International Journal of Scientific Research & Engineering Technology (IJSET)

Vol.1, pp.25-32, 2014

Comparative study between neural and neuro genetic classifier for ROI extraction

Mohamed Amine Bahri, Hassene Seddik, Anissa Selmani

Department of Electrical Engeneering, CEREP , ESSTT
5,Av. Taha Hussein,1008Tunis,Tunisia
a.minovic@hotmail.fr, hassene.seddik@estt.rnu.tn, anissaselmani02@gmail.com

Abstract— In this paper, a neuro-genetic clustering network is proposed for efficient ROI (Region Of Interest) extraction. Our goal is to be able to determine with accuracy different classes located in the image. To this end, we introduce a comparative study between neural and neuro-genetic classifier, i.e. genetic algorithms (GA) are applied to optimize internal parameters of the network structure (weights and bias) through a fitness function. The classification results proved that the combination between a feed forward neural network (FFNN) and genetic algorithms generates better results in terms of high accuracy and reliability than other methods based only on FFNN.

Keywords— Image classification, neuro-genetic classifier, genetic algorithms, roads detection**.**

1. Introduction

Image classification analyses the numerical properties of various image features and organizes data into categories.

In the literature, a large number of approaches have been proposed and few of them were interested with the neuro-genetic method: Sulaiman and al. [1] used a genetic algorithm to optimize the number of neurons in the hidden layer, the learning rate, the momentum rate, the type of activation function and the learning algorithm of a multi-layer feed forward neural network. Loan Ileanà and al. detailed in [2] the different steps showing the optimization of feed forward neural networks structure using genetic algorithms. Also, Philipp Koehrn examined in his thesis [3] how genetic algorithms can be used to optimize the network topology. In [4], David Montana and al. used the genetic algorithms to train feed forward neural networks. Whereas recently, Deepak Dhanwani and al. presented in [5] a study of a new hybrid model of neural networks and genetic algorithm to initialize and optimize the connection weights of ANN so as to improve the performance of the ANN and the same has been applied in a medical problem of predicting stroke disease. The proposed approach is divided in three essential steps:

A pre-processing step: it is a very important step consisting in finding and computing the most discriminative features able to distinguish the ROI of the image. Using non discriminative features may introduce confusion in the network training [6,7].

Feed forward neural network structure designing: in the literature, there are no formal methods for optimal choice of the neural network's structure or initial features [2]. However, experiments have shown that the use of two layer network (one hidden layer) is able to approximate most of the non linear functions and using two hidden layers can approximate any non linear function. Concerning the number of neurons in a neural networks hidden layer, and according to many tests, the choice can be made as follows: N inputs plus N outputs divided by two. Other rules relate to the number of examples available: use at most so many hidden units that the number of weights in the network times 10 is smaller than the number of examples. Weights and bias optimization through GA: perform fitness evaluation of internal network feature. The random set of bias and weights are optimized through it. The optimal values will be able to generate the best pattern learning and faster error rate reached. In this paper Section 4 discusses the proposed neuro-genetic approach and section 5 deals about the experimental results and the comparative study.

2. Feed Forward Clustering Network

2.1 Network classification

For reliable supervised classification, sufficient patterns have to be introduces represented by discriminative features. The first phase is the pattern learning. Wherein, the network is driven from features vectors containing N samples extracted from each sliding window with size (5×5) belonging to various areas of the image (ROI, edge, etc…). The second phase is the test; it is done with 50% of the database. The third is an evaluation phase exploiting the network outputs applied on similar images.

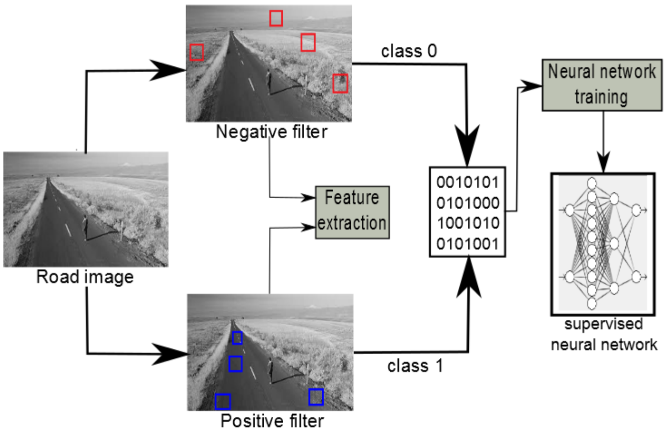


Fig. 1 Overview of the Neural Network training step.

Most units in neural network transform their inputs by using a scalar-to-scalar function called an activation function, producing a value called the unit's activation. In order to calculate the output of a single neuron, we start by computing the weighted sum of the neuron inputs. We shall define the inputs and neurons weights as follows: $I_v = [x_1, x_2, x_3, \ldots, x_n]^T$ and $W_v = [w_1, w_2, w_3, \ldots, w_n]^T$.

Where I_v is the input vector of the network and W_v is the weights vector for each neuron. The bias are added to the weighted sum and feed the activation function of the neuron as summarized by the following equation:

$$Y = \sigma\left(\sum_{k=1}^{n} w_k * I_k + B\right) \quad (1)$$

Where Y is the output of the neuron, σ is its activation function, w_k is the weight of the input I_k and B is the bias of the neuron.

2.2 Back-propagation learning algorithm (BPLA):

The procedure of a BPLA is described as follows:

All the introduced data is normalised. The obtained data is mapped to the bound [0;1] so as to avoid the saturation of neurons [9,10].

$$V_{new} = \frac{(V_{old} - V_{\min})}{(V_{\max} - V_{\min})} \times (D_{\max} - D_{\min}) + D_{\min} \quad (2)$$

Where: $V_{\min}$ is the minimal value of the introduces features, $V_{\max}$ is their maximal value, $D_{\max}$ is the maximum value after normalization, D_{min} is the minimum value after normalization, V_{new} is the new transformed value and V_{old} is the old value before normalization. The network structure is composed by only one hidden layer; The number of neurons in this layer is taken equal to 20 neurons. For this structure we choose a Learning rate (η) in the range between 0.1 and 1.0 whereas the momentum coefficient (α) is taken within the range of 0.01 and 1.0. A sigmoid Transfer defined by the following equation is applied on each neuron:

$$f(x) = \frac{1}{(1 + \exp(-x))} \quad (3)$$

The output of each neuron's is calculated using (equation1). Whereas the output layer is expressed as follows:

$$net_k = TV_k + \delta_k^l \quad (4)$$

Where TV_k is the target value of the output neuron k and δ_k^l is the error of neuron k . The error of each hidden layer is:

$$\delta_j^l = \sum_{i=1}^{n} (\delta_i^{l+1} \times W_{ji}^l \times f'(net_j^l)) \quad (5)$$

Where W_{ji} is the weight on the connection from neuron i to j and f' is the first derivative of the sigmoid function.

The incremental change for every weight for each learning interaction is computed by equation 6:

$$\Delta W_{ji}^l = (\eta \times \delta_j^{l+1} \times f'(net_j^l)) + \alpha \times \Delta W_{ji}^{l-1} \quad (6)$$

The error between the real and desired output, is computed using the sum of the squared error represented by the following equation:

$$SSE = \sum_{i=1}^{n} (T_i - Y_i)^2 \quad (7)$$

Where T_i is the actual value and Y_i is the estimated value.

3. Genetic Algorithm (GA)

3.1 General description

A genetic algorithm proceeds iteratively by generating new populations of individuals from the old ones. Every individual is the encoded (binary, real, etc.) version of a uncertain solution [5]. The canonical algorithm applies stochastic operators such as selection, crossover and mutation on an initially random population in order to compute a new population (each individual is represented by a set of chromosomes, and a population is a set of individuals).

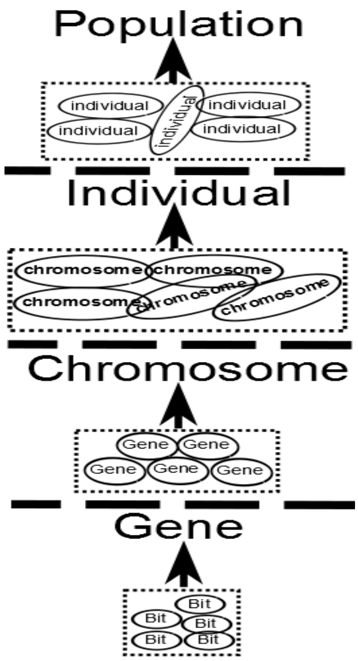


Fig. 2 The organization levels of a genetic algorithm

3.2 Computation steps of the GA

Firstly, we must choose a method of chromosomes encoding that ensures coding all possible solutions and facilitates the implementation of breeding operations. In this study, we used the binary coding because it is simple and effective. Computation steps of the GA are as follows [8]:

Step 1: Generate random population of N chromosomes. Their genes are of real and generated randomly.

Step 2: Evaluate the fitness of each chromosome in the population.

Step 3: Create a new population as follows:

- *Selection*: select two parent chromosomes from a population having the better fitness and the bigger chance to be selected.
- *Crossover:* cross over the parents to form a new offspring (children). Offspring can be an exact copy of parents if the crossover operation was not performed. The arithmetical type of crossover is the most used and is defined as follows: [13]

$$B_i^k = (b_1^k, .. b_i^k, ..., b_n^k), \text{ where } k = 1,2$$

$$b_i^1 = \lambda \cdot c_i^1 + (1-\lambda) \cdot c_i^2 \qquad (8)$$

$$b_i^2 = \lambda \cdot c_i^2 + (1-\lambda) \cdot c_i^1$$

Where B is the new offspring, c represents each gene and λ is a positive constant.

- *Mutation:* mutate the new children at each locus (position in chromosome). The non uniform type of mutation is the most used and is defined as follows: [13]

$$c_i^1 = \begin{Bmatrix} c_i + \Delta \cdot (t, d_i - c_i)\, if\ \tau = 0 \\ c_i - \Delta \cdot (t, c_i - \alpha_i)\, if\ \tau = 1 \end{Bmatrix} \qquad (9)$$

$$\Delta(t, y) = y \cdot (1 - r^{(1 - \frac{t}{g_{max}}) \cdot d}) \qquad (10)$$

Where τ is a binary number is randomly chosen, α is set as 0.5, g_{max} is the maximum generation, $[0, y]$ will be the value given by the function.

- Place new children into a new population.

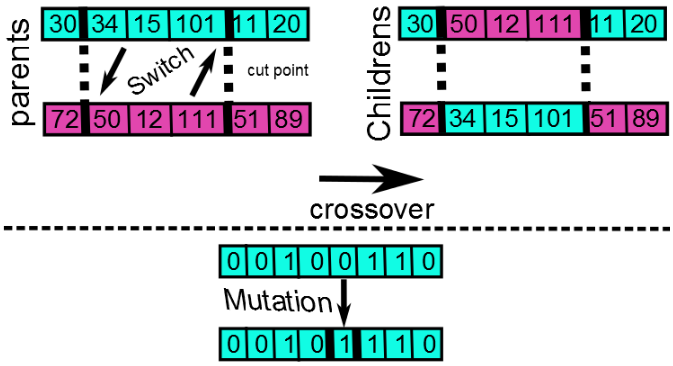


Fig. 3 Basic operators of the genetic algorithm: crossover and mutation operator

Step 4: if the termination condition is satisfied, stop, and return the best solution in current population.

Step 5: repeat from “step 2” until the algorithm converge.

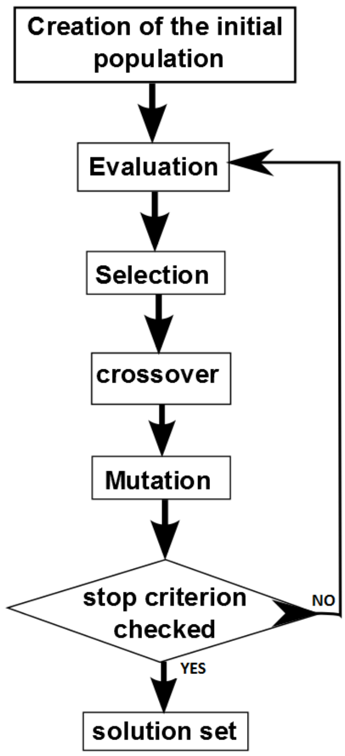


Fig. 4 Cycle of standard genetic algorithm

4. THE PROPOSED NEURO-GENETIC METHOD:

In this paper, we propose a new approach for image classification based on the combination between genetic algorithms and neural network: genetic algorithm used for neural network weights optimization.

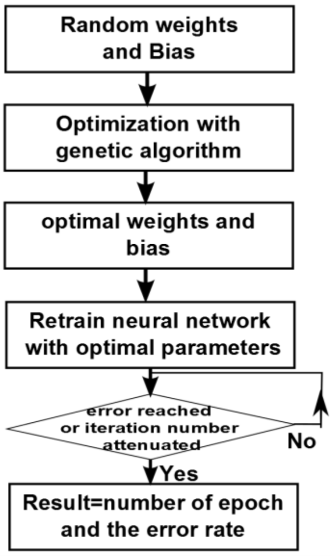


Fig. 5 The block diagram of the proposed method

To represent the elements of a research space $RS = RS_1 \times \ldots \times RS_n$ using the binary alphabet, a function $encod\_bin_i : RS_i \rightarrow \{0,1\}^{L_i}, L_i \in N$ must be specified, which encodes each element in RS_i using binary strings of length L_i. We recall that RS is the research space and L is the length of the binary string.

An element $x = (x_1, \ldots, x_n) \in RS$ is represented by linking together the coding of each one of its components:[14]

$$encod\_bin(x) = \{encod\_bin_1(x_1) .. encod\_bin_n(x_n)\}$$

The proposed method is described as follows:

Firstly, a pre-processing step is conducted for the selection of the training window: to have reliable classification, it is very important to determine an adequate amount of information that is able to characterize ROI with highest accuracy [11]. Then, we define the fitness function that will be used with genetic algorithm to determine the new weights. Once the parameters of the algorithm are set, we retrain the network with new weights and finally, we simulate the outputs delivered by the network.

Otherwise, a sequence of input vectors is fed to neural network and the output signal is compared with its corresponding target. The absolute difference is calculated, and the sum of all errors (MSE) for the whole sequence is used as a measure of fitness for the particular network under consideration shown in (Fig. 6). Genetic operator is applied to create a new population.

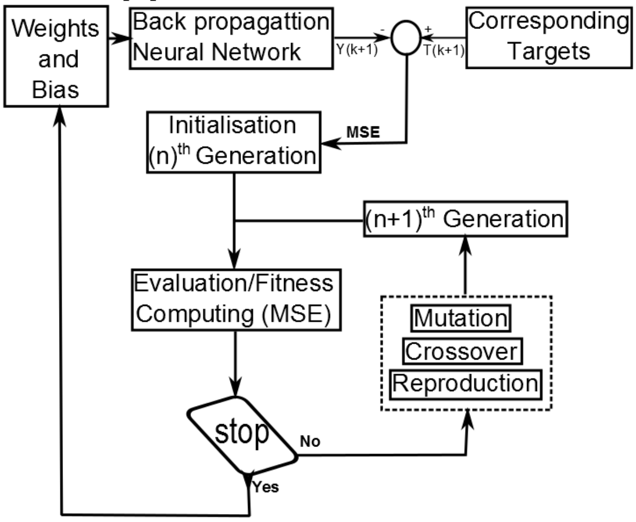


Fig. 6 Schemes for training feed forward neural network to identify a plant

5. SIMULATION RESULTS:

5.1 Experimental results:

In the conducted experiments, we applied the proposed approach to classify gray level images, detect and extract roads. For each image, we will use training windows having size (5×5). The structure of the neural network is presented by the following figure.

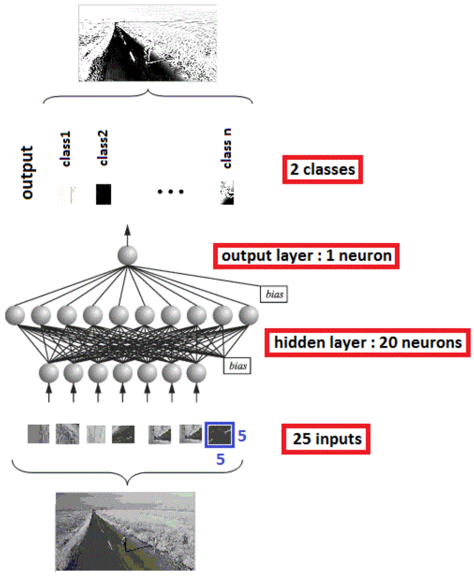


Fig. 7 The used network structure

The initial set of outputs considered as solutions generated by the GA is produced by a random number generator. We start by encoding the chromosome, i.e, the weights and bias are encoded as vectors of real numbers. After that, we trained the used network and we calculated the mean square error in each iteration defined by the next formula:

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (T_i - Y_i)^2 \qquad (11)$$

Where T_i is the actual value and Y_i is the estimated value.

In our genetic algorithm, we tested a large number of different types of genetic operators and we finally decided to use the following operators. Stochastic selection lays out a line in which each parent corresponds to section of the line of length proportional to its expectation. The algorithm moves along the line in steps of equal size, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size [12].

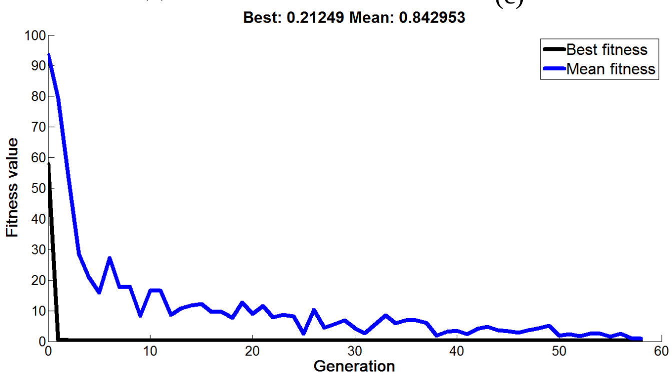
Mutation function: constraint dependent, i.e, chooses Gaussian if there are no constraints and adaptive feasible otherwise.

Crossover function: scattered. It creates a random binary vector. Then, it selects the genes where the vector is a “1” from the first parent, and the genes where the vector is a “0” from the second parent, and combines the genes to form the child [12].

Once the parameters of the algorithm are set, we minimize the RMSE function and we initialize the neural network with the new weights obtained corresponding to the minimum error of the genetic algorithm. The best fitness value reached by the GA is shown by the following figure (Figure 8.b) and equal to 0.21249. This minimum threshold allows generating the optimal features able to improve the network behaviour in learning and classification. Using the new generated weights and bias, we will classify the image into two classes (Figure 8.c):


(a)


(c)



(b)

Fig. 8 (a) Original image; (b) MSE optimization according to the original image; (c) classified image

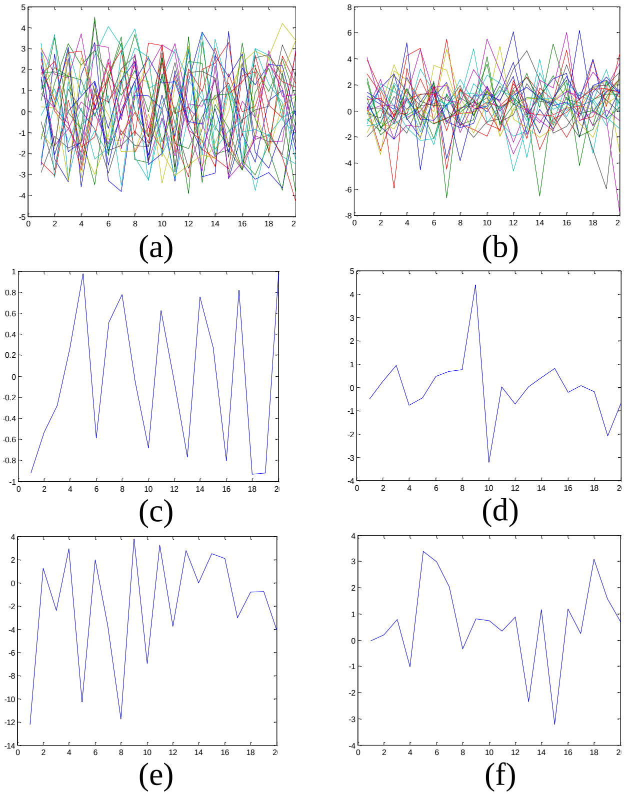


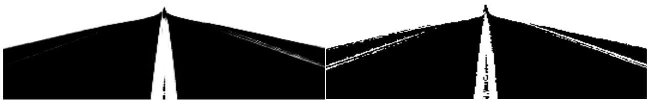
Fig. 9 Weight evolution of the input layer (a) before optimization, (b) after optimisation; Weight evolution of the hidden layer (c) before optimization, (d) after optimisation; Bias evolution of the input layer (e) before optimization, (f) after optimisation;

The figure above shows that the evolution shape of weights and bias, before and after optimization, changes radically. Genetic optimization requires a variation in their values randomly initialized to another shape having values more reduced.

5.2 Comparative study:

In order to test the classification performance of our proposed method, we compared our approach with another technique which uses only a feed forward neural network having the same design as the proposed approach. The following set of figures show an original image classified respectively by a neural network with random features initialisation and a neural classifier optimized by GA. The road is well detected and clearly extracted using the neuro-genetic classifier.

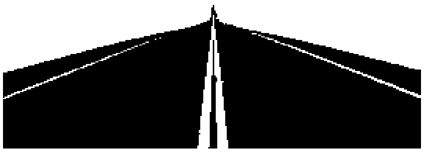
(a)



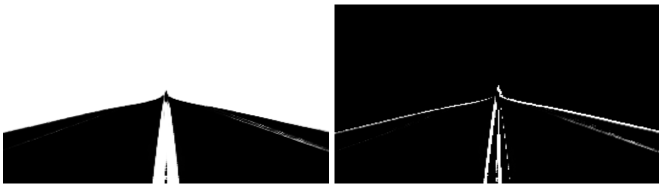
(b) (c)

Fig. 10 (a) Original image; (b) Result of classification by Neural network approach (c) Neuro-Genetic approach

Always in the same context, we evaluate the classification results using the criterion BER (Bit Error Rate)



(a)



(b) (c)

(d) (e)

Fig. 11 (a) Perfect classification; (b,d) neural and neuro-genetic classification respectively (thresholded); (c,e) BER evaluation

The following table shows the evaluation results of the image classified by two methods.

TABLE I: COMPARATIVE STUDY BETWEEN NEURAL AND NEURO-GENETIC CLASSIFICATION

Type of classification	Neural classification	Neuro-genetic classification
Image size	49 128 pixels	49 128 pixels
Defect	1881 pixels	31 pixels
Error rate (%)	3.82 %	0.06 %

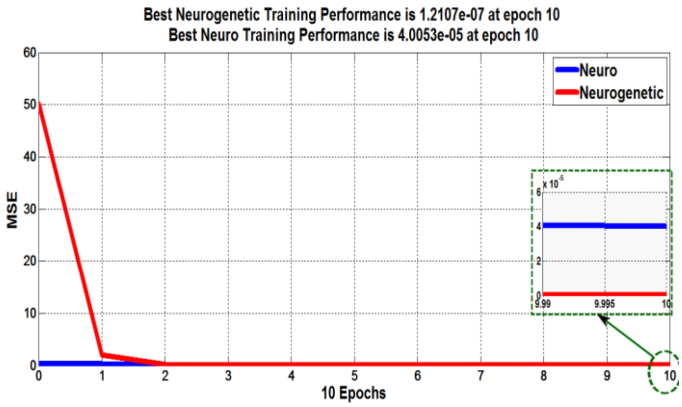


Fig. 12 comparison of the MSE between BPNN and GA-based BPNN for the same number of iterations.

The previous figure prove that the neuro-genetic classifier reaches a lower error rate that the neural one. After 10 iterations the error corresponding to the neuro-genetic classifier is 4.10^2 times lower than the neural classifier.

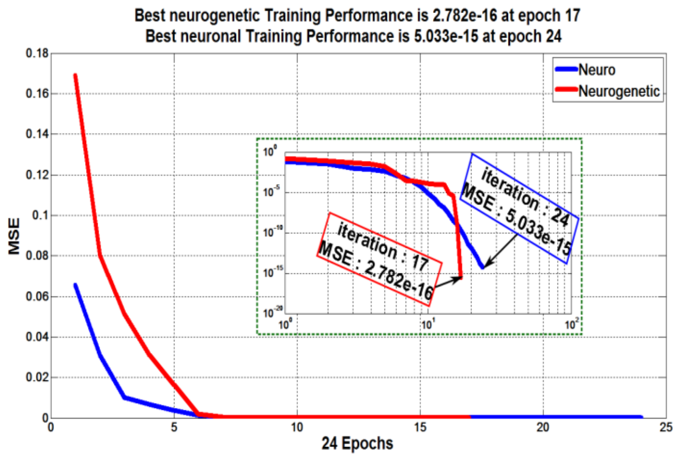


Fig. 13 comparison of the MSE speed convergence between BPNN and GA-based BPNN (convergence error = 10^{-14}).

The results show clearly that our new approach provides better accuracy and faster convergence: for the same number of epoch (10 epochs), achieving error (1.211e-07) with GA-based BPNN is lower than the BPNN's error (4.005e-05) (Figure12). Also, according to tests, we could note that the optimized neural network converges faster (17 epochs) than the BPNN (24 epochs) with random feature initialisation (Figure13).

(a)

(b) (c)

Fig. 14 (a) Original image; Result of classification by (b) Neural network approach (c) Neuro-Genetic approach

(a)

(b) (c)

Fig. 15 (a) Original image; Result of classification by (b) Neural network approach (c) Neuro-Genetic approach

Figure 14 and 15 show clearly that using the neuro-genetic classifier generates best classification results. The ROI are well identified and extracted, the road edges and direction is clearly and precisely set as illustrated in figure 14. c.

6. CONCLUSION:

In this paper, a comparative study between neural and neuro-genetic classifier is proposed for ROI extraction. Based on current results, the neuro-genetic approach seems to be more suitable for image classification than supervised neural network. Also, the proposed GA based BPNN is the fastest in the learning process and reaches the best error rates.

REFERENCES

[1] S. Sulaiman, T. Abdul Rahman and I. Musirin, “A genetic algorithm-based hybrid multi-layer feed forward neural network for predicting grid-connected photovoltaic system output”, IPCSIT, vol. 25, LACSIT press, Singapore, pp. 147-150, 2012.

[2] I. Ileanà, C. Rotar and A. Inccze, “the optimization of feed forward neural networks structure using genetic algorithms”, ICTAMI, Thessaloniki, Greece, pp. 223-234, 2004.

[3] P. Koehn, “Combining genetic algorithms and neural networks: the encoding problem”, thesis presented for the master of science degree, the university of Tennessee, Knoxville, 67p, December 1994.

[4] D. Montana and L. Davis, “training feed forward neural networks using genetic algorithms”, journal of machine learning, pp. 762-767, 1989.

[5] D. Dhanwani and A. Wadhe, “study of hybrid genetic algorithm using artificial neural network in data mining for the diagnosis of stroke disease”, IJCER, vol.03, Issue. 4, pp. 95-100, April 2013.

[6] Hughes, G. F. 1968. On the Mean Accuracy of Statistical Pattern Recognizers. IEEE Transactions on Information Theory, IT-14:55-63.

[7] Mohammed talibi-alaoui and abderrahmane sbihi, “fractal features classification for texture image using neural network and mathematical morphology,”, WCE, VOL. I, 2012

[8] Z. Che, T. Chiang and Z. Che, “feed forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm”, IJICIC, vol.7, number 10, pp. 5839-5850, October 2011.

[9] M. H. Hassoun, “fundamentals of artificial neural networks”, MIT press, Cambridge, London, MA,1995.

[10] T. Masters, “practical neural network recipes in C++”, Academic press professional, INC., San diego, CA, 1993.

[11] M. Bahri, H. Seddik and A. Selmani, “Efficient texture classification using a kohonen clustering network and the LNLBP attributes”, IJECCE, vol. 3, Issue 3, pp. 900-905, 2013.

[12] F. Kussener, “Active contour: a parallel genetic algorithm approach”, ICSI, Cergy, France, June 14-15, 2011

[13] V. Pallavi and V. Vaithiyanathan, “combined artificial neural network and genetic algorithm for cloud classification”, IJET, vol. 5, NO. 2, pp. 787-794, May 2013.

[14] F. Herrera, M. Lozano and J. L. Verdegay, “tackling real-coded genetic algorithms: operators and tools for behavioural analysis”, Artificial Intelligence Review, Granada, Spain, Pp. 265-319, 1998.

AUTHOR'S PROFILE

M. Bahri

was born in Tunisia in 07 June 1989. He received the M.S. degree from National higher school of engineer (ENSIT) in Tunis, Tunisia, in 2013.

He is author of 3 conference papers and currently working owards the Ph.D. Degree in the division of Electronic and digital technologies at ENSIT. His current research focuses on texture and image analysis.

H. Seddik

was born in Tunisia in 15 October 1970, he has obtained the electromechanical engineer degree from the national higher school of engineer (ENSIT) in Tunis, Tunisia in 1995 and followed by the master degree in "signal processing: speaker recognition" and the thesis degree in image processing "watermarking using non conventional transformations".

He is currently an assistant professor in ENSIT and works in the field since 17 years. He has authored more than 10 papers and book chapters in international journals and about 47 conference papers. His domain of interest is: Audio-image and video processing applied in filtering, encryption and watermarking.

Dr Seddik belongs to the CEREP research unit and supervises actually 04 thesis and 08 masters in the field.

A. Selmani

was born in Tunisia in 20 February 1988. She received the M.S. degree from National higher school of engineer (ENSIT) in Tunis, Tunisia, in 2012.

She is currently working towards the Ph.D. Degree in the division of Electronic and digital technologies at ENSIT. She has authored 6 conference papers. Her research interests include image, speech and video processing.