Analytica Chimica Acta, 233 (1990) 31–39
Elsevier Science Publishers B V., Amsterdam – Printed in The Netherlands

Processing of signals from an ion-selective electrode array by a neural network

M. BOS *, A BOS and W E VAN DER LINDEN

University of Twente, Department of Chemical Technology, P O Box 217, 7500 AE Enschede (The Netherlands)

(Received 22nd February 1989)

ABSTRACT

Neural network software is described for processing the signals of arrays of ion-selective electrodes The performance of the software was tested in the simultaneous determination of calcium and copper(II) ions in binary mixtures of copper(II) nitrate and calcium chloride and the simultaneous determination of potassium, calcium, nitrate and chloride in mixtures of potassium and calcium chlorides and ammonium nitrate The measurements for the Ca^{2+}/Cu^{2+} determinations were done with a pH-glass electrode and calcium and copper ion-selective electrodes, results were accurate to ±8%. For the $K^+/Ca^{2+}/NO_3^-/Cl^-$ determinations, the measurements were made with the relevant ion-selective electrodes and a glass electrode; the mean relative error was ±6%, and for the worst cases the errors did not exceed 20%.

The on-line simultaneous determination of ionic concentrations by means of ion-selective electrodes (ISEs) is currently hampered by lack of selectivity, non-Nernstian response, and the influence of the ionic strength of the medium on the relationship between the wanted concentration and the activity of the ions to which the ISEs respond. Various computer methods have been described in the literature for the processing of ISE signals that address one or more of these problems [1–3]. A recent review [4] outlines the application of sophisticated chemometric techniques like partial least squares in the calibration problem for multiple ISEs. This paper describes an investigation of the applicability of neural network theory to solve the aforementioned problems.

Neural network theory [5–9] extends the concept of the linear learning machine (LLM) [10], the first artificial-intelligence pattern-recognition technique introduced to analytical chemistry by Jurs, Kowalski and Isenhour in the early seventies [11–13]. Greater sophistication of the training algorithm for the LLM allowed its successful use in quantitative analysis [14]. The major problem in this approach is the handling of data sets that are not linearly separable. For such sets, only an approximate solution is found, although it can be improved by introducing quadratic and cross-product terms of the input signals [15]. Neural networks are capable of dealing with such data sets because interactions and nonlinear behaviour are automatically taken care of [5] and therefore they should be ideally suited for the processing of the signals of an ISE array.

NEURAL NETWORK ALGORITHMS

Calculation of output results from input signals

Neural network theory has been fully explained by Rumelhart and McClelland [5]. Only a working description of the algorithms used in this work is given here. A more extensive description of the backward error-propagation algorithm is available [9].

0003-2670/90/$03 50

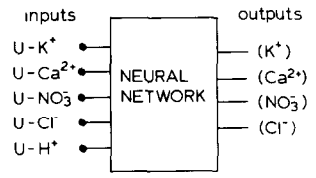

Fig. 1 Black box representation of the neural network for ISE signal processing.

Represented as a black box, the neural network receives its input signals from the ISEs and produces the wanted concentrations at its outputs (Fig. 1). Internally, the network consists of a number of simple processing units, the neurons. These neurons receive inputs coming either from external sensors or from the outputs of other neurons. Each neuron produces one output value that can be used as an input to other neurons, or it can represent a wanted result. The way in which the neurons are interconnected is called the topology of the network (Fig. 2).

In the current literature, the processing that is done in each of the neurons is subdivided into three stages [5,9]. The first is the calculation of the input function, *net*, which is done here by weighted summation of its inputs:

$$net_i = \sum_j w_{ji} o_j \qquad (1)$$

In this equation net_i is the input function of neuron i, w_{ji} is the weight factor associated with input j to neuron i, and o_j is the input j to the

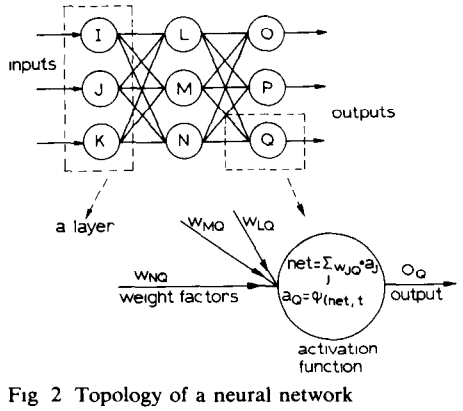

Fig 2 Topology of a neural network

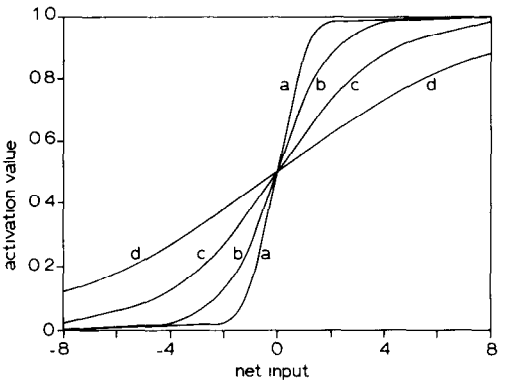

Fig. 3. Activation function for different values of the "temperature," T. (a) 0.5, (b) 1 0, (c) 2 0; (d) 4 0

neuron i. The latter is either the output of a neuron of a preceding layer or an external input signal if one is dealing with the first layer.

The second step is the calculation of the activation function from the result of this input function and the preceding activation state:

$$a_i(t) = F[net_i(t),\ a_i(t-1)] \qquad (2)$$

Here $a_i(t)$ denotes the state of activation of neuron i at time t and $a_i(t-1)$ is its activation state at the preceding time $(t-1)$. The activation function of the neuron, F, is chosen here as

$$a = 1/[1 + \exp(-net/T + \theta)] \qquad (3)$$

in which θ is a bias factor and T a factor that is called the "temperature". In Eqn. 3, the preceding activation is not taken into account, so that the neurons have no time-dependency here. This activation function has a sigmoidal shape; the position of the inflexion point is governed by the value of θ, and the steepness by the factor T. Figure 3 shows a graph of this function for several T values.

The third and last step of the processing done by the neurons is the calculation of the output value from the activation value. In this work, the output of a neuron is set equal to its activation value:

$$o_i(t) = a_i(t) \qquad (4)$$

Equations 1–4 account for the way by which a correctly trained network calculates its output values for a given set of input measurements.

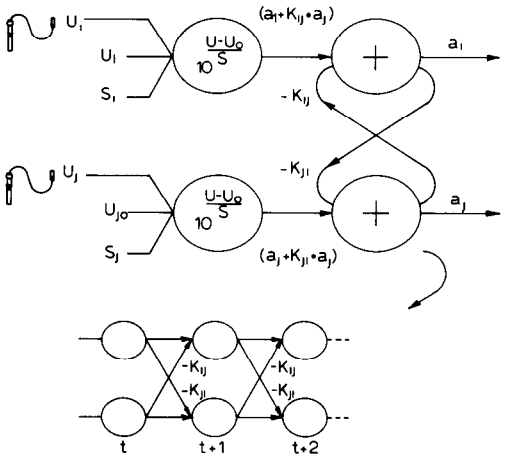

Fig. 4. Simulation of feedback in a recurrent network

Training of the network

Training of the neural network consists of finding the right combination of all weight factors w_{ji} and the values of θ_i and T_i that produces the correct output values for every input pattern (set of measurement values). This training is done with a data set consisting of sets of measurements on samples with known composition, i.e., for which the required output values are known. The algorithm that was used for this purpose is known as the generalized delta rule or backward error-propagation rule. Its use is restricted to feed-forward layered networks. Its goal is to minimize an error function at the outputs. A pattern p is presented to the inputs of the first layer of the network and activation values are calculated for every neuron, layer by layer, until the last layer has been reached. The error E_p for this pattern p is defined as

$$E_p = 1/2 \sum_j (t_{pj} - o_{pj})^2 \tag{5}$$

The value t_{pj} is the required output of neuron j in the last layer for the pattern p; o_{pj} is the output generated by this neuron for this pattern. The total sum of square errors for all patterns is then $E_{tot} = \Sigma_p E_p$. To minimize this error, the weight factors of the inputs of the neurons are adjusted in proportion to the error gradient. This is done for each known pattern and repeated for the whole set of these known patterns until an acceptable total error is obtained. The equation for this adjustment is

$$\Delta w_{ji} = -\epsilon \delta E_{pi}/\delta w_{ji}$$

in which the arbitrarily chosen proportionality constant ϵ is called the learning rate.

For the last layer of neurons that produces the wanted outputs, these corrections can easily be calculated from

$$\partial E_{pi}/\partial w_{ji} = (\partial E_{pi}/\partial o_{pi})(\partial o_{pi}/\partial w_{ji}) \tag{6}$$

Combining this with Eqn. 5 gives

$$\partial E_{pi}/\partial o_{pi} = -(t_{pi} - o_{pi})$$

The last term at the right hand side of Eqn. 6 is calculated from the activation function (Eqn. 3). Equation 4 can be left out because $o_i = a_i$ is used here. Thus

$$\partial o_i/\partial\, net_i = o_i(1 - o_i)1/T \tag{7}$$

From Eqn. 1, it follows that

$$\partial\, net_i/\partial w_{ji} = o_j$$

in which o_j is the output of neuron j of the preceding layer.

For the correction of the temperature factor, the following derivative is used:

$$\partial o_i/\partial T = o_i(1 - o_i)(1/T)\ \ln[(1 - o_i)/o_i]$$

In the generalized delta rule, the adjustment of the weight vectors is governed by a term delta:

$$\Delta w_{ji} = \epsilon\ \text{delta}_{pi} o_{pj} \tag{8}$$

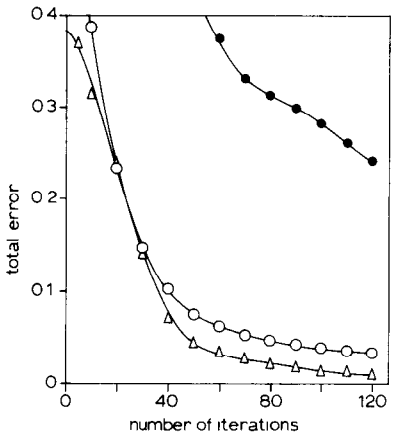

Fig. 5. Learning behaviour for various numbers of layers: (○) 1 layer, (△) 2 layers; (●) 3 layers

TABLE 1

Measurements with Cu^{2+}/Ca^{2+} ISEs

Sample	$[Cu^{2+}]$ (mmol l^{-1})	$[Ca^{2+}]$ (mmol l^{-1})	Temp (°C)	Cu-ISE (mV)	Ca-ISE (mV)	Glass (mV)
1	0 100	0 100	34 3	226 0	−21.5	110 5
2	0 498	0 100	34 3	245 1	−22 5	116 8
3	0.990	0.099	34.3	253 0	−23 5	121 3
4	4.988	0.095	34 3	271 0	−27.2	136 8
5	9.991	0 090	34 3	278 1	−29 1	145 4
6	0 100	0.500	34 6	224.3	−1 9	125 2
7	0.498	0.498	34.6	243 3	−2.4	128 2
8	0 990	0.495	34 6	252 3	−3.1	130.6
9	4 988	0 475	34 6	270 7	−6 5	140.9
10	9.991	0.450	34 6	277.8	−8.7	148 1
11	0 100	0 999	33 2	223 8	6 7	139 0
12	0 498	0 995	33 2	240.2	6 5	140.2
13	0 990	0.990	33 2	252.0	5.5	140 9
14	4.988	0.950	33 2	270 0	2 8	146.6
15	9 991	0 900	33 2	277.1	0 1	151 0
16	0 100	5 245	33 7	222 0	25.9	183 7
17	0.498	5.224	33 7	241.0	25.7	183 7
18	0.990	5.198	33 7	249.8	25 4	183.5
19	4 988	4 988	33 7	269 7	23 4	182.2
20	9 991	4.725	33.7	276.4	21 2	181 0
21	0.100	11.089	34 5	216 1	32 1	198.7
22	0.498	11 045	34 5	235.4	32 1	198.7
23	0 990	10 990	34.5	242 7	31.8	198 9
24	4 988	10.546	34 5	261 9	30.3	197.7
25	9.991	9 991	34 5	270.0	28 9	196.5

The delta term for a neuron in the last layer which provides an output is defined as

$$\text{delta}_{pi} = (t_{pi} - o_{pi}) f_i'(net_{pi}) \tag{9}$$

With the use of Eqn. 7, this can be rewritten as

$$\text{delta}_{pi} = (t_{pi} - o_{pi}) o_{pi}(1 - o_{pi})/T \tag{10}$$

For the so-called hidden neurons in the layers preceding the last layer, the target output t_{pi} is unknown, so that this delta value cannot be calculated. The generalized delta rule now defines this delta term for a hidden neuron as

$$\text{delta}_{pi} = f_i'(net_{pi}) \sum_k \text{delta}_{pik} w_{ik} \tag{11}$$

The error is thus propagated backwards through the layers and adjustments can be made.

From Eqns. 8 and 10, the correction factor can be calculated for the weights of the last layer of

TABLE 2

Recognition results for some Ca^{2+}/Cu^{2+} samples

Sample	$[Cu^{2+}]$ (mmol l^{-1})	Rel.error (%)	$[Ca^{2+}]$ (mmol l^{-1})	Rel.-error (%)
1	0 101	+1	0 097	−3
2	0.504	+2	0.091	−1
6	0.096	−4	0.462	−8
10	9.306	−7	0 426	−5
22	0 529	+6	11 128	+1

TABLE 3

Characteristics of Cu^{2+} and Ca^{2+} ISEs calculated with the recurrent network

ISE	U_0		Slope		K	
	N [a]	LG [b]	N	LG	N	LG
Cu	323 6	330 6	25 7	26 3	−0 031	−0 021
Ca	79.2	84 3	26 7	26 1	−0.009	−0.0015

[a] Network, [b] Linear regression

TABLE 4

Training set for the $K^+/Ca^{2+}/NO_3^-/Cl^-/H^+$ system

No	Conc. (mmol l^{-1})				Potential (mV)					Temp (°C)
	K	Ca	NO_3^-	Cl^-	K	Ca	NO_3^-	Cl^-	H^+	
1	0.100	0.100	0 100	0.300	−147 8	−22.5	228 2	227 0	128.6	33 8
2	0.990	0 099	0.099	1.188	−90.8	−23.3	227 2	195.5	142 2	33 8
3	9.991	0 090	0 090	10 171	−31 4	−27 4	209 8	141.2	188 1	33 8
4	0.498	0.498	0 100	1.493	−109 8	−3 3	226 5	189.1	138.0	33 8
5	4.988	0 475	0.095	5 938	−49.6	−5.6	217.12	154 5	173 1	33 8
6	0.100	0.999	0.100	2.098	−146.9	3 0	219.4	180 5	147 6	31 3
7	0 990	0.990	0.099	2 970	−91.6	4 3	219.3	172 2	154 2	31 6
8	9 991	0.900	0 90	11.791	−31 4	0 8	204 6	137 7	189 1	31.8
9	0 498	5.224	0 100	10.945	−110 8	23 2	205.6	140 4	184 2	32.1
10	4.988	4.988	0.095	14.964	−50.8	22.0	201 2	132.6	193 0	32 2
11	0 100	11 089	0 100	22 278	−150.3	27 6	194 8	123 2	201 6	33.3
12	0.990	10 990	0 099	22 970	−93 8	30 6	196.5	122 5	203 4	33.3
13	9.991	9.991	0.090	29.973	−33 6	28 9	190 8	115.1	211 2	33 3
14	0 498	0 100	0.498	0 697	−109.3	−25.2	188 4	207.5	142 9	33 5
15	4 988	0.095	0.475	5 178	−50 8	−27 4	187.9	157 6	173 9	33 7
16	0.100	0.500	0.500	1.099	−146.1	−5.8	187.6	196 5	144 6	34 6
17	0.990	0 495	0.495	1.980	−92 3	−5 1	188.6	181 2	153.0	34 0
18	9.991	0 450	0 450	10.891	−32 3	−8.5	185 9	138 5	190 3	34 0
19	0.498	0.995	0 498	2.488	−109 0	3.5	188 6	176 6	158.6	33 9
20	4 988	0 950	0 475	6.888	−50 5	2.1	187.6	150 8	179.3	33.9
21	0 100	5.245	0.500	10.589	−147.6	20.7	184.2	143.4	182.5	33.9
22	0.990	5.198	0.495	11 386	−93.1	21 5	185.2	140.9	184 9	33 9
23	9 991	4.725	0.450	19.442	−32 6	18 8	182.5	125.2	200.9	33 9
24	0.498	11.045	0.498	22.587	−111 0	29 1	180.0	122.5	203 1	33.9
25	4.988	10 546	0.475	26.081	−51 5	29 3	179.5	118 3	207 3	33 9
26	0.100	0 100	0 999	0.300	−142.5	−24 2	169.9	227.9	165 8	31 7
27	0.990	0 099	0.990	1.188	−91.1	−24 0	170.9	195 5	169.0	31 7
28	9.991	0 090	0.900	10.171	−29 9	−27.7	171.4	140.0	192 8	31 7
29	0 498	0.498	0.995	1.493	−108 3	−4 1	171.7	185.8	158 9	32 4
30	4 988	0.475	0.950	5 938	−48 1	−5.8	171 9	152 5	179 5	32 5
31	0.100	5.245	0.999	10 589	−145.2	22 0	170.9	140 9	187 1	32 7
32	0.990	5.198	0.990	11 386	−92.8	22.5	171 4	138 7	189.1	32 8
33	9.991	4 725	0 900	19 442	−32.6	20 3	171 2	124.7	202.6	32.8
34	0.498	0 995	0 995	2.488	−107 6	3 8	171 4	175.1	163.3	32.9
35	4 988	0 950	0.950	6.888	−48 1	2 6	172.2	156 7	181 1	32 9
36	0.100	11.089	0.999	22.278	−144 9	30 8	169.2	122.0	205 3	34 0
37	0.990	10.990	0.990	22 970	−92 3	30.8	169.4	121.3	206 3	34 0
38	9.991	9.991	0.900	29 973	−32.6	29.1	168.7	113.9	213 2	34 0
39	0.498	0 100	5 224	0.697	−105.6	−26.9	129 9	207 5	190 8	34 1
40	4 988	0 095	4 988	5.178	−50.0	−27 9	132.2	157 9	197 7	34.1
41	0 100	0.500	5.245	1.099	−134.1	−8.0	129 1	196 7	190 6	34 0
42	0 990	0.495	5.198	1.980	−89 4	−6 3	131.3	182 7	192 8	34 0
43	9 991	0.450	4.725	10 891	−32.1	−9 5	133 6	138 5	205.1	34.0
44	0.498	0.995	5.224	2 488	−105 3	2 1	131.1	177.3	193.8	34.0
45	4.988	0.950	4 988	6 888	−48 8	1 1	132 8	151.0	200.7	34.0
46	0.100	5.245	5 245	10.589	−134 3	20 0	130.4	140.7	203.6	34.0
47	0 990	5 198	5 198	11.386	−90.8	21 5	131.8	139.0	204.6	34.0
48	9 991	4 725	4.725	19.442	−32.3	19.8	134.3	124.7	212 5	34.0
49	0 498	11.045	5.224	22.587	−103.1	33 5	133 6	124 0	216 4	32 4
50	4 988	10.546	4.988	26.081	−47.6	31.8	134 5	119 8	218.4	32 4
51	0.100	0.100	11.089	0.300	−121.1	−26 9	112 4	229 9	209 0	32.6
52	0.990	0.099	10 990	1 188	−85 7	−26 2	113.4	198.2	210 0	32 6

(Continued overleaf)

TABLE 4 (continued)

No.	Conc (mmol l^{-1})				Potential (mV)					Temp. (°C)
	K	Ca	NO_3^-	Cl^-	K	Ca	NO_3^-	Cl^-	H^+	
53	9.991	0 090	9.991	10 171	−30.1	−28.7	116 4	142 7	215 9	32.7
54	0.498	0.498	11 045	1 493	−100 9	−6.8	112 7	191 3	209 8	32 8
55	4 988	0.475	10 546	5.938	−148 8	−7 3	114.4	156.7	213.2	32 8
56	0 100	0 999	11.089	2 098	−121.8	1 1	112 7	182 7	210 5	32.8
57	0.990	0 990	10 990	2 970	−86 7	2 3	113 9	174 6	211 5	32.8
58	9.991	0.900	9.991	11.791	−30 4	0 1	116.6	138.2	217.4	32 8
59	0.498	5 224	11.045	10 945	−102 1	21.5	113 9	140 9	217 1	33 3
60	4.988	4 988	10.546	14.964	−49.6	21 0	115 6	132 6	219 3	33.3
61	0.100	11.089	11 089	22.278	−123 5	29 8	114 6	124 0	223.3	33 5
62	0 990	10.990	10 990	22 970	−88 6	30 1	115.6	123.0	224.0	33 5

neurons. The new temperature factor is calculated from

$$\Delta w_{ji} = (\epsilon/T)(t_{pi} - o_{pi})\, o_{pi}(1 - o_{pi})\, o_{pj} \qquad (12)$$

$$\Delta T_i = (\epsilon_T/T)(t_{pi} - o_{pi})\, o_{pi}(1 - o_{pi}) \times \ln\left[(1 - o_{pi})/o_{pi}\right] \qquad (13)$$

For the layers which contain hidden neurons, Eqn. 11 is used together with Eqn. 8 to calculate the correction:

$$\Delta w_{ji} = (\epsilon/T)\, o_{pi}(1 - o_{pi})\, o_{pj} \sum_k \text{delta}_{pik} w_{ik} \qquad (14)$$

in which the summation is over the k neurons to which the neuron is connected. For the correction term of the temperature factor, the following equation is used:

$$\Delta T_i = (\epsilon/T)\, o_{pi}(1 - o_{pi})\, \ln\left[(1 - o_{pi})/o_{pi}\right] \times \sum_k \text{delta}_{pik} w_{ik} \qquad (15)$$

For multi-layered networks, local minima can occur in E_{tot}. To prevent the algorithm of being trapped in these local minima, the weight factors are corrected not only with the correction term from Eqn. 8 but also with a part of the preceding correction:

$$w_{ji}(t) = w_{ji}(t-1) + \Delta w_{ji}(t) + m\, \Delta w_{ji}(t-1) \qquad (16)$$

where m is called the momentum factor.

In the calculations, the value of the bias factor θ is taken into account as an extra input with a value of one to the neurons; θ thus becomes just another extra weight factor to be trained with the use of Eqns. 12 and 14.

Simulation of feedback

The backward error-propagation rule cannot be applied to networks in which there is feedback between the outputs of neurons to the input side of the network.

Feedback in a one-layer network can be simulated with a multi-layered network with one set of weight factors for all layers in which the sequence of the layers represents the time axis in discrete steps. Such a recurrent network simulation can be used to represent a model of the behaviour of ion-selective electrodes as given in the Nikolsky equation:

$$U_i = U_{i0} + S \log\left[a_i + \sum_j K_{ij} a_j^{n_i/n_j}\right]$$

where U_i is the potential of the ISE, U_{i0} its standard potential, S the usual slope term, a_i the activity of the ion that is measured, K_{ij} the selectivity coefficient for the required ion versus the interfering ion j, a_j the activity of this ion j, and n_i and n_j are the charges of ion i and j.

A schematic diagram of such a model is given in Fig. 4.

EXPERIMENTAL

Chemicals and equipment

Copper nitrate, calcium chloride, potassium chloride, ammonium chloride and ammonium

TABLE 5

Test set for the $K^+/Ca^{2+}/NO_3^-/Cl^-/H^+$ system

No.	Conc. (mmol l^{-1})				Potential (mV)					Temp (°C)
	K	Ca	NO_3^-	Cl^-	K	Ca	NO_3^-	Cl^-	H^+	
1	0 498	0 100	0.100	0.697	−107.8	−22.3	227 9	208 8	135.3	33.8
2	4 988	0 095	0.095	5.178	−48.8	−25.0	217.9	158.9	171.7	33 8
3	0.100	0 500	0.100	1.099	−148.1	−2 4	227 9	198 2	132.6	33 8
4	0 990	0 495	0.099	1 980	−92 1	−3.8	225 2	182 2	144.6	33 8
5	9.991	0 450	0 090	10.891	−31 6	−7.8	209.0	138 5	188 9	33 8
6	0.498	0 995	0 100	2 488	−108.8	4.3	219.8	176 6	150.8	31 5
7	4.988	0.950	0 095	6 888	−49 3	2.6	212.0	151.5	175 3	31.7
8	0.100	5.245	0 100	10 589	−151 3	21 5	203.6	141.4	183 0	32 0
9	0.990	5.198	0.099	11 386	−92 1	23.2	205.6	139.5	185 2	32.2
10	9.991	4.725	0.090	19.442	−32 3	21 0	196 5	125.7	200 7	32.3
11	0.498	11.045	0.100	22.587	−111 0	30 8	196 5	123 0	202 6	33 3
12	4.988	10 546	0.095	26.081	−51 5	30 1	190 8	115 1	211 2	33.3
13	0 100	0 100	0 500	0.300	−148.6	−27 9	185 9	227 2	138.0	33 5
14	0.990	0.099	0.495	1.188	−91.6	−25 0	189 1	194 3	148.3	33 6
15	9.991	0.090	0.450	10.171	−32.1	−29.1	185.9	140 2	188.9	33 7
16	0.498	0.498	0.498	1.493	−109.5	−5.3	188.1	188.4	148 3	34 0
17	4.988	0.475	0.475	5.938	−50.0	−6.5	187.4	154 2	175.8	34 0
18	0.100	0.999	0 500	2 098	−146 4	3.0	188.1	180.5	155.7	33.9
19	0.990	0.990	0 495	2 970	−91 8	3 8	189.1	172.2	162 1	33.9
20	9 991	0.900	0 450	11 791	−32 1	0 3	185.9	136.5	192 1	33.9
21	0.498	5.224	0.498	10 945	−110 3	21.0	185.4	142.4	183 7	33.9
22	4 988	4.988	0 475	14 964	−51.3	19.8	183.9	132.6	193.0	33.9
23	0.100	11.089	0.500	22.278	−146.9	27.9	179.0	122.7	202.1	33.9
24	0.990	10.990	0.495	22.970	−94.0	30.6	180.5	122.0	203.1	33.9
25	9.991	9 991	0.450	29.973	−33.3	28.4	178 5	114 6	211.5	33 9
26	0.498	0.100	0 995	0.697	−108.5	−24.0	170 8	208 5	167.0	31 7
27	4.988	0.095	0.950	5.178	−47.3	−25.7	171.7	157 6	182.0	31 7
28	0 100	0.500	0.999	1.099	−142 9	−3.8	171.4	195 7	156 4	32 4
29	0 990	0.495	0.990	1 980	−90 8	−4.3	171.9	180 3	161 8	32 4
30	9 991	0.450	0.900	10.891	−30.4	−7.5	171.9	137.5	192 3	32.5
31	0.498	5 224	0.995	10.945	−109 8	22 7	171.7	140.0	188.1	32.7
32	4.988	4 988	0.950	14.964	−50.3	21 7	171.4	131.6	195 7	32.8
33	0.100	0 999	0.999	2.098	−143.9	3 0	171.2	180.0	160.8	32.9
34	0.990	0.990	0.990	2.970	−89.9	3 5	172.2	170.4	166.3	32.9
35	9.991	0.900	0.900	11 791	−30.4	1.3	172.2	136.0	193.8	32.9
36	0.498	11 045	0.995	22.587	−109.8	31.1	169.4	121.8	205.8	34.0
37	4.988	10.546	0.950	26.081	−50.8	30.3	169 2	117.6	209.5	34.0
38	0.100	0.100	5.245	0.300	−134.1	−28.4	129 4	227 7	189.8	34 1
39	0.990	0.099	5 198	1 188	−89.9	−26 7	130 6	194.3	191 6	34.1
40	9 991	0.090	4.725	10 171	−31.9	−29.1	133 6	139.7	204.3	34 1
41	0.498	0 498	5 224	1 493	−105.1	−6.5	130.6	189.4	191 8	34.0
42	4.988	0.475	4.988	5.938	−49.3	−7.5	132 6	154.5	199.2	34 0
43	0 100	0.999	5 245	2.098	−133.4	0.3	129 6	181.0	192.3	34 0
44	0 990	0.990	5.198	2.970	−189 9	2.1	131.3	172 9	194.8	34 0
45	9 991	0.900	4.725	11.791	−31 6	−0.4	134.1	136 8	206.3	34 0
46	0.498	5.224	5.224	10.945	−106 1	21.2	131.1	139 7	204.1	34.0
47	4.988	4 988	4.988	14.964	−50 0	21 0	132.8	131 6	208.5	34 0
48	0.100	11 089	5.245	22.278	−131.4	32 3	133.1	125 0	215 9	32.4
49	0.990	10.990	5.198	22.970	−87.4	32 5	133.8	123.5	216 4	32.4
50	9.991	9.991	4 723	29 973	−30.6	30 8	135.3	115.9	220 6	32 4
51	0.498	0.100	11.045	0.697	−99.7	−26 0	112.9	211.2	209 8	32.6
52	4.988	0.095	10.546	5.178	−47.6	−27 4	114 6	160 4	212.7	32 7

(Continued overleaf)

TABLE 5 (continued)

No	Conc (mmol l^{-1})				Potential (mV)					Temp (°C)
	K	Ca	NO_3^-	Cl^-	K	Ca	NO_3^-	Cl^-	H^+	
53	0 100	0 500	11.089	1.099	−122.5	−7.3	111.9	198 2	209 3	32 8
54	0 990	0 495	10 990	1.980	−186.4	−6.3	113.4	184.2	210 5	32 8
55	9.991	0 450	9 991	10 891	−30 6	−8.5	115.9	140 2	216 1	32.8
56	0.498	0.995	11.045	2 488	−100 4	2 3	113.2	178 8	211 0	32 8
57	4 988	0 950	10.546	6 888	−47 8	1 3	114.9	152 7	214.2	32 8
58	0 100	5 245	11.089	10.589	−123 3	21 2	112.9	141 9	216 6	33.3
59	0 990	5 198	10 990	11 386	−88 4	21 7	114 4	140.0	217 4	33 3
60	9 991	4 725	9 991	19 442	−31 9	20.0	117.1	125 9	221 8	33 3
61	0.498	11.045	11 045	22 587	−102 1	30 1	115 1	123 2	223 5	33 5
62	4 988	10.546	10 546	26.081	−50 0	29 6	116 6	119 8	225 2	33 5

nitrate were used as received (Merck, p.a.). All solutions were prepared with deionized water that had been filtered through Millipore Q2 filters.

The following ISEs were used: for copper(II), Metrohm 6.0502.140; for calcium(II), Metrohm 6.0504.100; for chloride, Orion 94-17A; for potassium(I), Ingold 15.730.90; and for nitrate, Metrohm 6.0504.120; a combined pH-glass/reference electrode (Metrohm) was also used. Its reference electrode served for all the ISEs.

The potentials were measured with a home-made, computer-controlled, multiplexing instrument with an input impedance greater than 1 GΩ. This high impedance was obtained with the use of an isolating amplifier (type CA3130) for each channel. A 16-channel analog multiplexer was used to feed a selected input to an amplifier. The amplified signal was then converted to digital form by a 12-bit ADC (type AD574). This instrument was calibrated with a Knick calibration voltage source. The results indicated an accuracy of the electrode voltage readings of ±0.2 mV.

Noise in the potential readings was reduced by integration and averaging of the signals over a period of 1 s by computer control of the instrument.

Procedures

Samples were prepared by adding calculated amounts of stock solutions of the compounds from a burette to the measuring vessel which contained 40 ml of distilled water. The stock solutions were standardized titrimetrically (with a relative error of < 0.3%). The electrodes were rinsed with distilled water before use, wiped dry and then introduced into the sample. Electrode readings were taken after the temperature of the solution became stable (±0.2° C). During the measurements, the solution was stirred magnetically.

The software provided the following functions: dimensioning of the network and specification of the number of inputs and outputs, setting of the learning rate, the moment and the temperature adjustment and choice between adjustments for every pattern separately or for the whole training set after summation. Several options allowed the operator to store temporary results, display weights, temperatures, delta terms, activation values and concentration values calculated by the network for specific input patterns. The program is written in Pascal and runs on any MS/DOS machine.

RESULTS AND DISCUSSION

The Cu^{2+}/Ca^{2+} system

Table 1 shows the measured data for the Cu^{2+}/Ca^{2+} system. It includes the readings for the glass electrode. Although this signal is not directly related to the calcium and copper ion concentrations, it is nevertheless taken into account because this signal will certainly be influenced indirectly by the composition of the sample. The data are preprocessed before entering the training procedure of the network by centering the

inputs around a mean value of 0.5 and a span of 0.25, whereas the logarithms of the concentrations are used as the output values.

Networks with dimensions 1×5, 2×5 and 3×5 were trained with this data set. As can be seen from Fig. 5, the two-layer network performs better than the network with only one layer of neurons, whereas the training of the three-layer network needs much more iterations. The performance of the two-layer network is given in Table 2 which shows that the errors are in the range that is normal for ISE measurements.

The data set from Table 1 was used with the recurrent network software to determine the characteristics of the Ca^{2+} and Cu^{2+} ISEs. Table 3 shows a comparison of these characteristics evaluated by the recurrent network software and by linear regression from the Nikolsky equation. As the results of the network were found with the use of all the binary mixtures, they can be considered as more realistic.

The $K^+/Ca^{2+}/NO_3^-/Cl^-/H^+$ system

For this system, the complete set of data was divided into a training set (Table 4) and a set to evaluate the predictive performance of the trained network (Table 5).

Test runs with various sizes of network again showed that a two-layer network performs satisfactorily (Table 6). The minimum number of neurons needed per layer in this situation was found to be seven. Table 7 shows the worst-case results for the predicted concentrations.

The training calculations take a lot of computer time; generally about 10 000 iterations were needed to obtain the results of Table 6. This takes from 24 to 48 h on an 8-Mc machine equipped with a 8087 mathematical coprocessor.

TABLE 6

Mean prediction errors for the $K^+/Ca^{2+}/NO_3^-/Cl^-/H^+$ system with various sizes of network

Layers	Error (%) for different widths				
	4	5	6	7	10
1	13.8	–	–	–	–
2	16 6	8.2	7 4	5.6	5 6
3	8 3	7.7	6.5	5 8	4.4

TABLE 7

Worst-case prediction errors for the $K^+/Ca^{2+}/NO_3^-/Cl^-/H^+$ system with the 7×2 network

Sample	Error (%)			
	K	Ca	NO_3^-	Cl^-
5	0	−8	−19	+7
11	+1	+1	−16	+3
12	+13	−1	−11	+2
22	+1	+11	+8	+2
23	+3	+12	−5	−1
27	+18	+5	+6	0
28	+14	+11	−12	+5
34	+13	+2	−7	+5
43	+14	+6	+7	0
47	+8	0	+15	+7
49	+26	+4	+3	−2
52	+16	+3	+7	+3

REFERENCES

1 A.F Isbell, R.L Pecsok, R.H Davies and J H Purnell, Anal. Chem., 45 (1973) 2363
2 J W Frazer, D.J. Balaban, H.R Brand, G A Robinson and S.M Lanning, Anal. Chem , 55 (1983) 855.
3 R. Jain and J S Schultz, Anal Chem , 56 (1984) 141
4 M Otto and J D R. Thomas, Ion-Selective Electrode Rev., 8 (1986) 55
5 D E Rumelhart and J L. McClelland, Parallel Distributed Processing, Vols 1 and 2, MIT Press, Bradford, 1986.
6 R.P Lippmann, IEEE ASSM Magazine, April (1987) 4
7 P.D Wasserman and T Schwarz, IEEE Expert, Winter (1987) 10; Spring (1988) 10.
8 T Kohonen, Self-Organization and Associative Memory, Springer, Berlin, 2nd edn , 1988
9 W P. Jones, Back-Propagation, Byte, October (1987) 155.
10 N J Nilsson, Learning Machines, McGraw-Hill, New York, 1965
11 P C Jurs and T.L Isenhour, Chemical Application of Pattern Recognition, Wiley, New York, 1975
12 C L Wilkins and T.L Isenhour, Anal. Chem , 47 (1975) 1849
13 T.L Isenhour, B R. Kowalski and P C Jurs, CRC Crit Rev Anal. Chem , (1974) 1
14 M. Bos and G Jasink, Anal. Chim Acta, 103 (1978) 151
15 M Bos, Anal. Chim Acta, 166 (1984) 261