

## GENETIC ALGORITHM FOR NEURAL NETWORK PARAMETERS OPTIMIZATION

Yelena Gambarova

INTEGRIS Company, Baku, Azerbaijan  
YGambarova@azuni.net

**Abstract:** This study investigates the effectiveness of the genetic algorithm evolved neural network classifier and its application to the classification of remotely sensed multispectral imagery. Our methodology adopts a real coded GA strategy using datasets in a series of experiments that evaluate the effects on network performance of different choices of network parameters. The genetic operators are carefully designed to optimize the neural network, avoiding premature convergence.

### 1. DATA USED AND METHODOLOGY

This study was carried out in Gobustan, located between the southern outcrops of the Caucasus Mountain range and the Caspian Sea, some 60 km south of the capital Baku.

SPOT5 images in 2.5m and 5m resolutions, acquired between 2004 and 2007 were used for the delineation and classification of rare vegetation communities (Figure 1).



Figure 1. SPOT5 Images

Artificial neural networks (ANNs) were used for rare vegetation communities' classification using remotely sensed data. The genetic algorithm was used to optimize the parameters within the neural network. The most common parameters to optimize are the number of hidden neurons, the learning rates and momentum.

### 2. EXPERIMENTAL DESIGN

#### 2.1. Parameters used in the Neural Network (ANN)

Multilayer Perceptron (MLP) classifier was used for rare vegetation classification [1]. The input layer consisted of 3 neurons, corresponding to three spectral channels of SPOT satellite scanner: the green, red and near infrared (NIR) channel. The hidden layer had 25 neurons and the output layer had 5 neurons. An activation function was hyperbolic tangent. The backpropagation algorithm was used for neural network training.

A network structure of 4-25-5 was trained with the parameters listed in Table 1.

Table 1. Parameter settings for training neural network

Parameters	Value
Initial weight range	[0, 0.05]
Number of input nodes	3
Number of hidden nodes	25
Learning rate between input and hidden layers	0.5
Momentum term between input and hidden layers	0.7
Momentum term between hidden and output layers	0.7

## 2.2. Parameters used in the Genetic Algorithm (GA)

According to the evolutionary metaphor, a genetic algorithm starts with a population (collection) of individuals, which evolves toward optimum solutions through the genetic operators (*selection, crossover, mutation*), inspired by biological processes [2]. Each element of the population is called chromosome and codifies a point from the search space. The search is guided by a fitness function meant to evaluate the quality of each individual. The efficiency of a genetic algorithm is connected to the ability of defining “good” fitness function. The parameter settings for GA are presented in Table 2.

Table 2. Parameter settings for the genetic algorithm

Parameters	Value
Crossover probability	0.8
Mutation probability	0.07
Population size	20
Number of generations	70

The chromosomes will codify the network dimension (number of layers, number of neurons on layer) and the fitness function will integrate the training error and the entire number of neurons of the network. The fitness of an individual is determined by the total MSE [3]. The higher the error, the lower the fitness.

Select parents for reproduction based on their fitness. A *roulette wheel selection* scheme is adopted in our experiments [4]. The population of current generation is mapped onto a *Roulette wheel*, where each chromosome is represented by a space that proportionally corresponds to its fitness. Apply search operators in conjunction with the crossover and/or mutation operators, to parent chromosomes to generate offspring, which form the next generations. An asexual reproduction operator, an *One-point crossover operator* and a *Uniform mutation operator* is applied in the experiments of this article.

## 3. EXPERIMENTAL RESULTS

The algorithm of local optimization is founded on selecting and changing, step by step, the values of the various parameters: only one parameter at a time is changed, maintaining unchanged all the others. Particularly, after starting from a solution determined with the genetic algorithm, the first parameter, with a preset discretization step, was varied in all its field of variability and a “better” value of it was determined (i.e. the value which allows to obtain a greater percentage of coverage). Then, the second parameter was varied and so on. After having examined all the parameters, the algorithm started again from the first one. The algorithm stops as soon as a termination criterion was satisfied. When the network terminates due to one of these criteria, the best parameters (those that produced the lowest cost) were automatically loaded into the network.

The Table 3 displays each of the optimized parameter settings for each training pass, along with the fitness (the best average cost of the training). The Figure 2 displays *Non-Optimized*

MLP Learning Curve (Average Cost per Epoch) and Figure 3 displays Optimized MLP Learning Curve (Average Cost per Epoch).

Table 3. Optimization of parameters of Artificial Neural Networks by Genetic Algorithm

Generation	Chromosome	n supervised & non. rows	ropGradient. stepSize	hidden & non. rows	Fitness
0	0	24	0.100000	32	0.019918
0	1	4	0.047296	37	0.019362
0	2	27	0.213115	32	0.019340
0	3	8	0.089977	41	0.021533
0	4	7	0.390141	36	0.020039
0	5	16	0.074156	10	0.019945
0	6	7	0.071713	23	0.020418
0	7	28	0.495828	40	0.019655
0	8	25	0.397872	36	0.019481
0	9	13	0.349958	3	0.021043
0	10	17	0.350784	34	0.019756
0	11	18	0.216103	9	0.023240

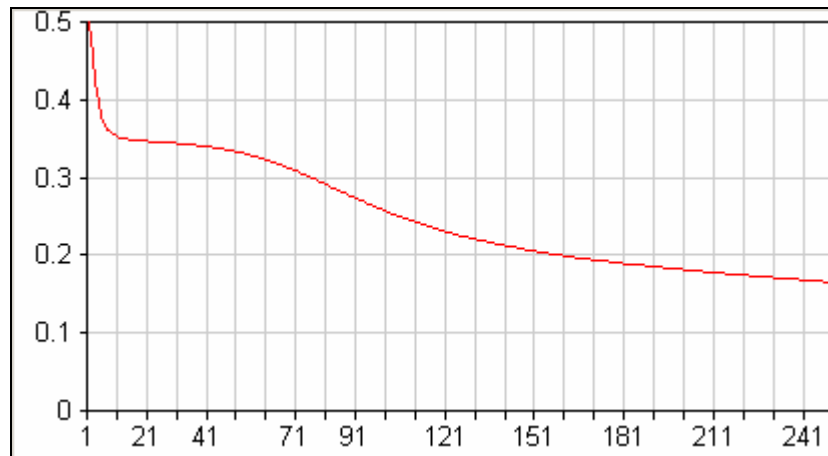


Figure 2. Non-Optimized MLP Learning Curve (Average Cost per Epoch)

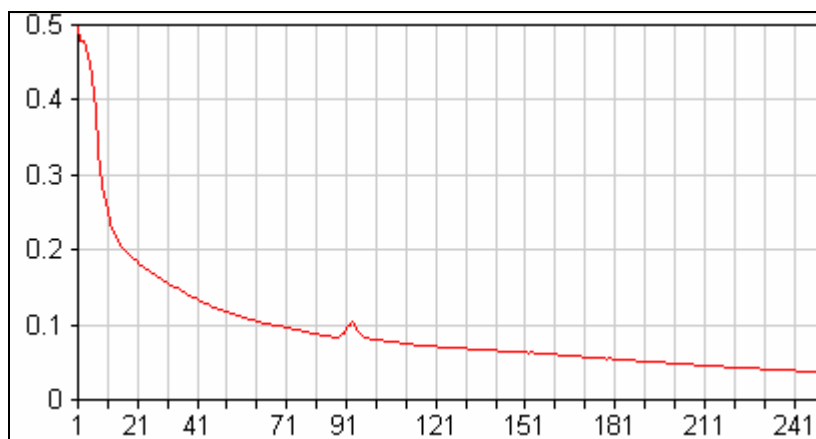


Figure 3. Optimized MLP Learning Curve (Average Cost per Epoch)

We compiled a confusion matrix with the results of recognition of samples from training sets and estimated the common degree of correctness (*CDC*) by the following formula:

$$CDC = 100\% \times (N_{CCS} / N_{total})$$

Where,  $N_{CCS}$  is the number of correctly classified samples;  
 $N_{total}$  is the total number of samples

Experimental results have shown that GA algorithm-based neural network classifier has better overall classification accuracy (measured in common degree of correctness) of the MLP:

$$CDC = 93.24\%$$

#### **4. CONCLUSION**

Before using the Neural Networks, it is necessary to define its structure, the number of hidden layers and the optimal number of neurons in each layer. To solve this problem and to build a NN with optimal structure we proposed in this work a technique, which uses GA.

This approach uses genetic computing for the establishment of the optimum number of neurons on layer, learning rate and momentum, for a given problem of classification of remotely sensed multispectral imagery.

#### **References**

1. E. Gambarova, H. Hasanov and T. Suleymanov. An investigation of the design and use of artificial neural networks in the classification of remotely sensed images. The Second International Conference "Problems of Cybernetics and Informatics", Baku, Azerbaijan, September 10-12, (2008), pp. 171-174.
2. C.Coello. An Updated Survey of Evolutionary Multiobjective Optimization Techniques. State of the Art and Future Trends, In 1999 Congress on Evolutionary Computation, Washington, (1999).
3. I.Ileană, C. Rotar, A. Incze. The Optimization of feed forward neural networks structure using genetic algorithms. Proceedings of the "International Conference on Theory and Applications of Mathematics and Informatics (ICTAMI 2004)", Thessaloniki, Greece, (2004), pp. 223-233.
4. D. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, New York, (1989).