of acid gas-alkanolamine-water systems to be correlated so that the solubility predictions can be made confidently where data do not exist or where they are of poor precision (Kundu, 2004). Besides thermodynamic modelling, the applicability of ANN based models can also be explored. There are a few reported applications of ANN in VLE predictions (Naidu, 2004; Mehmet, 2004; Scalabrin *et al*., 2000; Sharma et al., 1999; Petersen et al., 1994; Guimaraes and McGreavy, 1995; Ganguly, 2003).

The high cost of development and validation and large CPU time associated with complex numerical simulations have accelerated the endeavour to search for an alternative tool for VLE prediction instead of rigorous thermodynamic modelling. Conventional thermodynamic approach to predicting VLE data of acid gas-alkanolamine systems is based on 'activity coefficient' based models, hence, encourages a certain amount of empiricism and assumptions regarding the various interaction parameters involved in the equilibrated liquid phase (mixture of week electrolyte solutions). ANN process models extract the desired information directly from data, thus becomes cost effective. Long time required for training the net and over fitting the data are some of their disadvantages.

In the present work, for a wide range of temperature, partial pressure of CO_2 and for a wide range of concentration of MDEA and AMP solvents, the equilibrium liquid phase

CO_2 loading (α_{CO_2}) have been predicted based on ANN model. The equilibrium solubility determines the minimum circulation rate of the solvent through the absorber, determines the maximum allowable concentration of the acid gases in the regenerated solution in order to meet the product gas specification, and provides the boundary conditions for solving the partial differential equations describing mass transfer coupled with chemical reactions. The architecture of the network used in this work is multilayer feedforward network with backpropagation training algorithm. The effect of network architecture including the number of neurons in each hidden layers, number of hidden layers was studied on the prediction accuracy of the model.

2. ANN MODEL

2.1 Network Architecture

Neural network are typically organized in layers. Layers are made up of number of interconnected nodes, which may contain an activation function as described further. Apart from input and output layers, the intermediate ones are termed as hidden layers. The architecture of a network consists of a description of how many layers a network has, the number of neurons in each layer, each layer's transfer function and how the layers are connected to each other. The best architecture to be used is problem specific. Each input is weighted with an appropriate weight (w). Activation function defines the output of a neuron in terms of the activity level at its input, in other words it just operates on the weighted sum of the input. It introduces nonlinearity into the network, without which hidden units would not make effective nets. Almost any nonlinear function can be selected as activation function, but for back propagation learning it must be differentiable. The types of transfer function identified for the feed forward network is 'Log-sigmoid' and 'Trans-sigmoid' transfer function for first hidden layers of feed forward network and linear transfer function to the output layer. The output layer contains neurons, which are used as linear approximators in "*Linear filters*". Use of a nonlinear transfer function makes a network capable of storing nonlinear relation between input and output. Multiple feed forward layers give a grater freedom to the network. Network with biases can represent relationships between inputs and outputs more easily than networks without biases. Properly trained backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. There are generally four steps in the training process namely assembling the training data, creating the network object, training the network, and simulating the network response to new inputs.

Even though a number of techniques have been present for network topology selection, it still remains an iterative trial and error procedure (Sharma et al., 1999). Sharma et al., to reduce this trial and error selection process, used one heuristic approach. They determined the optimum architecture within 50-100 iterations without traversing the entire graph of absolute error as a function of the number of iterations for each topology. In the present work the optimum architecture evolved out through an elaborate trial and error procedure. NEWFF function of MATLAB 7 is used to create a new network. The input vector contains molar concentration of AMP / MDEA, operating temperature (Kelvin) and partial pressure of CO_2 (kPa.). The training of the network is done by abstracting experimental data (Kundu, 2004) is as follows:

For CO_2-AMP-H_2O system, concentration of AMP = {3.4, 2.0, 2.8}; Temperature (K) = {303, 313, 323, 333, 343, 353, 373}; Partial pressure of CO_2 (kPa) = { $P_{CO_2 Maximum} =$ 1000 and $P_{CO_2 Minimum} =$ 0.373}. For CO_2-MDEA-H_2O system, concentration of MDEA = {1.69, 2.0, 2.53, 3.04, 4.28}; Temperature (K) = {303, 313, 323, 333, 343, 353, 373}; Partial pressure of CO_2 (kPa) = { $P_{CO_2 Maximum} =$ 0.1 and $P_{CO_2 Minimum} =$ 5500}. The various combinations of these inputs results the experimental value of the '*loading*' (α_{CO_2}), corresponding to each

combinations. Since the data fed has to be normalized to equalize the magnitude of change observed in the experimentation and to train the network in accordance with this change. The normalization is done as per the following:

$$R_1 = \frac{r_{exp.} - r_{min.}}{r_{max.} - r_{min.}} \quad (1)$$

'*R*' is the normalized value of the parameter. '*r*' represents the various experimental results taken for consideration with the available maximum and minimum values in the experimental data stock. Hence three input vectors were framed using the normalized reported experimental values against concentration, temperature and pressure.

2.2 Network Training

During training the weights and biases of the network are iteratively adjusted to minimize the network performance function. The default performance function for feedforward networks is mean square error (mse) the average squared error between the network outputs *a* and the target outputs *t*. Standard backpropagation is a gradient descent algorithm in which the network weights are moved along the negative of the gradient of the performance function (Demuth and Beale, 1996). There are different batch steepest descent training functions, which are 'Batch gradient descent' (TRAINGD), 'Batch gradient descent with momentum' (TRAINGDM). The faster training algorithms fall into two main categories; heuristic techniques and standard numerical optimization techniques. Resilient back propagation (TRAINRP), was used in the present work. Multilayer networks typically use sigmoid transfer functions in the hidden layers. Sigmoid functions are characterized by the fact that their slope must approach zero as the input gets large. This causes a problem when using steepest descent to train a multilayer network with sigmoid functions, since the gradient can have a very small magnitude; and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. The purpose of the resilient backpropagation training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate updated value. The training parameters for TRAINRP are 'epochs', 'show, goal', 'time', 'min_grad', 'max_fail', 'delt_inc', 'delt_de'c, 'delta0', 'deltamax'. TRAINRP is generally much faster than the standard steepest descent algorithm (Neural Network Toolbox). It also has the nice property that it requires only a modest increase in memory requirements. We do need to store the update values for each weight and bias, which is equivalent to storage of the gradient. In construction of the neural model for VLE data of CO_2-MDEA-H_2O and CO_2-AMP-H_2O systems; 'TRAINRP' Matlab function is used as a network training function that updates weight and bias values according to the resilient back propagation algorithm (RPROP). The training data set required for adequate mapping was determined by a trial and error procedure. For CO_2-AMP-H_2O system 102 numbers and for CO_2-MDEA-H_2O system 217 numbers of data points were used for training the networks. The trained net was then used to simulate the VLE data, which were not used in training.

3. RESULTS AND DISCUSSIONS

3.1 CO_2-MDEA-H_2O System

For CO_2-MDEA-H_2O system, the best trained feed forward network containing three hidden layers, each layer having 40 neurons and LOGSIG transfer function used in all hidden layers followed by a PURELINE transfer function in the output layer (40 40 40 1) simulates 61 liquid phase CO_2 loading values, which were not used for training, with an average absolute deviation (AAD %) of 5.7 % when compared with the experimental results (Fig. 1). The training

performance versus epochs for a (30 30 30 1) network configuration is shown in Figure 2. TRAINRP was used with training parameters goal: 1e-005 and epochs 10000. The effect of different combinations of number of neurons and number of hidden layers on the prediction accuracy of the network is presented in Tables 1 and 2.

Table 1: Effect of number of neurons in Two-hidden layered Network performance

Epoch set=10000

Number of neurons for hidden layers: H1, H2

Sl. No.	H1	H2	% Prediction Error
1	25	25	11.0874
2	30	30	10.54
3	40	40	8.78
4	50	50	10.118

Table 2: Effect of number of neurons and hidden layers on Network performance

Epoch set=10000

Number of neurons for hidden layers: H1, H2, H3

Sl. No.	H1	H2	H3	% Prediction Error
1	20	20	20	6.18
2	25	25	25	12.5
3	30	30	30	6.180
4	40	40	40	5.744
5	50	50	50	9.5

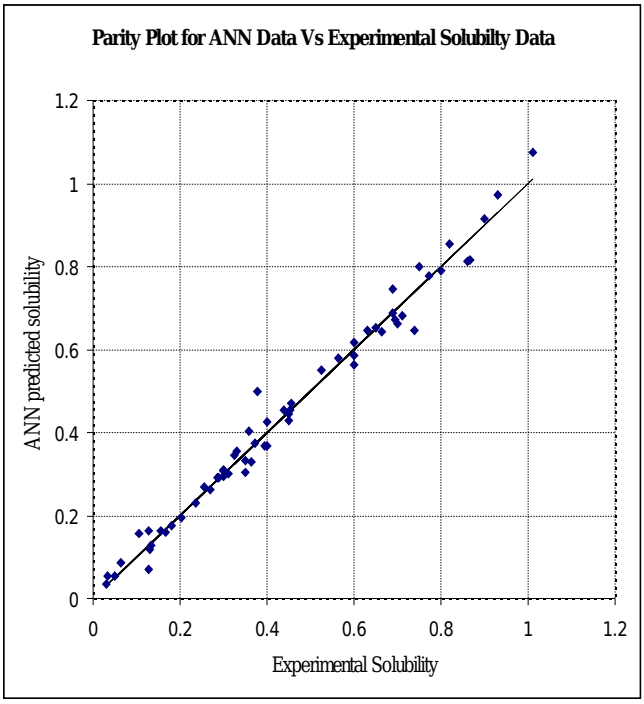

Fig 1: Experimental versus ANN predicted loading (α_{CO_2})

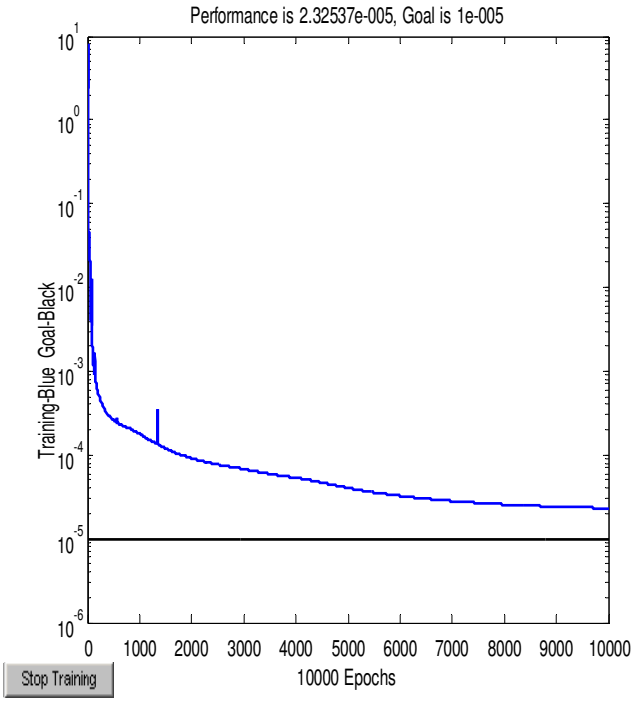

Fig. 2: Performance versus epochs in '30 30 30 1' network.

3.2 CO_2-AMP-H_2O System

For CO_2-AMP-H_2O system the trained

network using TRAINRP and having 10 LOGSIG neurons in two hidden layers was employed for simulating the following results, which were not used for training (Table 3). Figure 3 presents the typical structure of the network used for VLE prediction

Table 3 ANN predictions for the VLE of (CO_2-AMP-H_2O system)

Sl No.	Data Reference	[Conc] of AMP	Temp. in [K]	Average error % (on three runs basis)
1	Teng & Mather	2	343	*0.526066667*
2	Kundu.,M	2	303	*2.838933333*
3*	Kundu.,M	3.4	303	*0.3352*
4	Kundu.,M	3.4	313	*2.739166667*
5	Kundu.,M	3.4	323	*20.99233333*
6	Kundu.,M	2.8	303	*0.539633333*
7	Kundu.,M	2.8	313	*0.912033333*
8	Kundu.,M	2.8	323	*0.698333333*
9	Teng & Mather	2	313	*10.56223333*
AAD				4.46

It has been observed that Sl. No. 3* in the aforementioned table using 9 data points with a concentration of AMP 3.4 M and temperature of 373 K, the error of 0.3352 % is obtained which represents the least value. The effects of increasing the LOGSIG neurons are shown in Tables 4 and 5, respectively.

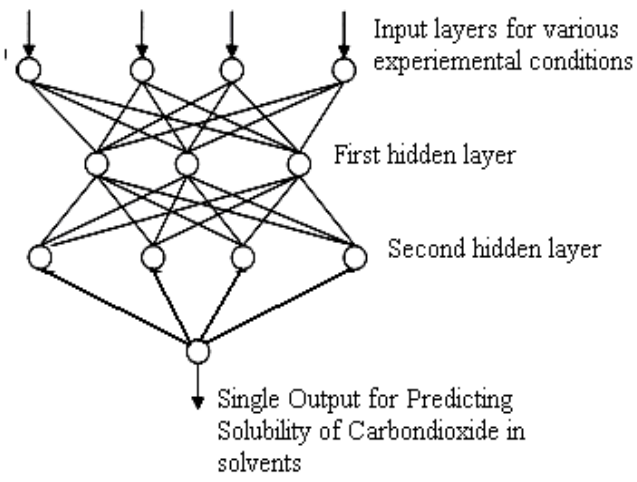

Fig 3: Typical Structure of network used as predictive VLE data (CO_2-AMP-H_2O) system

Table 4: Effect of number of neurons on Network performance for CO_2-AMP-H_2O system

Epochse=10000			
Number of neurons for hidden layers: H1, H2,H3			
Sl. No.	**H1**	**H2**	**% Prediction Error**
1	5	5	1.026
2	5	20	0.2979
3	5	50	1.2219
4	10	10	0.5585
5	20	5	0.3994
6	20	20	0.4839
7	50	50	0.22425

The performance of simulation (prediction using the trained network on such data sets, which were not used for training) also depends on the transfer function of the neurons connecting the hidden layers and the output layer. It was estimated that using LOGSIG as the transfer function for the single output neuron can enhance the prediction by 120 % for H1=10; H2=10 network and 130% for H1=20; H2=20 network. The network was observed stabilized with two hidden layers as from the insensitivity with the increase in the hidden layers (Table 5):

Table 5: Hidden layer sensitivity and the associated number of neurons to them on the performance of the Network for (CO_2-AMP-H_2O) system

Epoch set=10000				
Number of neurons for hidden layers: H1, H2,H3				
Sl. No.	H1	H2	H3	% Prediction Error
1	10	10	10	0.3731
2	5	10	20	0.435
3	20	20	20	0.1257
4	10	20	30	0.2036

It has been found that ANN predictions are in excellent agreement with the experimental results available in the open literature both for **(**CO_2-AMP-H_2O) and (CO_2-MDEA-H_2O) systems, which are regarded as a highly non-linear, involving multi-component; multiphase equilibria. Keeping in mind that the availability of a moderate to large number of data points for training and the non-linearity of the systems have been considered here, the present authors think that inclusion of three hidden layers for CO_2-MDEA-H_2O and two hidden layers for CO_2-AMP-H_2O system may not be completely undesirable for accurate prediction. While training the network using simpler architectures for CO_2-MDEA-H_2O system, most of the time performance goals were not met within a stipulated number of iterations. The reason behind it, could be the most non-ideal behavior of aqueous MDEA solution among the alkanolamine family.

4. ACKNOWLEDGMENT

The financial support by the Department of Science and Technology (DST), New Delhi, India, is gratefully acknowledged.

5. REFERENCES

Demuth, H.B., Beale, M.H. (1997). *Neural network toolbox users guide*, MathWorks, Inc. version 3.0.

Demuth, H.B., Beale, M.H. (1996). *Neural Network Design*, PWS Publishing Company, Boston, MA. Chapters 11-12.

Kundu, M. (2004). *Vapour – liquid equilibrium of CO_2 in alkanolamines*. Ph. D Thesis, Indian Institute of Technology, Kharagpur, India.

Teng, T. T., and Mather, A.E. (1989). *Solubility of H_2S, CO_2 and their mixtures in an AMP solution.* The Canadian Journal of Chemical Engineering, 67, 846 - 850.

S. Naidu, L. Santosh (2004). *Neural network surrogate model for multidisciplinary design optimization*. M.Tech Dissertation Report, Department of Aerospace Engineering, Indian Institute of Technology, Bombay, India.

Mehmet B. (2004). *The use of neural networks on VLE data prediction*. Journal of Scientific & Industrial Research, JSIRAC (63) (4) 203-312.

Mehmet B. (2004) *Isobaric vapour–liquid equilibrium calculations of binary systems using a neural network*. J. Serb. Chem. Soc., 69 (8–9) 669–674.

Scalabrin, G., Piazza, L., Cristofoli, G. (2000) *Application of Neural Networks to a predictive extended corresponding states model for pure halocarbon.* Thermodynamics Paper presented at the Fourteenth Symposium on Thermo physical Properties, June 25-30, 2000, Colorado, U.S.A.

Sharma, R., Singhal, D., Ghosh, R., Dwivedi, A. (1999). *Potential application of artificial neural network to thermodynamics: vapour-liquid equilibrium prediction*. Computers and Chemical Engineering ,23, 385-290.

Petersen, P., Fredenslund, A., Rasmussen, P. (1994). *Artificial neural networks as a predictive tool for vapour-liquid equilibrium.* Computers and Chemical Engineering, 18, S63-S67.

Ganguly, S. (2003). *Prediction of VLE Data using Radial Basis Function Network*. Computers and Chemical Engineering, 27,

1445-1454.

Guimaraes, P. R. B., McGreavy, C. (1995). *Flow of Information through an Artificial Neural Network*. Computers and Chemical Engineering, 19 (S1), 741.