DESIGNING HOPFIELD TYPE NETWORKS USING GENETIC ALGORITHMS AND ITS COMPARISON WITH SIMULATED ANNEALING

SANKAR K. PAL*, SUSMITA DE and ASHISH GHOSH
Machine Intelligence Unit, Indian Statistical Institute
203 B. T. Road, Calcutta 700 035, India

An application of Genetic Algorithms (GAs) to evolve Hopfield type optimum neural network architectures for object extraction problem is demonstrated. Different optimizing functions involving minimization of energy value of the network, maximization of percentage of correct classification of pixels (*pcc*), minimization of number of connections of the network (*noc*), and a combination of *pcc* and *noc* have been considered. The *noc* value of the evolved (sub)optimal architectures is seen to be reduced to two-third of that required for the fully connected version. The performance of GA is seen to be better than that of Simulated Annealing for this problem.

Keywords: Genetic algorithms, object extraction, Hopfield type networks, simulated annealing.

1. INTRODUCTION

Genetic Algorithms (GAs)[1-3] are adaptive computational procedures modeled on the mechanics of natural genetic systems. They express their ability by efficiently exploiting the historical information to speculate on new offspring with expected improved performance.[1] GAs are executed iteratively on a set of coded solutions, called a *population*, with three basic operators: *selection/reproduction, crossover,* and *mutation*.[1,2] They use only the payoff (objective function) information and probabilistic transition rules. GAs are theoretically and empirically proven to provide robust search in complex spaces.

Since GA works simultaneously on a set of coded solutions, it has very little chance to get stuck at local optima. This feature also helps to introduce a large amount of implicit parallelism in the computational procedure. Again, GAs do not need any sort of auxiliary information, like the derivative of the optimizing function. The resolution of the possible search space can be controlled by varying the coding length of the parameters.

Neural networks (NNs) are designated by the network topology, connection strengths between pairs of neurons/nodes, node characteristics, and rules for updating status. In spite of the wide range of applicability of NNs, there is no formal procedure to design an optimum neural network for a given problem. Recently, some attempts are made in this regard using GAs.[4-9] These investigations are mainly concerned with multi-layer networks; no attempt is made (as to the knowledge of the authors) on Hopfield type models.[10]

*Corresponding author; e-mail: sankar@isical.ernet.in

International Journal of Pattern Recognition and Artificial Intelligence Vol. 11 No. 3 (1997) 447–461

In this article we describe a method where the searching capability of GAs is exploited to evolve Hopfield type optimum neural network architectures, particularly for the problem of object extraction. The use of Hopfield type neural network for object background classification is described in Ref. 11 where a neuron is assigned corresponding to each pixel of the image. The connectivity of each neuron with its neighbors was considered to be fixed and full. Note that, the connectivity of a neuron allows the neighborhood information to be used for deciding the class of the corresponding pixel as object or background. So if most of the neighbors belong to object class then the possibility of that pixel being classified as object will increase. In case, some of the neighbors are corrupted by noise and turned into background class, then full connectivity (which considers maximum neighborhood information) may result in misclassification of the said pixel. It is therefore necessary, particularly in a very noisy environment, to deal with variable connected networks for improving the quality of the segmented output. This will also enable one to have less connectivity (i.e., less expensive networks) as compared to the fully connected one, besides the improved performance.

The present investigation attempts to evolve such a variable connected network using GAs, and it consists of three parts. At first, networks are evolved for providing optimum segmented output irrespective of the number of connections in the network. Here the output performance is evaluated in terms of both energy value of the (converged) network and the percentage of correct classification of pixels. In the second part, we have obtained architectures for providing maximum number of correct classification of pixels with less number of connections. The final part consists of determining the architecture with minimum number of connections for a desired output. The merit of using the variable connectivity over the corresponding fixed full connectivity is established for a set of images. The searching capability of GA is also compared extensively with that of the Simulated Annealing (SA) for the said problem.

2. EVOLUTION OF HOPFIELD TYPE NEURAL NETWORK ARCHITECTURES FOR OBJECT EXTRACTION

In this section we first of all describe the principle of object extraction using Hopfield type neural networks as described in Ref. 11. This is followed by evolutionary schemes based on GA and SA for designing such networks.

2.1. Principle of Object Extraction using Hopfield type Neural Network

To use a Hopfield type neural network for object background classification,[11] a neuron is assigned corresponding to every pixel. Each neuron can be connected to its neighbors (over a window) only. The connection can be full (a neuron is connected with all of its neighbors) or can be partial (a neuron may not be connected with all of its neighbors). The network topology for a fully connected third order neighborhood is depicted in Fig. 1. Here the maximum number of connections of a neuron with

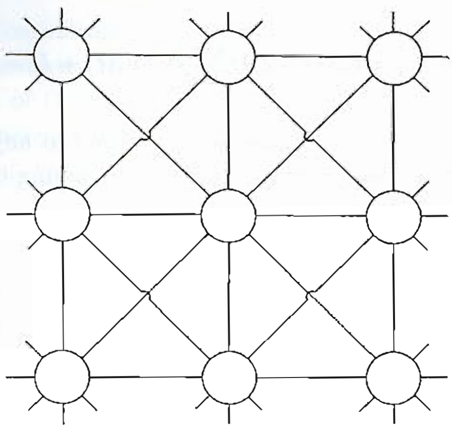


Fig. 1. Topology of the neural network with third order connectivity (in the proposed system all connections may not exist).

its neighbors is 8. In practice, all these connections may not exist. Again, different neurons may have different connectivity configuration within its neighbors.

The energy function of this model has two parts. The first part is due to the local field or local feedback and the second part corresponds to the input bias of the neurons. In terms of images, the first part can be viewed as the impact of the gray levels of the neighboring pixels, whereas the second part can be attributed to the gray value of the pixel under consideration. The total energy contributed by all pixel pairs will be $-\sum_i \sum_j W_{ij} V_i V_j$, where V_i, V_j are the status of ith and jth neurons, respectively and W_{ij} is the connection strength between these two neurons.

For every neuron i, the initial input bias I_i and the initial state V_i are taken to be proportional to the actual gray level of the corresponding pixel. If a gray level of a pixel is high (low), the corresponding intensity value of the scene is expected to be high (low). The input bias value is taken in the range $[-1, 1]$. Under this framework an ON (1) neuron corresponds to an object pixel and the OFF (-1) one as background. So the threshold between object and background can be taken as 0. Thus the amount of energy contributed by the input bias values is $-\sum_i I_i V_i$. So the expression of energy for the object extraction problem takes the form

$$E = -\sum_i \sum_j W_{ij} V_i V_j - \sum_i I_i V_i. \tag{1}$$

Stable states of the network (the local minima of its energy function) are assumed to correspond to the partitioning of a scene into compact regions. So from a given initial state, the status of a neuron is modified iteratively to attain a stable state.

It is to be noted that in order to get the input to all neurons of the network (of size $N_1 \times N_2$) at an instant $(t + \Delta t)$ one has to solve $N_1 \times N_2$ differential equations with given initial values at time t. For this, the Euler method is used here i.e., we iterated

$$U_i(t+\Delta t) = U_i(t) + \Delta t(\sum_j W_{ij} V_j(t) + I_i - U_i(t)) \tag{2}$$

until convergence; $U_i(t)$ is the total input to a neuron i at any instant t. Numerical solutions for these differential equations require a stopping criterion which can be taken as

$$\mid U_i(t+\Delta t) - U_i(t) \mid < \varepsilon, \forall i \tag{3}$$

where ε is a preassigned small positive quantity. The network is assumed to attain a stable state if for every neuron i, $\mid V_i(t) - V_i(t+\Delta t) \mid < \varepsilon^1$, where ε^1 is another preassigned small positive quantity.

2.2. (Sub)optimal Architecture Evolution using GAs

A chromosome of the GA represents a network architecture. For an $m \times n$ image, each pixel (neuron) being connected to at most k of its neighbors, the length of the chromosome is $m \times n \times k$. If a neuron is connected to any of its neighbors, the corresponding bit of the chromosome is set to 1, else 0. The initial population is generated randomly. Each network is then allowed to run for object extraction as described in Sec. 2.1. The energy value (Eq. (1)) obtained at the converged state of a network is taken as the index of fitness of the corresponding chromosome (minimum energy corresponds to maximum fitness) for its selection for the next generation. Crossover and mutation operations are performed on these selected chromosomes to get new offspring i.e., to generate new architectures. The whole process is continued for a number of generations until the GA converges. The best chromosome of the final population represents a (sub)optimum architecture (with respect to energy value) for object extraction.

Further, if the desired output images (i.e., the target values of pixels) are known, we can measure the percentage of correct classification of pixels (*pcc*) of the converged (evolved) network and use it as a fitness value for selecting networks. *pcc* is defined as,[11]

$$pcc = (tcc \times 100)/(m \times n), \tag{4}$$

where, *tcc* is the total number of pixels correctly classified for an $m \times n$ image. The higher the *pcc* value, the better is the chromosome.

In the first part of our investigation, we have considered energy and *pcc* based fitness measures to evolve networks for providing optimum segmented output. In the second part, to obtain architectures for providing maximum number of correct classification of pixels with minimum number of connections of the network, the fitness value is taken as

$$fitness = tcc + (max\_noc - noc)/100, \tag{5}$$

where, *max_noc* is the maximum number of connections i.e., the number of connections of a fully connected network. (Note that, one could have used some other combinations of *tcc* and *noc*.) Finally, to obtain a desired output quality with minimum number of connections, the *noc* value is taken as the fitness value of the network so that the minimum value corresponds to maximum fitness.

For the simulation of GA, following steps are adopted. The population size is kept fixed at 30. Generational replacement technique[2] is used. Both the elitist model (i.e., by copying the best member of each generation into the next one, replacing the lowest fitted string)[2] and the standard GA (SGA) i.e., without elitism, are implemented.

Linear normalization selection procedure[2] (which works better in a close competitive environment) is adopted. The difference between successive fitness values is taken as 1 and the minimum fitness value is kept to 1. The number of copies produced by the ith individual (chromosome) with normalized fitness value f_i in a population of size n is taken as round(c_i); where $c_i = n \times f_i / \sum_{i=1}^{n} f_i$.

The experiment is carried out with three different sets of crossover and mutation probabilities e.g. (p_c and p_m) = (0.8, 0.02), (0.9, 0.02), and (0.8, 0.05). Mutation operation has been kept embedded within the crossover operation. We have run the algorithm for 200 iterations (it is seen that the average fitness value of the population is not changing much after this).

2.3. (Sub)optimal Architecture Evolution using SA

In a part of the experiment we compared the results obtained by GA with that of simulated annealing (SA). The implementational aspect of simulated annealing algorithm is described here, in brief.

Here each state of the system is referred to a chromosome. The meaning and representation of a chromosome are identical to that of the GA based methods. The initial chromosome is chosen randomly. A new chromosome (neighbor) is generated from the present one by mutation. The searching of the neighbor of a chromosome is guided by Metropolis algorithm.[12] The algorithm has been changed accordingly for maximization problems.

The parameters of the simulated annealing algorithm are selected so as to maintain parity with the proposed GA based methods as much as possible. The investigation is done with three different values of initial Temperature, T. They are 100, 50, and 10. The temperature decay rate is set to 0.95. Since GA is being executed 200 times for each run, we have also fixed the minimum T to 0.0036, 0.0018, and 0.00036, respectively, so that SA can also execute with 200 different T values. Further, since the population size of the GA is taken as 30, the simulated annealing algorithm is also executed 30 times for each value of T. This ensures that the same number of chromosomes is being searched using both GA and SA based methods.

3. ANALYSIS OF RESULTS

The effectiveness of the proposed technique has been demonstrated using some synthetic images (Figs. 3(a), 4(a), 5(a)) and two real images of 'BIPLANE' (Fig. 6(a)) and 'Blurred Chromosome' (Fig. 7(a)). The synthetic input images are generated by adding N(0, $\sigma_i{}^2$) noise to each pixel of the binary (two-tone) image shown in Fig. 2. The size of each image is 64×64.

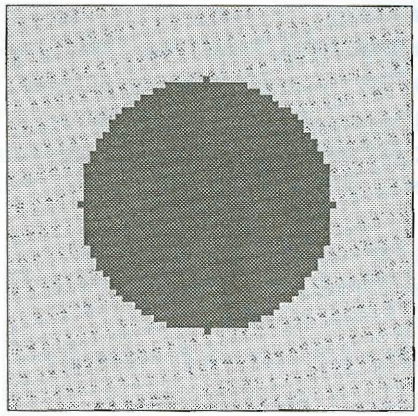
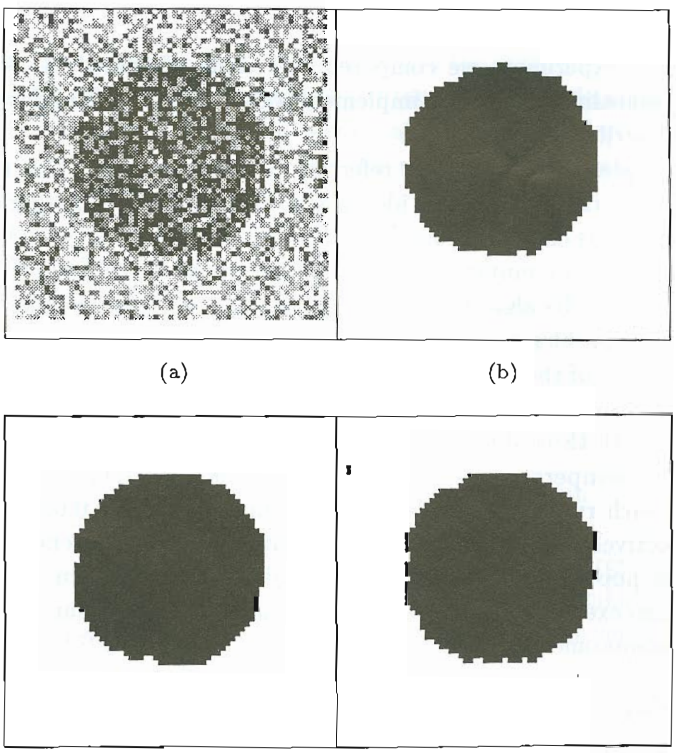


Fig. 2. Original synthetic image.



(a) (b)

(c) (d)

Fig. 3. Results for a noisy version ($\sigma = 15$) of the synthetic image. (a) Input. (b) Output using SGA with energy as fitness measure. (c) Output using fully connected architecture. (d) Output using SA.

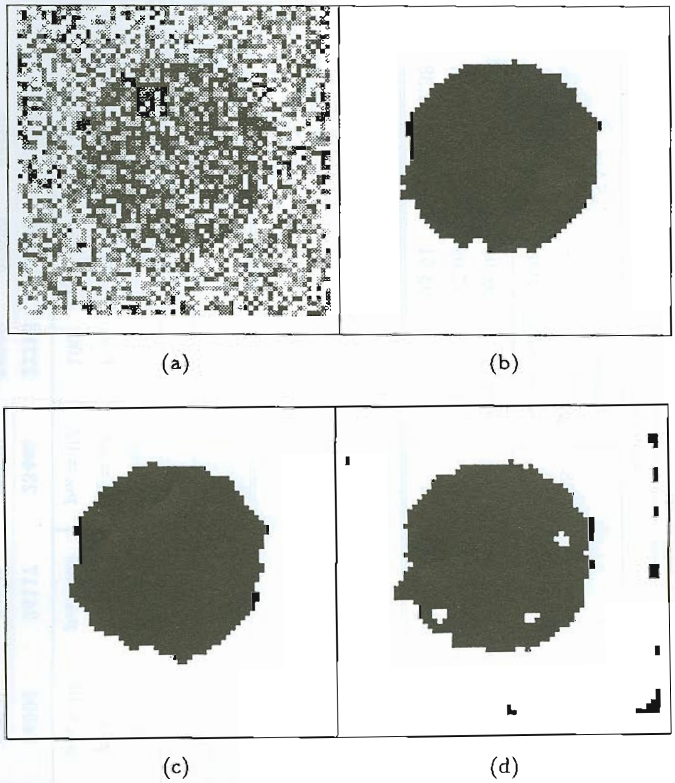


Fig. 4. Results for a noisy version ($\sigma = 24$) of the synthetic image. (a) Input. (b) Output using SGA with energy as fitness measure. (c) Output using fully connected architecture. (d) Output using SA.

The percentage of correct classification of pixels of the best chromosome in the last generation using *pcc* and energy based fitness measures for different noisy versions (Figs. 3(a), 4(a), 5(a)) of the synthetic image is depicted in Table 1. The number of connections of these evolved architectures is put in Table 2. Note that the *noc* of the corresponding fully connected architecture is $64 \times 64 \times 8 = 32{,}768$; which is almost one and a half times larger than that of the proposed GA based methods.

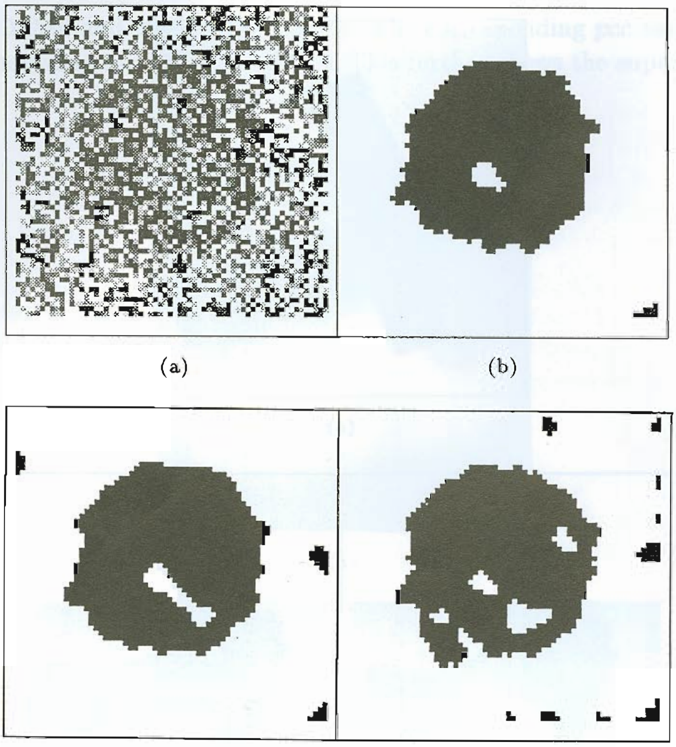
It is seen from Table 1 that the *pcc* based fitness measure performs better than the corresponding energy based one (except the case of SA with $T = 100$ for $\sigma = 32$). This is due to the fact that the former one considers information about the actual image (original two-tone version) for the evaluation of the fitness of chromosome, whereas the latter one does not. Moreover, standard GAs mostly yielded superior output compared to the corresponding elitist version. This may be due to the following reason: the adopted selection procedure produces more copies of the best chromosome; elitism adds one more copy of the same, and thus accelerates premature convergence. From the table it is also found that the GA based methods are better than the SA based ones (except the case of elitist model with energy based fitness measure, and $p_c = 0.08$, $p_m = 0.02$ for the image with $\sigma = 15$). In order

Table 1. Percentage of correct classification of pixels for the noisy (synthetic) images.

Image with	fixed fully connected network as used in Ref. 11	Percentage of correct classification of pixels (*pcc*) with fitness measure based on									
		pcc			energy						
		SGA	elitism	SA	SGA			elitism	SA		
		$p_c = .08$ $p_m = .02$	$p_c = .08$ $p_m = .02$	T = 100	$p_c = .08$ $p_m = .02$	$p_c = .09$ $p_m = .02$	$p_c = .08$ $p_m = .05$	$p_c = .08$ $p_m = .02$	T = 100	T = 50	T = 10
$\sigma = 15$	99.32	99.78	99.78	98.95	99.07	98.97	99.12	98.71	98.95	98.75	98.97
$\sigma = 24$	98.46	99.41	99.00	97.46	98.75	98.27	98.63	98.27	97.09	97.07	97.44
$\sigma = 32$	95.63	98.24	97.90	92.75	96.39	96.48	96.44	96.17	93.51	93.02	92.38

Table 2. Total number of connections.

Image with	Total number of connections (*noc*) with fitness measure based on									
	pcc			energy						
	SGA	elitism	SA	SGA			elitism	SA		
	p_c=.08 p_m=.02	p_c=.08 p_m=.02	T = 100	p_c=.08 p_m=.02	p_c=.09 p_m=.02	p_c=.08 p_m=.05	p_c=.08 p_m=.02	T = 100	T = 50	T = 10
$\sigma = 15$	21866	21934	20837	24138	24096	24117	23443	22758	23035	23648
$\sigma = 24$	21965	21817	20732	23603	23621	23605	22810	22808	23079	23508
$\sigma = 32$	21860	21927	20839	23243	23285	23221	22491	22748	22993	23457



(a) (b)

(c) (d)

Fig. 5. Results for a noisy version ($\sigma = 32$) of the synthetic image. (a) Input. (b) Output using SGA with energy as fitness measure. (c) Output using fully connected architecture. (d) Output using SA.

to demonstrate the effect of different parameter values, we have included, as an illustration, the results corresponding to three sets of p_c, p_m and T values (using energy based evaluation) for SGA and SA.

As a typical illustration, output images obtained by SGA (with energy based evaluation and $p_c = 0.08$, $p_m = 0.02$) are presented in Figs. 3(b), 4(b), and 5(b). The corresponding outputs obtained by the fully connected architectures (as employed in Ref. 11) and SA based technique (with $T = 100$) are shown in Figs. 3(c), 4(c), 5(c) and 3(d), 4(d), 5(d), respectively for comparison. It is seen from these segmented outputs that for highly corrupted images, GA based technique performs significantly better than those using fully connected architecture[11] and simulated annealing.

Results corresponding to maximization of Eq. (5) for obtaining architectures in order to provide maximum number of correct classification of pixels with minimum number of connections, are put in Table 3. The terms inside the brackets associated with the fitness values represent the corresponding *tcc* and *noc* values. This table also shows that the fitness values for the GA based method are much higher than those of SA.

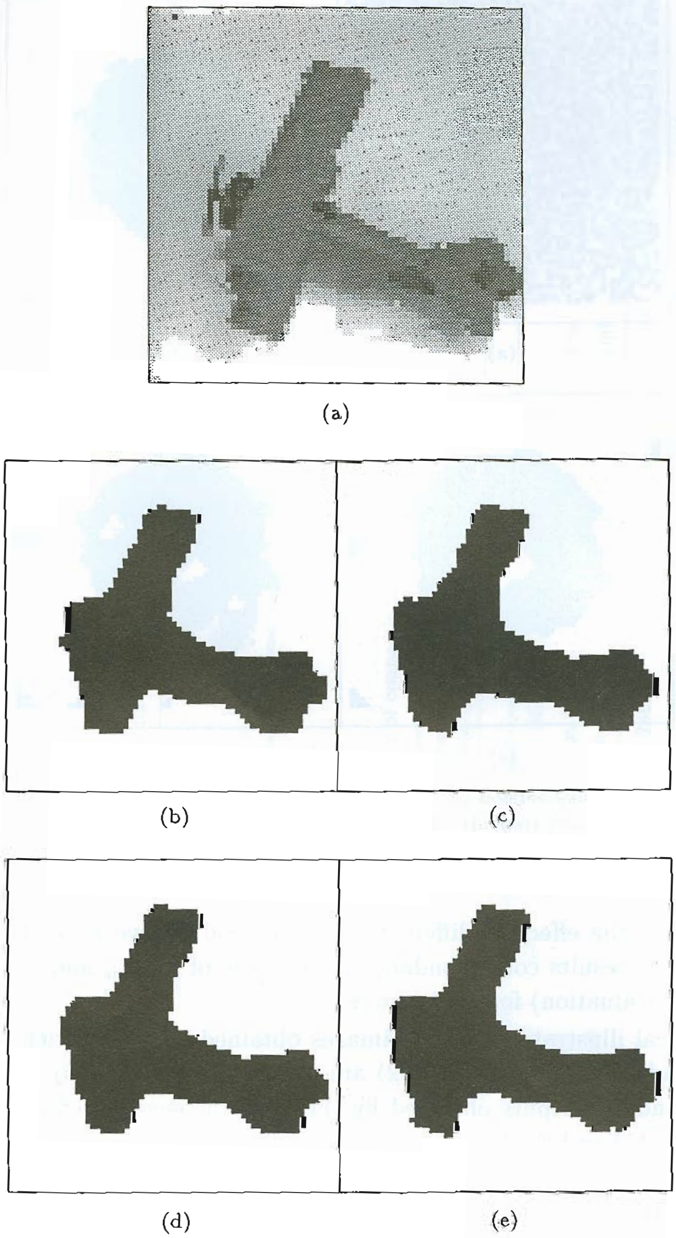


Fig. 6. Results for 'BIPLANE' image. (a) Input. (b) Output using SGA with energy as fitness measure. (c) Output using elitism with energy as fitness measure. (d) Output using fully connected architecture. (e) Output using SA.

Finally, for a given *pcc* value we have obtained architectures by minimizing number of connections. Here we have considered only the noisy image of Fig. 5(a) as input. The *pcc* value is prefixed to 88.0. It is found that the *noc* values of the evolved networks using SGA ($p_c = 0.08$, $p_m = 0.02$) and SA ($T = 100$) based method are 19586 and 20469, respectively. The corresponding *pcc* values obtained from these networks are 88.94 and 88.09. This further shows the superiority of GA over SA.

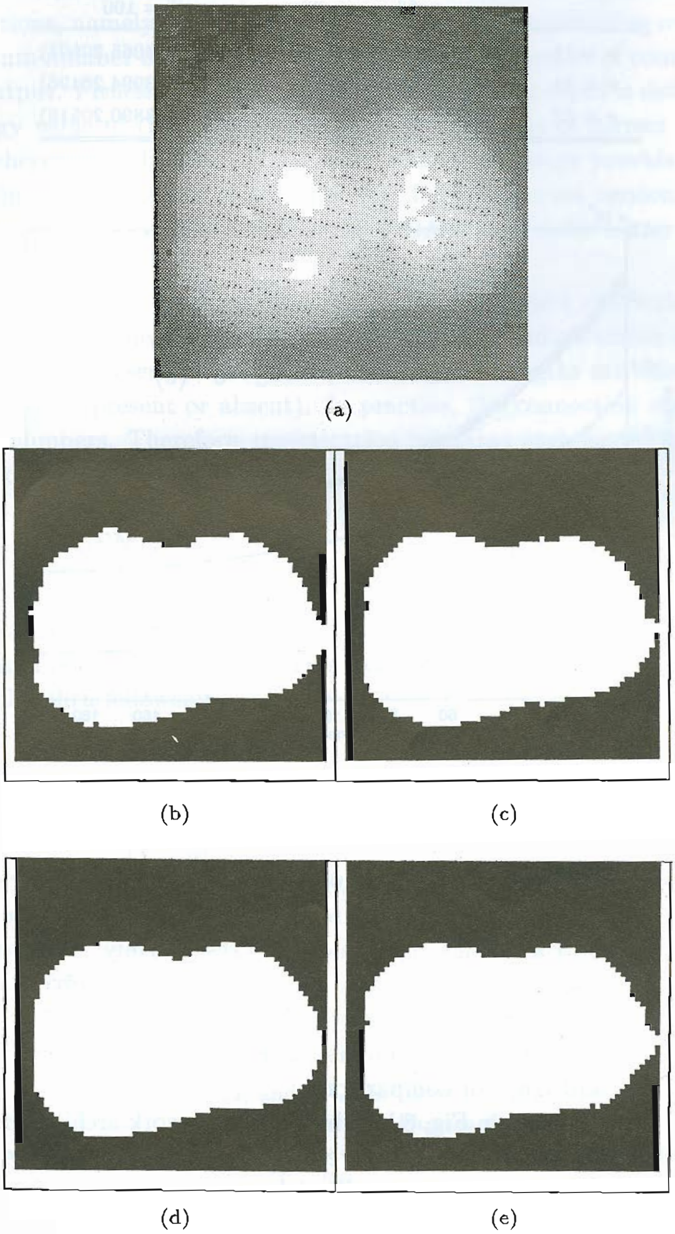


Fig. 7. Results for 'Blurred Chromosome' image. (a) Input. (b) Output using SGA with energy as fitness measure. (c) Output using elitism with energy as fitness measure. (d) Output using fully connected architecture. (e) Output using SA.

Table 3. Results corresponding to maximization of Eq. (5) for the noisy (synthetic) images.

Image with	Fitness (tcc,noc) values using	
	SGA	SA
	$p_c = .08, p_m = .02$	$T = 100$
$\sigma = 15$	4211.74 (4091,20694)	4187.67 (4065,20501)
$\sigma = 24$	4182.07 (4065,21061)	4116.72 (3994,20496)
$\sigma = 32$	4160.19 (4046,21349)	4012.50 (3890,20518)

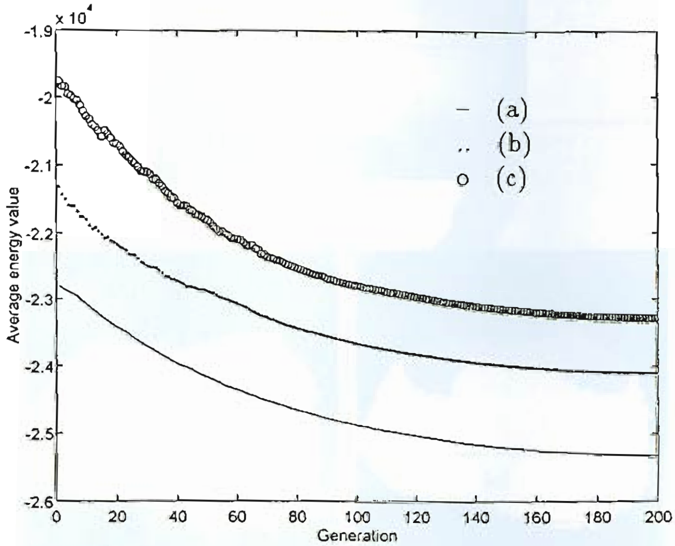


Fig. 8. Variation of average fitness value using SGA (energy based evaluation): (a) for noisy image with $\sigma = 15$, (b) for noisy image with $\sigma = 24$ and (c) for noisy image with $\sigma = 32$.

Objects extracted using SGA ($p_c = 0.08$, $p_m = 0.02$) and elitist strategy for the 'BIPLANE' (Fig. 6(a)) and 'Blurred Chromosome' (Fig. 7(a)) images are shown in Figs. 6(b) and 6(c) and Figs. 7(b) and 7(c), respectively. Here energy value is considered as the fitness measure, since the target values of pixels are unknown (unlike the synthetic images). The corresponding outputs obtained by the fixed fully connected version and SA based technique (with $T = 100$) are shown in Figs. 6(d) and 6(e) and 7(d) and 7(e), for comparison.

Finally, we demonstrate in Fig. 8 how better the network architectures are gradually evoluted with generation. Here we consider, as an example, the variation of average energy value (when SGA is used) with generation for the input images of Figs. 3(a), 4(a), and 5(a). Energy values are seen to decrease gradually with generation, attaining a stable state (representing optimum architecture) around 200th generation.

The experiment is performed on a Sun SPARC Classic Workstation (frequency 59MHz). To execute each generation, CPU time requires 2 minutes (approximately).

4. CONCLUSIONS AND DISCUSSION

The effectiveness of GAs and SA for evolving Hopfield type optimum neural network architectures for object extraction has been demonstrated. The proposed techniques are able to find out automatically the network architectures subject to different objective functions, namely, optimizing segmented output, maximizing output quality with minimum number of connections, and minimizing number of connections for a desirable output. Fitness function to optimize segmented output is defined in terms of the energy value of the network and/or the percentage of correct classification of pixels, wherever applicable. The evolved networks always provide less number of connections compared to the corresponding fully connected version. For the parameters considered here, the performance of GA is seen to be better than that of SA.

Although we have considered here the problem of object extraction, other application areas of Hopfield type networks can also be handled within the proposed framework. In the present investigation, connection strengths are taken as 1 and 0 (i.e., connection is present or absent). In practice, the connection strengths could also be real numbers. Therefore, investigation involving such networks may constitute a part of future work.

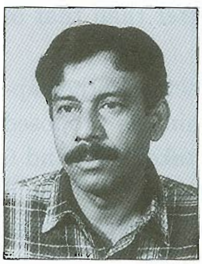
ACKNOWLEDGMENTS

The authors gratefully acknowledge Dr. C. A. Murthy for his valuable discussion. Ms. Susmita De is grateful to the Department of Atomic Energy, Govt. of India, for providing a research fellowship. This work was done when Prof. S. K. Pal held a Jawaharlal Nehru fellowship.

REFERENCES

1. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Massachusetts: Addison-Wesley, 1989.
2. L. Davis (ed.), *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold, 1991.
3. Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Berlin: Springer-Verlag, 1992.
4. D. Whitley, T. Starkweather, and C. Bogart, "Genetic algorithms and neural networks: optimizing connections and connectivity," *Parallel Computing* **14** (1990) 347–361.
5. J. D. Schaffer, R. A. Caruana, and L. J. Eshelman, "Using genetic search to exploit the emergent behavior of neural networks," *Physica D* **42** (1990) 244–248.
6. S. Saha and J. P. Christensen, "Genetic design of sparse feedforward neural networks," *Information Sciences* **79** (1994) 191–200.
7. S. Harp and T. Samad, "Genetic synthesis of neural network architectures," in *Handbook of Genetic Algorithms*, ed. L. Davis, Van Nostrand Reinhold, 1991.
8. V. Maniezzo, "Genetic evolution of the topology and weight distribution of neural networks," *IEEE Transactions on Neural Networks* **5**, 1 (1994) 39–53.
9. S. K. Pal and D. Bhandari, "Selection of optimal set of weights in a layered network using genetic algorithms," *Information Sciences* **80** (1994) 213–234.

10. J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two state neurons," in *Proc. National Academy of Science*, (USA), 1984, pp. 3088–3092.
11. A. Ghosh, N. R. Pal, and S. K. Pal, "Object background classification using Hopfield type neural network," *Int. J. of Pattern Recognition and Artificial Intelligence* **6**, 5 (1992) 989–1008.
12. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science* **220**, 4598 (1983) 671–680.



Sankar K. Pal is a Professor and Founding Head of Machine Intelligence Unit at the Indian Statistical Institute, Calcutta. He obtained M. Tech. and Ph.D. in radiophysics and electronics in 1974 and 1979 respectively, from the University of Calcutta, India. In 1982 he received another Ph.D. in electrical engineering along with DIC from Imperial College, University of London. In 1986 he was awarded *Fulbright Post-doctoral Visiting Fellowship* to work at the University of California, Berkeley and the University of Maryland, College Park, USA. In 1989 he received an *NRC-NASA Research Award* to work at the NASA Johnson Space Center, Houston, Texas, USA.

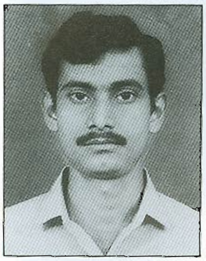
He received the *1990 Shanti Swarup Bhatnagar Prize* in Engineering Sciences, 1993 *Jawaharlal Nehru Fellowship*, 1993 Vikram Sarabhai Research Award, 1993 NASA Tech Brief Award, 1994 IEEE Transaction on Neural Networks outstanding paper, Award and 1995 NASA Patent Application Award.

His research interests mainly include pattern recognition, image processing, neural nets, genetic algorithms, and fuzzy sets and systems. He is a co-author of the book *Fuzzy Mathematical Approach to Pattern Recognition*, John Wiley & Sons (Halsted), N.Y., 1986 and a co-editor of two books *Fuzzy Models for Pattern Recognition*, IEEE Press, N.Y., 1992 and *Genetic Algorithms for Pattern Recognition*, CRC Press, Boca Raton, 1996.

Prof. Pal is a *Fellow* of the IEEE, USA, Indian National Science Academy, Indian Academy of Sciences, National Academy of Sciences, India, Indian National Academy of Engineering, Institute of Engineers, India, and the IETE, a Member, *Executive Advisory Editorial Board*, IEEE Trans. Fuzzy Systems and International Journal of Approximate Reasoning, North Holland, and an *Associate Editor*, IEEE Trans. Neural Networks, Pattern Recognition Letters, Neurocomputing, Applied Intelligence, Information Sciences: Applications, Int. J. Knowledge Based Intell. Engg. Syst., and Far-East Journal of Mathematical Sciences.

Susmita De received the B.Sc. degree with Honors in physics and B. Tech. degree in computer science and engineering from the University of Calcutta, India, in 1988 and 1991, respectively. She obtained the M.Tech. degree in computer science and engineering from the Indian Institute of Technology, Bombay, India in 1993. She worked as a Dr. K. S. Krishnan Senior Research Fellow of the Dept. of Atomic Energy, Govt. of India, from 1994 to February, 1997 in the Indian Statistical Institute, Calcutta towards the Ph.D. degree.

At present, Ms. De is a Lecturer at the Dept. of Computer Science and Engineering, Jadavpur University, Calcutta, India. Her research interests include genetic algorithms, neural networks, image processing, pattern recognition, and soft computing.



Ashish Ghosh is a Lecturer of the Machine Intelligence Unit at the Indian Statistical Institute, Calcutta. He received the B.E. degree in electronics and telecommunication from the Jadavpur University, Calcutta in 1987, and the M.Tech. and Ph.D. degrees in computer science from the Indian Statistical Institute, Calcutta in 1989 and 1993, respectively. He received the prestigious and most coveted *Young Scientists* award in engineering sciences from the Indian National Science Academy in 1995; and in computer science from the Indian Science Congress Association in 1992. His name has been cited in *International Directory of Distinguished Leadership* of American Biographical Institute Inc. in 1993; *International Who's Who of Contemporary Achievement* in 1994; and in *World's Who's Who of Men and Women of Distinction* in 1994. He visited the Osaka Prefecture University, Japan with a Post-doctoral fellowship during October 1995 to March 1997.

His research interests include evolutionary computation, neural networks, image processing, fuzzy sets and systems, and pattern recognition.