Evolving Neural Networks using Moment Method for Handwritten Digit Recognition

H. El Fadili, K. Zenkouar, and H. Qjidaa

***Abstract*—**This paper proposes a neural network weights and topology optimization using genetic evolution and the backpropagation training algorithm. The proposed crossover and mutation operators aims to adapt the networks architectures and weights during the evolution process. Through a specific inheritance procedure, the weights are transmitted from the parents to their offsprings, which allows re-exploitation of the already trained networks and hence the acceleration of the global convergence of the algorithm. In the preprocessing phase, a new feature extraction method is proposed based on Legendre moments with the Maximum entropy principle MEP as a selection criterion. This allows a global search space reduction in the design of the networks. The proposed method has been applied and tested on the well known MNIST database of handwritten digits.

***Keywords*—**Genetic algorithm, Legendre Moments, MEP, Neural Network.

I. Introduction

MULTIPLE layer perceptron networks (MLP) trained with backpropagation algorithm are frequently used and have shown good capabilities to solve a wide variety of real-world problems. However, their performance is strongly affected by the quality of the representation of the patterns i.e. features, and the architecture of the network used as a classifier. Consequently, The main concern of the present paper is twofold, on the one hand we present an efficient feature extraction method using the orthogonal moments known for their invariance, high robustness in the presence of noise and their use of global instead of local information of an image [1], [2], [3]. The proposed approach investigates the application of moment method to evaluate a set of candidate features and to select an informative subset to be used as input data for a neural network classifier. for this, we introduce the Maximum Entropy Principle (MEP) as a selection criterion.

On the other hand, as pointed out in [4], the network topology has a crucial impact on the speed and performance of Backpropagation trained networks. Choosing an appropriate topology for a given problem depends on personal experience of the human designer. This manual neural network design is something of black art [5] and it can be assumed that most of such network designs are not optimal. In order to adapt the network topology to the problem at hand, we propose an automatic design procedure based on genetic algorithms (GAs). The proposed algorithm aims to generate a near optimal feedforward neural networks dynamically for the task of handwritten digit recognition, using an automatic topology optimization by a proposed genetic operators. The basic idea of the proposed algorithm is a global sampling by GA over the space of alternative solutions with respect to network topologies while the backpropagation algorithm proceeds by locally searching the immediate neighborhood of a current solution. We demonstrates that with the use of an efficient selection strategy, a new crossover and mutation operators, the generated networks reach classification performances near optimum with high recognition rates and good generalization ability.

As a summary, the proposed contribution for object recognition is addressed following two steps : preprocessing and recognition. In the first one, we propose a novel method that extracts optimal object features using the MEP as a selection criterion [3]. Our objective is to reduce the input dimensionality of the classification problem by eliminating features with low information content or high redundancy with respect to other features. The second step is achieved by using GA for automatic performance optimization .The networks adaptation changes architectures and weights by a presented crossover and mutation operators. The weights are transmitted to the produced offsprings by specific inheritance procedure which allows the reutilization of the already trained networks and hence the acceleration of the global convergence of the algorithm. The feature extraction method introduces a beneficial prior knowledge, and allows a global search space reduction in the design of the networks. The resulting genetic algorithm is applied and tested using the well known MNIST database of handwritten digits [6].

Our paper is organized as follows: in Section II, some basic definitions are given including Legendre moments and their approximation. Sections III presents the optimal Moment selection method. Section IV points out the proposed neural network evolution and the details of the proposed crossover and mutation operators. Finally, section V and VI deal with the summary of important results and conclusions of the paper.

H. El Fadili is with Faculté des science dhar el mehraz Département de physique, LESSI B.P. 1796 Fès Maroc (phone: 21265964574 e-mail: el_fadili_hakim@hotmail.com).

K. Zenkouar is with Faculté des science dhar el mehraz Département de physique, LESSI B.P. 1796 Fès Maroc (e-mail: kzenkouar@hotmail.com).

H. Qjidaa is with Faculté des science dhar el mehraz Département de physique, LESSI B.P. 1796 Fès Maroc (e-mail: qjidah@yahoo.com).

II. Legendre Moments

The Legendre moments of order $(p+q)$ is defined for a given object function $f(x,y)$ as:

$$\lambda_{p,q} = \frac{(2p+1)(2q+1)}{4} \int_R \int_R P_p(x)\ P_q(y)\ f(x,y)dxdy \tag{1}$$

where $f(x,y)$ is assumed to have bounded support, the Legendre polynomials $P_p(x)$ are defined like:

$$P_p(x) = \frac{1}{2^p p!} \frac{d^p}{dx^p}(x^2-1)^p \tag{2}$$

In practice, the Legendre moments have to be computed from sampled data, i.e., the rectangular sampling of the original object function $f(x,y)$, producing the set of samples $f(x_i,y_j)$ with an (M,N) array of pixels. The piecewise constant approximation of $f(x,y)$ in (1), proposed by Liao and Pawlak [2] yields the following approximation of $\lambda_{p,q}$:

$$\hat{\lambda}_{p,q} = \sum_{i=1}^{M} \sum_{j=1}^{N} H_{p,q}(x_i,y_j) f(x_i,y_j) \tag{3}$$

where

$$H_{p,q}(x_i,y_j) = \frac{(2p+1)(2q+1)}{4} \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} \int_{y_j-\frac{\Delta y}{2}}^{y_j+\frac{\Delta y}{2}} P_p(x)\ P_q(y)dxdy \tag{4}$$

represents the integration of the polynomial $P_p(x)\,P_q(y)$ around the (x_i,y_j) pixel. This approximation allows a good quality of reconstructed images by reducing the reconstruction error [2], [7]. For this, this approximation will be adopted in the following sections.

III. Optimal Moment Selection Method

A. Object Description using Legendre Moment

The object function $f(x,y)$ can be approximated from $\hat{\lambda}_{p,q}$ up to a given order θ as truncated series [2]:

$$\hat{f}_\theta(x,y) = \sum_{p=0}^{\theta} \sum_{q=0}^{p} \hat{\lambda}_{p-q,q} P_{p-q}(x) P_q(y) \tag{5}$$

The number of moments used in the reconstruction of the object for a given order θ is defined by:

$$N_\theta = \frac{(\theta+1)(\theta+2)}{2} \tag{6}$$

Using N_θ moment, an object can be represented as a point in an N_θ-dimensional vector space. Determining the moment vector size depends only on the expansion order θ, which will be selected using MEP in the next section.

B. Optimal Subset Moments Selection using MEP

In this section, we determine the order of the truncated expansion of $f_\theta(x,y)$ which provides a good quality of the reconstructed object. The moments used in this reconstruction process will constitute the optimal subset for representing this object. For this, we introduce the Maximum Entropy Principle (MEP) to extract relevant moments that uniquely represent the pattern [3], [7], [8]. By applying the maximum entropy principle the optimal $\hat{p}_\theta^*$ is such that

$$S(\hat{p}_\theta^*) = \mathrm{MAX}\{S(\hat{p}_\theta)\ /\ \hat{p}_\theta \in G_W\} \tag{7}$$

where $\hat{p}(x_i,y_j)$ is the estimated probability density function obtained by normalizing $\hat{f}(x_i,y_j)$:

$$\hat{p}(x_i,y_j) = \frac{\hat{f}(x_i,y_j)}{\sum_{x_i,y_j \in \Omega} \hat{f}(x_i,y_j)} \tag{8}$$

and The Shannon entropy of $\hat{p}_\theta^*$ is defined as :

$$S(\hat{p}_\theta) = -\sum_{x_i,y_j \in \Omega} \hat{p}_\theta(x_i,y_j) \log(\hat{p}_\theta(x_i,y_j)) \tag{9}$$

The moment extraction algorithm uses the entropy as a measure of moment order selection, when sufficient information about the pattern have been recovered.

IV. Genetic Generation of Neural Networks

A. Genetic Representation of Neural Network

Neural network is widely used as a classifier in many handwritten character recognition systems. Representing the structure of a neural network is not as straightforward.

In this paper, we use the high level encoding of the Neural network structure due to its better scalability, desired regularities and biological plausibility [9]. The neural network is represented in a chromosomal string of values containing the number of nodes in each layer. The string has variable length which code the total layers number information. The search space of the GA is limited to a finite range of architectures exploring a maximum of two hidden layers.

B. Genetic Algorithm Operators

The basic GA operators are selection, crossover and mutation. The replacement strategy adopted and The choice of our evaluation function is addressed in the following sections.

1) Evaluation Function

The fitness function is used to selectively reproduce the most fit individual to produce new offsprings for the next generation. In order to give more chances to low neural network structures, we introduce a biais towards smaller networks by allocating the more training time the smaller the network. This approach have been introduced in [10]. A performance set of 5.000 samples is derived from the initial training set in order to evaluate the GA performance. That is, the fitness function is the Net's performance on evaluation set over the total number of epochs.

2) Selection scheme

Selection in GAs aims at giving higher probabilities to fitter individuals in a population so that they can produce hopefully fitter offspring. The genetic algorithm used in our experiments is based on the following strategy: after all creatures have been evaluated and assigned a score, a set of elite creatures (having highest scores and representing 20% of the population) is directly copied in the next population. This strategy ensures that best creatures are always duplicated in the next generation. The rest of the produced population is then generated by crossover using rank-based selection based on the cumulative probability for each chromosome. The generated offsprings replace their parents in the next generation after the application of our mutation operator.

3) *Crossover*

The essence of any crossover operator is to exchange components of the two parents to form new offsprings. In our work the one-point crossover is used, it's realized by cutting the strings at a randomly chosen position. The child is hence generated by taking one segment part from each parent. The genetic operators must produce correct and complete offsprings. That is, the choice of the cross point position is important in order to generate a valid offspring operating on the same interface units (input and output) as their parents. Fig. 1(b) illustrates an example of the proposed crossover operator mating two parent networks with different number of hidden layers. The offspring 1 and 2 inherit incoming hidden weights from parent 2 and 1 respectively, the weights of the other layers of the parent 1 and parent 2 are copied into the child 1 and 2 respectively with a deletion or an addition of connections. The added connections to the offsprings receive random weights.

4) *Architecture Mutation*

Often, the mutation is controlled by the mutation rate, which is very low for GAs in comparison with the crossover rate. In contrast, evolutionary programming often uses mutation almost exclusively. In this paper the mutation is performed more frequently than in traditional genetic algorithms. In [11] Xin yao proposes to apply mutation when the algorithm fails to improve the error of the parent network.

In this paper the mutation is applied following the three conditions:

1) If among the offsprings generated by crossover two child are similar to their parents.

2) At least two individuals in the current generated population are similar with respect to internal structure of the genotype.

3) If the fitness of the best individual in the current population is more than a fixed threshold: $f^j(ai) < \alpha 1$ or $f^j(ai)$ - $f^{j-1}(ai) < \alpha 2$. Where $f^j(ai)$ is the fitness of the best individual ai for the generation j. The mutation is applied, according to previously cited conditions, to (a) one of the similar individuals in the population or (b) the best individual in a population where the fitness is great than a fixed threshold.

In this paper the mutation is applied by randomly add an arbitrary value to the string element corresponding to hidden nodes, the sign of the generated value practically corresponds to deletion or addition of nodes in the parent structure. The deletion of a node involves the complete removal of the node and all its incident connections. The addition of nodes in the parent structure is simulated trough splitting existing nodes into the resulting nodes in the mutated structure. Xin yao in [11] limits to two, the number of mutated hidden nodes, here we present a general procedure to an N number of mutated hidden nodes: if an existing node i is splitted into N nodes the weights of the new nodes have the following values:

$$w^m_{ki} = w_{ki} \qquad \text{for} \quad j<i \quad \text{and } m=1..N$$

$$w^1_{ki} = (1+\frac{N}{2}\beta) w_{ki} \qquad \text{for } k>i$$

$$w^m_{ki} = -\frac{N}{2}\beta w_{ki} \qquad \text{for} \quad k>i \quad \text{and} \quad m=2..N$$

where w_{ki} is the weight vector of the existing node i, w^m_{ki} is the weight vector of the new m^th node produced by splitting, and β is a mutation parameter taken in our case as $|\beta| \prec 1$.

5) *The proposed Genetic Algorithm*

1. Generate initial population of size $N_p = 10$.
2. Evaluate each member of the population.
3. Apply the selection operator by the Rank-based method.
4. Save the elite creatures for the next generation.
5. Apply the crossover to the selected population.
6. Possibly Mutate with respect to the previously cited conditions.
7. Replace the parents by the produced offspring.
8. Evaluate the offsprings.

Repeat step 3 to step 8 until an acceptable network has been produced or N_g generation has been reached.

V. Experimental Results

In this paper, multilayer feedforward neural network (MFNN) is used to classify the patterns. In our algorithm, The stochastic gradient algorithm as a minimization procedure, is used during the learning phase. The input of the MFNN are feature vectors derived from the proposed feature extraction method described in section.III The number of nodes in the output layer is set to the number of digit classes. The networks in the GA starting points were conducted using the initial weight vectors that have been randomly chosen from a uniform distribution in (-1,1). The initial weights vectors in the next generations is inherited from parents. The method is tested using the MNIST database of handwritten digits. Table I shows optimal orders θ obtained by our moment extraction algorithm.

TABLE I Some digits in the feature subset database with the corresponding optimal Moment order

Digit	0	1	2	3	4	5	6
θ	10	9	9	10	10	9	9

In this study, the classifier error rate τ (%) is considered as the number of misclassifications in the training (test) phase over the total number of training (test) images.

Several fixed topologies were tested, figure 1 (a) shows the behavior of the error rates on the test set during the training phase. To start the evolution process we designed 10 different nets randomly chosen. These Nets have various numbers of units and hidden layers (HL) see Table II, We restrict the research space by taking the maximum hidden layers equal to two.

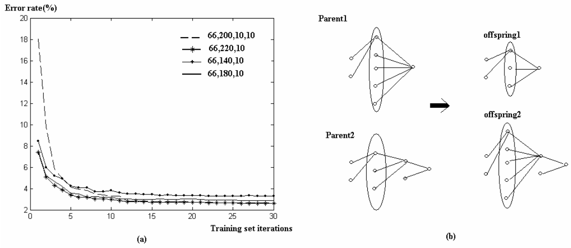

Figure 1 (a) Error rates on the test set of the MFNN versus 30 iterations through 60.000 MNIST pattern training set with fixed topologies. For example the name 66,220,10 indicates 66 input units, 10 output units and 220 hidden units. (b) Example of crossover applied on two MFNN individuals with different number of layers.

TABLE II CHARACTESTICS OF THE STARTING NETS OF THE PRESENTED ALGORITHM.

	Net1	Net2	Net3	Net4	Net5	Net6	Net7	Net8	Net9	Net10
Units	256	286	276	236	176	86	266	326	296	226
H.L.	2	3	2	3	3	2	3	3	2	2

In order to accelerate the evolution process, we don't use the complete training set, instead of that 20.000 samples were chosen for training and 5.000 examples for GA evaluation. For each session, The global learning rate η was decreased using the following schedule: 0.01 for the first iteration, 0.005 for the next three, 0.002 for the next seven and 0.0005 thereafter. The mutation parameter is set to 0.1.

The achieved rates on the evaluation set during the evolution process are shown in Fig. 2(a). Only the values of the best and the worst nets of each generation are represented. The best net reaches a classification performance of 4.98% misclassified patterns with reduced number of epochs. The development of the network's complexity of the best individual is demonstrated in Fig. 2(b) .

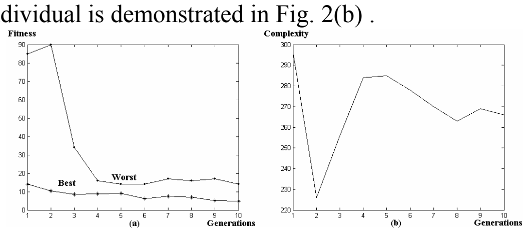

Fig. 2 (a) The achieved classification performance during the evolution process. (b) the behavior of the network complexities versus the generations.

It should be noted that :

1) There is a great enhancement in the behavior of the generated networks with respect to the fitness of the best and the worst individuals in each population.

2) The difference between the best and the worst individuals tends to become smaller through the generations.

3)The algorithm gives more powerful individuals with networks having small complexity.

Table III compares the results of the nets created by evolution with the results of the fixed architectures when trained with the complete MNIST training set. The table represents the achieved classification performance with respect to the training set, and the test set. Furthermore, it shows the total number of units in order to compare the network's complexity. We can see that the adapted nets are more efficient although they are smaller. They train faster and generalize better. To achieve a similar classification performances with the fixed nets, many additional passes would be necessary.

TABLE III: COMPARISON BETWEEN THE FIXED AND THE GENERATED NETWORKS TRAINED WITH THE WHOLE TRAINING SET.

	Units	Training rate (%)	Test rate (%)
Fixed nets			
66,140,10	216	2.23	3.21
66,180,10	256	2.16	2.86
66,200,10,10	286	2.01	2.61
Evolved Nets			
Net1 (gen_6)	278	2.09	2.54
Net1 (gen_8)	263	1.91	2.11
Net2 (gen_9)	269	1.98	2.21
Net1 (gen_10)	266	1.54	1.87
Net2 (gen_10)	263	1.61	1.98

VI. CONCLUSION AND OUTLOOK

A new genetic learning algorithm is proposed to generate a near optimal feedforward neural networks dynamically for the task of handwritten digit recognition. The training process changes network architectures and weights by efficient crossover and mutation operators. In order to test the generalization ability of the proposed algorithm, we plan to implement this method to further tasks, especially for face detection which is difficult problem in the context of artificial intelligence.

REFERENCES

[1] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transaction on Information Theory*, vol. 8, no. 2, pp. 179-187, 1962.

[2] S. X. Liao and Miroslaw Pawlak, "On image analysis by moments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 254-266, 1996.

[3] H. Qjidaa and L. Redouane, "Robust line fitting in a noisy image by the method of moments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1216-1223, 1999.

[4] W. H. schiffmann and K. Mecklenburg, "Genetic Generation of Backpropagation Trained Neural Networks," *Proc. of Parallel Processing in Neural Systems and Computers(ICNC)*, Eckmiller R. et al. (Eds.) pp. 205-208, Elsevier, 1990.

[5] G. Miller P. M. Todd and S. U. Hegde, Designing Neural Networks using Genetic Algorithms, *Proc. Of the third Intern. Conference on Genetic Algorithms (ICGA)*, San Mateo (CA), 1989, pp. 379-384.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no.11, pp. 2278-2324, November 1998.

[7] H. El Fadili, K. Zenkouar and H. Qjidaa, "Lapped Block Image Analysis Via the Method of Legendre Moments," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no.9, pp. 902-913, August 2003.

[8] X. Zhunang, R. M. Haralick, and Y. Zhao, "Maximum entropy image reconstruction," *IEEE Trans. Signal Processing*, vol. 39, no. 6, pp. 1478-1480, 1991.

[9] D. Parisi, A.Cangelosi and S. Nolfi, "cell division and migration in a genotype for neural networks," Network: computation in neural systems, vol. 5, no. 4, 1994.

[10] D. Whitley, T. Starkweather, and C. Bogart, "Genetic algorithms and neural networks: optimizing connections and connectivity," Parallel Computing,vol. 14, pp. 347-361, 1990.

[11] Y. Liu and X. Yao (1996), "A population-based learning algorithm which learns both architectures and weights of neural networks," *Chinese Journal of Advanced Software Research* (Allerton Press, Inc., New York, NY 10011), vol. 3, no. 1, pp. 54-65, 1996.