Artificial Neural Network Classification of Phase Equilibrium Methods – Part 2

S. Oreški[a], J. Zupan[b] and P. Glavič[a,*]

[a]Faculty of Chemistry and Chemical Engineering, University of Maribor, Smetanova 17, P.O.Box 219, SI-2000 Maribor, Slovenia

[b]National Institute of Chemistry, Hajdrihova 19, P.O.Box 34–30, SI-1000 Ljubljana, Slovenia

Original scientific paper
Received: December 4, 2001
Accepted: April 8, 2002

A further study of the neural network application for predicting appropriate methods of phase equilibrium on the basis of known physical properties is presented. Kohonen neural networks are used to classify objects into none, one or more possible classes. The classes in the study represent possible methods of phase equilibrium. The trained neural network estimates the reliability of its predictions – the adequacy of individual methods of phase equilibrium for further efficient chemical process design and simulation. The analysis of the preliminary, less accurate results confirms the hypothesis to use Kohonen networks for classification of the classes as a correct one. Therefore, the Kohonen network architecture yielding the best separation of clusters was chosen for further analysis. It has been adapted and the training continued until the conflicting situations were resolved. Out of the several Kohonen networks trained the best one was analyzed. The maps of individual physical properties and the probability maps were obtained for each specific phase equilibrium. The correlation among maps is shown.

Keywords:

Physical properties; phase equilibrium; artificial neural networks; artificial neural network classification; learning procedure.

Introduction

A further study of a neural network application for classification and prediction of appropriate methods of physical equlibrium on the basis of known physical properties is presented. Many papers presenting tools to help an engineer in choosing a suitable phase equilibrium method have been published. In the article by *Baňares-Alcantara* and *Westerberg*[1] the steps followed to develop a prototype expert system CONPHYDE are described. Using the framework of the existing expert system PROSPECTOR, CONPHYDE is designed to aid engineers in the selection of an appropriate vapour-liquid equilibrium method. Similar is the hybrid system of *Kelly*, *Holste* and *Hall*[2] but it handles much broader set of properties and applications. Another knowledge based system for the selection of thermodynamic models, TMS,[3,4] has similar objectives but has an alternative structural methodology. All the systems mentioned need inference networks to be previously built up. An exception is the expert system PHYP by *Oreski* and *Glavic*,[5] which has the same purpose but uses rules automatically generated by its shell instead of the inference networks.

In the field of phase equilibria, several applications of neural networks exist. We shall mention only some of them. Artificial neural networks are used as a part of/or a complete predictive tool for vapour-liquid equilibrium. For example, *Petersen*, *Fredenslund* and *Rasmussen*[6] used artificial neural network as a new group contribution method. *Habiballah*, *Startzman* and *Barrufet*[7] used it for prediction of vapour-liquid equilibrium *K*-values. The possibility of applying neural networks for vapour-liquid equilibrium data prediction/estimation of methane-ethane and ammonia-water systems has been explored using the back propagation algorithm by *Sharma*, *Singhal*, *Ghosh* and *Dwivedi*.[8] Neural networks were used as a part of predictive tool by *Alvarez*, *Riverol*, *Correa* and *Navaza*,[9] where a combined mixing rule for the prediction of vapour-liquid equilibrium was designed using neural networks. There are some applications of neural networks for prediction of liquid-liquid equilibrium. An article by *Bogdan*, *Gosak* and *Vasic-Racki*[10] can be mentioned where mathematical modeling of liquid-liquid equilibrium using neural network is represented. Some applications in solid-liquid equilibrium predictions exist.[11,12] *Buenz*, *Braun* and *Janowsky*[13] involved neural networks in correlating and predicting physical properties of pure components and mixtures from molecular structure. Appli-

*Corresponding author. Tel.:+386 2 22 94 451; fax:+386 2 25 27 774; e-mail: glavic@uni-mb.si

cations of neural networks to predict melting points exist, too.[14,15] It can be resumed that in the field of phase equilibrium the neural networks have been used for prediction only and not for classification so far.

Previous work

The basic idea to apply a neural network for classification in the field of phase equlibrium was first introduced in the previous article[16]. There, a neural network is used to classify and predict appropriate methods of phase equlibrium on the basis of known physical properties. Four main characteristics were exposed for the application:

– A large number of data exists regarding different combinations of physical properties with an appropriate method of phase equilibrium attached.

– The available data do not describe all the possibilities – the domain is not covered by all the possible combinations of chemical components, concentrations, working temperatures, pressures, etc. (some of these combinations are not yet known, some of them are not possible).

– The classification is to be made by the neural networks – they must be able to attach none, one or more possible phase equilibrium methods to the data known.

– Neural networks must also be able to estimate the reliability of the phase equilibrium methods proposed.

The number of data are to be correlated with the number of neurons needed and the time for training. Supervised training is not appropriate owing to the large number of data. An incomplete domain requires the unsupervised approach because all the responses are not available. The classification can be carried out in a supervised or an unsupervised manner if the number of objects is not too large. But for the classification of a large number of objects, unsupervised learning strategy is more efficient. According to the nature of the problem we were trying to solve, the Kohonen neural network was employed among several different neural networks as one with the most appropriate architecture and learning strategy.

In the previous article,[16] several smaller Kohonen neural networks of a size appropriate for the given number of objects were trained at different learning steps (epochs) after the data preprocessing procedure. Each object was associated with a multi-dimensional target whose components represented classes of phase equlibria. When increasing the number of epochs in the training the number of activated neurons increased, while errors of learning and the number of conflicting situations were reduced. Regardless of the prolongation of the training time the number of conflicts in any neural network could not be diminished below 16 %. This is due to very similar representations of objects that "excite" the same neuron, i.e., those objects that cause the conflicts. Anyway, a few conclusions could be made:

– After training, the final weights of the 'winning' neuron in the Kohonen neural network were proportional to the probabilities of the corresponding classes to which the unknown object should belong.

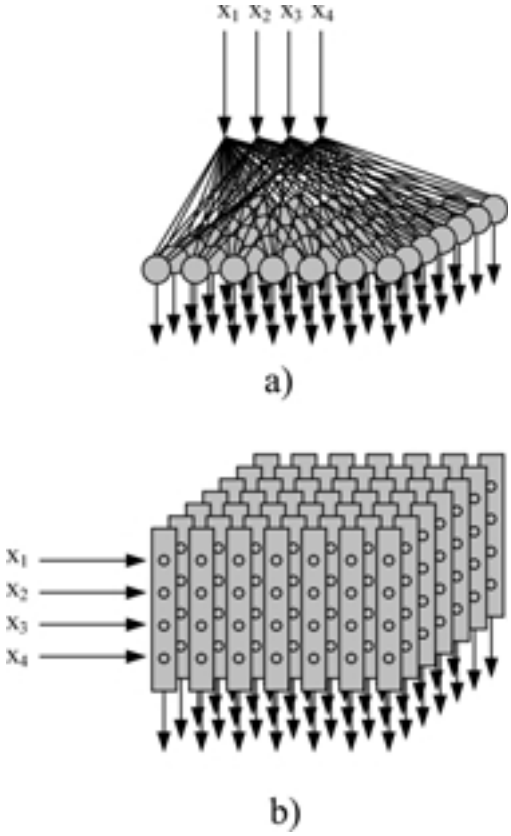
– The trained Kohonen neural network is able to classify objects into none, one or more possible classes where the classes represent possible methods of phase equilibrium.

– The trained neural network can estimate the reliability of the predictions, indicating the adequacy of individual methods of phase equilibrium.

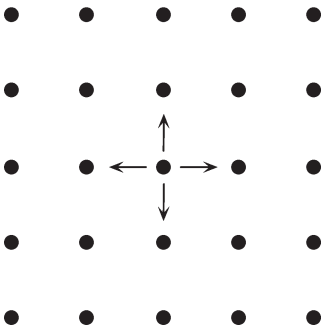
Kohonen network architecture and learning algorithm

Analysis of the preliminary, less accurate results, confirmed that our hypothesis to use Kohonen networks for separation of the classes was correct. Therefore, the network architecture yielding the best separation of clusters was chosen for further analysis. In the previous article the Kohonen network architecture and the learning algorithm were represented in detail[16]. Therefore, only a brief description of the architecture and learning strategy chosen for the given problem is presented here.[17]

The Kohonen network is based on a single layer of neurons arranged in a two-dimensional plane. Two different visual presentations of Kohonen networks are possible. Figure 1 shows both presentations. We have chosen the matrix one. The neighbourhood of a neuron is arranged in squares. In the Kohonen conception of neural networks, the signal similarity is related to the topological relations among the neurons in the network. Therefore, each neuron has four nearest neighbours (Figure 2). From the practical point, each neuron in the net is ensured with the same number of first-, second-, etc. neighbours (Figure 3). The Kohonen learning tries to map the input so that similar signals excite neurons that are very close together – the aim is to map similar signals to similar neuron positions. In a Kohonen network there are two layers: inactive input and active output one. The latter is arranged as a two-dimensional grid (Figure 1b). All the neurons in the active layer receive the same multidimensional input. However, the output of each neuron is



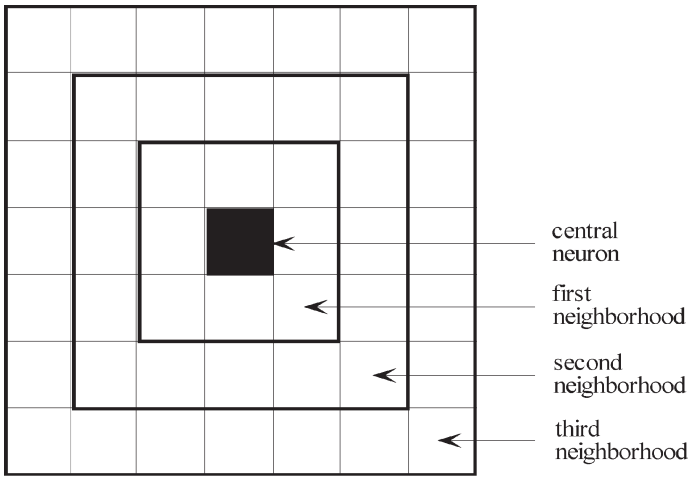
F i g . 1 – *Two-dimensional layout of a Kohonen neural network can be represented a) in a "flow-chart" manner or b) in a matrix manner. The matrix description for the Kohonen network is much more convenient because it shows the relation between the input data* $\boldsymbol{X}_s$ *(*x_{s1},x_{s2}...x_{sm}*) and the planes of weights very clear.*



F i g . 2 – *Square layout of neighbours*

not connected to all other neurons in the plane, but only to a small number of them, those being topologically close to it. The local feedback of possible corrections makes the topologically close neurons behave similarly when inputting similar signals.

The Kohonen learning procedure is unsupervised learning. It is called a competitive learning or the "winner takes all" method. In summary, the algorithm for one cycle of Kohonen learning for the problem given is as follows:



F i g . 3 – *The square neighbourhood having 8, 16, 24, etc. neighbours in concentric neighbourhoods*

– an m-dimensional object $\boldsymbol{X}_s$ enters the network;

– the responses of all neurons (each having m weights) are calculated;

– the position c is found for the neuron whose output is the most similar to the input:

$$c \leftarrow \min\left\{\sum_{i=1}^{m}(x_{si} - w_{ji})^2\right\}, \quad j = 1, 2, \ldots, n \quad (1)$$

(The index j refers to a particular neuron, n is the number of neurons; m is the number of weights per neuron; s identifies a particular input.)

– the weights of the neuron c are corrected to improve its response for the same input $\boldsymbol{X}$ in the next cycle;

– the weights of all the neurons in the neighbourhood of the c-th neuron are corrected by a fraction that decreases with increasing topological distance from c:

$$w_{ji}^{(new)} = w_{ji}^{(old)} + \eta(t)\, a(d_{c-j})(x_{si} - w_{ji}^{(old)}) \quad (2)$$

(Here, x_i is a component of the input $\boldsymbol{X}_s$; the central neuron is designated by c, and the one being corrected by j; a particular weight of the neuron j (and a particular input) is designated by i; t is (related to) the iteration cycle in question.)

– The next m-variate object $\boldsymbol{X}_s$ is input and the process repeated.

In the Equation 2, the corresponding scaling function $a(d_{c-j})$ is a topology dependent function, where d_{c-j} is the topological distance between the central neuron c and the current neuron j, and has a triangular form (Figure 4). Corrections are decreasing with each iteration step what is represented with another monotonically decreasing function $\eta(t)$. In

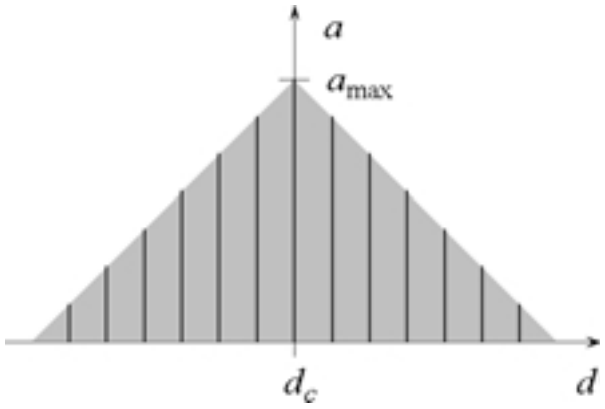


Fig. 4 – *A triangular function for scaling corrections on neighbourweights*

this function, t is the number of objects entered into the training process; t can be associated with time as well, since the time used for training is proportional to the number of objects entering the network.

Training of larger neural networks with the original database

In the previous article[16] the database consisted of 3780 objects $\boldsymbol{X}_s$ arranged in 41-dimensional vectors appropriate for training neural networks. The first twenty-six dimensions are representing nine original physical properties of samples (objects), describing chemical bonds, structure of the components, working conditions, further calculations desired, accuracy of the methods, simplicity and speed of calculations, and data availability. The last fifteen dimensions of the vector $\boldsymbol{X}_s$ are representing a target vector $\boldsymbol{Y}$ of the fifteen phase equilibrium methods, most often used in practice. Six of them: *Soave-Redlich-Kwong*[18] and *Peng-Robinson*[19] cubic equations, *Benedict-Webb-Rubin*,[20] *Starling*[21–23] and *Lee-Kesler*[24] multi-parametric equations and theoretical *Virial*[25] equation belong to the equations of state. The remaining nine: *Margules-1* and *2* methods,[26,28] slightly more complex *van Laar*[27,28] and complex *Wilson*,[29,30] *NRTL*,[31] *UNIQUAC*,[32] *ASOG*[33] and original *UNIFAC*[34,35] methods as well as the *Regular Solution Theory*[36] belong to the activity coefficient methods. Besides the original papers and books about the methods of phase equilibria encoded into the bank of objects, some useful rewiews can be found in Oellrich et al.,[37] Prausnitz et al.,[38] Reid et al.,[39] Malanovski and Anderko,[40] etc.

Several square Kohonen networks of dimensions from 50×50 (space for 2500 neurons) to 70×70 (space for 4900 neurons) with 41 weights in each neuron were trained at different learning steps (epochs) according to the criterion that the winner c is the neuron having the weight vector $\boldsymbol{W}_j(w_{j1}, w_{j2}\ldots w_{jm})$ most similar to the input signal $\boldsymbol{X}_s(x_{s1}, x_{s2}\ldots x_{sm})$ (Eq. 1). During the training, the weights of all the neurons in the neighbourhood of the c-th neuron were corrected according to the correction function for the criterion with the output signal most similar to the input one, $\boldsymbol{X}_s$ (Eq. 2). The learning rate term $\eta(t)$ was decreasing linearly from 0,5 at the beginning of the training to 0,01 at the end of it. The triangular neighbourhood function $a(d_{c-j})$ was used for scaling corrections on neighbours' weights (Fig. 4).

The results are collected in Figure 5. The number of activated neurons is increasing with the ris-

Network's configuration	No. of epochs	No. of neurons	No. of active neur.	Total RMS	RMS/neuron	RMS/weight	No. of conflicts
50x50	200	2500	929	56,603	0,921	0,144	630
50x50	500	2500	1075	52,567	0,855	0,134	563
50x50	900	2500	1164	47,961	0,780	0,122	701
50x50	1500	2500	1253	47,863	0,778	0,122	602
60x60	200	3600	1069	56,837	0,924	0,144	583
60x60	500	3600	1389	48,656	0,791	0,124	651
60x60	900	3600	1485	45,656	0,743	0,116	636
60x60	1500	3600	1601	43,805	0,712	0,111	553
60x60	2000	3600	1547	42,348	0,689	0,108	573
65x65	500	4225	1420	49,038	0,798	0,125	650
70x70	900	4900	1609	43,016	0,700	0,105	562
70x70	2000	4900	1966	36,771	0,598	0,093	653

Fig. 5 – *Results of training Kohonen neural networks on the base of original object.*

ing number of epochs at the neural networks of the same dimensions. The number of the activated neurons is also increasing when neural networks of bigger dimensions are trained. The more neurons are activated the less objects activate the same neuron. It is also evident that the sum of errors in one epoch (total RMS value), the average error at one object (RMS value/neuron), and the average error at one weight (RMS/weight) can be reduced by increasing the number of epochs when neural networks of the same dimensions are trained. Also, all the three kinds of errors are diminishing with increasing dimensionality of neural networks trained in the same number of epochs. For instance, when training the 50x50 Kohonen network with 200 epochs the sum of errors in one epoch is 56,603, the average error at one object is 0,921 and the average error at one weight is 0,144. Training the Kohonen network with 2000 epochs the values 36,771, 0,598 and 0,093 are obtained, respectively. On the other hand no specific rule for conflicting situations could be deduced. The number of conflicting situations (the situations where expected and unexpected objects activate the same neuron) vary between 553 in the 60x60 Kohonen neural network trained with 1500 epochs and 701 in the 50x50 Kohonen network trained with 900 epochs. The fraction of conflicting situations could not be diminished below 14,63 %.

Class	No. of conflicting objects
SRK	1
PR	19
BWR	2
STARLING	10
LKP	10
VIRIAL	0
MARGULES-1	6
MARGULES-2	18
VAN LAAR	26
WILSON	18
UNIQUAC	400
NRTL	43
ASOG	20
UNIFAC	240
REGULAR SOLUTION	2

Fig. 6 – *A distribution of conflicting objects among classes*

The results were more precisely analyzed for the 70x70 Kohonen neural network trained with 900 epochs. It had 562 conflicting situations, what means that a certain number of activated neurons were "excited" with non-conflicting objects and one or more conflicting ones (altogether there are 562 conflicting objects). Distribution of conflicting objects among classes is shown in Figure 6. From the figure it can be seen that equations of state participated 42 times and activity coefficient methods 773 times, the latter having 95,34 % of conflicting objects. *Five neurons* activated with non-conflicting and conflicting objects have been chosen randomly for illustration.

• *First example* is the neuron in position (1,1) of the network:

```
neuron( 1, 1)          Label(Kohonen): L
        737 conflict   Label(origin.): S
        745 conflict   Label(origin.): S
        749 conflict   Label(origin.): S
        753 ok         Label(origin.): L
        761 ok         Label(origin.): L
        765 ok         Label(origin.): L.
```

The label (Kohonen) is determinated by the Kohonen neural network. The label (origin.) is set by the user. When the original label is not the same as the Kohonen label the learning object is the conflicting one. The neuron(1,1) is activated by six learning objects. Objects 737, 745 and 749 are conflicting objects while objects 753, 761 and 765 are non-conflicting ones. The learning objects are:

```
737 3 1 1 2 4 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
745 3 1 1 2 2 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
749 3 1 1 2 1 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
753 3 1 1 1 4 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
761 3 1 1 1 2 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
765 3 1 1 1 1 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0.
```

The first column contains the object number, the following nine columns represent the nine physical properties, and the last fifteen ones indicate the methods of phase equilibrium.

• *Second example* is the neuron(1,17):

```
neuron( 1,17)          Label(Kohonen): L
        145 conflict   Label(origin.): S
        157 ok         Label(origin.): L
        169 conflict   Label(origin.): S
        181 ok         Label(origin.): L
        193 conflict   Label(origin.): S
        205 ok         Label(origin.): L
        217 conflict   Label(origin.): S
        229 ok         Label(origin.): L,
```

which is activated by eight learning objects:

```
145 5 2 2 2 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
157 5 2 2 1 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
169 5 2 1 2 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
181 5 2 1 1 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
193 5 1 2 2 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
205 5 1 2 1 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
217 5 1 1 2 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
229 5 1 1 1 3 7 2 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0.
```

regarding the time needed for training. The training time increases proportionally when increasing the number of epochs, but increases exponentially when increasing the dimensions of the neural networks trained. Conflicting neurons remain the unsolved problem. Better success can be achieved with interventions in the bank of objects to improve their information content or increase the selectivity of the representation.

• *Third example* is the neuron (5,70):

```
neuron( 5,70)          Label(Kohonen): L
         28 conflict   Label(origin.): S
         32 conflict   Label(origin.): S
         40 ok         Label(origin.): L
         44 ok         Label(origin.): L,
```

activated by four learning objects:

```
28 5 5 1 2 3 1 1 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
32 5 5 1 2 2 1 1 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
40 5 5 1 1 3 1 1 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0
44 5 5 1 1 2 1 1 2 1  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0.
```

• *Fourth example* is the neuron (7,18):

```
neuron( 7,18)          Label(Kohonen): L
       2413 conflict   Label(origin.): S
       2457 ok         Label(origin.): L
       2533 conflict   Label(origin.): S
       2577 ok         Label(origin.): L,
```

also activated by four learning objects:

```
2413 1 5 2 2 3 5 2 1 1  0  0  0  0  0  0  7  8  9  0  0  0  0  0  0
2457 1 5 2 1 3 5 2 1 1  0  0  0  0  0  0  7  8  9  0  0  0  0  0  0
2533 1 5 1 2 3 5 2 1 1  0  0  0  0  0  0  7  8  9  0  0  0  0  0  0
2577 1 5 1 1 3 5 2 1 1  0  0  0  0  0  0  7  8  9  0  0  0  0  0  0.
```

• And the *last* example is the neuron (14,1):

```
neuron(14, 1)          Label(Kohonen): L
        371 conflict   Label(origin.): S
        415 ok         Label(origin.): L,
```

which is agitated by two learning objects:

```
371 3 5 1 2 1 1 1 2 2  0  0  0  0  0  0  0  0  0  0  0 13 14  0
415 3 5 1 1 1 1 1 2 2  0  0  0  0  0  0  0  0  0  0  0 13 14  0.
```

The examples illustrate that the conflicting object differs from the non-conflicting one only in the fourth dimension x_4 of the $\boldsymbol{X}_s$ vector representing the temperature. When inspecting several other conflicting neurons it was found out that they were activated by the objects $\boldsymbol{X}_s$, differing mutually only in values of the 'no-binary' variable x_4 (the temperature) again. The difference in the value of one variable (temperature) only is obviously not informative enough to enable the Kohonen neural networks to be more precise.

A conclusion can be made that some success was achieved when larger Kohonen neural networks with more epochs were trained with the original objects. But a gain on increasing the number of activated neurons and decreasing learning errors has too little effect

Information content improvements and preprocessing of data

When analyzing the conflicting neurons it was found out that almost all of them were located in saturated vapour-liquid regions and associated with the activity coefficient methods. Usually, the Kohonen learning proposes a liquid phase when a two-phase vapour-liquid region was expected. Because the activity coefficient methods are applicable to the liquid part of the two-phase region (the vapour phase is ideal or it is simulated with an appropriate equation of state), these conflicting neurons were not considered as being very controversial. Opposite to the activity coefficient methods, the equations of states are useful in both phases of saturated regions, vapour and liquid.

This additional knowledge is added to the original vectors $\boldsymbol{X}$ (y,x_1…x_9) using two new dimensions, x_{10} and x_{11}. The corrected vectors $\boldsymbol{X}$ (y,x_1…x_9,x_{10}, x_{11},) include a precise information about applicability of all the methods of phase equilibrium in saturated regions. Now, the variables x_i of vector $\boldsymbol{X}$ (y,x_1…x_9,x_{10},x_{11},) are representing physical properties of samples (objects), and the dimension y is one appropriate method of phase equilibrium out of the fifteen possible attached. The same methods of phase equilibrium as in the original bank of objects were retained as possible classes. Similar preprocessing procedure as in the previous article,[16] including 'bubble' and 'quick' sort techniques, detection and elimination of identical objects, and simple object classification, was to be executed again. In the same manner the refined and improved user-friendly objects were transformed into the objects suitable for training Kohonen neural networks. The transformation itself is explained in details in the previous article. After the transformation each object was represented as a multidimensional vector

$\boldsymbol{X}_s$ ($x_{s1} \ldots x_{s31}$, $y_1 = x_{s32} \ldots y_{15} = x_{s46}$). Altogether, the final representation consisted of 46 variables x_{si} (31 plus 15 representing 11 different variables x_i and a target vector $\boldsymbol{Y}$, respectively). Finally, in the database of objects 4228 objects $\boldsymbol{X}_s$ arranged in the 46-dimensional vectors appropriate for training neural networks, were obtained. In Figure 7, a structure of the improved bank of objects is represented in dimensions of vectors $\boldsymbol{X}$, $\boldsymbol{X}_s$ and target vector $\boldsymbol{Y}$.

Training of neural networks with improved objects

According to the number of objects, several Kohonen neural networks, having sizes from 50x50 to 70x70 with 46 weights in each neuron, were estimated to be appropriate for training. The neural networks were trained at different epochs using competitive learning according to the criterion that weight vector $\boldsymbol{W}_j(w_{j1}, w_{j2} \ldots w_{jm})$ was the most similar to the input signal $\boldsymbol{X}_s(x_{s1}, x_{s2} \ldots x_{sm})$ (Eq. 1). A corresponding correction function was used for correcting weights (Eq.2). Again, the triangular neighbourhood function $\boldsymbol{a}(\boldsymbol{d}_{c-j})$ was used for scaling corrections on weights of neighbours (see Figure 4) and the learning rate term $\eta(t)$ was decreasing linearly between 0,5 and 0,1. Besides, the labels "V" for representing homogeneous vapour regions, "S" for representing heterogeneous vapour-liquid, and "L" for homogeneous liquid phase regions, a new label "l" was introduced to locate heterogeneous liquid-liquid regions on the Kohonen map. When analysing the results it is found out that in global the errors of training are decreasing by increasing the size of networks and the number of epochs used for training. The errors are in the limits typical for Kohonen type of neural networks. No conflicting neurons are detected.

Results for a typical trained Kohonen neural network

Explanation of maps

In Figure 8 the Kohonen map is containing four different (coloured in the computer output) labels "V", "S", "L", "l", and no label is represented. The map is a result of the 70x70 Kohonen neural network trained with 4228 46-dimensional objects after 900 epochs. In the map labels represent active neurons activated by one or more objects. It can be seen that active neurons are distributed rather evenly through the whole Kohonen map. Labels "V", "S", "L", and "l" are grouped in separate clusters representing regions of homogeneous vapour

Phase equilibrium method	X	X_s	Y
SRK	x_1	x_{s32}	y_1
PR	x_2	x_{s33}	y_2
BWR	x_3	x_{s34}	y_3
STARLING	x_4	x_{s35}	y_4
LKPKP	x_5	x_{s36}	y_5
VIRIAL	x_6	x_{s37}	y_6
MARGULES-1	x_7	x_{s38}	y_7
MARGULES-2	x_8	x_{s39}	y_8
VAN LAAR	x_9	x_{s40}	y_9
WILSON	x_{10}	x_{s41}	y_{10}
UNIQUAC	x_{11}	x_{s42}	y_{11}
NRTL	x_{12}	x_{s43}	y_{12}
ASOG	x_{13}	x_{s44}	y_{13}
UNIFAC	x_{14}	x_{s45}	y_{14}
REGULAR SOLUTION	x_{15}	x_{s46}	y_{15}

Physical property	X	X_s		
CHEMICAL BOND	x_1	x_{s1}	x_{s2}	x_{s3}
NonPolar	1	1	0	0
Slightly Polar	2	2	0	0
Polar	3	3	0	0
Electrolyte	4	0	1	0
Polymer	5	0	0	1
.	.		.	
.	.		.	
.	.		.	
PRESSURE	x_3	x_{s9}	x_{s10}	
Low	1	1	0	
Medium	2	2	0	
High	3	3	0	
p_c	4	0	1	
$0<p_r<10$	5	0	2	
TEMPERATURE	x_4	x_{s11}		
$T<T_b$	1	1		
$T_b \le T \le T_d$	2	2		
$T_d<T<T_c$	3	3		
$T=T_c$	4	4		
$T_c<T<4T_c$	5	5		
.	.		.	
.	.		.	
.	.		.	

Fig. 7 – *Structure of improved bank of objects, expressed by* $\boldsymbol{X}$, $\boldsymbol{X}_s$ *and target vector* $\boldsymbol{Y}$

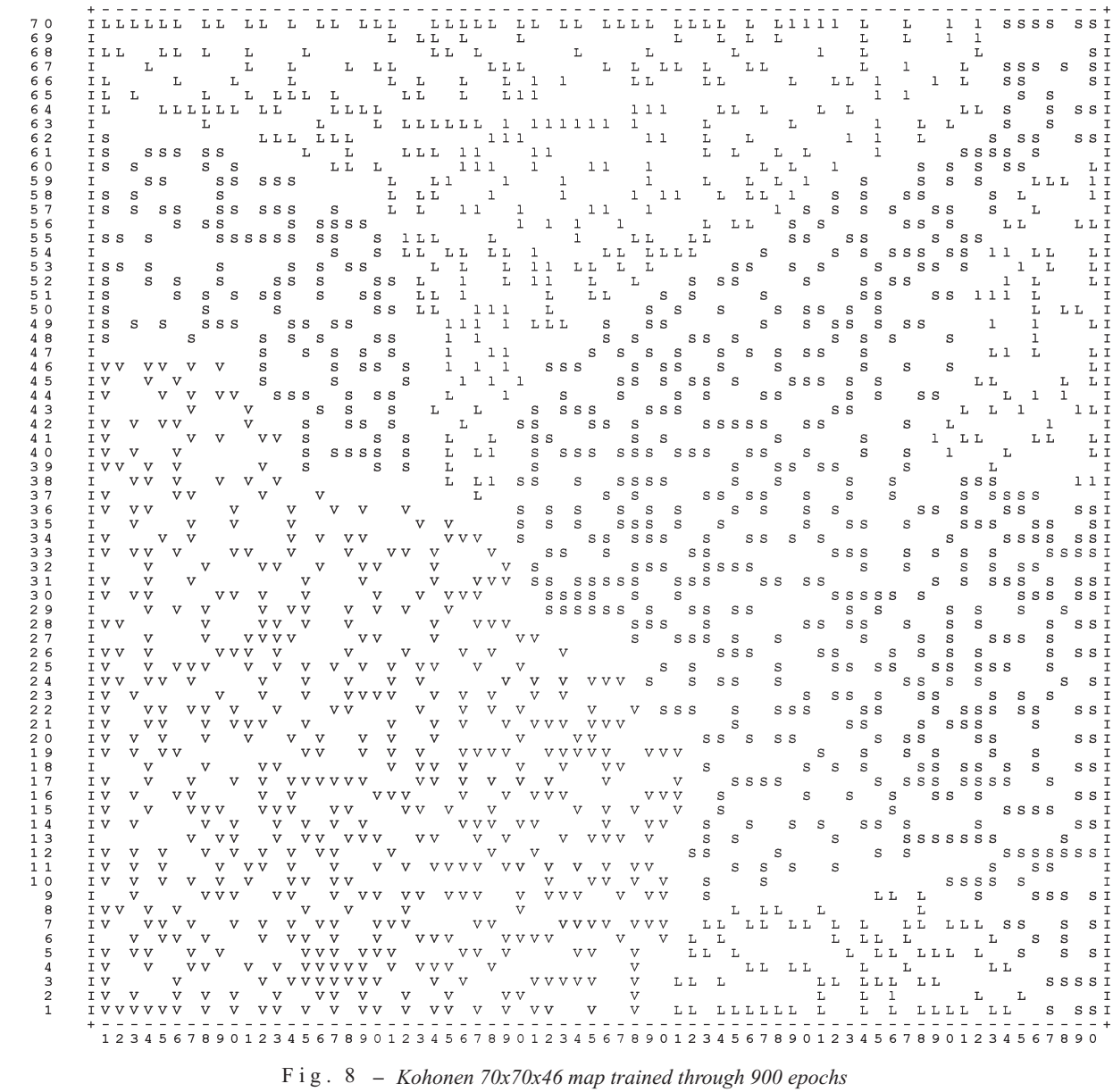


Fig. 8 – *Kohonen 70x70x46 map trained through 900 epochs*

phase, heterogeneous vapour-liquid phase, homogeneous liquid phase and heterogeneous liquid-liquid phase, respectively. Every labeled neuron carries information about the adequacy of at least one phase equilibrium method for specific combination of physical properties. Neurons not excited by any of the 4228 objects are represented by vacancies.

In the Kohonen neural network the weights affected by each variable are lying on a single and well-defined level of weights. There are as many weight levels as there are input variables describing the objects for which the network is designed. Each level of weights can be represented as a map. In our case Kohonen neural networks were trained with 46-dimensional input objects. 46 maps can be obtained for each input variable separately: the first 31 two-dimensional maps for weights w_{j1} to w_{j31} are representing physical properties, the last 15 two-dimensional maps for weights w_{j32} to w_{j46} are probability maps representing the 15 methods of phase equilibrium.

Maps of physical properties

There are two kinds of maps representing physical properties. If there is no need for binary substitution, all the possible values of the physical property attributes are represented in one map. Such an example is the physical property *temperature* with five possible attributes (see Figure 9). The attributes indicate the increase of temperature from $T<T_b$ to $T_c<T<4T_c$, hence they could be coded in one variable with values 1, 2, 3, 4, and 5, respectively. Binary substitution was needed when attributes of cer-

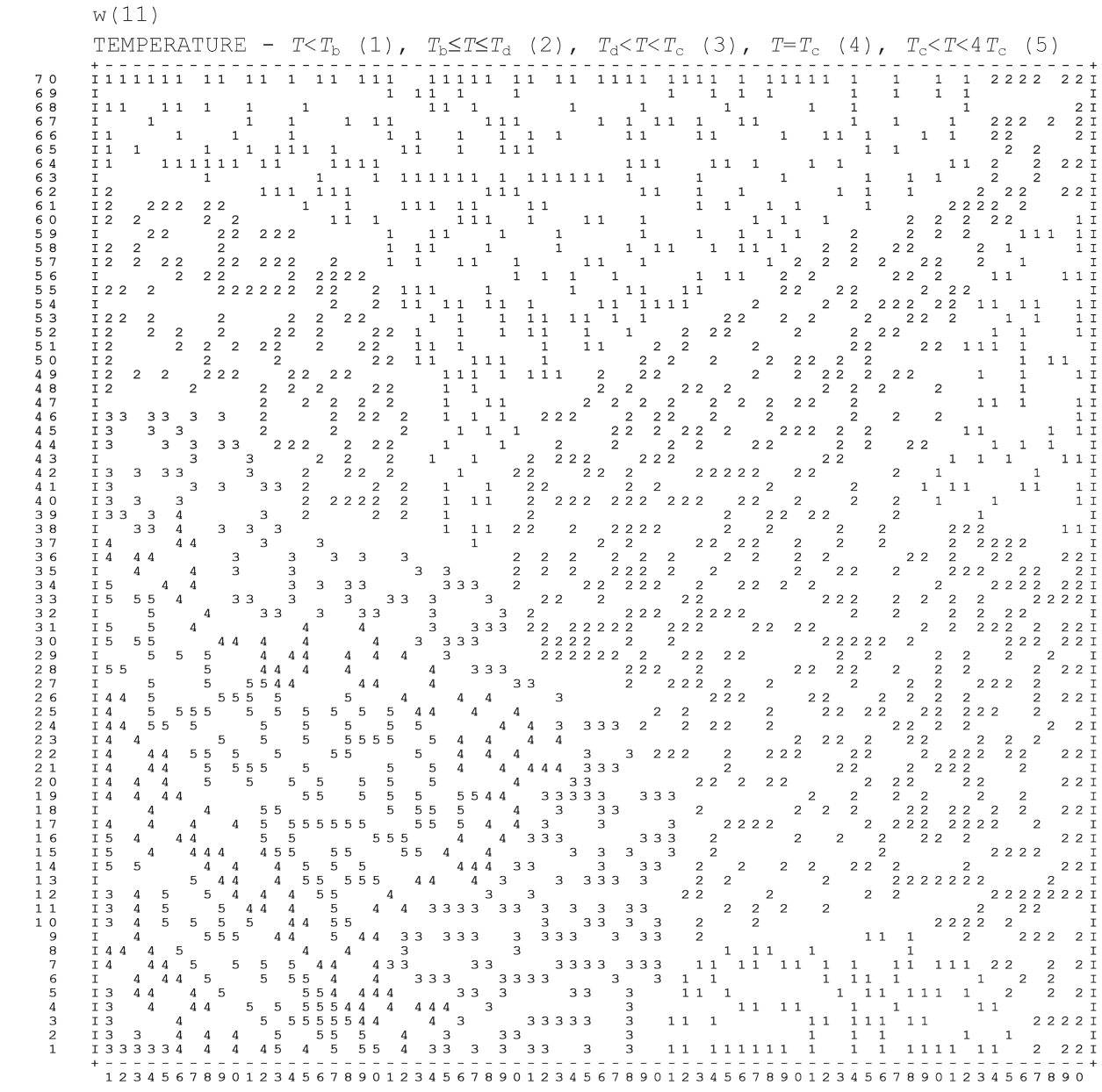


Fig. 9 – *Map representing physical property temperature.*

tain physical property or some of them were not correlated mutually or with the others.

For example the physical property *chemical bond* has attributes of both types. The first three attributes indicate the increase of polarity from non-polar to polar. Hence, they were coded in one variable with values 1, 2, and 3 and represented in one map. The last two attributes (mixture with the presence of an electrolyte, and mixture with the presence of a polymer, respectively) are not correlated neither among themselves nor with the first three ones. Therefore, they were transformed binary and are represented on two separate maps with values 1. In Figure 10 a separate map for a mixture with the presence of a polymer is represented. Altogether, the physical property *chemical bond* is represented with three separate maps for dimensions w_{j1}, w_{j2}, and w_{j3} of the vector $\boldsymbol{W}_j$.

Another example is the physical property *composition* with attributes representing systems consisting of normal fluids (rare gases, N_2, O_2, CO_2, hydrocarbons), systems containing mixtures with wide boiling temperatures, systems containing H_2 (causing deviations from the real behaviour with some phase equilibrium methods), systems consisting of a large range of components (for example: hydrocarbons, ketones, esters, water, amines, alcohols, nitriles, etc.), and systems consisting of other organic components. The attributes are not correlated mutually. *Composition* is represented with five

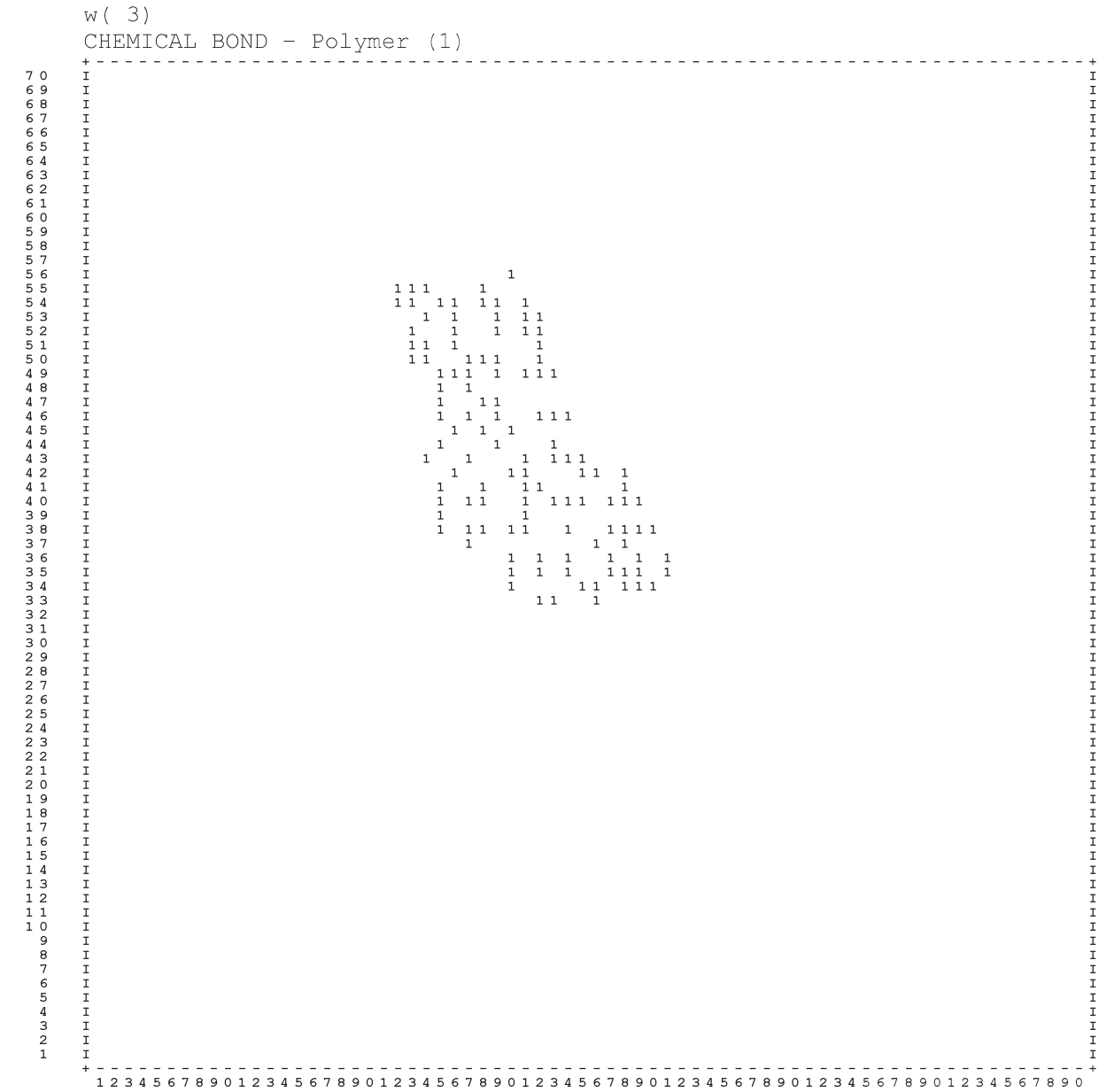


Fig. 10 – *Separate map representing one of the attributes of physical property – chemical bond in polymer*

separate maps for dimensions w_{j4}, w_{j5}, w_{j6}, w_{j7}, and w_{j8} of the vector $\boldsymbol{W}_j$.

Probability maps

The trained weights in probability maps carry original values between 0,0 and 1,0. The original values of weights are scaled in the interval (1–9) for easier physical presentation of maps (one digit of adequate power on one neuron). The outlook of the main Kohonen map and the corresponding probability maps is different from the maps of the trained, less accurate Kohonen neural network represented in the previous article[16]. Despite the dissimilarity, inspecting all the probability maps the former observations can be confirmed:

– When overlapping a separate probability map with the Kohonen map it can be seen in which region (vapour, vapour-liquid, liquid or liquid-liquid) the neurons are activated for the phase equilibrium method defined.

– Equations of state appear in regions “V”, “S”, and “L” (an exception is *Virial* equation, the use of which is limited only to vapour phase).

– The activity coefficient methods appear in regions “S”, “L” and “l”.

– The applicability of a phase equilibrium method is increasing by rising the number of activated neurons on one probability map and vice versa.

– Specific clusters on a probability map indicate specific use of a phase equilibrium method.

– When having similar phase equilibrium methods similar neurons on the probability maps are activated.

– Equations of state and activity coefficient methods can involve the same activated neurons, when they are appropriate for the same objects.

– The same neurons can be activated in different probability maps with the same or different strengths. In that way information about appropriate methods of phase equilibrium and their reliability is obtained.

Examples

As examples of the probability maps, two maps for the two phase equilibrium methods frequently used in practice are represented, the *Soave-Redlich-Kwong* method and the *UNIFAC* method.

1. SRK method

The probability map for the *Soave-Redlich-Kwong* method, one of the equation of state is represented in Figure 11. When overlapping the probability map with the main Kohonen map it can be seen that this equation of state method appears in regions “V”, “S”, and “L”. Comparing the probability map with maps of physical properties, more detailed information about correlation between phase equilibrium method and physical properties, can be obtained. When components are non-polar or slightly polar, the method is applicable to systems

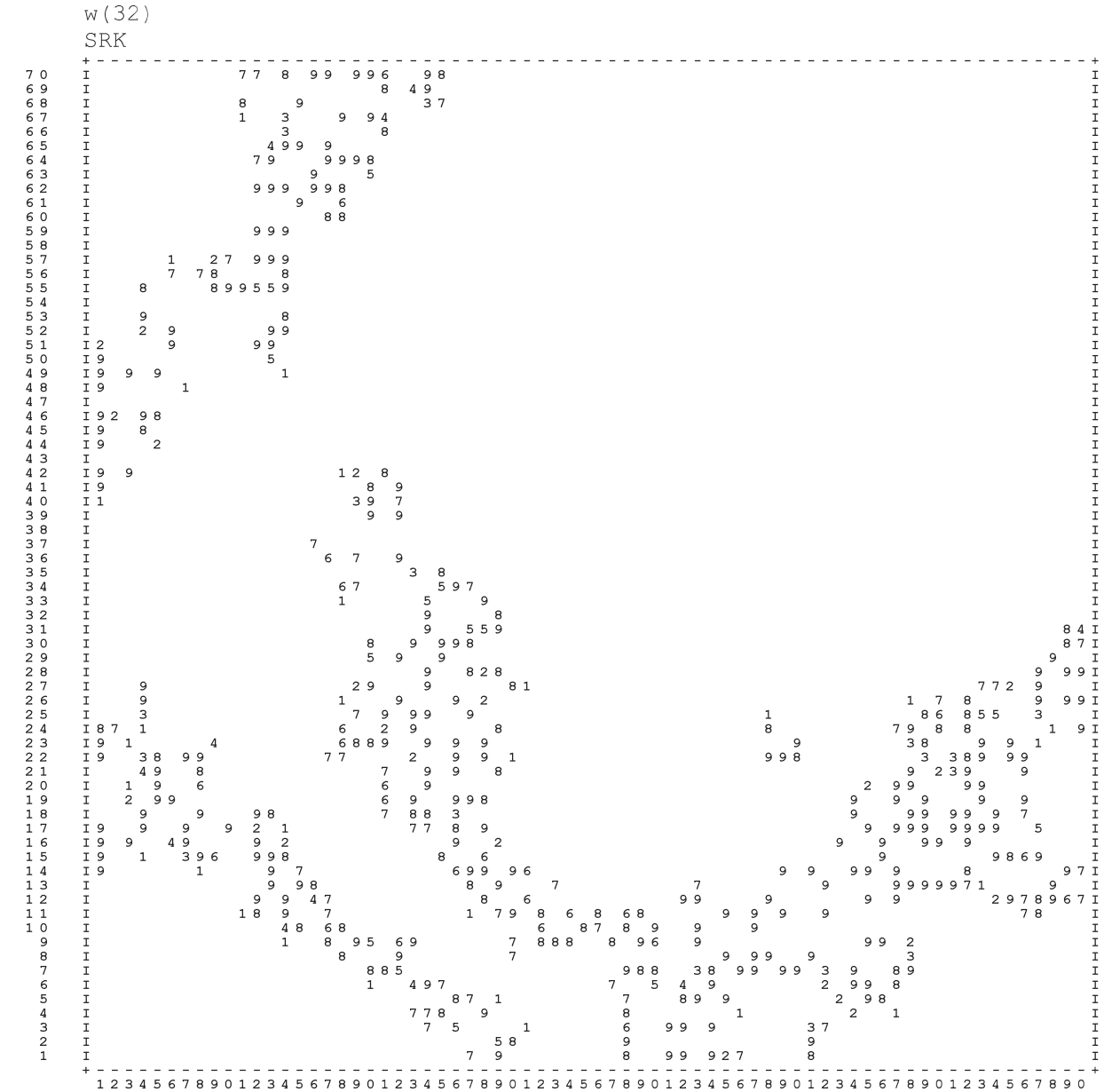


Fig. 11 – *Probability map representing Soave-Redlich-Kwong method*

consisting of normal fluids and systems containing mixtures with wide range of boiling temperature. Working pressure can have all the values of attributes except the last one $0<p_r<10$. Working temperatures can be in regions $T<T_b$, $T_b \leq T \leq T_d$, $T_d<T<T_c$, $T=T_c$, and $T_c<T<4T_c$.

The method is applicable in homogeneous regions of vapour and liquid phases. It can be used for calculations of *K*-values, vapour pressures, partial molar properties, and vapour-liquid equilibrium. It is not very accurate when calculating liquid densities. The accuracy of the method depends from the other values of physical properties and varies from accurate to less accurate. The neural network identifies the method as a fast one compared to the other equations of state of the *van der Waals* type. Data for all components and interaction coefficients are expected for good simulation. The maps also indicate that the *Soave-Redlich-Kwong* equation of state is applicable in vapour and in liquid phase of saturated regions. To illustrate correlation between the method and the physical property temperature, maps from Figures 9 and 11 must be overlapped. The correlation between the method and the polarity of the system is obtained by overlapping maps from Figures 11 and 12.

2. UNIFAC method

The probability map for the *UNIFAC* method as a representative of activity coefficient methods is

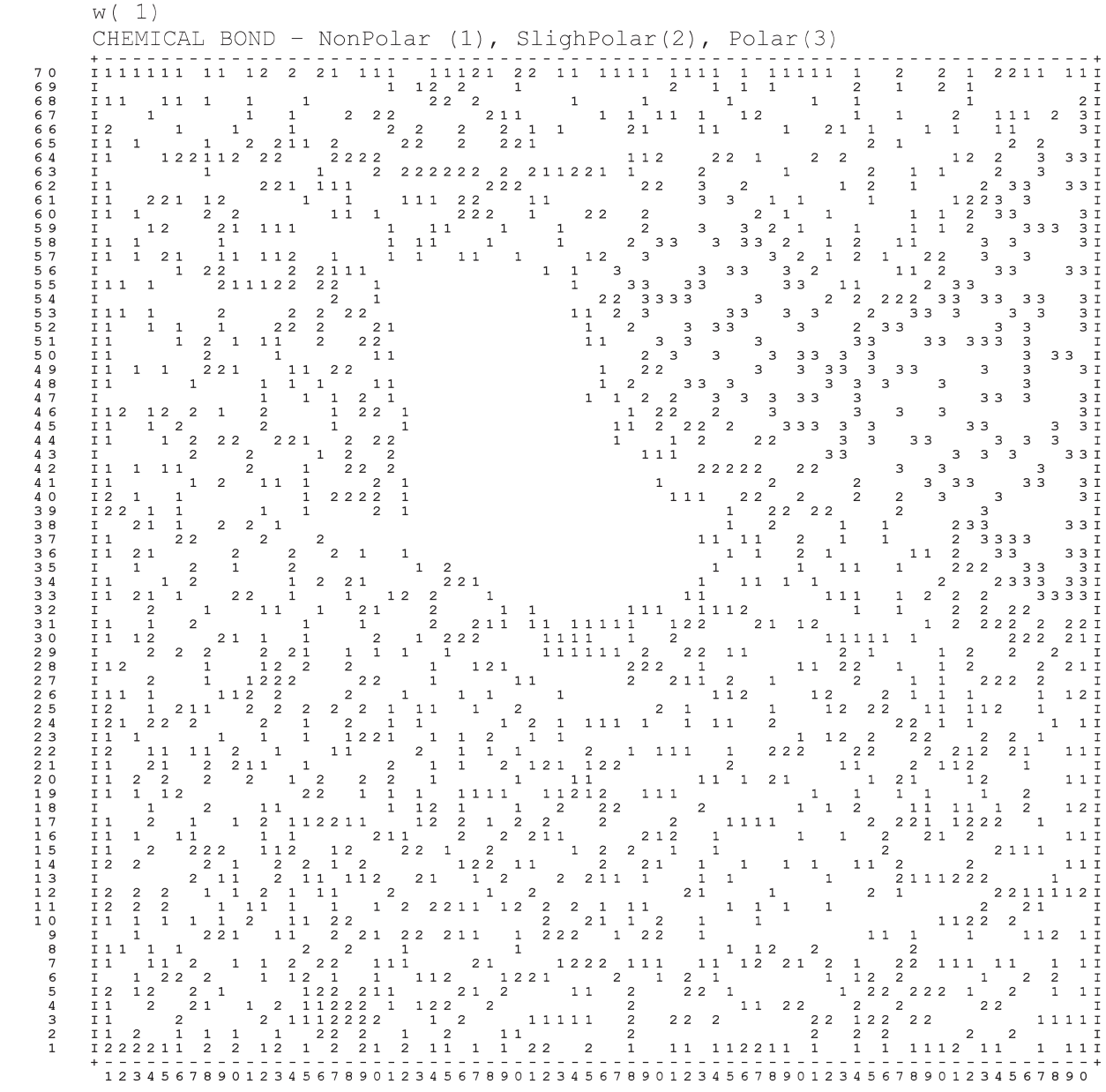


Fig. 12 – *Map representing physical property chemical bond for non-polar, slightly polar and polar mixtures*

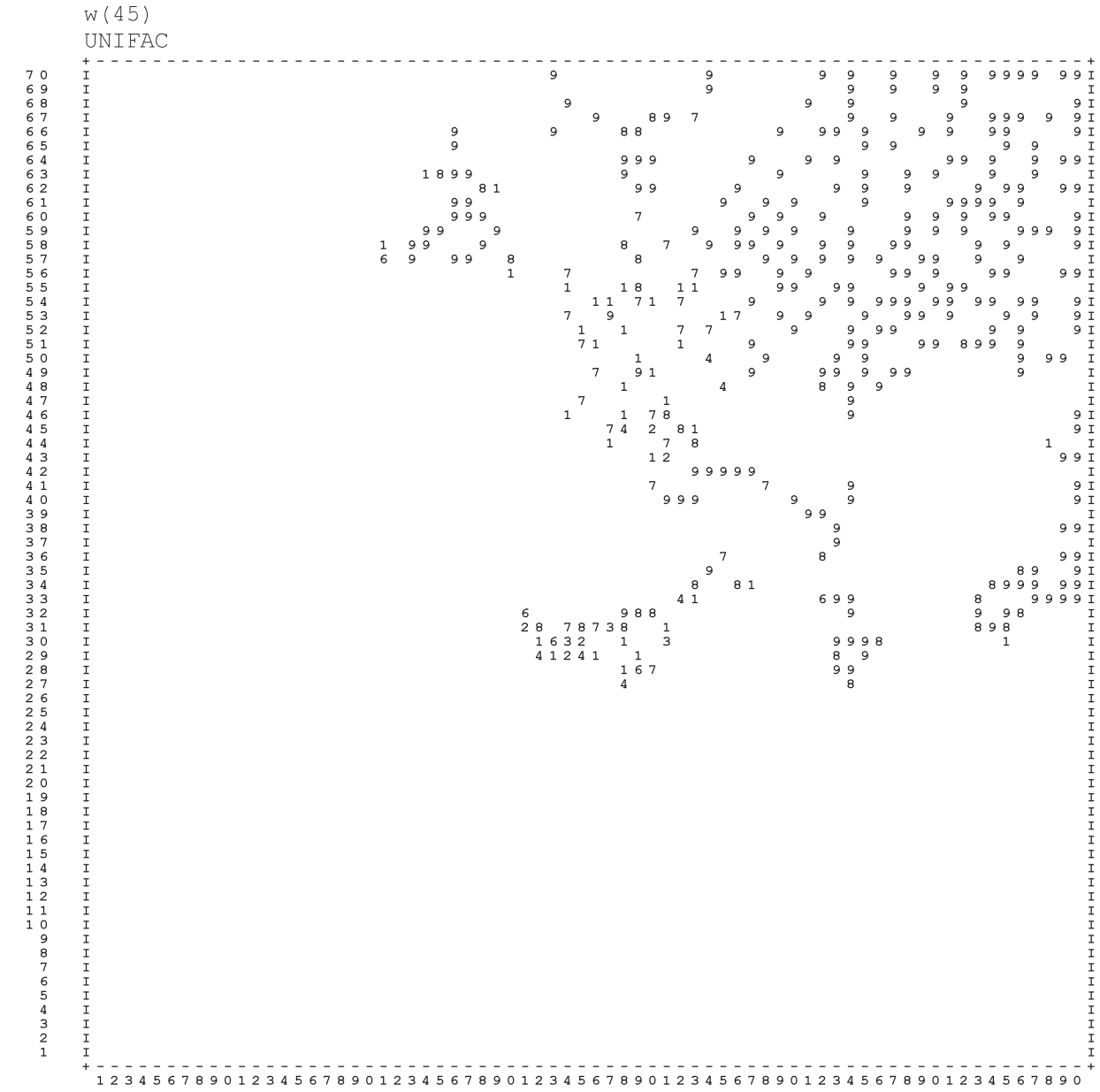


Fig. 13 – *Probability map representing UNIFAC method*

represented in Figure 13. When overlapping the probability map of the *UNIFAC* method with the main Kohonen map it can be seen that the method appears in regions “L”, “l”, and “S”. Further inspection of the *UNIFAC* probability map and the maps of physical properties show correlation between them. The method is applicable to systems consisting of a large range of components and other organic components when components are non-polar, slightly polar, or polar. Working pressure can be low or medium. Working temperatures can take values in regions $T<T_b$ and $T_b \leq T \leq T_d$.

The method is applicable to one-phase regions and also to phase splitting. It is capable of predicting azeotropes. It is useful for non-ideal mixtures and operating conditions in the diluted region with respect to alcohol. It can be used for calculations of vapour-liquid, vapour-liquid-liquid or liquid-liquid equilibrium and excess enthalpies. The method is accurate for calculating equilibria and less accurate when estimating excess enthalpies. The neural network indetifies the method as slower because of the complex *UNIFAC* model. It can be used for predicting phase equilibria in systems for which no experimental data are available. The maps also indicate more precisely that the *UNIFAC* method is applicable in regions of liquid, liquid-liquid phase, and in liquid part of the vapour-liquid phase of saturated regions. For illustration, the correlation between the *UNIFAC* method and the physical property *temperature* can be found when comparing Figures 9 and 13. Figures 12 and 13 demonstrate the correlation

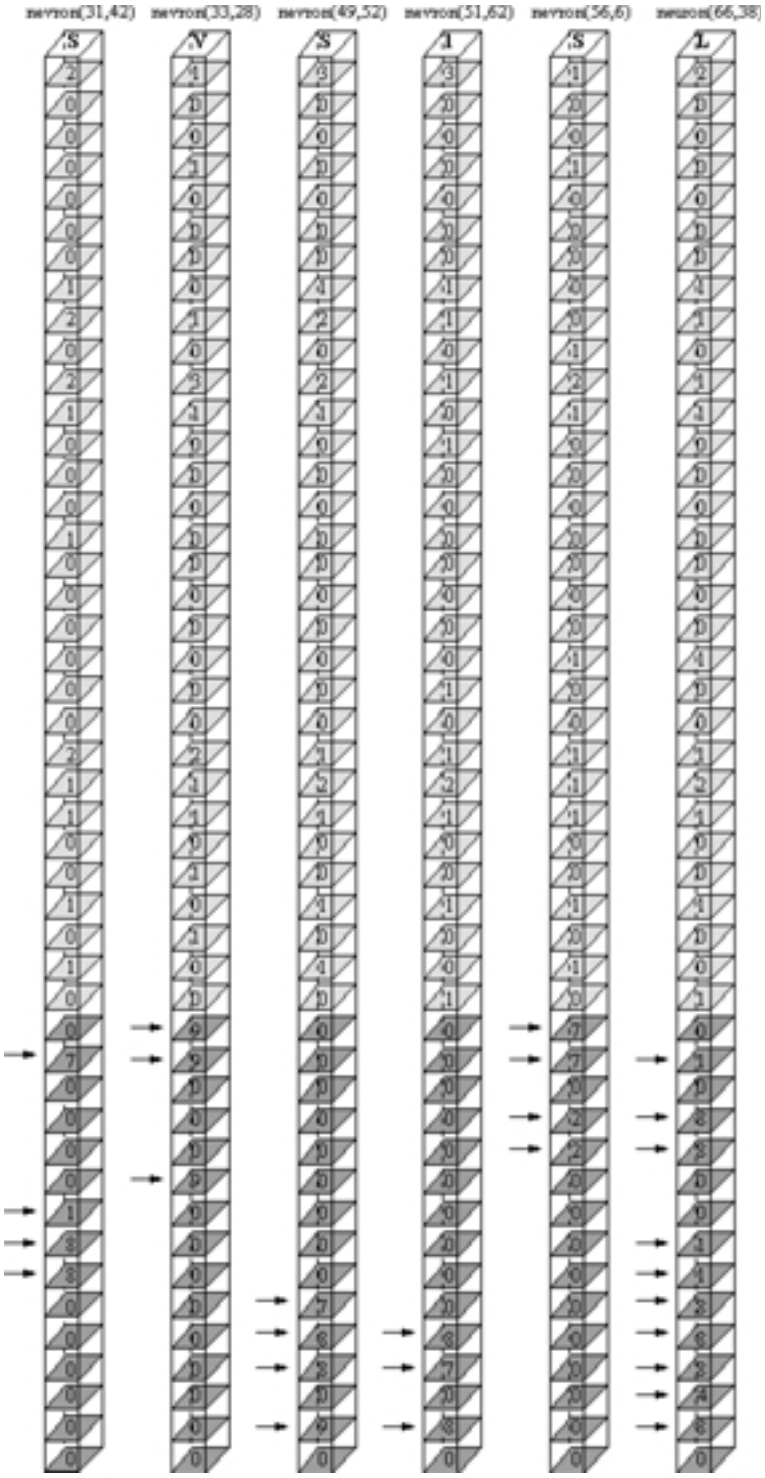
Fig. 14 – *Map representing presence of azeotrops in mixture*

between the polarity and the *UNIFAC* method. Figures 13 and 14 show the correlation between the *UNIFAC* method indicating the presence of azeotropes in a mixture.

Presentation of some randomly chosen column-like neurons

If we visualize Kohonen neural network as a block composed of overlapped maps, the neurons have a form of columns, and the entire network is represented as a block of matrices. With column-like neurons a clear presentation of the weights in individual neurons, and their interconnection in the network when handling the same input variable, are obtained. In Figure 15 some randomly chosen column-like neurons are represented: neuron(31,42), neuron(33,28), neuron(49,52), neuron(51,62), neuron(56,6) and neuron(66,38).

The neuron(31,42) has the label "S" on the main Kohonen map. The label represents a saturated region. The neuron was activated for slightly polar organic components at medium working pressure in temperature range $T_b \leq T \leq T_d$. No phase splitting is expected. In this particular neuron the weight w_{j16} is affected by the signal x_{s16} representing *K*-values. Methods of phase equilibrium can be less accurate and fast with all the necessary coefficients. The liquid phase of a saturated region must be simulated with them. The neural network has found four methods of phase equilibrium, *Peng-Robinson*, *Margules-1, Margules-2* and *van Laar* one with the corresponding weights of 7, 1, 8



F i g . 1 5 – *Some randomly chosen column-like neurons with presentation of weights*

and 8. The weights indicate that *Margules-2* and *van Laar* methods are the most appropriate to solve the problem given. Somewhat less appropriate is the *Peng-Robinson* method, much less appropriate is the *Margules-1* method.

The Kohonen label for the neuron(33,28) is "V", representing a vapour region. The neuron was activated for non-polar mixtures of normal fluids at low working pressures and temperature range $T_d<T<T_c$ in one phase region. Methods of phase equilibrium can be less accurate and fast with all the necessary coefficients. Neural networks are proposing *Soave-Redlich-Kwong*, *Peng-Robinson* and *Virial* methods of phase equilibrium with equal weights of 9.

The neuron(49,52) has the label "S" on the main Kohonen map. It covers polar mixtures of or-

ganic components at medium working pressures and working temperatures between T_b and T_d. No phase splitting is supposed to appear in liquid phase of saturated region. The methods of phase equilibrium are expected to be accurate with all the coefficients necessary. The methods can be slow and must be appropriate for liquid phase of vapour-liquid saturated region. The neural network has found four methods of phase equilibrium as appropriate: *UNIFAC, NRTL*, *UNIQUAC* and *Wilson*. Their trained weights are 9, 8, 8 and 7, respectively.

The neuron(51,62) has the label "l" on the main Kohonen map. The label indicates the liquid-liquid region. The neuron is active for polar organic components at low working pressure and temperatures lower then T_b. Phase splitting is expected in liquid region. The methods must be accurate but slow, appropriate to simulate liquid-liquid equilibrium. The neural network has found *UNIQUAC*, *UNIFAC* and *NRTL* methods with the corresponding trained weights 8, 8 and 7 as the most appropriate ones.

The neuron(56,6) was activated for mixtures of non-polar normal fluids having working pressure in the range $0<p_r<10$ and working temperature in the range $T_b \leq T \leq T_d$. No phase splitting can occur in liquid phase of the saturated region, marked with the label "S" on the main Kohonen map. The methods of phase equilibrium are expected to be accurate and fast. Vapour-liquid equilibrium should be simulated with them. For such a combination of physical properties, the neural network recommends the *Soave-Redlich-Kwong* or *Peng-Robinson* equations of state far more then the *Starling* or *Lee-Kesler-Plocker* equations with the trained weights 7, 7, 2 and 2, respectively.

The neuron(66,38) has the label "L" on the main Kohonen map. It was activated for slightly polar organic mixtures at low working pressures and temperatures below T_b. No phase splitting should appear in liquid region. The methods of phase equilibrium are expected to be accurate, they can be slow. The neural network proposes *Starling*, *Lee-Kesler-Plocker*, *Wilson*, *UNIQUAC*, *NRTL* and *UNIFAC* rather then *ASOG* and far more then *Peng-Robinson*, *Margules-2* and *van Laar* methods of phase equilibrium. The corresponding trained weights are 8 for the first six methods of phase equilibrium, 4 for the *ASOG* method, and 1 for the last three ones.

Conclusions

Preliminary, less precise results for smaller Kohonen neural network were improved in two attempts. The first one was the increase of the dimensions of Kohonen neural networks and the number of epochs for training. According to the fact that training time is increasing proportionally with the size of the neural network and exponentially with epochs, a compromise has to be found. As a result a higher number of active neurons was obtained and training errors decreased. On the other hand, no specific rule for resolving conflicting situations could be obtained. The second attempt was an intervention in the bank of objects to improve their information content. With two new dimensions of vectors $\boldsymbol{X}_s$ representing objects, more precise definition of phase equilibrium methods applicability in saturated region was added to the objects. Conflicting situations were resolved. Using labels "V", "S", "L" and "l" Kohonen neural networks were able to cluster vapour, saturated, liquid and liquid-liquid regions respectively. Besides the main Kohonen map indicating different phase regions, separate maps of physical properties and probability maps of phase equilibrium methods, were obtained. Maps show transparent correlation between physical properties and methods of phase equilibrium. Presenting neurons in a form of columns, the values of trained weights were obtained at individual neurons, showing how weights handling the same input variable were connected together in the network. In separate columns, the scaled values of weights from probability maps represent the certainty of each phase equilibrium method for specific combination of physical properties. In that way the neural networks estimates the applicability of the phase equilibrium methods, their adequacy for efficient chemical process design, and simulation.

Symbols

- a – topology dependent scaling function or neighbourhood function, –
- c – central neuron, –
- $\boldsymbol{d}_{c-j}$ – topological distance between central neuron c and neuron j, –
- j – current neuron, –
- m – number of weights, –
- n – number of neurons, –
- s – particular input, –
- t – number of objects which can be associated with time, –
- $\boldsymbol{W}_j$ – weight vector, –
- w_{ji} – weight of neuron j on level i, –
- w_{ci} – weight of central neuron c on level i, –
- $\boldsymbol{X}$ – user's object, –
- xi – physical property i, –
- $\boldsymbol{X}_s$ – input signal to the Kohonen neural network, –
- $\boldsymbol{Y}$ – target vector, –
- y_i – phase equilibrium method i, –

T – working temperature, K
T_b – boiling point of mixture, K
T_d – dew point of mixture, K
T_c – critical temperature, K
p – working pressure, bar
p_c – critical pressure, bar
p_r – reduced pressure, 1

Greek symbols

$\eta(r)$ – learning rate term, dependent on t, –

Abbreviations

RMS – the root-mean-square of a variate or the square root of the mean squared value
SRK – Soave-Redlich-Kwong equation of state
PR – Peng-Robinson equation of state
BWR – Benedict-Webb-Rubin equation of state
LKP – Lee-Kesler-Plocker equation of state

References

1. *Baňares-Alcantara*, *R*., *Westerberg*, *A. W*., *Rychener M. D*., Comput. Chem. Engng. **9** (1985) 127
2. *Kelly*, *E. B*., *Holste*, *J. C*., *Hall*, *K.R*., Paper presented in the AIChE 1987 Annual Meeting, Session No. 138
3. *Gani*, *R*., *O'Connell*, *J. P*., Comput. Chem. Eng. **13** (1989) 397
4. *Nielsen*, *J. M*., *Gani*, *R*., *O'Connell*, *J. P*., TMS: A Knowledge Based Expert System for Thermodynamic Model Selection and Application, In: Puigjaner, L. & Espuńa, A. (Eds.), Computer-Oriented Process Engineering, p 29, Elsevier Science Publ BV, Amsterdam, 1991
5. *Oreski*, *S*., *Glavic*, *P*., Hung. J. Ind. Chem. **25** (1997) 161
6. *Petersen*, *R*., *Fredenslund*, *A*., *Rasmussen*, *P*., Comput. Chem. Engng. **18** (1994) S63
7. *Habiballah*, *W. A*., *Startzman*, *R. A*., *Barrufet*, *M. A*., SPE Reservoir Engineering **11** (1996) 121
8. Sharma, *R*., *Singhal*, *D*., *Ghosh*, *R*., *Dwivedi*, *A*., Computers Chem. Engng. **23** (1999) 385
9. *Alvarez*, *E*., *Riverol*, *C*., *Correa*, *J. M*., *Navaza*, *J. M*., Ind. Eng. Chem. Res. **38** (1999) 1706
10. *Bogdan*, *S*., *Gosak*, *D*., *Vasic-Racki*, *D*., Comput. Chem. Engng. **19** (1995) S791
11. *Wei*, *W*., *Luming*, *Y*., *Ruiwu*, *P*., *Nianyi*, *C*., *Jinwei*, *Z*., Materials Science & Engineering B: Solid-State Materials for Advanced Technology **B31** (1995) 305
12. *Liuming*, *Y*., *Wei*, *W*., *Jingwei*, *Z*., *Nianyi*, *C*., *Ruiwu*, *P*., Pan Tao Ti Hsueh Pao / Chinese Journal of Semiconductors **16** (1995) 603
13. *Buenz*, *A. P*., *Braun*, *B*., *Janowsky*, *R*., Fluid Phase Equilibria **158–160** (1999) 367
14. *Chonghe*, *L*., *Jin*, *G*., *Pei*, *Q*., *Ruiliang*, *C*., *Nianyi*, *C*., J. Phys. Chem. Solids **57** (1996) 1797
15. *Nianyi*, *C*., *Gang*, *L*., *Chonghe*, *L*., *Pei*, *Q*., *Honglin*, *L*., J. Phys. Chem. Solids **58** (1997) 731
16. *Oreški*, *S*., *Zupan*, *J*., *Glavič*, *P*., Chem. Biochem. Eng. Q. **15** (2001) 3
17. *Zupan*, *J*., Gasteiger, *J*., Neural Networks in Chemistry and Drug Design, Wiley – VCH, Weinheim, 1999
18. *Soave*, *G*., Chem. Eng. Sci. **27** (1972) 1197
19. *Peng*, *D. Y*., *Robinson*, *D. B*., Ind. Eng. Chem. Fundam. **15** (1976) 59
20. *Yamada*, *T*., AIChE J. **19** (1973) 286
21. *Starling*, *K. E*., *Han*, *M. S*., Hydrocarbon Proc. **51** (1972) 129
22. *Brule*, *M. R*., *Lin*, *C. T*., *Starling*, *K. E*., AIChE J. **28** (1982) 616
23. *Watanasari*, *S*., *Brule*, *M. R*., *Starling*, *K. E*., AIChE J. **28** (1982) 626
24. *Lee*, *B. I*., *Kesler*, *M. G*., AIChE J. **21** (1975) 510
25. *Mason*, *E. A*., *Spurling*, *T. H*., The Virial Equation of State, Pergamon, New York, 1968
26. *Margules*, *M*., *Sitzber*, *M*., Akad. Wiss. Wien, Math. Naturw (2A) **104** (1895) 1234
27. *Van Laar*, *J. J*., Z. Phys. Chem. **72** (1910) 723
28. *Chien*, *H. H. Y*., *Null*, *H. R*., AIChE J. **18** (1972) 1177
29. *Wilson*, *G. M*., J. Am. Chem. Soc. **86** (1964) 127, 133
30. *Renon*, *H*., *Prausnitz*, *J. M*., AIChE J. **15** (1969) 785
31. *Renon*, *H*., *Prausnitz*, *J. M*., AIChE J. **14** (1968) 135
32. *Prausnitz*, *J*., *Anderson*, *T*., *Grens*, *E*., *Eckert*, *C*., *Hsieh*, *R*., *O'Connell*, *J*., Computer Calculations for Multicomponent Vapor-Liquid and Liquid-Liquid Equilibria, Prentice-Hall, Inc., New Jersey, 1980
33. *Kojima*, *K*., *Tochigi*, *K*., Prediction of Vapor-Liquid Equilibria by the ASOG Method, Kodansha Ltd. and Elsevier Scientific Publishing Company, Tokyo, 1979
34. *Fredenslund*, *A*., *Gmehling*, *J*., *Rasmussen*, *P*., Vapor-Liquid Equilibria Using UNIFAC, Elsevier Scien. Pub. Comp., Amsterdam, 1977
35. *Tiegs*, *D*., Gmehling, *J*., *Rasmussen*, *P*., *Fredenslund*, *A*., Ind. & Eng. Chem. Res. **26** (1987) 159
36. *Hildebrand*, *J. H*., *Prausnitz*, *J. M*., *Scott*, *R. L*., Regular and Related Solutions, Van Nostrand Reinhold, New York, 1970
37. *Oellrich*, *L*., *Plocker*, *U*., *Prausnitz*, *J. M*., *Knapp*, *H*., Int. Chem. Engng **21** (1981) 1
38. *Prausnitz*, *J. M*., *Lichtenthaler*, *R. N*., and *Azevedo*, *E. G*., Molecular Thermodynamics of Fluid-Phase Equilibria, Prentice-Hall, Englewood Cliffs, 1986
39. *Reid*, *R. C*., *Prausnitz*, *J. M*., *Poling*, *B. E*., The Properties of Gases & Liquids, McGraw-Hill Book Company, New York, 1987
40. *Malanowski*, *S*., *Anderko*, *A*., Modelling Phase Equilibria, Thermodynamic Background and Practical Tools, John Wiley & Sons, New York, 1992