Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neurocomputing 51 (2003) 237–247

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Pattern recognition using multilayer neural-genetic algorithm

Yas Abbas Alsultanny*, Musbah M. Aqel

Computer Science Department, College of Computer and Information Technology, Applied Science University, P.O. Box 17, Amman 11931, Jordan

Received 2 July 2001; accepted 8 May 2002

Abstract

The genetic algorithm implemented with neural network to determine automatically the suitable network architecture and the set of parameters from a restricted region of space. The multilayer neural-genetic algorithm was applied in image processing for pattern recognition, and to determine the object orientation. The library to cover the views of object was build from real images of (10×10) pixels. Which is the smallest image size can be used in this algorithm to recognize the type of aircraft with its direction

The multilayer perecptron neural network integrated with the genetic algorithm, the result showed good optimization, by reducing the number of hidden nodes required to train the neural network (the number of epoch's reduced to less than 50%). One of the important results of the implemented algorithm is the reduction in the time required to train the neural network.

Keywords: Neural network; Genetic algorithm; Image processing; Pattern recognition

1. Introduction

Computer imaging separated into two fields:

- *Computer vision.*
- *Image processing.*

Computer vision is computer imaging where the application does not involve a human being in the visual loop. Image analysis involves the examination of the image

* Corresponding author. Fax: 5232899.
E-mail address: alsultanny@hotmail.com (Y.A. Alsultanny).

PII: S0925-2312(02)00619-7

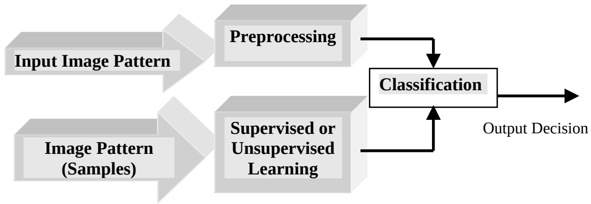


Fig. 1. Block diagram of decision-theoretical pattern recognition system.

data to facilitate solving a vision problem. The image analysis process involves two other topics: *feature extraction* is the process of acquiring higher-level image information, such as shape or color information, and pattern classification is the act of taking this higher-level information identifying objects within image. The block diagram of decision-theoretical pattern recognition shown in Fig. 1 represented by three primary stages [11,6]:

- *Preprocessing*
- *Learning*
- *Classification*

Preprocessing is used to remove noise and eliminate irrelevant, visually unnecessary information. *Learning* stage involves either supervised or unsupervised learning. *Classification* stage classified the pattern, which is preprocessed in the first stage, by computing the output and considering the maximum output is the more similar image in database library to the unknown input pattern image.

2. Multilayer perecptron neural network (MLP)

The three layers network with sigmoid activation function is capable to produce a decision function with enough hidden units is shown in Fig. 2.

The *input layer* contains the input nodes interacts with the outside environment. The input units are only buffer units that pass the signal without changing it.

The *hidden layer* size is left to the appreciation of the user that will estimate the number of hidden nodes by his experience or by trial and error. This number must not be too large to avoid waste of computations and slow convergence process and not too low to make the network capable to absorb the set of training pixels.

The *output layer* represent the number of nodes, which is equal to the number of classes, each output node representing a class [12,7,4,8].

3. Genetic algorithm (GA)

The parameters which needs to be searched by GA in the training are the range of initial weights (R), initial values of learning rate and momentum rate, and the network architecture. The initial weights are generated using a uniform random number generator

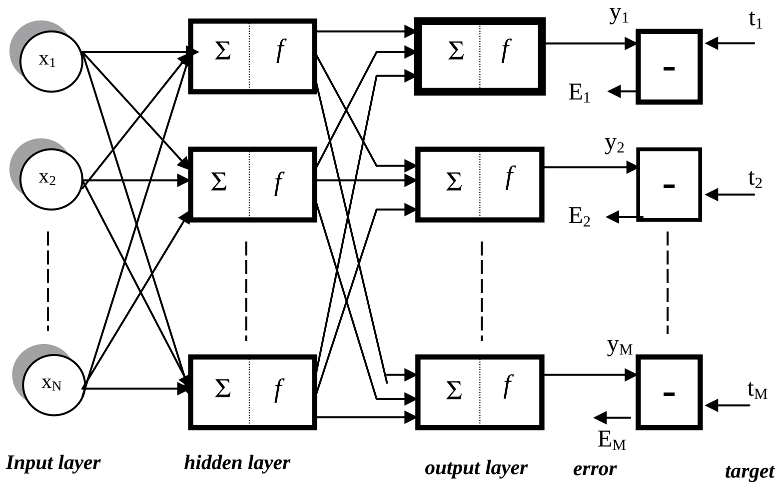


Fig. 2. Multilayer neural network architecture $x_1, x_2, \ldots, x_N$ = input nodes, $y_1, y_2, \ldots, y_M$ = output nodes, $E_1, E_2, \ldots, E_M$ = error, $t_1, t_2, \ldots, t_M$ = target.

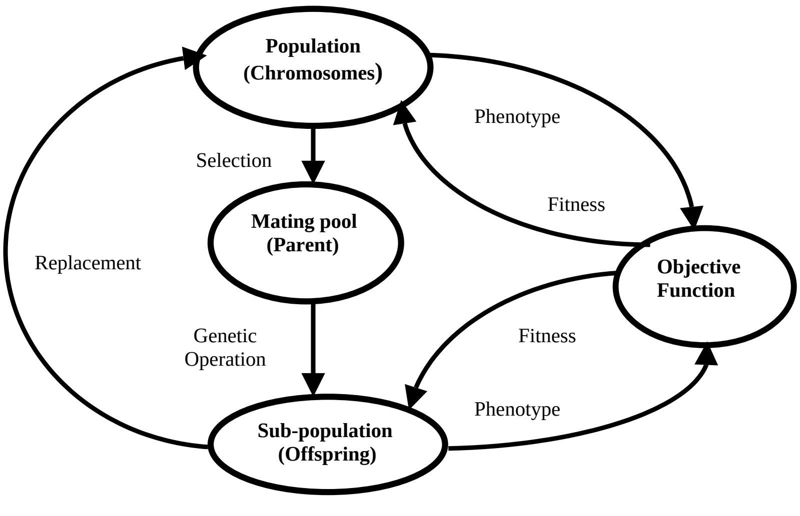


Fig. 3. Cycle of GA.

in the range $[-R, R]$. The network architectures described by the number of hidden layer (h), the number of nodes in each layer (n_i, $i = 0, 1, \ldots, h$). The GA proposed as searching process based on the laws of the natural selection and genetics. This algorithm which is used firstly to obtain the optimal values of the hidden nodes and the initial values of learning rate (R) and momentum top-level description of a simple GA is shown below [5,3]:

1. Randomly generate an initial population $P^0 = (a_1^0, a_2^0 \ldots a_\lambda^0)$.

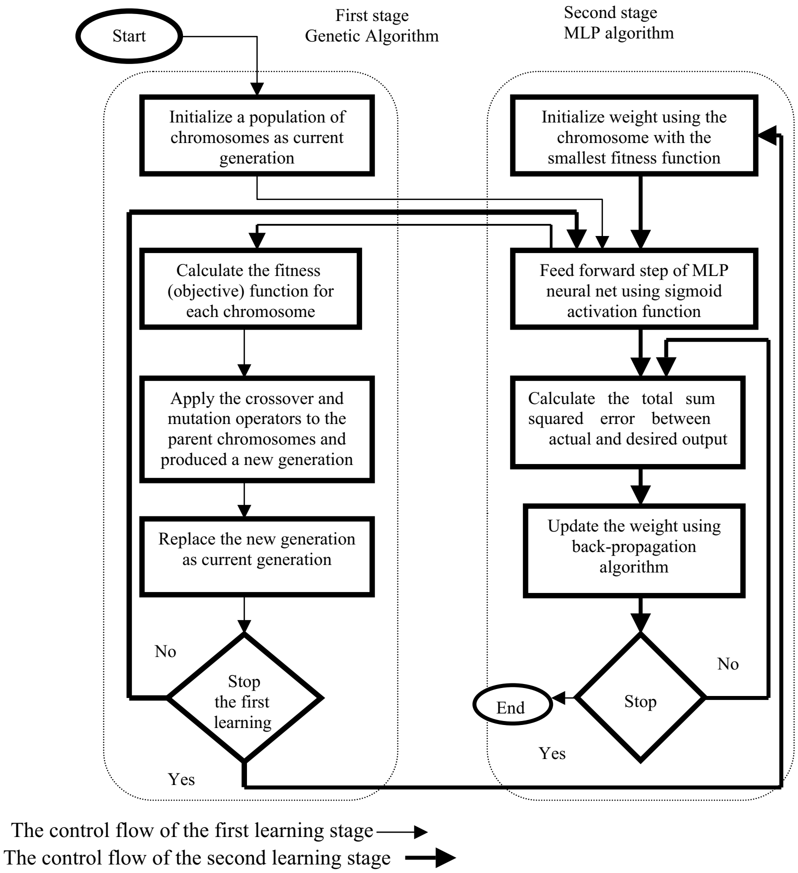


Fig. 4. The neural-genetic learning algorithm.

2. Compute the fitness $f(a_i^t)$ of each chromosome a_i^t in the current population P^t.
3. Create new chromosome P'^t of mating current chromosomes, applying mutation and recombination as the parent chromosomes mate.
4. Delete numbers of the population to make room for new chromosomes.
5. Compute the fitness of $f(a_i'^t)$, and insert these into population.
6. Increment number of generation, if not (end-test) go to step 3, or else stop return the best chromosome.

The GA cycle is shown in Fig. 3. From which the phenotype represents the coding which is used to represent the chromosomes. The mating pool represents the summing of chromosomes that represents the solution of the problem. Offspring represents new chromosomes result by mutation and crossover operation. Lastly, the objective function represents the function that measured the performance of the origin function [10,9,1].

4. Neural-genetic network learning algorithm

The proposed learning algorithm using genetic algorithm with multilayer neural networks for pattern recognition is presented in Fig. 4. It consists of two learning stages. The first learning using genetic algorithm with the feed-forward step uses stage to accelerate the whole learning process. The genetic algorithm performs global search and seeks near-optimal initial point (weight vector) for the second stage. Where, each chromosome is used to encode the weights of neural network. The fitness (objective) function for genetic algorithm is defined as the total sum squared system error (TSSE) of the corresponding neural network. Therefore, it becomes an unconstrained optimization problem to find a set of decision variables by minimizing the objective function. The best fitness function value in a population is defined as smallest value of the objective in the current population. Let us consider the neural-genetic algorithm as follows:

/* First learning stage */

Initialize the chromosomes randomly as current generation, initialize working parameters, and set the first chromosome as the *best_chromosome*.

a- For $i = 0$ to population size, do concurrently:
   - Initialize *sub_total_fitness* to zero and *sub_best_chromosome* as null.

b- For $j = 1$ to Length of chromosomes, do sequentially:
   - Perform feed-forward procedure of the MLP (using sigmoid activation function).
   - Calculate the objective function (system error for the neural network).

   Next j
   - Calculate the *total_fitness* by accumulating *sub_total_fitness*.

c- Store the *best_chromsome* as *sub_best_chromosome*.

   Next i

Guarded section 1-entry

d- Compare the *sub_best_chromsome* with each other and set the best *sub_best_chromosome* as the *best_chromosome*.

Guarded section 1-end

Do

e- For $i = 0$ to (population size/2), do concurrently:
   - Initialize *sub_total_fitness* to zero and *sub_best_chromosome* as null.
   - For $j = 1$ to length of chromosome, do sequentially:
     * Select parent using roulette wheel parent selection.
     * Apply two-point, multipoint, or uniform crossover and mutation to the parents.

- For $k = 1$ to length of chromosome, do sequentially:
  ∗ Perform feed-forward procedure of the MLP to parent chromosomes.
  ∗ Calculate the objective function (system error for the neural network) for parent chromosomes.

Next k
- Calculate the *sub_total_fitness* by accumulating objective of each chromosome.
- Store the *best_chromosome* as *sub_best_chromosome*.

Next j

g- Replace the old generation by the new generation WHILE (stopping criteria).

/* Second learning stage */

- Set the *best_chromosome* as the initial weight vector, set up the topologic structure of neural network.
- Compute the actual output of the feed-forward MLP neural network.
- Compute the error between desired output and actual.
- Updates the weight using back-propagation algorithm.

END

5. Implementation and results

Multilayer neural networks are trained to (12) images of sizes 10×10 pixels, which represents the smallest image size required for the aircraft recognition, where we can recognize the aircraft with their direction. The initial values such as the number of hidden nodes, learning techniques, and input coding in addition to initialization weight have been defined. To test the ability of net to pattern recognition in recall or testing phase, the failure rate proposed as the measurement to the sensitivity of neural network is defined as

$$\text{Failure rate (FR)} = \frac{\text{Total No. of tested images} - \text{No. of classified images}}{\text{Total number of test image}} \times 100\%.$$

Fig. 5 shows the images of 256 gray levels used to test the proposed algorithm.

Test 1: MLP network

Training parameters:

Image size $= 10 \times 10$ pixels
Number of images $= 12$
Desired Accuracy $= 90\%$
Hidden nodes $= 60$

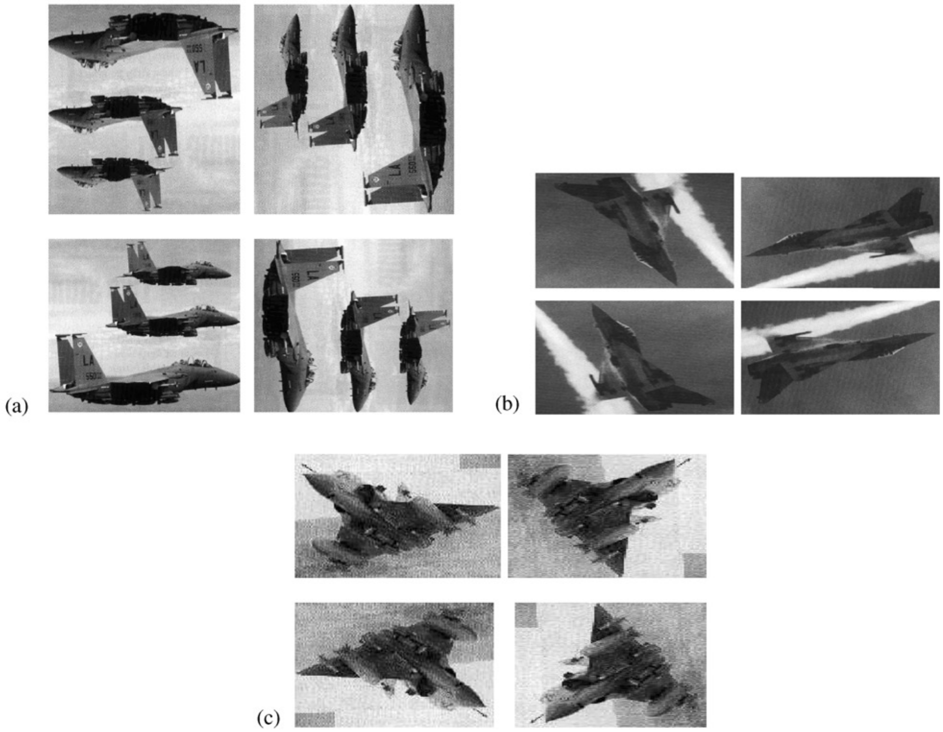


Fig. 5. Groups of images of 256 gray level used in testing of the neural-genetic algorithm. (a) Four images to four directions of the same groups of aircraft; (b) Four images to four directions of the same aircraft with missile; (c) Four images to four directions of the same aircraft.

Learning rate = 0.6
Momentum = 0.06

Training result are:

Epoch's	75	150	225	300	366
Failure rate	83.33%	58.33%	50%	50%	0.00%
TSSE	0.7252	0.6747	0.3647	0.3613	5.7×10^{-5}

Test 2: Neural-genetic network learning algorithm

The neural-genetic network learning algorithm applied to the same images. A three-layer neural network with one hidden layer was used to learn this problem. The total number of links in these three-layer neural networks is 6600.

Training parameters:

Population size = 12
Mutation rate = 0.05
Crossover rate = 0.46

Real coding to the chromosomes

Length of chromosomes = 6600
Number of generation = 8
Desired accuracy = 90%

Training results are:

No. of generation	2	4	6	8
Total Fitness	6.5653	6.4254	5.9028	5.6283

Training results are:

Epoch's	25	50	75	100	125	158
Failure rate	83.33%	75%	66.67%	58.33%	50%	0.00%
TSSE	0.9831	0.8616	0.5054	0.3651	0.216	0.0008

The system error on the first test by using the MLP learning neural network algorithm was 5.73×10^{-5} after 366 iteration. After 8 generations, the first learning stage of neural-genetic network learning algorithm met stopping criteria. The best average fitness function has value of about 5.6283. The result of the first learning stages is used as an initial weight vector for the second learning stage. After 158 iterations, the system error in the neural-genetic neural network learning algorithm converges to a value of less than 0.0008 and the algorithm achieves 100% to train all the images. The performance of hybrid neural-genetic networks and MLP neural net are shown in Fig. 6.

6. Conclusions

The neural network depends on the training conditions (hidden node number, learning rate, momentum, and initial weight). The minimum size used to recognize the type of aircraft and its direction is (10×10) pixels. This technique can be used to detect the stealth aircraft, which cannot be detected by the radar signal. The learning time depends strongly on the resolution of image, hidden nodes and training pixels. The proposed genetic algorithm presented to find suitable network, which was able to perform well within the specified tolerance e.g. zero failure rate for MLP net. As

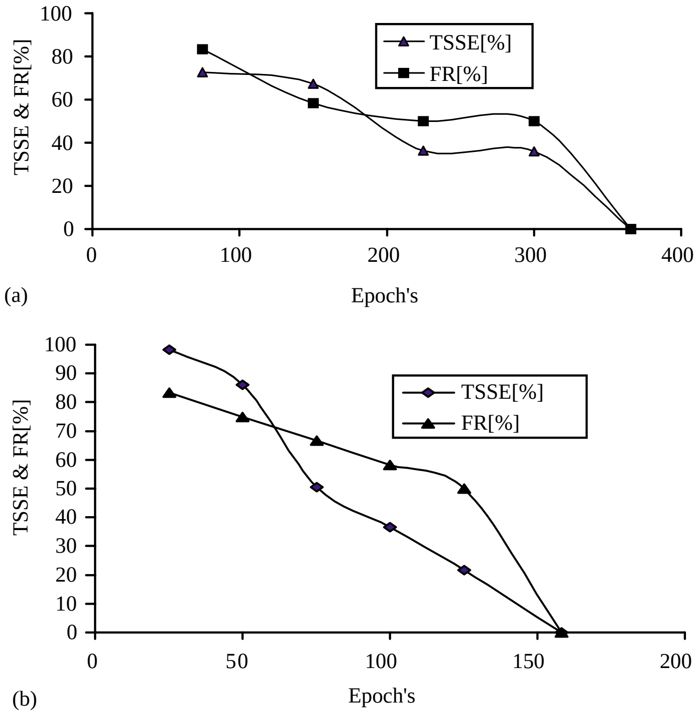


Fig. 6. (a) Performance of MLP neural net. (b) Performance of neural-genetic learning algorithm.

far as training procedures are concerned, we have identified some important learning phenomenon e.g. the effect caused by learning rate, number of hidden units, training of under units, training sets, etc. Thus, the ratio of reduction on the number of epoch's is least or equal to 50% if we used the genetic algorithm. Integrating the genetic algorithm with MLP network, the following observations are the conclusions can be drawn:

- The results of neural network learning are sensitive to the initial value of the weight vector. A genetic algorithm is employed to perform global search and to seek a good starting weight vector for subsequent neural network learning algorithm. The result is an improvement in the convergence speed of the algorithm.
- The problem of entrapment in a local minimum is encountered in gradient-based neural network learning algorithm. In the neural-genetic learning algorithm presented, this problem is circumvented using a genetic algorithm, which is guided by the fitness function of population rather than gradient direction. After several iterations of the global search, the first learning stage returns a near-global optimum point that is used as the initial weight vector for the second learning stage.

7. For further reading

The following reference may also be of interest to the reader: [2].

Acknowledgements

The authors are very grateful to the useful comments given by anonymous reviewers, which helped greatly in revising this work.

References

[1] D. Anthony, E. Hines, The use of genetic algorithms to learn the most appropriate inputs to neural network, Application of the International Association of Science and Technology for Development-IASTED, June, 1990, 223–226.

[2] T.N. Bui, B.R. Moon, Genetic algorithms and graph partitioning, IEEE Trans. comput. 45 (7) (1996) 841–855.

[3] L. Davis, Hand book of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[4] L. Fauselt, Fundamentals of Neural Networks, Prentice-Hall, International Inc., Englewood Cliffs, NJ, 1994.

[5] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Pub. Comp. Inc., Reading, MA, 1989.

[6] R. Gonzalez, P. Wintz, Digital Image Processing, Addison-Wesley Pub. Comp. Inc., Reading, MA, 1992.

[7] R.L. Harvey, Neural Networks Principles, Prentice-Hall, Englewood Cliffs, NJ, 1994.

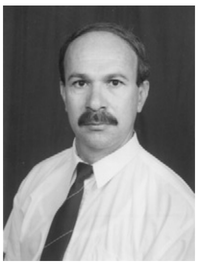
[8] R.P. Lippmann, Introduction to computing with neural nets, IEEE Trans. Acoust. Speech Signal Process. 4 (87) (1987) 3–22.

[9] M. Mitchill, An introduction to genetic algorithms, Massachusetts Institute of Technology, England, 1996.

[10] S.c. Ng, S.H. Leung, C.Y. Chung, A. Luk, W.H. Lau, The genetic search approach, IEEE Signal process. Mag. 13 (6) (1996) 38–46.

[11] S.E. Umbaugh, Computer Vision and Image Processing A practical approach using CVIP tools, Prentice-Hall PTR, Englewood Cliffs, NJ, 1998.

[12] J.M. Zurada, Introduction to Artificial Neural Systems, Jaico Pub. House, Bombay, India, 1997.



Yas Abbas K. Alsultanny received the B.S. degree from the Technology University, Control and Systems Engineering Department, Iraq in 1982, the M.S. degree, in Computer Engineering from the Technology University, Iraq 1991, and the Ph.D. degree, in image processing from Baghdad University, Iraq, in 1997. From 1997 to 2000, he was an assistant professor at the Computer Engineering Department in Baghdad, and a member in the scientific research council in Baghdad. He supervised 13, M.S. thesis. From 2000 he was an assistant professor at the computer Science department, in the Computer Science and Information Technology Department in the Applied Science University, in Jordan, he has been a plenary speaker at various conferences. His research interested, include, image processing, pattern recognition, fuzzy set, neural network, wavelet, and GIS.

Musbah M. Aqel was born in Hebron/West Bank, in 1959, He received the B.E. degree in electronics & Communication from regional engineering college, University of Kashmir, India, in 1987, and the M.Tech. and Ph.D. Degrees in Computer Science from School of Computer and Systems Sciences, Jawaharal Nehru University, New Delhi, India, in 1989 and 1993, respectively. From 1994 to 1999 he was an Assistant Professor at the Department of Computer Science and Information Systems, Faculty of Basic Science at Applied Science University, Amman, Jordan, From 1999 to 2001 was a chairman of Computer Science and Information systems Department. He is a member of organization steering, and technical Committees of many national and international conferences, the advisory Board of Jordan Journal of Applied Science, Jordan Engineering Association, and Jordan Computer Society. He is Currently Dean of the Faculty of Computer Science and Information Technology and Associate Professor with the Department of Computer Science at Applied Science University, Amman, Jordan. His research interests include Algorithms Design and Analysis, Database Systems, Simulation and Modeling, Pattern Recognition, and Image Processing.