

NEURAL NETWORK SYNTHESIS BASED ON EVOLUTIONARY OPTIMIZATION

A.A. OLIINYK, S.A. SUBBOTIN

The evolutionary approach for neural network structural synthesis is considered in this paper. The new method of multimodal evolutionary search with a chromosome clustering is offered. The developed method is based on the idea of simultaneous search of several optimums, thus chromosomes are grouped in clusters on their arrangement in a search space. So stable subpopulations in different clusters are formed, diversity of search is provided, and convergence to different local minima is reached that allows to find closer to optimal architectures of neural networks. Software implementing proposed method is developed. The experiments with proposed method in practical problem solving were conducted.

INTRODUCTION

Nowadays neural networks, with their ability to derive meaning from complicated or imprecise data, are widely used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques [1, 2].

It is known that the architecture of a neural network determines its information processing capability [3]. So architecture design has become one of the most important tasks in neural network research and application.

The architecture of a neural network includes its topological structure, the transfer and discriminant function of each node in the network.

Before present times architecture design is still a human expert's job. It depends heavily on the expert experience and a tedious trial-and-error process. There is no systematic way to design a near-optimal architecture for a given task automatically.

The synthesis of neural network is concerned with the optimization of some criterion like a sum of squared error. However, solving of this optimization task is engaged with problems caused by high dimension of training sample, multiextremeness of criterion function, nondifferentiability of activation functions [2], that complicates or makes impossible application of traditional optimization methods [1].

Research on constructive and destructive algorithms represents an effort towards the automatic design of architectures. A constructive algorithm starts with a minimal network (network with minimal number of hidden layers, nodes, and connections) and adds new layers, nodes, and connections when necessary during training while a destructive algorithm does the opposite, i.e., starts with the maximal network and deletes unnecessary layers, nodes, and connections during training. However, such structural hill climbing methods are susceptible to be-

coming trapped at structural local optima. In addition, they only investigate restricted topological subsets rather than the complete class of network architectures [3].

For the synthesis of neural network it is expedient to use methods of evolutionary search that are a family of computational models inspired by evolution. These methods differ from more traditional optimization techniques in that they involve a search from a population of solutions, not from a single point. Each iteration of an evolutionary method involves a competitive selection that weeds out poor solutions. The solutions with high fitness are recombined with other solutions by swapping parts of a solution with another [4].

However, the result of evolutionary optimization is the set of equal or few distinguished decisions. Therefore, the optimum structure of neural network can be not found because classical evolutionary methods can not uniformly cover search space, and large areas in space of variables can appear not investigated for the limited amount of iterations.

Therefore, the purpose of this work is a development of a multimodal method of evolutionary search which raises a diversity of a population and allows to cover in regular more intervals space of search which result is a set of various decisions (structures of neural network), that allows to choose architecture of neural network, in the best way satisfying external criteria.

PROBLEM STATEMENT

Let A be a maximal allowable quantity of neurons in the network, $\langle X, Y \rangle$ is a sample of training data, where $X = \{X_i\}$ is a set of feature values describing considered object or process; $Y = \{y_p\}$ is a set of target values; $X_i = \{x_{ip}\}$ is an i -th feature in the sample, $i = 1, 2, \dots, L$; x_{ip} is a value of i -th feature for p -th observation of the sample, $p = 1, 2, \dots, m$; y_p represents the value of the predicted parameter for p -th observation of the sample; L is a quantity of features in the sample; m is a quantity of observations.

So the problem of structural synthesis of neural network $NN = NN(C)$ can be formulated as a search problem $\xi = (NN, X, Y) \rightarrow \text{opt}$ in architecture space, where $C = C(L, A)$ is a matrix determining presence or absence of the connections between elements in the network NN , $\xi = (NN, X, Y)$ is an optimality criteria, e.g., lowest training error, lowest network complexity, etc.

STRUCTURAL SYNTHESIS OF NEURAL NETWORK BASED ON EVOLUTIONARY OPTIMIZATION

For application of evolutionary search for neural network synthesis it is necessary to determine a scheme of representation of network structure in a chromosome and to choose a fitness-function for estimation of chromosomes.

There are following methods of encoding of the information on neural network structure in chromosomes [3]: direct encoding, parametric representation, developmental rule representation, fractal representation, etc. One of the most effective encoding method is a direct encoding at which presence of each possible connection is directly described in a binary matrix of connections C , where value c_{ij} corresponds to presence ($c_{ij} = 1$) or absence ($c_{ij} = 0$) of connection from i -th to j -th neuron. Thus, the neural network is represented as a connectivity matrix.

The chromosome (Fig. 1) in direct encoding scheme is represented by the bit line containing the information about presence of connections.

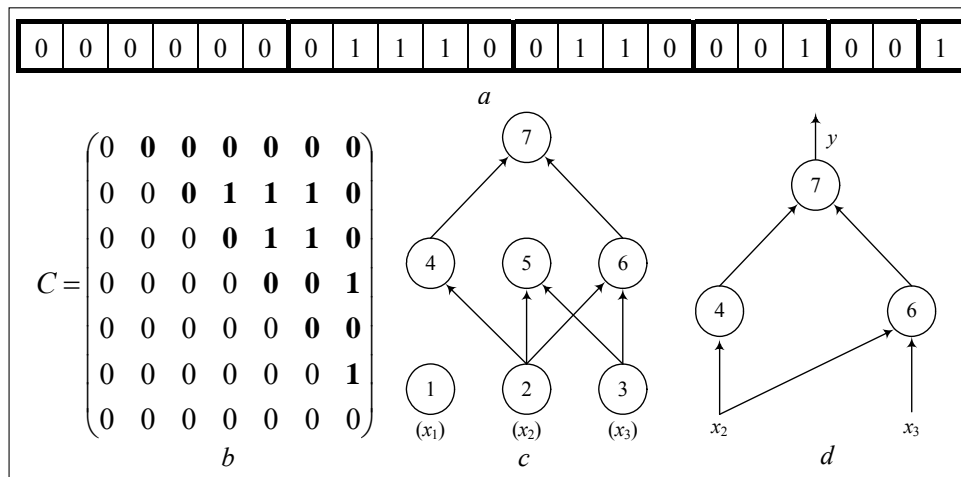


Fig. 1. An example of a chromosome and its decoding: a — chromosome, b — connectivity matrix, c — graph (architecture), d — synthesized neural network

Chromosome decoding to the structure of neural network occurs as follows.

Step 1. Generate connectivity matrix (Fig. 1, b) of neural network, corresponding to a chromosome (Fig. 1, a).

Step 2. Construct a graph (Fig. 1, c) based on the connectivity matrix.

Step 3. Synthesize a neural network (Fig. 1, d) on the basis of the graph constructed on the previous step, having removed thus neurons, not having target connections with neurons of the subsequent layers.

In case of need a choice of neuron activation function at structural synthesis it is possible to enter in a chromosome the additional genes containing the information of a kind of activation function for each neuron.

Structural synthesis of neural network based on the evolutionary approach can be executed as the following sequence of steps [5–8].

Step 1. Generate the initial population of chromosomes containing the information of network’s structure.

Step 2. Compute the fitness of each chromosome in the current population.

Step 2.1. Decode each chromosome in the population into architecture of the neural network.

Step 2.2. Train each neural network by the chosen rule using the data from training sample.

Step 2.3. Calculate value of the fitness-function considering a training error and complexity of constructed neural network.

Step 3. Check up search termination criteria. In the case of their satisfaction go to a step 7.

Step 4. Select the most fitted chromosomes for their crossing and mutation.

Step 5. Execute crossover and mutation operators on chromosomes selected earlier.

Step 6. Create new generation from the chromosomes obtained on the previous step and the elite chromosomes of the current generation. Go to a step 2.

Step 7. End.

A MULTIMODAL EVOLUTIONARY METHOD FOR STRUCTURAL SYNTHESIS OF NEURAL NETWORKS

It is shown in [3] that the surface of performance function for neural networks structural synthesis is nondifferentiable, noisy, complex and multimodal since different architectures may have similar performance.

The result of usage of classical evolutionary methods is the population of a few distinguished solutions therefore the found decision can appear a local optimum of multiextreme function. Such a decision (structure of neural network), as a rule, is inefficient at its usage in practice.

Therefore, for structural synthesis of neural networks it is expedient to use evolutionary methods capable to find several suboptimum decisions. The main problem of usage of traditional evolutionary methods for optimization of multimodal functions is a premature convergence to a local optimum. For overcoming this problem two groups of methods are developed: avoid strategies and repair strategies [9–13].

In avoid strategies method, the main idea is to prevent premature convergence to a local optimum [9–12]. The algorithms attempting to slow down genetic convergence aim at maintaining the population's diversity for a longer period and thereby avoid stagnation in a local optimum. Algorithms in this category either use a replacement scheme for updating the population or try to reduce the spread of genes by introducing a spatial population topology. The strategies trying to prevent overlap of solutions using penalty functions for reduction of the probability of occurrence in a population of similar solutions that attracts necessity of penalty function calculation for each chromosome in the population, hence, considerably slows down process of evolutionary search.

In repair strategies method, algorithms either maintain diversity by mass extinction techniques or by introducing new genetic material when population convergence is detected, that also demands significant time expenses [9, 13].

In the developed method of multimodal evolutionary search with chromosome clustering it is offered to group solutions (chromosome) in cluster on their arrangement in a search space.

The suggested method during evolutionary search defines the groups of similar chromosomes and raises a variety of a population by reducing the of fitness function values of chromosomes depending on a closeness to the center of their group.

The developed polymodal evolutionary search with the chromosome clustering assumes the execution of the following steps.

Step 1. Set: the quantity of optimums (the quantity optimum architectures of neural network) k which is required to be found during evolutionary search; N represents the quantity of chromosomes in a population, $N \gg k$.

Step 2. Set the counter of iterations: $t = 1$.

Step 3. Set the quantity of elite chromosomes: $k_e = k$.

Step 4. Initialize an initial population with chromosomes H_i ($j = 1, 2, \dots, N$) with length L (the quantity of features).

Step 5. Calculate the fitness function value $f(H_j)$ for each chromosome H_j .

Step 6. Group chromosomes in k clusters based on their fitness function values and an arrangement in an architecture space.

Step 6.1. For each chromosome H_j calculate Hamming distance [2] to all other chromosomes in a population. Hamming distance d between chromosomes H_j and H_l is calculated by the formula:

$$d = \sum_{u=1}^L |h_{ju} - h_{lu}|,$$

where h_{ju} and h_{lu} are the values of genes of chromosomes H_j and H_l , respectively

Step 6.2. Set the counter of generated clusters: $m = 1$.

Step 6.3. Choose a chromosome with the best fitness function value as the center of m -th cluster. Thus the chromosomes which yet have been not grouped in clusters are considered.

Step 6.4. Add in cluster $(N/k - 1)$ chromosomes nearest on Hamming distance to a chromosome, being the center of current m -th cluster.

Step 6.5. If $m = k$ then go to a step 7.

Step 6.6. Set: $m = m + 1$. Go to a step 6.3.

Step 7. Reduce fitness function values of the chromosomes which are not being the best in cluster using the formula:

$$f_{n,j} = \left(\frac{d_j}{d_{\max,j}} \right)^s f_j,$$

where f_j is the fitness function value before j -th chromosome changing; $f_{n,j}$ is a new fitness function value of j -th chromosome; d_j is a Hamming distance from j -th chromosome to the center of its group; $d_{\max,j}$ is a maximal Hamming distance in the cluster of j -th chromosome; s is the parameter determining a degree of fitness function reduction of chromosomes, not being the centers of cluster, $s \geq 1$.

Step 8. Apply crossover and mutation operators.

Step 9. Generate new population. Thus the best (elite) chromosomes in every cluster are guaranteed pass in the new generation.

Step 10. If $t = T$ (T is the maximum possible quantity of iterations), then go to a step 13.

Step 11. Set: $t = t + 1$.

Step 12. Go to a step 5.

Step 13. Estimate each of k the chromosomes being the centers of clusters, with the help of the data of test sample. Choose the best chromosome. Neural network, corresponding to such chromosome, is the solution.

Step 14. End.

The developed method of multimodal evolutionary search with chromosome clustering raises a variety of a population and allows to cover in regular more intervals search space, raising thus an opportunity of search of a global optimum and increasing probability of successful execution of estimation procedure of the founded solutions with the help of the external criteria on test sample.

EXPERIMENTS AND RESULTS

The suggested method of multimodal evolutionary search with the chromosome clustering has been realised as computer program. The experimental research of the offered method of neural network synthesis was carried out based on the decision of a vehicle classification problem by 2d gray-scale images.

The initial sample contained the transformed graphic representations of vehicles received from video cameras at streets in Zaporozhye, Ukraine. Sample consisted of 1062 vehicle images, each of which was characterized by 4096 features representing normalized values of the image points intensity projected on a sensor matrix of 64×64 pixels. Using these 4096 features there were calculated 26 generalizing features. Vehicles were classified on cars, minibuses, motorcycles, trucks and buses. For each class of transport the model was constructed. Thus the problem has consisted in synthesis of four classification models of each type of vehicles based on 26 generalizing features [14].

The coding of a potential solution was performed like at Fig. 1. There are evolutionary operators were used: roulette wheel selection, uniform crossover, simple mutation. The initial parameters of all evolutionary methods were established by the following: population size $N = 100$; crossover probability $p_{cr} = 0,8$; mutation rate $m_r = 0,02$. Stopping criteria: maximum number of iterations $T = 100$; achievement of comprehensible value of the fitness function equal to 0.01.

The purpose of the experiments was to synthesize the optimal neural network model. The maximal allowable sum of squared error on this model for training data and test sample is 0,01 and 0,02, respectively. Results of experiments for different quantity of clusters are presented in the table 1 (SSE_{ls} represents a sum of squared error for learning sample, SSE_{ts} is a sum of squared error for test sample, n represents the quantity of the obtained models providing sufficient value of sum of squared error for test sample).

Table 1. Results of Experiments

Quantity of clusters	SSE_{ls}	SSE_{ts}	n
1	0,0098	0,0272	0
2	0,0095	0,0227	0
3	0,0096	0,0183	1
4	0,0094	0,0142	1
5	0,0095	0,0137	2

The example of running different evolutionary methods for the decision of a vehicles classification problem is shown in Fig. 2.

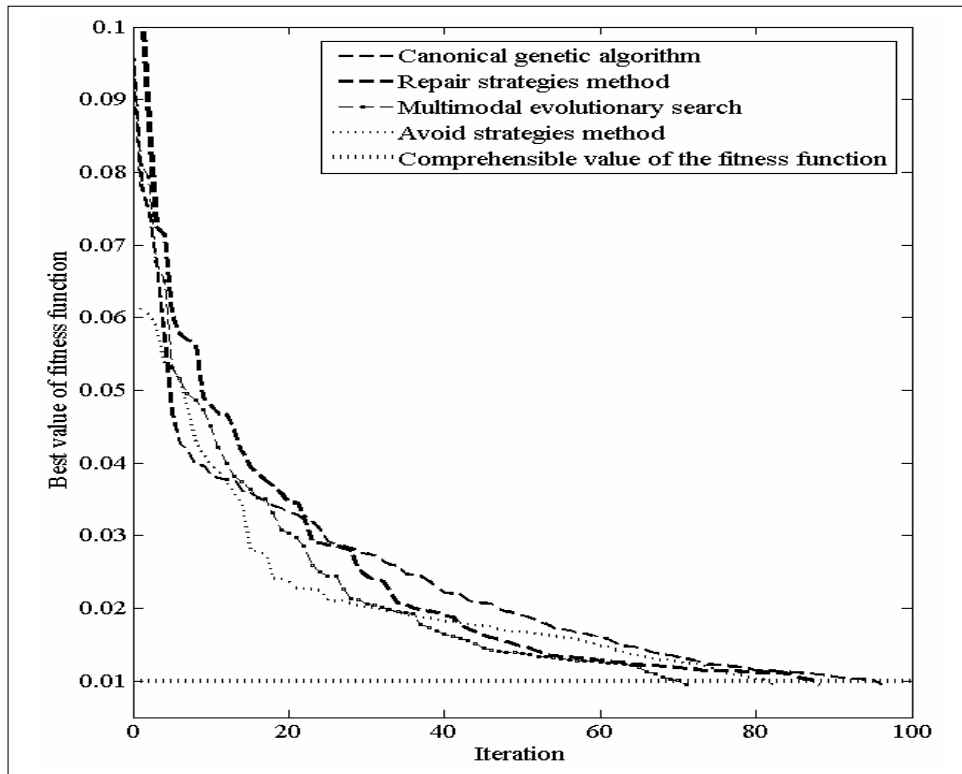


Fig. 2. An example of evolutionary methods running

The comparison of the proposed method with other evolutionary methods is presented in the table 2 (τ represents a time for evolutionary optimisation, f_{count} is the quantity of fitness function calculation).

The best solution (neural network structure) found by the proposed method is shown in Fig. 3.

Table 2. Comparison of Evolutionary Methods

Method	τ	f_{count}	SSE_{ls}	SSE_{ts}
Canonical genetic algorithm	709,3	9619	0,0098	0,0272
Avoid strategies method	587,9	8018	0,0095	0,0191
Repair strategies method	629,8	8714	0,0097	0,0158
Multimodal evolutionary search	521,7	7092	0,0095	0,0137

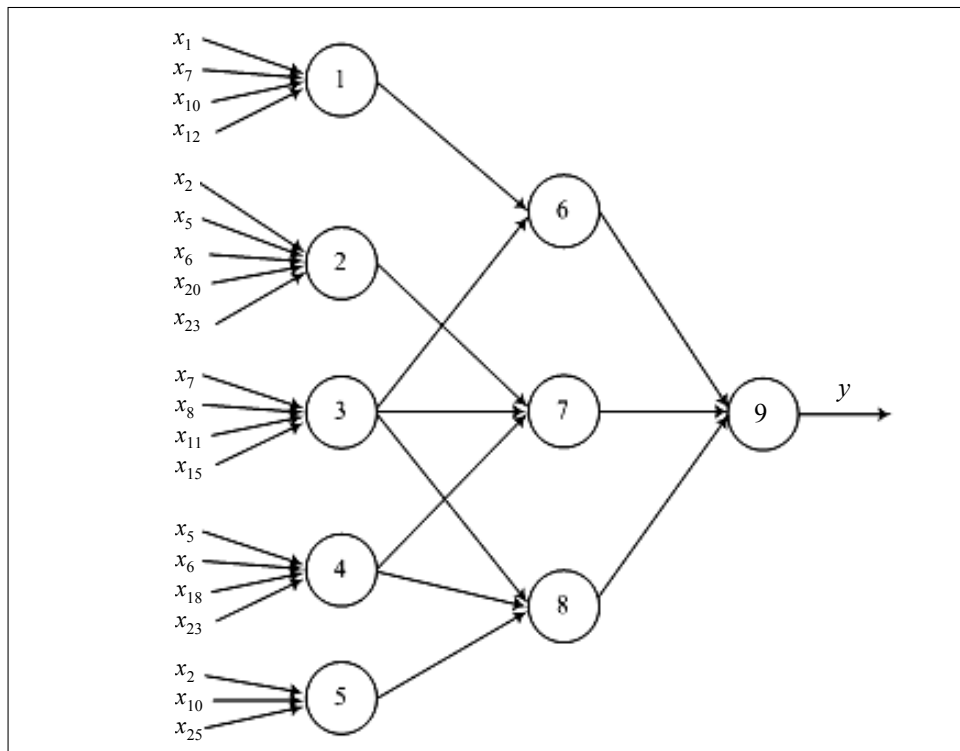


Fig. 3. The best solution (neural network structure)

The weight matrix of synthesized neural network is presented in the table 3, where μ represents the number of layer in the network; ρ is the number of neuron in the layer; b is the value of bias; node is the first node in the connection; w is the value of weight.

The experiments have shown that as a result of application of multimodal evolutionary search with the chromosome clustering stable subpopulations are formed in different clusters, heterogeneity of search is provided, and also convergence to different local minima is reached.

CONCLUSION

The new method of multimodal evolutionary search with a chromosome clustering is offered in this paper. The developed method is based on idea of simultaneous search of several optimums, thus solutions (architectures) are grouped in clusters on their arrangement in architecture space that results in more uniform covering of search space. Comparison of the results obtained with the help of the developed method with results of application of classical evolutionary methods shows that the offered method allows to synthesize closer to optimum neural networks because of more uniform covering of search space. Thus, the suggested method can be recommended for application in practice for solving different problems in pattern recognition and computational diagnosis.

Table 3. The Weight Matrix of Synthesized Neural Network

μ	ρ	b	node	w
1	1	-0,1852	feature x_1	-1,7264
			feature x_7	2,2621
			feature x_{10}	0,8386
			feature x_{12}	-0,1721
	2	-0,5291	feature x_2	-0,7502
			feature x_5	-1,1064
			feature x_6	2,1628
			feature x_{20}	1,8531
	3	1,2674	feature x_{23}	1,0372
			feature x_7	0,8580
			feature x_8	-0,6082
			feature x_{11}	-0,5810
	4	-0,9620	feature x_{15}	-1,1310
			feature x_5	0,5502
			feature x_{16}	1,3051
			feature x_{18}	1,2109
5	0,4281	feature x_{23}	-2,0945	
		feature x_2	1,0845	
		feature x_{10}	-1,6093	
		feature x_{25}	0,8803	
2	6	-1,6528	neuron 1	0,7726
			neuron 3	-0,6190
	7	2,9381	neuron 2	-1,5278
			neuron 3	-1,7429
	8	0,7804	neuron 4	0,9726
			neuron 3	0,7904
3	9	0,3174	neuron 4	0,4086
			neuron 5	-1,0462
			neuron 6	0,8054
			neuron 7	-1,7960
			neuron 8	0,9467

REFERENCES

1. Ripley B. Pattern Recognition and Neural Networks. — Cambridge: Cambridge University Press, 2008. — 416 p.
2. Yao X. Evolving Artificial Neural Network // Proceedings of the IEEE. — 1999. — **87**, № 9. — P. 1423–1447.
3. Haykin S. Neural Networks: A Comprehensive Foundation. — New Jersey: Prentice Hall, 1999. — 842 p.
4. Haupt R., Haupt S. Practical Genetic Algorithms. — Hoboken: John Wiley & Sons, 2004. — 272 p.
5. Siebel N.T., Kassahun Y. Learning Neural Networks for Visual Servoing Using Evolutionary Methods // Hybrid Intelligent Systems: Sixth International Conference, 13–15 December 2006, Auckland: Proceedings. — Los Alamitos: IEEE, 2006. — P. 6–14.

6. Rocha M., Cortez P., Neves J. Simultaneous evolution of neural network topologies and weights for classification and regression // Computational Intelligence and Bioinspired Systems. — Berlin: Springer, 2005. — P. 59–66.
7. Thammano A., Meengen A. A New Evolutionary Neural Network Classifier // Advances in Knowledge Discovery and Data Mining. — Berlin: Springer, 2005. — P. 249–255.
8. Siebel N., Krause J., Sommer G. Efficient learning of neural networks with evolutionary algorithms // Pattern Recognition: 29th DAGM conference on: proceedings. — Berlin: Springer, 2007. — P. 466–475.
9. Ursem R.K. Multinational Evolutionary Algorithms // Evolutionary Computation: CEC 99 Congress: proceedings. — Los Alamitos, IEEE, 1999. — P. 1633–1640.
10. Shimodaira H. A Diversity Control Oriented Genetic Algorithm (DCGA): Development and Experimental Results // Genetic and Evolutionary Computation. — Orlando: Morgan Kaufmann, 2000. — P. 603–611.
11. Thomsen R., Rickers P., Krink T. A Religion-Based Spatial Model For Evolutionary Algorithms // Parallel Problem Solving from Nature. — Berlin: Springer, 2000. — P. 817–826.
12. Tsutsui S., Fujimoto Y., Ghosh A. Forking Genetic Algorithms: GAs with Search Space Division Schemes // Evolutionary Computation. — 1997. — 5. — P. 61–80.
13. Ursem R.K. Diversity-Guided Evolutionary Algorithms // Parallel Problem Solving from Nature. — Berlin: Springer, 2002. — P. 462–471.
14. Олейник А.А. Выбор системы информативных признаков для классификации транспортных средств на основе эволюционного поиска // Комп'ютерне моделювання та інтелектуальні системи: збірник наукових праць / За ред. Д.М. Пізи, С.О. Субботіна. — Запоріжжя: ЗНТУ, 2007. — С. 134–146.

Received 14.02.2014

From the Editorial Board: the article corresponds completely to submitted manuscript.