Classification and Clustering: Using Neural Networks

Anil K. Sharma

L. C. Smith College of Engineering and Computer Science, Syracuse Univesity, Syracuse, New York 13244-4100

Sohail Sheikh*

Department of Electrical Engineering, Widener University, Chester, Pennsylvania 19013

István Pelczer and George C. Levy

NMR and Data Processing Laboratory, Chemistry Department, CST Building, Syracuse University, Syracuse, New York 13244-4100

Received March 28, 1994®

The connectionist approach uses two learning paradigms: supervised and unsupervised. Neutral networks which are based on these two different types of learning paradigms have been tested for their classification and clustering abilities in normal conditions as well as in hostile conditions. We combined both paradigms together with feature enhancement, and the *test case* has been chosen from the field of nuclear magnetic resonance (NMR) spectroscopy. Our research indicates that neural networks based upon both paradigms together with feature enhancements exhibit better capabilities than the neutral networks based on a single paradigm alone without feature enhancement. We also observed that *neural network tuning* is one of the most difficult aspects of the connectionist§pos approach. For achieving optimum results, some of the suggestions are incorporated. The results obtained from classification and clustering are also shown in this paper.

1. INTRODUCTION

Recently, there has been a great deal of interest in neural networks in many areas of science and technology such as engineering, biology, and in the field of cognitive sciences.[1] The applications of the neural networks also appeared in several areas of chemistry,[2] including the prediction of protein secondary structure,[3-6] structure–activity relationships,[7,8] studies of chemical reactions,[9] chemical engineering process control,[10,11] and others.[12-15] Neural networks have also been reported for use in a few analytical chemical studies including ^{1}H NMR spectroscopy,[16,17] mass spectroscopy,[18] infrared spectral interpretation,[19-21] spectroscopic parametrization,[22,23] classification of fuels from chromatographic data,[24] and ^{13}C NMR spectroscopy.[25-28]

The *connectionist approach* [which applying neural networks] is based on the supervised or *unsupervised learning paradigms*.[29] But there are some situations, like in a *battlefield zone*, where these paradigms cannot be used in isolation because the scenario changes continuously. In this paper we describe supervised and unsupervised learnings together with feature enhancements at every step, on different neural networks, and discuss classification and clustering tasks. In order to test these neural network models, we chose a problem from the field of *nuclear magnetic resonance (NMR) spectroscopy*. In recent years, multidimensional nuclear magnetic resonance spectroscopy has become established as a powerful method for determining three-dimensional structures of proteins and other bimolecules.[30-33] We used *test data* from ^{1}H two-dimensional spectroscopy for classification and clustering purposes using neural networks. In our case, we used chemical shifts $\langle\delta\rangle$ of protons from amino acids ^{1}H data.[34] In the previous applications, generally, supervised and unsupervised learning paradigm-based neural networks[35-38] have been used with a difference. Especially, the idea of applying *Hopfield neural networks* for feature enhancements is missing.

An artificial neural network functions by receiving a set of input facts, processing those facts, and then producing a set of output values calculated from these facts. The basic principles of neural networks can be found in the literature.[29,39-41] The training of a network is important, which is the major distinction criterion for different neural networks. The neural networks used here are based on two different paradigms; supervised [In supervised learning, the network is given both input data and desired output data. After each trial, the network compares its own output with the right output supplied to it, corrects any differences, and tries again. The neural network goes into an iterative learning phase until the desired output can be produced.] and unsupervised [In unsupervised learning, there is no target vector, and the neural network is assumed to learn the input without the help from the outside, self-organizing itself until it produces a consistent output.] learning[29] paradigms. In a *learning phase*, initially random connection strengths (weights) and thresholds (biases) are modified through repeated cycles by use of a training data set. In our case, ^{1}H NMR chemical shifts $\langle\delta\rangle$ of amino acids are used. In each cycle, adjustments are made to the weights and biases. At the end of the training phase, the *knowledge* in the network consists of connection strengths and thresholds that have been derived from the training data.[4] We used back-propagation of error[29] to train multilayer feed-forward neural networks which are based upon the supervised learning and Kohonen's learning vector quantization neural network which is based on the unsupervised learning.[41] The Hopfield neural network[39] was used for feature enhancements. The neural networks based on *supervised learning* are more suitable for classification tasks, whereas for clustering problems *unsupervised learning* is used.

Neural Networks Based on the Supervised Learning Paradigm. Some of the neural networks of this type are perceptron, back-propagation trained multilayer perceptron, continuous

* To whom correspondence should be addressed.

® Abstract published in *Advance ACS Abstracts*, July 15, 1994.

0095-2338/94/1634-1130$04.50/0

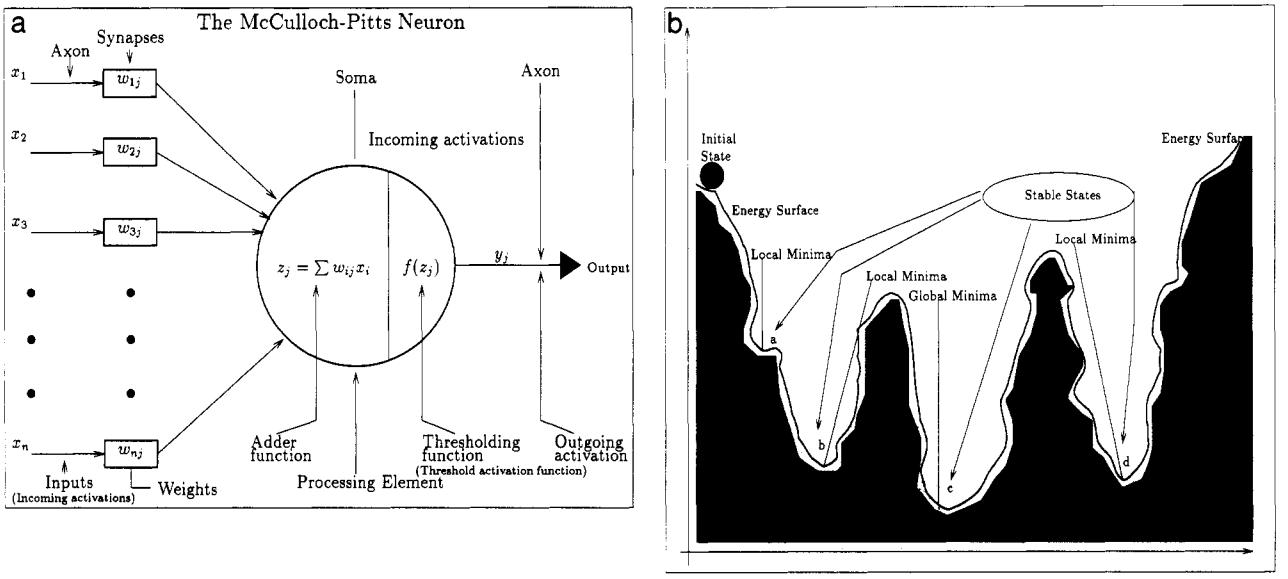


Figure 1. (a, left) An artificial neuron and (b, right) different types of energy minima.

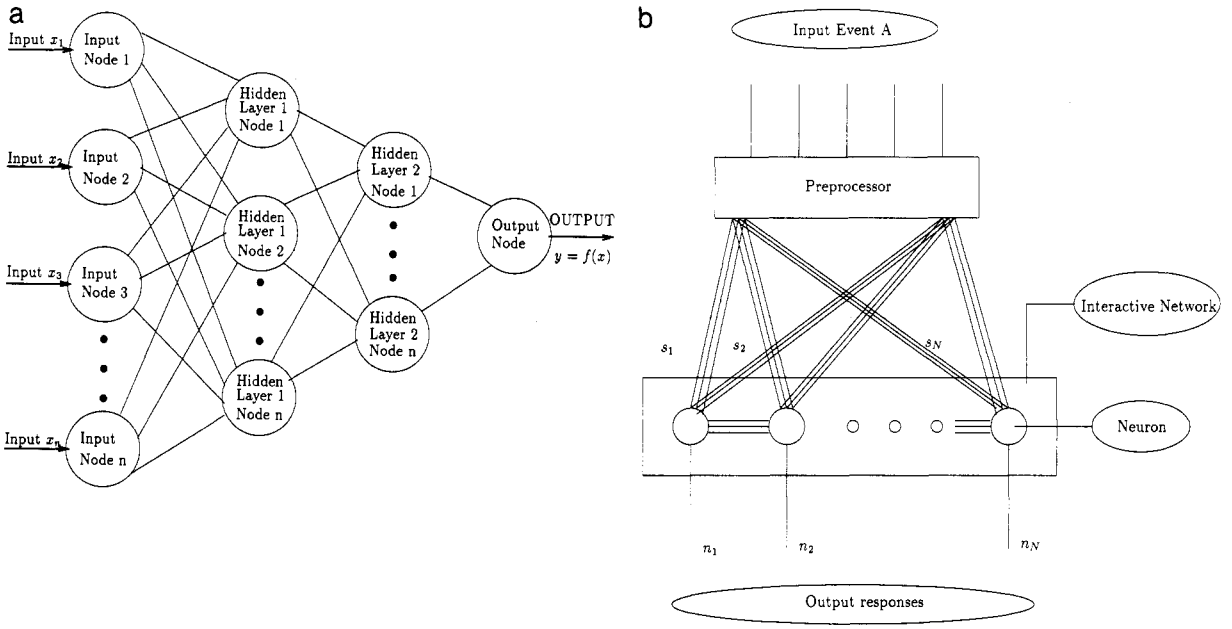


Figure 2. (a, left) Neural network based on the supervised learning and (b, right) a neural network based on the unsupervised learning.

Hopfield network, brain-state-in-a-box network, adaptive heuristic critic network, associative reward penalty network, and avalanche matched filter network, etc. In our case we are using back-propagation network. McCulloch–Pitts neurons are the basic units of neural networks, shown in Figure 1a.

The basic configurations of the neural networks based on the *supervised learning paradigm* is shown in Figure 2a and that based on the unsupervised learning paradigm in Figure 2b. In the case of the supervised learning paradigm based neural networks, all information flows forward from the input neurons, through the hidden or middle layer(s) and from there, to the output neurons. A sigmoidal function is used for thresholding. This learning procedure provides a deterministic mechanism for adjusting the connection weights for hidden (internal) units. Such units are essential for neural networks to deal with problems involving input pattern that are not linearly separable.[29]

There are two distinct operations which take place during the training phase; the first one is the feedforward computation and update of the weights based upon the error of the network's output. Typically each neuron in the network uses the same activation function and threshold value, although this is not necessary. First, the connection weights between the hidden and output layers are adjusted along with the activation thresholds in the output nodes. In the second stage, the connection weights between the input and the hidden layers are adjusted along with the activation thresholds in the hidden layer to reduce error in the output. This error is defined as

$$E_p = \frac{1}{2}\sum_j (t_{pj} - o_{pj})^2 \qquad (1)$$

where E_p is a measure of error on input/output pattern **p** and $E = \sum E_p$ is the overall mesure of error. Here t_{pj} is the value of target output, and o_{pj} is the output value of the network.

With hidden units, however, it is not so obvious to compute the derivatives because there is a possibility of getting stuck in local minima. Figure 1b shows different types of energy minima. The derivatives can be calculated by *generalized δ rule*.[29] The *generalized δ rule* works for layered feedforward networks with any number of hidden layers between the input and output layers. The following equation is used in the learning algorithm to calculate the change in weight (ξ) whenever the node (neuron) is an output node. Here, w_{kj} are

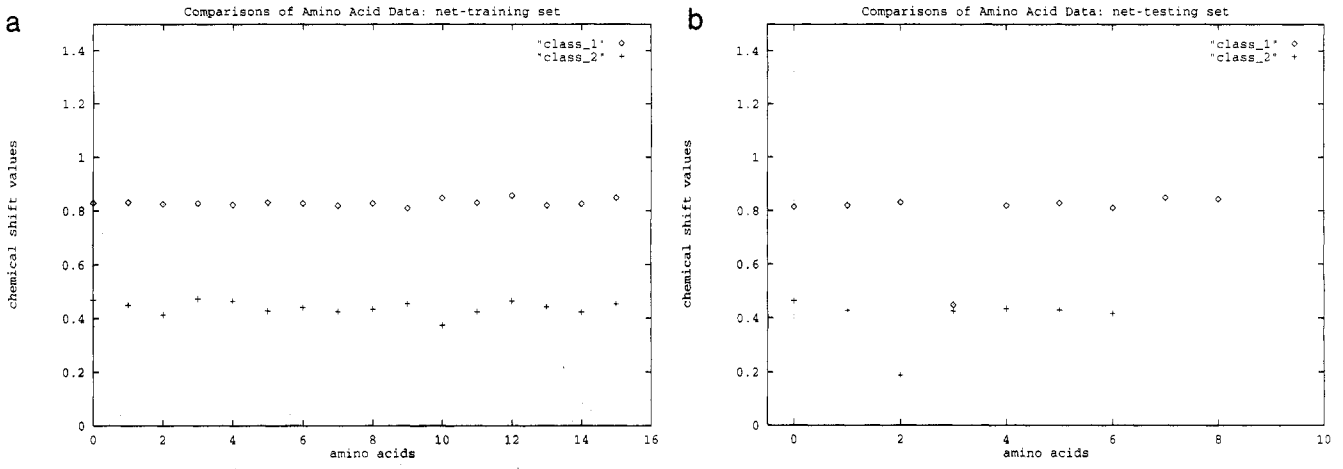


Figure 3. The dot diagrams of the data sets, based on chemical shift parameters of NH and H_α, (a) used for training and (b) for testing data files.

Table 1. All 20 Amino Acids, Divided into Eight Different Groups, Based on Their Atomic Structural Connectivities

group	typical structure feature	amino acids
1	secondary amino group	Pro
2	aliphatic side chain	Gly, Ala, Val, Leu, Ile
3	aliphatic hydroxyl side chain	Ser, Thr
4	aromatic side chain	Phe, Tyr, Trp
5	basic side chain	Lys, Arg, His
6	acidic side chain	Asp, Glu
7	amide side chain	Asn, Gln
8	sulfur-containing side chain	Cys, Met

Table 2. Statistical Parameters for *Data Set*

parameters	for NH	for H_α	values
range	0.470	0.990	
mean value	8.290	4.390	
variance	0.143	0.057	
std deviation	0.378	0.239	
covariance (NH, H_α)			+0.022
correlation coefficient (r_{NHH_α})			+0.244

the weights between the layers.

$$\xi_{pj} = o_{pj}(1 - o_{pj})\sum_k \xi_{pk} w_{kj} \tag{2}$$

The application of the generalized delta rule involves two phases. During the first phase, called the *forward phase*, the input is presented to and propagated forward through the network to compute the output value o_{pj} for each output unit. For each processing element, all current outputs are compared with the desired output, and the difference, or error, is computed. The second phase, called the *backward phase*, involves a backward pass through the network (analogous to the initial forward pass) during which the error signal is passed to each unit in the network and the appropriate weight changes are made. This second backward pass allows the recursive computation of ξ as indicated above. Only when these two phases are completed, can the new inputs be presented. In addition to these phases, a momentum term and a learning term are also used and a change of weights is given by

$$\Delta w_{ji}(n + 1) = \beta(\xi_{pj} o_{pi} + \gamma \Delta w_{ji}(n)) \tag{3}$$

where β is the gain and γ is a constant which determines the effect of past weight changes on the current direction of movement in weight space. The learning term is given by

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \tag{4}$$

where

$$x = \sum_i w_{ij} o_{pi} - \Theta$$

Here α is the learning rate, and Θ is the threshold of the node. If α is greater than 1, then the slope of the sigmoid curve gets steeper. The design of an efficient algorithm includes all the factors such as gain, learning rate, and momentum.

Neural Networks Based on the Unsupervised Learning Paradigm. Some of the neural networks of this type are adaptive Grossberg network, shunting Grossberg network, analog adaptive resonance theory (ART) network, adaptive bidirectional associative memory (BAM) network, drive-reinforcement network, linear associative memory network, Kohonen's learning vector quantization (LVQ) network, and counterpropagation network, etc. Any of the above mentioned neural networks can be used for classification and clustering. We decided to use Kohonen LVQ net because standard and tested software package, written by Kohonen, is available. The Kohonen's LVQ has two layers (Figure 2b), an input layer and an output layer with extensive interconnections. When the input is presented, the node in the output layer with the minimum distance to the input is chosen. Then all the weights to this output node and its neighbors are adjusted to bring them even closer to the input. The self-organizing network adjusts itself with each input pattern. The learning procedure developed by Teuvo Kohonen[41] was inspired by learning in biological systems. These networks consist of a single layer of neurons known as a Kohonen layer. Each neuron in a Kohonen layer receives the input pattern and computes the dot product of its weight vector with the input vector. The neuron with the largest dot product is declared the winner. Only the winner neuron and its neighboring neurons are permitted to learn. Only the winner neuron and its neighboring neurons are permitted to learn. The Kohonen learning rule can be defined as

$$\mathbf{W}_{new} = W_{old} + \alpha \cdot (\mathbf{S} - \mathbf{W}_{old}) \tag{5}$$

where $\mathbf{W}$ is the weight vector, and $\mathbf{S}$ is the input pattern, it identifies the input pattern with a "codebook" vector, and α is the learning rate.

In Kohonen learning, the winner neuron and its neighboring neurons perform sililar adjustments. The input vector

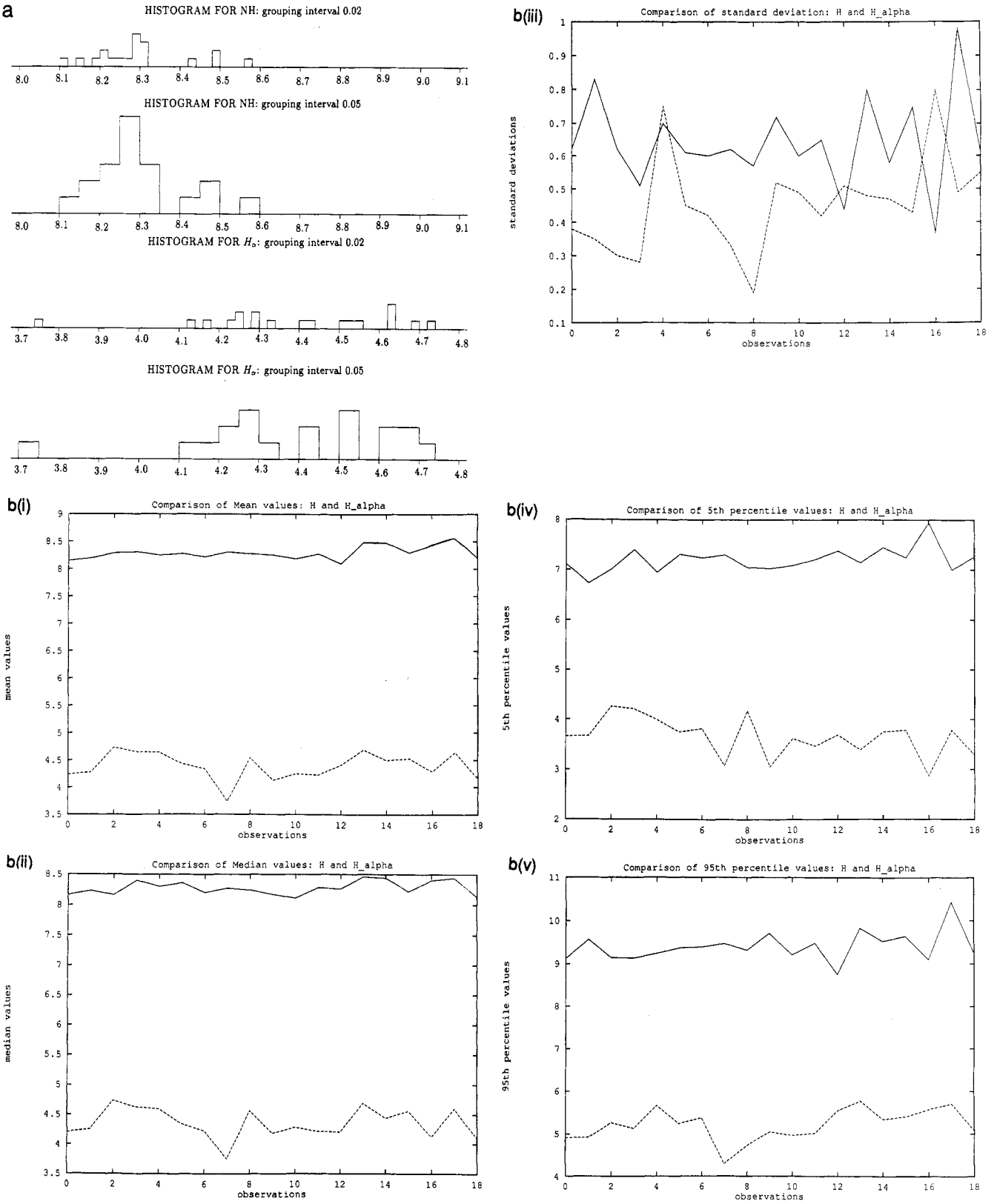


Figure 4. The statistical analysis based on NH and H_α chemical shifts showing (a) histogram and (b) comparative plots for mean, median, standard deviation, 5th percentile, and 95th percentile (The NH values are shown by solid lines and H_α by dotted lines.).

$$\mathbf{S} = s_1, s_2, \ldots, s_n$$

is connected to all the units. Each unit is associated with a variable vector denoted as

$$m_i(t) = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{in}] \quad (6)$$

In this case, the winning element is defined as the one that has the closest match to the input pattern. Generally with an increment of time, the size of the neighborhood reduces till it becomes similar in response and properties, to winning element, and then a global organization begins to take shape. The application of a neural network to a particular problem involves the choice of network topology—that is, the number of layers, the size of each layer, and the pattern of connections—and the assignment of connection strengths to each pair of connected units and of thresholds to each unit.[4]

Table 3. Tuning Parameters for Both the Learning Paradigms Based Neural Networks

parameters	neural network based on supervised learning paradigm	neural network based on unsupervised learning paradigm
bias	+1.0	0
learning rate	0.05	0.15
momentum	0.7	0
max. total error	0.00001	0
max. individual error	0.000001	0
no. of code vectors	0	7
no. of hidden layers	1	0
no. of nodes in hidden layers	12	0
max. no. of iterations	2000000	5000

For *feature enhancements* on every step, discrete Hopfield network is used. Hopfield network allows information to be stored distributively in the structure of the neural networks for the purpose of retrieval from partial or noisy information. For more details, refer to.[39,40]

The amino acids are the basic building blocks of proteins in all living beings on this plantet. They can be divided into eight groups on the basis of their structural characteristics,[42] shown in Table 1. In the next section we will discuss how these amino acids have been classified and clustered using their NH and $C_\alpha H$ ^1H chemical shift characteristics by applying above mentioned neural networks.

2. EXPERIMENTATION

Neural networks based on both paradigms were simulated and evaluated for performance. All the simulations were carried out on a SUN-4/260CXP workstation.

It is important to explore all possible characteristics of the input data set, before applying neural networks. The data set we used for testing consists of typical NH and H_α chemical shift values $\langle\delta\rangle$ of all amino acids[34] which is also shown in Table 9. It is a moderately difficult data set to deal with. We generated two different files: a *training data* file and a *testing data* file. The training data set consists of 32 entries of NH and H_α for amino acids. The testing data set consists of 14 entries of NH and H_α values. Different graphical techniques, such as *Gnuplot*, *Mathematica*, etc., were used to find spatial distributions of data items. The data set files based on NH and H_α of amino acids are shown in Figure 3 in forms of dots, and Figure 4a shows the histograms.

For systematic data analysis, *statistical tools* were also used. The comparative plots of mean, median, standard deviations, 5th percentiles, and 95th percentiles for NH and H_α chemical shift values are shown in Figure 4b. The statistical parameters are listed in Table 2.

Neural network tuning refers to the setting of different parameters such that the neural networks will perform the required task. Before applying neural networks to any task, the tuning should be the first and foremost step. Tuning plays a critical role for achieving the best possible results. The tuning parameters for neural networks based on both paradigms are tabulated in Table 3.

Once tuning was completed and we were satisfied with performance, both paradigms based neural networks tested for classification and clustering tasks. In the case of neural networks based on supervised learning paradigm, at the optimum tuning stage, the recognition accuracy could reach to 100%. The optimum tuning stage parameters are presented

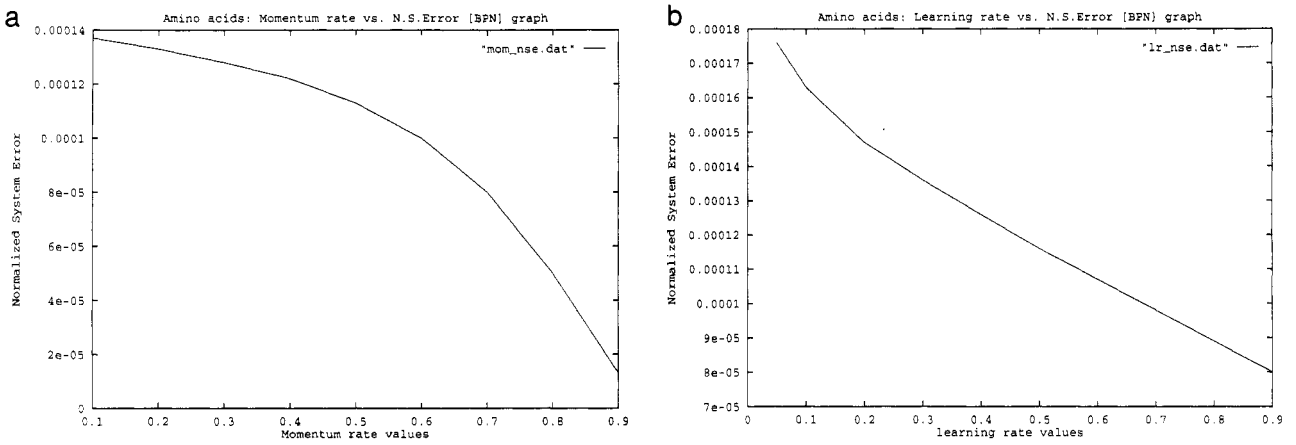


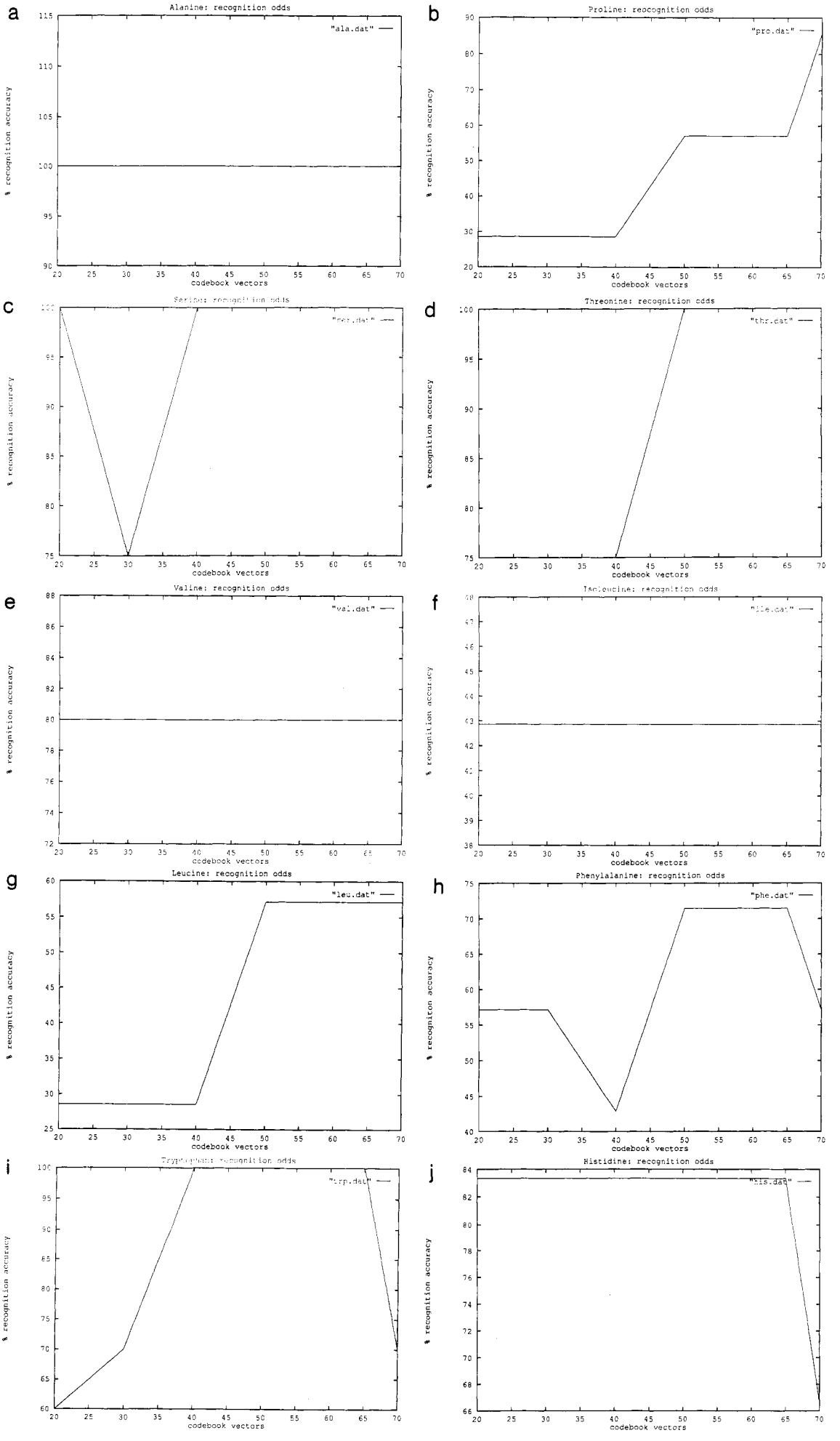
Figure 5. The rate of change of normalized system error (NSE) (a) with momentum (η) and (b) with learning rates (α) during classification process.

Table 4. Recognition Odds for Training Set

s. no.	actions	values
1	code vectors chosen	4, 5, 6, 7, and 8
2	initial learning rates	0.05, 0.10, 0.15, 0.20, 0.25, and 0.30
3	iterations	1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, and 10000
4	recognition accuracy of 1st entry (16 entries)	100%
5	recognition accuracy of 2nd entry (16 entries)	100%
6	total accuracy (32 entries)	100%

Table 5. Recognition Odds for Testing Set

s. no.	actions	values
1	code vectors chosen	4, 5, 6, 7, and 8
2	initial learning rates	0.05, 0.10, 0.15, 0.20, 0.25, and 0.30
3	iterations	1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, and 10000
4	recognition accuracy of 1st entry (7 entries)	100%
5	recognition accuracy of 2nd entry (7 entries)	100%
6	total accuracy (7 entries)	100%


a
Alanine: recognition odds
"ala.dat"
% recognition accuracy
codebook vectors
b
Proline: recognition odds
"pro.dat"
% recognition accuracy
codebook vectors
c
Serine: recognition odds
"ser.dat"
% recognition accuracy
codebook vectors
d
Threonine: recognition odds
"thr.dat"
% recognition accuracy
codebook vectors
e
Valine: recognition odds
"val.dat"
% recognition accuracy
codebook vectors
f
Isoleucine: recognition odds
"ile.dat"
% recognition accuracy
codebook vectors
g
Leucine: recognition odds
"leu.dat"
% recognition accuracy
codebook vectors
h
Phenylalanine: recognition odds
"phe.dat"
% recogniton accuracy
codebook vectors
i
Tryptophan: recognition odds
"trp.dat"
% recognition accuracy
codebook vectors
j
Histidine: recognition odds
"his.dat"
% recognition accuracy
codebook vectors

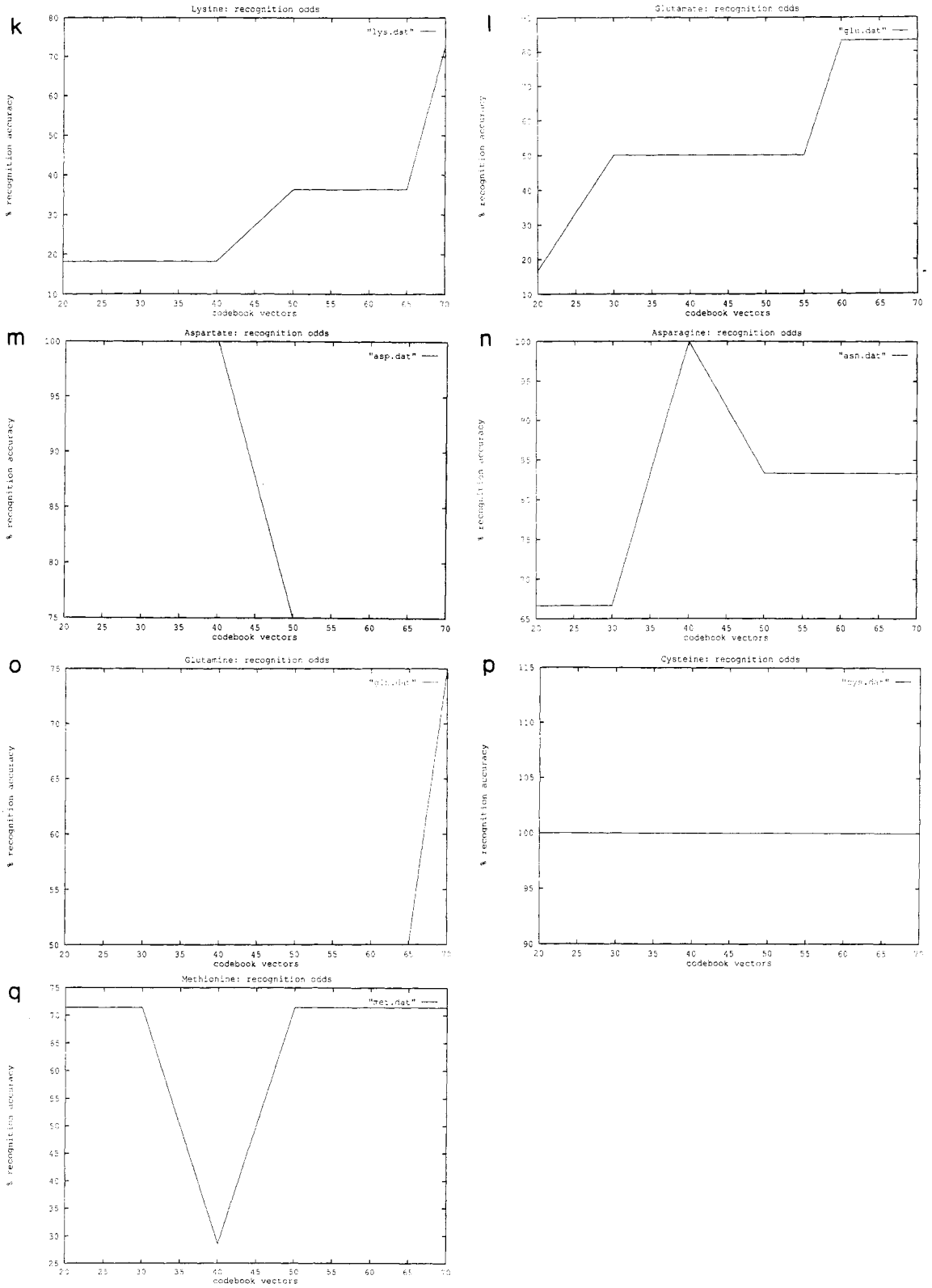


Figure 6. The percentages (%) of recognition accuracies for all eight groups of amino acids: (1) (a) alanine and (b) proline, (2) (c) serine and (d) threonine, (3) (e) valine, (f) isoleucine, and (g) leucine, (4) (h) phenylalanine and (i) tryptophan, (5) (j) histidine and (k) lysine, (6) (l) glutamate and (m) aspartate, (7) (n) asparagine and (o) glutamine, and (8) (p) cysteine and (q) methionine.

in Table 3. For neural networks based on unsupervised learning paradigm the results are shown in Tables 4 and 5.

The details study of change of tuning parameters on recognition accuracies were carried out for both paradigms based neural networks. In case of the neural networks based on supervised learning paradigm, some of the parameters changed were momentum rate (η), learning rate (α), and number of iterations. The rate of change of normalized system error (NSE) [NSE is the measure of difference between actual output and desired output.] with momentum and learning rates are plotted in Figure 5.

The similar studies were carried out for the neural networks based on unsupervised learning paradigm. In this case, the effects of learning rate on the total accuracy and threshold value input on the numbers of clusters formed were taken into account. The results are tabulated in Tables 4 and 5.

The scheme of combining neural networks based on both learning paradigms allows us to handle a large body of data

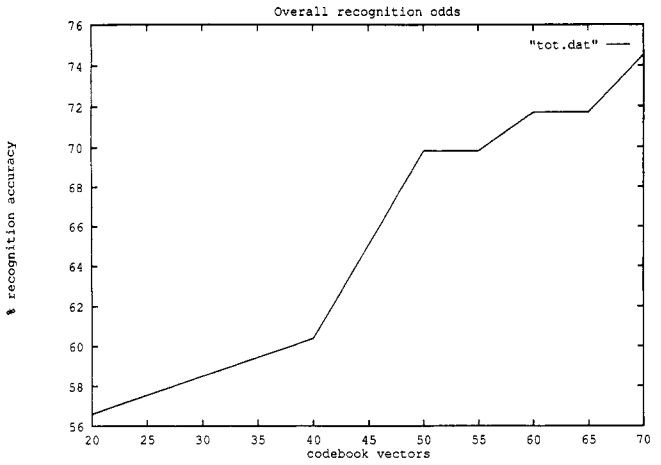


Figure 7. Rate of change of overall recognition accuracy for all amino acids.

Table 6. Threshold Value vs Number of Clusters for the *Data Set*

sequence	threshold value input	no. of clusters formed	% of elements in cluster
1	3	19	05.26
2	4	11	05.91
3	5	8	12.50
4	6	8	12.50
5	7	8	12.50
6	8	7	14.29
7	9	7	14.29
8	10	6	16.67
9	15	6	16.67
10	18	6	16.67
11	20	6	16.67
12	22	6	16.67
13	24	6	16.67
14	30	5	20.00
15	34	5	20.00
16	47	4	25.00
17	59	3	33.33

and performs classification and clustering more efficiently as compared to existing neural networks based on a single paradigm. All the amino acids were divided into eight groups, based on their chemical structure, shown in Table 1. Change of individual recognition accuracies for amino acids along the clustering process are shown in Figure 6. The rate of change of *combined recognition accuracies* for all amino acids is shown in Figure 7. It is helpful in zeroing optimum recognition accuracies of amino acids.

All the above mentioned studies were conducted in normal conditions. Further studies were also carried out using the hostile conditions. Some of the important parameters responsible for causing these conditions were pruning the nodes, altering neural network layers, and even altering the whole neural network structures. After the neural networks were tuned and trained, experiments were carried out while pruning the nodes and even altering the neural network structure by removing the layers. The neural networks were tested even after removal of two layers. In another test, we removed up to 20% of the nodes in steps of 2%, before applying the recognition task. We did not stop here, and neural network performances were tested even after altering the whole structures. By retuning the neural networks, it became possible to achieve almost the same results.

3. DISCUSSION

It is evident from data analysis that mean chemical shifts are sparsely distributed, not linearly separable, and there are clear indications of overlaps. Evaluation of the data in Figure 4 reveals that NH and H_α are not closely related. The histograms of both data distributions have larger tails on the left, showing that both the data sets are *negatively skewed.* The amount of scatter about the mean is more in case of NH values compared to H_α which is seen by comparison of the standard deviations.

The simulation programs were tested for optimization before applying them to these data. At this point, it appeared to be quite simple to perform *neural network tuning.* However, this task turned out to be one of the most difficult ones of the entire research process. In our case the job is to classify and perform cluster analysis. It is desirable to have neural network systems with many parameters, because this allows the user to customize the network for his own needs. But at the same time it gives too much freedom to the user, and, without any default or reference value, this customization may be very hard to do. This was exactly the case in our present project. None of the previous research contains pertinent advice to what the value of the different parameters should be. And perhaps, even if it would have been available, it would not apply to the use of these neural networks based on amino acid chemical shift $\langle\delta\rangle$ values.

The neural networks were tuned as well as possible. Graphical displays were done regularly, to observe the status of clustering, using *Sammon* [A graphical tool developed by Kohonen and his associates.]. Often the setting of a parameter was changed by trial and error until the desired results were obtained. A list of the final set of parameters, used for finding global minima, used for the *test data set* are given in Table 3.

Many times neural networks have the tendency to go to local minima. That is why it is important to find global minima. Different types of minima are shown in Figure 1b.

For the classification task, during normal conditions, both of the neural network paradigms were used. Both types of neural networks have achieved 100% recognition accuracies but with a difference—the neural networks based on the supervised learning paradigm could attain 100% recognition rates only using a specific set of tuning parameters (refer to Table 3), while networks based on the unsupervised learning paradigm gave 100% recognition rates over a wide range of tuning parameters. This means that in the second case the performance was more robust.

After the neural networks were tuned and trained, we altered the neural network structure by removing intermediate one to two layers. For example, for the neural network trained with four layers, recognition was carried out after removing one layer. The net effect was a reduced recognition rate. By retuning the parameters, the simpler network structure could provide almost the same recognition accuracy.

Neural networks based on the supervised learning paradigm have great generalized capabilities and are very powerful; however, they require extensive training. They have additional problems: one problem is local minima because they use a gradient descent algorithm. This may require time-consuming iterations of training. Another problem is paralysis of the networks because nonlinearity is sigmoidal. Weight changes are based on the difference in the outputs between what is desired and what is actually calculated by the network and the derivative of the sigmoid. When the weights get large, the output values will fall on the flat part of the sigmoid, and the derivatives will be very small, causing a very small change in the weights. Neural networks therefore can converge very slowly in these circumstances.

Table 7. Comparative Study of Supervised Learning and Unsupervised Learning Paradigms Based Neural Networks

parameters	supervised learning paradigm based neural networks	unsupervised learning paradigm based neural networks
training time	52 min	30 s
testing time	1 min	5 s
design constraints	yes	yes
stability	no	yes
accuracy	no	yes

Table 8. Applying Combinations of Both the Paradigms Based Neural Networks to *Test Case*

data set	recognition rate (%) in normal conditions	recognition rate (%) in hostile conditions
chemical shifts ⟨δ⟩ of amino acids	100.00	84.53

Table 9. Distribution of Chemical Shifts ⟨δ⟩ in ^{1}H NMR Spectra of Polypeptides and Proteins[42] [a]

residue	atom	δ (ppm)	s (ppm)	N	P5 (ppm)	median (ppm)	P95 (ppm)
Ala	*NH*	8.15	0.62	99	7.11	8.16	9.12
	H_α	4.24	0.38	102	3.66	4.21	4.91
Arg	*NH*	8.20	0.83	46	6.73	8.23	9.57
	H_α	4.28	0.35	49	3.67	4.26	4.92
Asn	*NH*	8.29	0.62	53	7.01	8.16	9.14
	H_α	4.73	0.30	54	4.26	4.74	5.26
Asp	*NH*	8.31	0.51	43	7.40	8.40	9.13
	H_α	4.65	0.28	44	4.21	4.63	5.13
Cys	*NH*	8.25	0.70	91	6.95	8.30	9.24
	H_α	4.64	0.75	93	4.00	4.60	5.66
Gln	*NH*	8.28	0.61	38	7.31	8.36	9.37
	H_α	4.43	0.45	39	3.75	4.35	5.24
Glu	*NH*	8.22	0.60	66	7.24	8.20	9.39
	H_α	4.34	0.42	66	3.82	4.23	5.37
Gly	*NH*	8.31	0.62	84	7.30	8.27	9.47
	H_α	3.74	0.33	91	3.08	3.75	4.29
His	*NH*	8.28	0.57	19	7.05	8.24	9.32
	H_α	4.54	0.19	19	4.18	4.57	4.72
Ile	*NH*	8.26	0.72	43	7.03	8.17	9.71
	H_α	4.13	0.52	45	3.06	4.19	5.05
Leu	*NH*	8.19	0.60	55	7.09	8.11	9.21
	H_α	4.25	0.49	57	3.62	4.29	4.96
Lys	*NH*	8.28	0.65	91	7.21	8.28	9.49
	H_α	4.23	0.42	95	3.47	4.22	5.02
Met	*NH*	8.10	0.44	16	7.38	8.26	8.75
	H_α	4.41	0.51	18	3.69	4.21	5.54
Phe	*NH*	8.49	0.80	37	7.14	8.46	9.83
	H_α	4.69	0.48	38	3.39	4.69	5.77
Pro	*NH*	0.00	0.00	00	0.00	0.00	0.00
	H_α	4.48	0.31	51	3.94	4.42	4.91
Ser	*NH*	8.48	0.58	58	7.45	8.44	9.53
	H_α	4.50	0.47	61	3.75	4.44	5.34
Thr	*NH*	8.30	0.75	76	7.24	8.22	9.64
	H_α	4.53	0.43	79	3.78	4.56	5.40
Trp	*NH*	8.43	0.37	8	7.95	8.40	9.10
	H_α	4.29	0.80	8	2.87	4.13	5.57
Tyr	*NH*	8.57	0.98	42	7.00	8.44	10.44
	H_α	4.64	0.49	44	3.78	4.60	5.70
Ala	*NH*	8.20	0.61	69	7.26	8.13	9.24
	H_α	4.16	0.55	72	3.29	4.10	5.08

[a] Only the *NH* and H_α values are listed.

While neural networks based on the supervised learning paradigm have high performance when given patterns similar to those on which they are trained, they do not have ability to recognize new categories of patterns. The key characteristics of these networks is that they perform an arbitrary nonlinear mapping from a set of input stimuli to a set of output nodes using features extracted from the input patterns [By features, we mean the correlation of activity among different input nodes.]. The mapping procedure is not well understood, and there is no guarantee that the system will converge. There is no evidence that this method is used in biological systems. Unlike the neural networks based on the unsupervised learning paradigm, these networks can store many more patterns than the number of dimensions.

The neural networks based on the unsupervised learning paradigm also require considerable self-training time (t), but much less, compared to the supervised learning paradigm based neural networks. Unsupervised learning paradigm based networks are generally useful when the amount of input data is large, relative to the number of clusters required. These networks are potentially very fast and can work in real time. This means the networks could learn continuously and adapt to changes over time.

During the process of research we also explored hardware implementation possibilities for both neural networks based on the two paradigms. Neural networks based on the supervised learning paradigm are easy to implement in hardware, but that is not yet the case with neural networks based on the unsupervised learning paradigm.

4. CONCLUSIONS

This study concludes that neural networks based on the supervised learning paradigm require a lot of training time, are unstable, very sensitive to tuning parameters, and generally get stuck in local minimas. On the other hand the neural networks based on the unsupervised learning paradigm are significantly faster and are less sensitive to tuning parameters. The neural networks based on both paradigms simulatneously along with feature enhancement on every step provide a much faster, robust solution which is less sensitive to tuning parameters. Results from comparative studies of the neural networks based on the supervised learning paradigm and the unsupervised learning paradigm are presented in Table 7.

Table 8 shows the results obtained by applying both the paradigms based neural networks together with feature enhancement on every step to the *test case* of recognition of amino acids for ^{1}H NMR spectra.

REFERENCES AND NOTES

(1) Caudill, M.; Butler, C. *Naturally Intelligent Systems*; MIT Press: Cambridge, 1990.

(2) Borman, S. Neural Network Applications In Chemistry Begin To Appear. *Chem. Eng. News* **1989**, *67*(17), 24–28.

(3) Qian, N.; Sejnowski, T. J. "Predicting the Secondary Structure of Globular Proteins Using Neural Network Models", *J. Mol. Biol.* **1988**, *202*, 865–884.

(4) Holley, L. H.; Karplus, M. Protein secondary structure prediction with a neural network. *Proc. Natl. Acad. Sci. U.S.A.* **1989**, *86*, 152–156.

(5) Kneller, D. G.; Cohen, F. E.; Langridge, R. Improvements in Protein Secondary Structure Prediction by An Enhanced Neural Network. *J. Mol. Biol.* **1990**, *214*, 171–182.

(6) Giacomini, M.; Parisini, T.; Ruggiero, C. Secondary structure of proteins from NMR data by neural nets. *Artificial Neural Networks* **1992**, *2*, 1603–1606.

(7) Aoyama, T.; Suzuki, Y.; Ichikawa, H. Neural Networks Applied to Quantitative Structure–Activity Relationship Analysis. *J. Med. Chem.* **1990**, *33*, 2583–2590.

(8) Aoyama, T.; Ichikawa, H. Obtaining the Correlation Indices between Drug Activity and Structural Parameters Using a Neural Network. *Chem. Pharm. Bull.* **1991**, *39*, 372–378.

(9) Mikhailov, A. S.; Tereshko, V. M. Global optimization by raction-diffusion systems. In *Proceedings of the 1991 International Conference. ICANN-91*; Kohonen, T., Makisara, K., Simula, O., Kangas, J., Eds.; **1990**, *30*, 447–484.

(10) Ungar, L. H.; Powell, B. A.; Kamens, S. N. Adaptive networks for fault diagnosis and process control. *Comput. Chem.* 4Eng. **1990**, *14*, 561–572.

(11) Egolf, L. M.; Jurs, P. C. Prediction of Boiling Points of Organic Heterocyclic Compounds Using Regression and Neural Network Techniques. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 616–625.

(12) Yee, E.; Ho, J. Neural network recognition and classification of aerosol particle distributions measured with a two-spot laser velocimeter. *Appl. Opt.* **1990**, *29*, 2929–2938.
(13) Brunak, S.; Engelbrecht, J.; Knudsen, S. Neural network detects errors in the assignment of mRNA splice sites. *Nucleic Acids Res.* **1990**, *18*, 4797–4801.
(14) Maggiora, G. M.; Elrod, D. W. Computational Neural Networks as Model-Free Mapping Devices. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 732–741.
(15) West, G. M. J. Predicting Phosphorus NMR Shifts Using Neural Networks. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 577–589.
(16) Thomsen, J. U.; Meyer, B. Pattern Recognition of the ^{1}H NMR Spectra of Sugar Alditols Using a Neural Network. *J. Magn. Reson.* **1989**, *84*, 212–217.
(17) Meyer, B. Identification of the ^{1}H-NMR Spectra of Complex Oligosaccharides with Artificial Neural Networks. *Science* **1991**, *251*, 542–544.
(18) Curry, B.; Rumelhart, D. A neural network which classifies mass spectra. *Tetrahedron Comput. Methodol.* **1990**, *3*, 213–237.
(19) Munk, M. E.; Madison, M. S.; Robb, E. W. Neural Network Models for Infrared Spectrum Interpretation. *Mikrochim. Acta* [*Wien*] **1991**, *II*, 505–514.
(20) Meyer, M.; Wiegelt, T. Interpretation of Infrared Spectra by Artificial Neural Networks. *Anal. Chim. Acta* **1992**, *265*, 183–90.
(21) Ricard, D.; Cachet, C.; Cabrol-Bass, D. Neural Network Approach to Structural Feature Recognition from Infrared Spectra. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 202–210.
(22) Gezelter, J. D.; Freeman, R. Use of Neural Networks to Design Shaped Radiofrequency Pulses. *J. Magn. Reson.* **1990**, *90*, 397–404.
(23) Long, J. R.; Gregoriou; Gamberline, P. J. Spectroscopic Calibration and Quantitation Using Artificial Neural Networks. *Anal. Chem.* **1990**, *62*, 1791–1797.
(24) Long, J. R.; Mayfield, H. T.; Henley, M. V.; Kromann, P. R. Classification of fuels from Chromatographic data. *Anal. Chem.* **1991**, *63*, 1256–1261.
(25) Anker, L. S.; Jurs, P. C. Prediction of Carbon-13 Nuclear Magnetic Resonance Chemical Shifts by Artificial Neural Networks. *Anal. Chem.* **1992**, *64*, No. 10, 1157–1164.
(26) Doucet, J. P.; Panaye, A.; Feuilleaubois, E.; Ladd, P. Neural Networks and ^{13}C NMR Shift Prediction. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 320–324.
(27) Kvasnicka, V.; Sklenak, S.; Pospichal, J. Application of Neural Networks with feedback connections in Chemistry: prediction of ^{13}C NMR Chemical Shifts in a Series of Monosubstituted Benzenes. *THEOCHEM* **1992**, *96*, 87–107.
(28) Kvasnicka, V.,; Sklenak, S.; Pospichal, J. Application of Recurrent Neural Networks in Chemistry. Prediction and Classification of ^{13}C NMR Chemical Shifts in a Series of Monosubstituted Benzenes. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, No. 6, 742–747.
(29) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vols. I, II, & III*; MIT Press: Cambridge, MA, 1986.
(30) Neidig, K.-P.; Saffrich, R.; Lorenz, M.; Kalbitzer, H. R. Cluster Analysis and Multiplet Pattern Recognition in Two-Dimensional NMR Spectra. *J. Magn. Reson.* **1990**, *89*, 543–552.
(31) Bax, A. Two-Dimensional NMR and Protein Structure. *Annu. Rev. Biochem.* **1989**, *58*, 223.
(32) Wagner, G. NMR Investigations of Protein Structure. *Prog. Nucl. Magn. Reson. Spectrosc.* **1990**, *22*, 101.
(33) Wüthrich, K. NMR of Proteins and Nuclear Acids; Wiley-Interscience: New York, 1986.
(34) Gross, K.-H.; Kalbitzer, H. R. Distribution of Chemical Shifts in ^{1}H Nuclear Magnetic Resonance Spectra of Proteins. *J. Magn. Reson.* **1988**, *76*, 87–99.
(35) Kjaer, M.; Poulsen, F. M. Identification of 2D ^{1}H NMR Antiphase Cross Peaks Using a Neural Network. *J. Magn. Reson.* **1991**, *94*, 659–663.
(36) Corne, S. A.; Johnson, P. A. An Artificial Neural Network for Classifying Cross Peaks in Two-Dimensional NMR Spectra. *J. Magn. Reson.* **1992**, *100*, 256–266.
(37) Anand, R.; Mehrotra, K.; Mohan, C. K.; Ranka, S. Analyzing Images Containing Multiple Sparse Patterns with Neural Networks; In *12th International Joint Conference on Artificial Intelligence*; 1993; pp 838–843.
(38) Carrara, E. A.; Pagliari, F.; Nicolini, C. Neural Networks for the peak-picking of nuclear magnetic resonance spectra. *Neural Networks* **1993**, *6*, 1023–1032.
(39) Hopfield, J. J. Neural networks and physical systems with emergent collective computational capabilities. In *Proc. Natl. Acad. Sci. U.S.A.* **1982**, *79*, 2554–2558.
(40) Hopfield, J. J. Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons. In *Proc. Natl. Acad. Sci. U.S.A.* **1984**, *81*, 3088–3092.
(41) Kohonen, T. Self Organization and Associative Memory; Springer-Verlag: Berlin, Heidelberg, 1987.
(42) Stryer, L. Biochemistry; W. H. Freeman and Company: New York, 1981.