A neural network approach to crystal structure classification

K. Ravindra Shetty*, Ashok Rao and K. Gopala†**

*LG Soft, 5th Floor, Golf View Homes, Bangalore 560 017, India
**CART, NIE, Mysore 570 008, India
†Department of Physics, University of Mysore, Mysore 570 006, India

This paper focuses classification of crystal classes in a periodic table using the known neural network (NN) learning algorithm, viz. generalized delta rule (GDR) by feeding the set of input features in max–min–max sub arrays. We have taken eighteen independent physical parameters for each element, trained the network from atomic number (AN) 1 to 84 and we validated the crystal class from AN 86 to 95 from the trained network and achieved 100 per cent accuracy, which was later extended from AN 96 to 120. Further, we have also evaluated the dependencies of the neural network in different confidence intervals and hidden layers. We would like to call this learning algorithm as max–min–max GDR.

ARTIFICIAL neural network (ANN) is one of the burgeoning areas of current research and it is attracting people from a wide variety of disciplines of science and technology. It is known that the human brain is built of cells called neurons. A collection of neurons are linked together in such a way that individual neurons can perform separate functions simultaneously. Such a collection of neurons is called a neural network (NN)[1].

In ANN the fundamental unit that we employ is an approximated electronic/mathematical model of a neuron. The connection strength between layers is called weight. The process of adjustment of weights is called learning or training[2]. Humans are intelligent because evolution has equipped them with a richly structured brain. This structure, while serving a variety of functions, enables them to learn. Learning procedure is constructing new representations, and the results of learning can be viewed as numerical solutions to the problem of whether to use local or distributed representations[3]. Basically there are three types of learning: Supervised learning (training with teacher), Unsupervised learning (training without teacher), and Hybrid learning (which falls between supervised and unsupervised learning). Reinforcement learning is a variant of supervised learning wherein the network is provided with only a critique on the correctness of the network outputs, not correct answers themselves[4].

The work on ANN models started around early 40s. Basically there are four main landmarks, viz. Mc Culloch and Pitt's pioneering work (1940s), Rosenblatt's perceptron rule – LMS algorithm (1960), the Back propagation algorithm (1974s) which has been modified a number of times, and Hopfield's energy approach (1982)[5].

*For correspondence. (e-mail: ravindrak@lgsi.co.in)

Figure 1 *a* depicts an example of a typical processing unit for an ANN. On the left are the multiple inputs to the processing unit, each arriving from another unit, which is connected to the unit shown at the center. Each interconnection has an associated connection strength, given as $W_1, W_2, \cdots, W_n$. The processing unit performs a weighted sum on the inputs and uses a nonlinear threshold function (f), to compute its output. The calculated result is sent along the output connections to the target cells shown at the right. In the above network output (Y_i) is given by

$$Y_i = f_s \sum_{j=1}^{n} W_{ji} X_i. \tag{1}$$

Commonly used activation (squashing) functions are: threshold function, piecewise linear, sigmoid (Figure 1 *b*), and gaussian. The sigmoid function is by far the most frequently used in ANNs. It is a strictly increasingly function that exhibits smoothness and has the desired asymptotic properties. This is a crude analogy to a biological neuron: wires and interconnections model axons and dendrites; connection weights represent synapses, and the threshold function approximates the activity in soma.

Capabilities of the network can be further enhanced by cascading a group of single ANNs, to form a multilayer neural network. Output of one layer is the input to the next layer as shown in Figure 1 *c*. Based on the connection pattern (architecture), ANNs can be grouped into two categories: (i) feedforward (static or memoryless), in which graphs have no loops, and (ii) recurrent (feedback or dynamic) networks, in which loops occur because of feedback connections.

There are three main areas of research on connectionist networks: search, representation, and learning. This paper focuses on learning. We have used neural networks for periodic table crystal classification because they have no mathematical models for prediction or classification. To increase capability of the network, we have used multilayer network. For better accuracy of the output we have used modified sigmoid function at the internal representation units, and the output at node j (Figure 1 *c*) is $f_j(x)$ and is given by,

$$f_j(x) = \frac{1}{1.005 + e^{-(\text{Net X} + \theta_j)/\theta_0}}, \tag{2}$$

where

$$\text{Net X} = \sum_{j}^{n} W_{ij} X_j, \ \theta_j,$$

serves as a threshold or bias, its function is to shift the activation function to the left along the horizontal

axis. θ_0 will modify the shape of the sigmoid function as shown in Figure 1 *b*.

We have trained the NN for different numerical constants (1.5, 1.45, · · ·, 0.50) at the denominator of equation (2), and achieved minimum error for 1.005. This function is highly nonlinear, continuous, and continuously differentiable. The back propagation algorithm (BPA) is an iterative gradient algorithm designed to minimize the mean square error between the actual output and the desired output. This algorithm is also known as 'The generalized delta rule'. The neurons in layers, other than the input and output layers are called hidden units or hidden nodes, as their outputs do not directly interact with the environment. With the BPA, the weights associated with hidden layers can also be adjusted and thus enable the ANN to learn. For our above-stated problem we tried with number of hidden layers from 1 to 4, but we got better performance when the hidden layer was one, and in that layer three nodes were taken. In the following section we discuss the working of NN.

In practice NNs cannot provide the solution working by themselves alone, rather, these need to be integrated into a consistent system engineering approach. Specifically, a complex problem of interest is decomposed into a number of relatively simple tasks, and NNs are assigned a subset of the tasks (e.g. pattern recognition, associative memory control) that match their inherent capabilities. In its simplest 'feedforward' form, a neural network is a collection of connected activatable units ('neurons'), wherein the connections are real value weights. The network is presented with an activation pattern on its input units, for example a set of numbers representing features of an image to be classified (say the pixels in an image of a letter M). Activation spreads in forward direction from the input units to output units through between layers (hidden layers) over the weighted connections. Typically, the activation coming into a unit from other units is multiplied by the weights on the links over which it spreads only in forward direction in the case of feedforward networks (both directions in the case of recurrent or feedback networks), and then is added together with other incoming activation. The result is thresholded (i.e. the unit 'turns on' if the resulting activation is above the units threshold). This

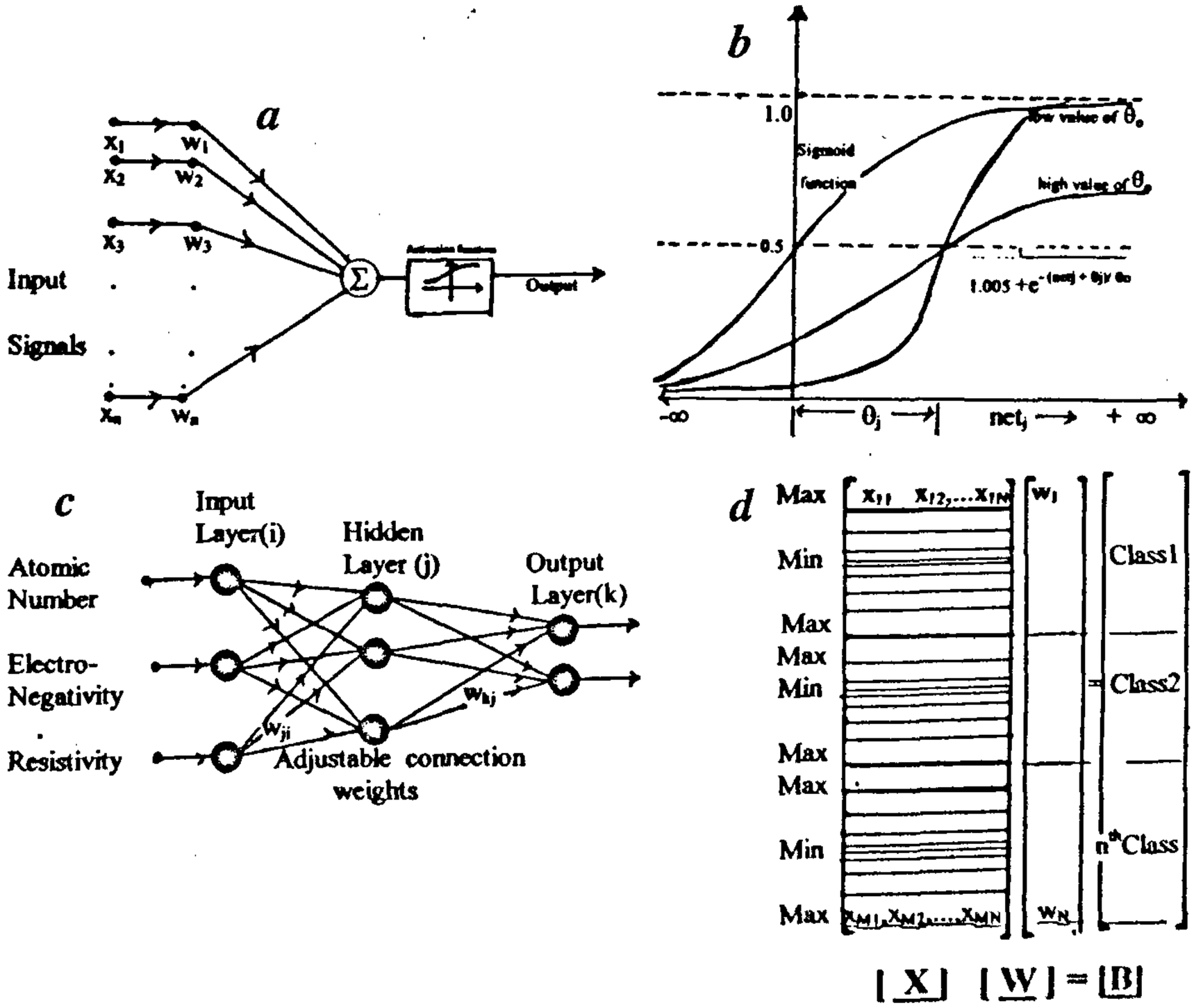

Figure 1. *a*, Nonlinear model of a neuron; *b*, The modified sigmoid function; *c*, Basic architecture of an ANN for crystal classification; *d*, Max–min–max subarrays of patterns in *N* dimension.

process is meant to roughly mimic the way in which activation spreads through the networks of neurons in the brain.

After activation has spread through a feedforward network, the resulting activation pattern on the output units encodes the networks 'answer' to the input (e.g. a classification of the input pattern as the letter M). In most applications, the network learns a correct mapping between input and output patterns via a learning algorithm. Typically the weights are initially set to small random values. Then a set of training inputs is presented sequentially to the network. In the back propagation learning procedure, after each input has propagated through the network weights are adjusted to reduce the difference between the networks output and the correct output. Each iteration is called a training cycle, and a complete pass of training cycles through the set of training inputs is called a 'training epoch'. This type of procedure is known as 'supervised learning', since a teacher supervises the learning by providing correct output values to guide the learning process. In contrast, in 'unsupervised learning' there is no teacher, and the learning system must learn on its own using less detailed environmental feedback on its performance.

The study of geometrical structure of an atom is essential, particularly in the branch of material science, solid state physics and electronics. It is very difficult to find a correct mathematical model for the crystal structure of an element in the periodic table. At present we know most of the parameters of elements up to AN 109. Discoveries of new elements are very slow owing to very small life time (few micro seconds) of higher atomic number elements. It has been hypothesized that elements up to atomic number 120 could exist (even if the life time is about pico second). Keeping this in mind we made an attempt to predict the crystal class of the elements up to atomic number 120, which in simulation is feasible and could yield meaningful results. Here we have used NN for two separate classification of crystals:

The first method involves the classification of the whole periodic table by training the network with data of elements from AN 1 to 84 into eight crystal classes, namely hexagonal close-packed (hcp), body-centered cubic (bcc), face-centered cubic (fcc), rhombus, diamond, cubic, complex, and hex (= hex + tetra).

In the second type, classification is based on the premise that 'the crystal structures of many materials of commercial importance are relatively simple and that they are either simple cubic (sc) or face-centered cubic (fcc) or body-centered cubic (bcc)', and for this type of classification reciprocal lattice technique[6] is employed.

An ideal crystal is composed of atoms arranged on a lattice defined by three fundamental translation vectors **a**, **b**, **c** such that an atom in its arrangement looks the same in every respect when viewed from any other point **r** as when viewed from a point given by

$$\mathbf{r}' = \mathbf{r} + u\mathbf{a} + v\mathbf{b} + w\mathbf{c}, \tag{3}$$

where u, v, and w are arbitrary integers. The set of points **r** specified by equation (3) for all values of the integers u, v, and w defines a lattice. A lattice is a mathematical abstraction: the crystal structure is formed only when a basis of atoms is attached identically to each lattice point, and is given by

lattice + basis = crystal structure.

Many elements occur in several crystal structures and undergo transformation from one to the other with varying temperatures and with pressure. Sometimes two structures coexist at the same temperature, although one may be stable. Sometimes the difference in the free energies of certain structures may be very small and thus it may not be possible to calculate these differences theoretically. For classification we have referred the compilation by Wyckoff[7].

For the computer implementation our main objective was to make the neural network classify a given input (crystal structure of an element in the periodic table) as follows: Neural network can predict and classify, and the classification is based on training. Thus effective training ensures that the NN output is accurate and acceptable. There is no general rule or particular reason for the choice of number of physical parameters at the input of a NN. When an element exists in nature, its physical, chemical, and electrical parameters are studied and determined. For the present work we were able to associate the following physical parameters with each element, i.e. atomic number, electro-negativity, Debye temperature, transition temperature (process-vaporization: liquid to gaseous), conversion energy, lattice parameters (a and c), density, atomic concentration of the element (in 10^{28} m^{-3}), nearest neighborhood, energy required to form separated neutral atoms from the solid at 0 K in terms of electron-volt per atom, kcal per mol, energy to remove one electron, energy to remove two electrons, bulk modulus, compressibility, conductivity, and resistance. (all in SI units)[8–13]. We extended this to relate to the unknown element from AN to 120. Even though compressibility-bulk modulus, conductivity-resistance are inverse of each other, the numerical values associated with each of them were different. NN showed sensitivity to numerical values, and was observed to be nonlinear in nature. Thus reciprocal parameters could exhibit in nature and showed nonrepeated responses in NN.

Training is done with data whose output is known. This training set is to be validated to ensure that the neural network has been subjected to the entire spectrum

of choice of inputs. This can be done if further inputs, whose outputs are known, are given at the input to check the effectiveness by training. The following steps give the computer implementation.

(i) Group the elements from atomic number (AN) 1 to 84 into eight classes (hexagonal close-packed (hcp), body-centered cubic (bcc), face-centered cubic (fcc), rhombus, diamond, cubic, complex, and hex (= hex + tetra))[6,7].
(ii) For each class (say hcp, which has twenty-four elements) calculate the average ($\bar{x}$) and standard deviation (σ).
(iii) Calculate $\bar{x} \pm \sigma/3$ for each (eighteen parameters) of them.
(iv) For each class (say hcp), each parameter values are normalized between 0 and 1. Step (ii) and (iii) are repeated for other seven classes.
(v) The above values are encoded as in max–min–max pattern with output features*.
(vi) This is used to train the neural network for 2000 iterations.
(vii) For validating the training, we fed the input datas from AN 86 to AN 95 as in step (v) without output features and we achieved 100% accuracy in classification. This trained and validated network was then used to classify data, the output for which was not known.
(viii) To predict the crystal structure of elements from AN 96 to 120, we took the overall average ($\bar{x}'$), and standard deviation (σ') of all the known elements and followed steps (ii), (iii), (iv) and (v) without any output at the input of the NN.
(ix) The process was repeated all over by first training the neural network for 5000 and then 10000 iterations, and then we studied the behaviour of the normalized system error (NSE) as a function of number of hidden layers (1 to 5), and number of units in a hidden layer (1 to 10). We got best accuracy for single hidden layer, and in that three nodes.

Experimentally we found that performance of the NN was highly dependent on training. Neural network was trained (AN 1–84) and validated (AN 86–95) from AN 1 to 95 without any error. Here we have avoided AN 9 and AN 85 as it has already been confirmed that crystal structure for these is non-determinable, even though these two elements exist physically[7]. Based on the validation, we predicted the crystal structure from AN 96 to 120. Figure 2 and Table 1 show the NN prediction and evaluation scores respectively with the following remarks.

(i) AN 109, 110, and 111 belong to the column class of noble (Cu, Ag, Au) and platinum (Pd and Pt) metals and Rh superconductor at low temperature[14]. Since all the known elements are fcc, the prediction of the elements with AN 109, 110, and 111 is compatible.
(ii) The last column elements (noble gases) are fcc. The neural network prediction of the same column AN 118 belong to fcc.

NN is extensively used for prediction. Even if we know the correct mathematical relationship for a pattern problem, we can test its validity using a number of tools. In the following section we have used NN for the classification of crystals using reciprocal lattice technique (X-ray diffraction), as we know most of the elements exist in cubic crystal structure.

Basically there are three different crystal structures, namely sc, fcc, and bcc. To find the above structure, reciprocal lattice technique is powerful. We have also classified the given crystal as sc or bcc or fcc. The data values are obtained from Scherrer method[6,7]. In this case, reciprocal lattice vectors are taken, based on the following analogy:

Every crystal structure has two lattices associated with it; the crystal lattice, and the reciprocal lattice. Any vector of the form

$$\mathbf{G} = h\mathbf{A} + k\mathbf{B} + l\mathbf{C} \tag{3}$$

(h, k, and l are integers or zeros) is called a reciprocal lattice vector, where

$$\begin{aligned} \mathbf{A} &= 2\Pi\ (\mathbf{b} \times \mathbf{c})/\mathbf{a}.\mathbf{b} \times \mathbf{c} \\ \mathbf{B} &= 2\Pi\ (\mathbf{c} \times \mathbf{a})/\mathbf{a}.\mathbf{b} \times \mathbf{c}. \\ \mathbf{C} &= 2\Pi\ (\mathbf{a} \times \mathbf{b})/\mathbf{a}.\mathbf{b} \times \mathbf{c} \end{aligned} \tag{4}$$

If **a, b**, and **c** are primitive lattice vectors of the crystal lattice, then **A**, **B**, and **C** are primitive vectors of the reciprocal lattice. The reciprocal lattice is a lattice in the associated Fourier space. Every position in Fourier space may have a meaning, but there is a special importance to the points defined by the set of Gs. Any function invariant under a lattice translation can be expanded in a Fourier series of the form

$$n(\mathbf{r}) = n \sum_{j=1}^{n} \exp(i\,.\,\mathbf{G}\,.\,\mathbf{r}). \tag{5}$$

In Debye Scherrer method

$$\frac{4a^2 \sin^2 \theta}{N} = \lambda^2 \quad \text{or} \quad \frac{\sin^2 \theta}{N} = \frac{\lambda^2}{4a^2},$$

where a/N is lattice constant λ the wavelength of the monochromatic X-ray used, and θ the diffraction angle from the respective reference line. For training, θ values have been taken from the standard reference manual and for prediction, θ values have been taken from the laboratory, and we are able to classify without any error for all classes. In this method we have calculated $N/\sin^2 \theta$.

*Available with the authors.

For sc, $N = 1, 2, 3, 4, 5, 6, 8, 9, \cdots$,

for fcc, $N = 3, 4, 8, 11, 12, 16, 19, \cdots$,

and for bcc, $N = 2, 4, 6, 8, 10, 12, 14, 15, \cdots$

Further, we took three crystal structures: sc, fcc, and bcc.

In the following section we have studied the dependency of the NN on different confidence intervals. The term 'confidence interval' has an intuitive meaning as well as a technical meaning. It is natural to expect it to mean 'an interval in which one may be confident that a parameter lies'. Therefore if the means of our target problems are plotted, they will form a frequency distribution which will be a close approximation to a normal probability curve: thus a plot of the standard error of the means (N) and sample size is shown in Figure 3 *a*. We have to select the confidence coefficient intervals as follows:

Let the means (μ s) and standard deviation or standard error (σ s) be taken from an approximately normal distribution statistic. We can expect to find an actual sample lying in the interval $\mu \pm \sigma$s, $\mu \pm 2\sigma$s, $\mu \pm 3\sigma$s about 68.27%, 95.45%, and 99.73% of the time respectively. Figure 3 *b* illustrates how the width of confidence interval varies with confidence coefficient for any given sample result.

Many confidence intervals can be discussed in terms of one-dimensional statistics, $T(X)$, which depends upon a vector of observations X. If $T(X)$ is a continuous random variable, given probabilities α_1 and α_2, it is possible to find $T_1(\theta)$ and $T_2(\theta)$ such that

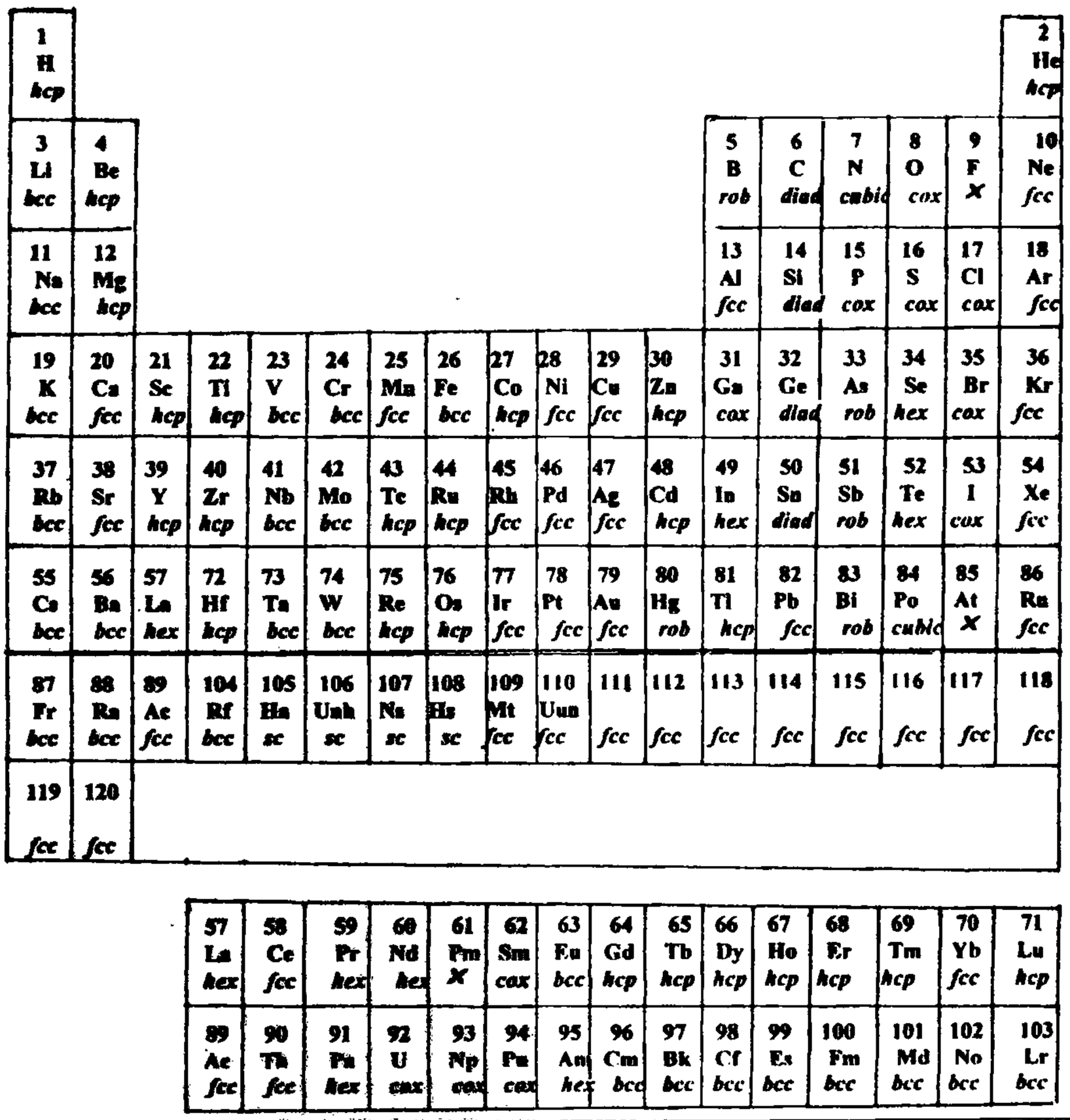

Figure 2. Neural network prediction of crystal class elements in a periodic table.

$$P_r[T(X) < T_1(\theta) \mid \theta_1] = \alpha_1, \quad (6)$$

and

$$P_r[T(X) > T_2(\theta) \mid \theta_2] = \alpha_2. \quad (7)$$

Confidence limits for θ-based statistic T are shown in Figure 3 *c*. For every particular value of θ the probability that T lies between $T_1(\theta) = T_2(\theta)$ is $1 - \alpha_1 - \alpha_2$. The basic idea of confidence intervals is to express confidence $1 - \alpha_1 - \alpha_2$ that the point (θ, T) lies in the confidence belt after T has been observed. Following are the results obtained on dependency of NN in a confidence interval and hidden layers.

(i) Normalized system error (NSE) decreases in all confidence intervals from $\bar{x} \pm \sigma/3$ to $\bar{x} \pm 3\sigma$, as shown in Figure 4 *a*.
(ii) At $\bar{x} \pm \sigma/2$ and $\bar{x} \pm \sigma$ confidence intervals NSE is minimum for 2000 iterations and the best fit line is given by $Y = 0.00435912 \ln(X) + 0.0136809$ (Figure 4 *b*).

Table 1. Known and neural network output of crystal class in a periodic table

Atomic number	Element	Crystal structure	
		KI	NNP
85	At	Uk	NC
86	Rn	fcc	fcc
87	Fr	bcc	bcc
88	Ra	bcc	bcc
89	Ac	fcc	fcc
90	Th	fcc	fcc
91	Pa	hex	hex
92	U	Complex	Complex
93	Np	Complex	Complex
94	Pu	Complex	Complex
95	Am	hex	hex
96	Cm	Uk	bcc
97	Bk	Uk	bcc
98	Cf	Uk	bcc
99	Es	Uk	bcc
100	Fm	Uk	bcc
101	Md	Uk	bcc
102	No	Uk	bcc
103	Lr	Uk	bcc
104	Rf	Uk	bcc
105	Ha	Uk	sc
106	Unh	Uk	sc
107	Ns	Uk	sc
108	Hs	Uk	sc
109	Mt	Uk	fcc
110	Uun	Uk	fcc
111		Uk	fcc
112		Uk	fcc
113		Uk	fcc
114		Uk	fcc
115		Uk	fcc
116		Uk	fcc
117		Uk	fcc
118		Uk	fcc
119		Uk	fcc
120		Uk	fcc

KI, Known information; Uk, unknown; NNP, neural network prediction; NC, not considered; bcc, body-centered cubic; fcc, face-centered cubic; sc, simple cubic.

(iii) As the number of units in a hidden layer is increased, from 1 to 10, the NSE decreases. The best fit line is given by $Y = -0.027989\,X + 0.8964$ (Figure 4 *c*).
(iv) As we increase the number of hidden layers, the NSE increases and its best fit line is: $Y = 0.105234\,X + 0.625$ (Figure 4 *d*).

Thus study of crystal structure and its class is based on modern algebra, particularly the group theory. We have attempted to classify crystals in a periodic table using a new tool: neural networks. We have shown the advancement and capability of neural networks in predicting the crystal class of high atomic number elements which are yet to be discovered, and perhaps might take decades before they are discovered. It would be interesting to extend this technique to solve other problems as well in physics or related fields.

In the present investigation we have taken eighteen physical parameters to determine the crystal structure

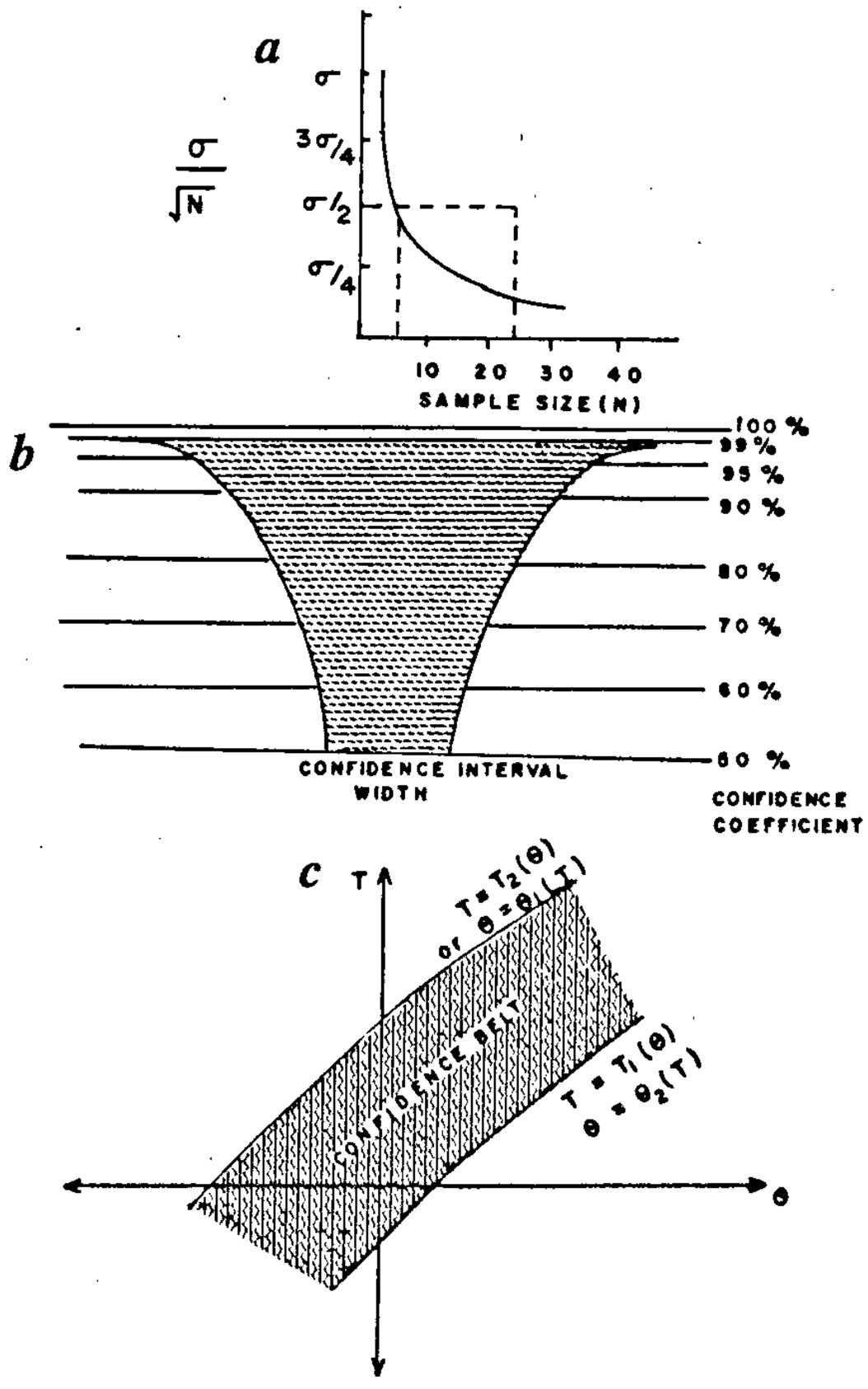

Figure 3. *a*, Relation between the standard error of the mean and sample size; *b*, Relation between confidence coefficient and confidence interval widths; *c*, Confidence limits for θ-based on the statistic T.

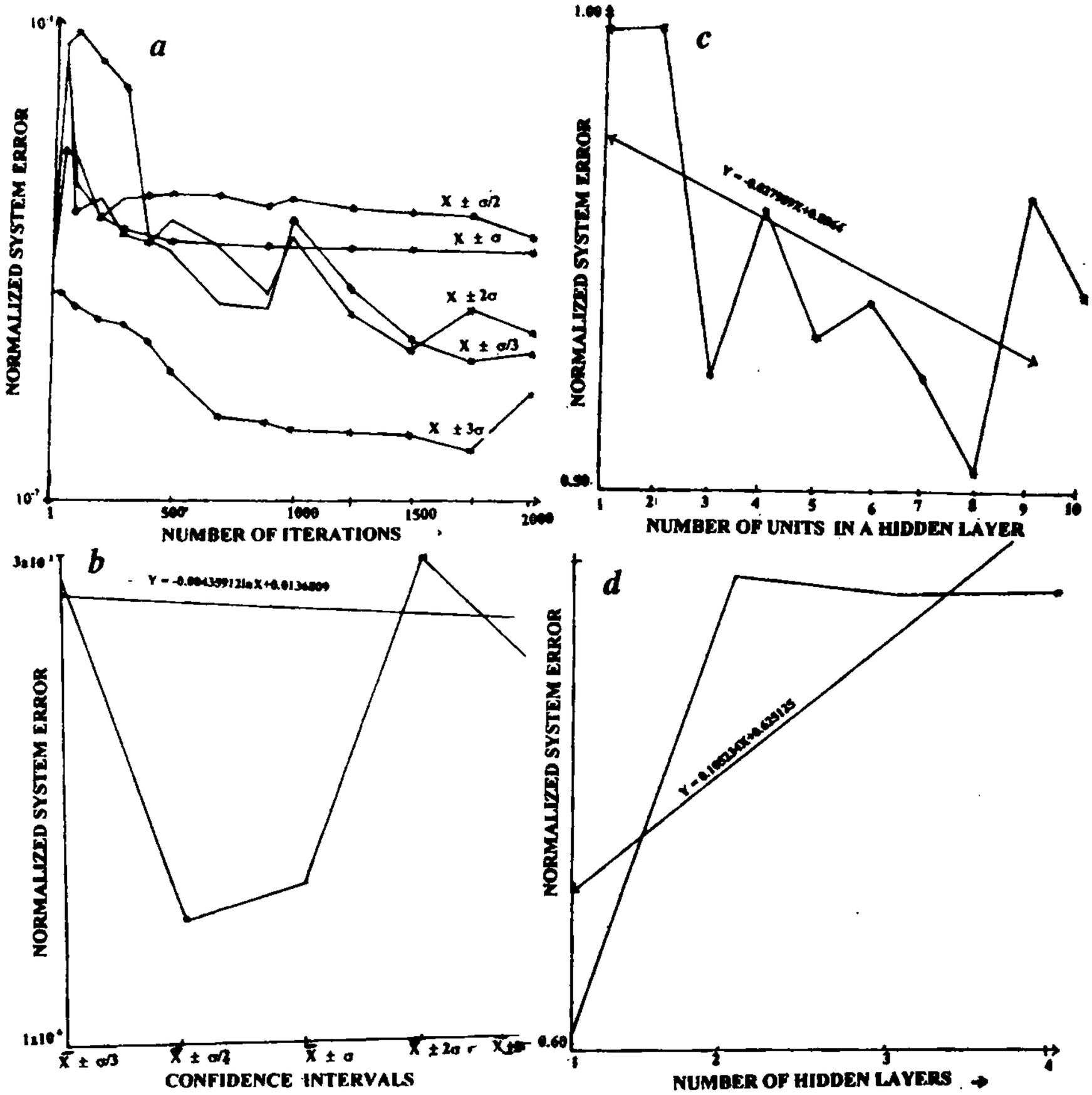

Figure 4. *a*, Normalized system error vs number of iterations at different confidence intervals; *b*, Normalized system error vs confidence intervals; *c*, Normalized system error vs units in a hidden layer; *d*, Normalized system error vs number of hidden layers.

using NN. To find the optimum number and the actual parameters sufficient for classification, there is a need to examine the performance of the network for different choices of the parameters included in the input vector. This question is open for future investigations.

1. Mueller, B., Reinhardt, J., *Neural Networks*, Springer–Verlag, Berlin, 1990.
2. Simon, Haykin, *Neural Networks*, McMillan College Publishing Company, New York, 1995.
3. Geoffrey, H. E., *Artificial Intelligence*, 1989, **40**, 185–234.
4. Jain, A. K. and Mao, J., *IEEE Computer*, 1996, March, 31.
5. Bernard, W., *Proc. IEEE*, 1990, **78**, 1415.
6. Kittel, C., in *Introduction to Solid State Physics*, Wiley Eastern Limited, New Delhi, 1984, fifth edition.
7. Table of Periodic Properties of Elements, Sargent-Welch-Scientific Company, Illinois 60077, Catalog No. S-18806, 1980.
8. Allred-Rochow, *J. Inorg-Nuclear Chem.*, 1958, **5**, 264.
9. Waggman, D. D., Jobe, T. L., Pomalski, E. S. and Schumn, R. H., *The National Bureau of Standards*, NY, 1978.
10. Gschneidner, K. J., *Solid State Phys.*, 1964, **16**, 275.
11. Birch, F., *Hand Book of Physical Constants*, Geological Society of America Memoir, 1966, **97**, 107.
12. National Bureau of Standards Circular, 1978, 467.
13. Pearlman, N., AIP Handbook, third edition, 1978.
14. Frank, P., *Phys. Today*, 1993, **34**.

Received 25 June 1998; revised accepted 28 December 1998