Sergey Zamay, Victor Okhonin, Oleg Yakubaylik
E-mail – oleg@icm.krasn.ru

Chapter _. Neuronetwork and GIS

1. Neuronetwork: an extra dimension in GIS

Artificial Intelligence Technologies

One of the main GIS problems identified by many researches is to obtain new knowledge and ideas concerning the nature of spatial data. However, users often pay the attention to the data representation mainly, and GIS potentials which may contribute decision-making are underestimated. Geographic data merit in decision-making support systems is of great importance when GIS is connected to the software based on Artificial Intelligence (AI) methodology and technologies, which have recently become considerably widespread. The importance of such AI tools as expert systems and neuronetwork is that they essentially extend the range of practically significant tasks which could be solved by means of computers. Besides, a solution of these problems results in substantial economic profits.

The successful evolution of AI methods and technologies result in the implementation of numerous end-user-oriented applications including those designed for experts in GIS. The integration of AI and GIS systems is especially efficient in evaluation, control and decision-making problems. In this context, the development of AI technologies such as neuronetworks, evolutionary computations (independent and adaptive behavior of computer applications and robot-technical devices), fuzzy logic, self-organizing database management systems, image processing, expert systems and etc. is connected to the expansion of functional capabilities in decision-making support. There is a good reason to believe that AI elements will be integrated into the new generation of GIS software.

Trends for the Development of Analytic and Prognostic GIS Properties

GIS technologies combine the conventional database operations, such as queries and statistical analysis, with the advantages of visualization and geographic analysis provided by digital maps. In principle, GIS technologies are applicable for multidimensional data analysis in abstract spaces of an arbitrary dimension. It makes promising the adoption of these technologies to the problems of analysis and forecast of territory-distributed events and processes. The demand for a complex evaluation of a current situation, a revelation of main factors, a forecast of possible after-effects, strategy planning force to go out of the common problems that could be solved by conventional GIS technologies.

Both scientists and politicians admit this issue [5]. The Digital Future of the Planet Forum has mentioned 3 key problems here [6]. They are the following:
– a development of the detailed multi-level and multi-scale representation of the Planet;
– the problem of the development of an adequate 3D Earth description;
– the problems of processing of great amounts of heterogeneous environmental geodata;

Various geodata heterogeneous in the structure, contents and/or representation are used to describe the World adequately. These are the data on landscape, on natural resources, environmental, weather and climate data, data on the import and export, on infrastructure, demography, criminality etc. These multilevel data of different detailization levels which start with the lot, city and regions data and end in descriptions of countries, continents, climate or hydrodynamics of continental platforms.

The intelligent technologies based on neuronetwork methods are regarded as the most rapidly developing Information Technologies for analysis, structuring and systematization of these data. The event records, data on time dynamics of processes and on variation of object state are important to solve the prognostic problems. The processes and their description may be considered as objects in a multidimensional space. The object is described by its geographic coordinates and quantitative indices of the state vector. Investigation of the processes developing in time results in the fourth axis (namely the time axis) appearance in addition to the three spatial ones. Set of other multidimensional space axes can be used to determine a measure of proximity between various objects and process parameters. Such approach is fruitful to analyze cause-and-effect relationship of the events, functional relations between the objects and the forecasting of processes development. The neuronetwork models based on the statistic analysis of spatial and time series of geodata are suitable for estimation of future with extrapolation methods.

GIS are user-friendly, making the territory-oriented data input/output easy for man. The software may be expanded with the tools of multidimensional data processing in abstract spaces of an arbitrary dimension. The geodata may be stored either within GIS itself or outside it. A deep relation between geoinformation systems and databases, from the point of view of their format and technological nature provides a user with a natural access to the raw source data. The use of Internet resources provides an access to that latter located virtually elsewhere. Due to a complexity of the problem of storage, transfer and searching of great data volumes kept in electronic media, GIS usually employs external database management systems. As a rule, the data stored in databases requires specific preprocessing. Spatial analysis of raw data is impossible without that latter or it consumes too much resources. The neuronetwork algorithms seem to be the most efficient methods of preprocessing of great volumes of raw heterogeneous data. High parallelism of neuronetwork algorithms results in an efficient and simple implementation of them in structures of parallel computation.

GIS has much in common with automatic control systems (CADs, ACSs). GIS may become a convenient interface for accessing raw database data, electronic tables and sensors when a complex object to be managed is represented by a structure functional scheme. Basically, the geometric GIS nature yields both a development of the scheme and the content-analysis of structure and topology of interconnections in that latter. Implementation of modern geometry and statistics methods enables automated graph and network construction, its quantitative analysis of relationships, construction of extrapolation, interpolation and adaptive models of technological processes. Neuronetwork models are an example of multilevel patterns of almost any reasonable complexity that are made up of non-linear elements. Such patterns can be used for automated construction of graphs and technological networks which can be regarded as the bases for construction of more complex models including direct mathematical modeling of technological processes. One can even develop a statistic model of technological processes based on the data obtained due to neuronetwork methods.

The Integration of Neuronetwork and Geoinformation Technologies.

Though GIS contain the geodata describing the geometric (topological) properties of objects (usually dots, lines and sites), the functional capabilities of these latter in space analysis are relatively low. Mathematical tools for multidimensional space analysis are developing permanently being well-provided with the methods taking its origin in geometry, topology and other disciplines. That is why integration of geoinformation technologies and methods into the space analysis looks natural and promising.

The steps of integration are the following:

1. an expansion of functional capabilities of conventional methods, technologies and software for space analysis into GIS by exploiting the potentials of advanced mathematical methodology of multidimensional data analysis;

2. a development of new methods based on intelligent computer technologies. These methods are regarded as a foundation for new generation of more advanced convenient software tools for geodata analysis with an increased amount of raw data;
3. a development of new data models, information technologies and software which can be integrated with conventional GIS and are implemented specially for multidimensional data analysis and for modeling and forecasting of territory distributed processes.

In fact, GIS grow up in the first direction since the moment of their emergence. Two other directions are closely related to the fundamental investigations in the area of cross-fields of mathematics, computer science and neurophysiology. A wide class of statistical and adaptive methods for analysis of multidimensional data which are neuronetwork methods in nature has been developed lately. These methods are applicable not only to data analysis but (and that is essential) also to a construction of the models of processes running in multidimensional spaces. New interesting classes of non-linear models based on statistic analysis of raw data are proposed now, where the means of information technologies are used to access and preprocess the raw data stored in GIS and databases. The statistic and adaptive methods of geodata analysis improve the source data quality and allow development of a neuronetwork model adequate both to the source data purpose and quality, and expert opinions, as well as the aims of a research.

The resources necessary for such information models are mainly determined by the aims of a user and hardly depend on the raw data volume, though the analytical and forecasting properties of models grow up essentially as an amount and completeness of the source geodata increases. The capacity of world data resources describing the social, economy, demographic, ecological, climate, geological and other processes grows permanently. One faces an urgent problem what to do with all these 'treasures'. Client-server and Internet technologies provide a distributed access to the source data. The neuronetwork solutions of geodata analysis are well-parallelized providing researchers with the tools for development of prognostic models based on heterogeneous data. GIS technologies yield an excellent cartography interface. These three factors support strongly a solution of extremely complex and resource consuming models of social economy processes and environmental phenomena. The importance and value of problems determine a steady interest in development of neuronetwork methods and stable prospective for intelligent computing technologies in implementation of world-scale GIS models based on Internet data and computing resources.

2. Neuronetworks in biology and engineering.

Modern computers are based on von Neumann scheme which rapidly performs the extended chains of binary operations. It seems that this approach was partly determined by the structure of mathematics in the first part of the 20^{th} century when higher mathematics relied on arithmetic, and that latter is based on binary logic. Evidently, the things could go on in other way if something other had been taken for a basis of mathematics.

An idea to implement the principles of biological neuronetwork functioning competed to the von Neumann approach. Rosenblatt perceptron, the first neuro-like system, was developed nearly at the same time when the first computer has been done. Both tendencies, i.e. Rosenblatt and von Neumann, were developing independently for some time, then the perceptional direction went into a crisis and revived later in 80s as neuronetwork. On this new stage, the binary logic and bionic principles were combined. It should be noticed that the crisis in bionic direction was not resulted from technical shortage or a lack of applications but from a mathematical investigation carried out by Minsky and Papert. They have shown that neither perceptron exists able to determine reliably such topological parameter of an object as the connectivity; it followed in a decay of enthusiasm.

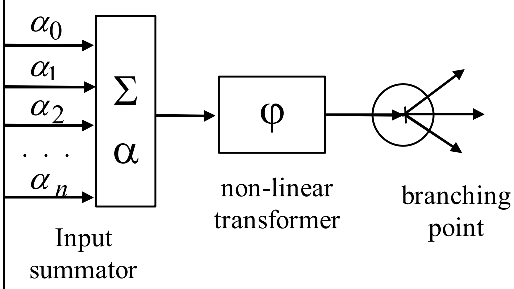
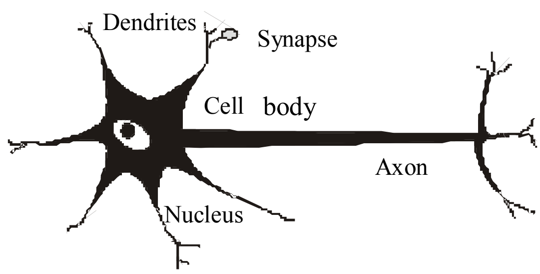
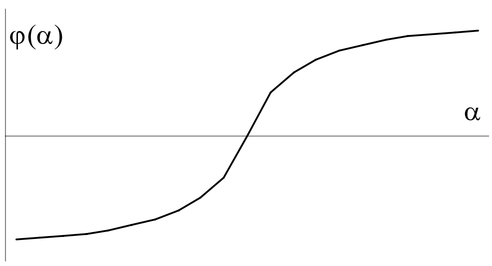
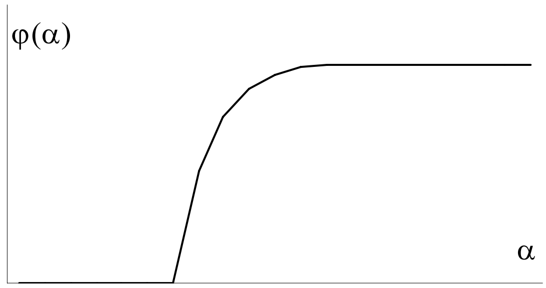
The table below presents a comparison between von Neumann computer and a biological neuronetwork.

	von Neumann computer	Biological Neuronetwork System
Processor	Complex	Simple
	High-speed	Low-speed
	One or few	Large in number
Memory	Separated from the processor	Integrated into the processor
	Localized	Distributed
	Not content-addressed	Content-addressed
Computations	Centralized	Distributed
	Successive [consecutive]	Parallel
	Stored programs	Self-training
Reliability	High vulnerability	Durability
Specialization	Numerical and symbolic operations	Perception problems
Operation environment	Strictly defined	Poorly defined
	Restricted	No limitations

Following are some parameters of a human brain: the cerebral brain cortex is formed by the neuron layer from 2 to 3 mm in depth, with total area about 2.2 dm^2 and contains approximately 10^{11} neurons, where each neuron connects 10^3 to 10^4 neurons.

Neurons interact each other with a series of short pulses of a few milliseconds (ms.) in duration each. The message is transmitted via impulse-frequency modulation. The frequency may vary from several peaks up to hundreds of Hertz which is a million times slower in comparison to the fastest switching electronic circuits. Nevertheless, man makes a decision concerning to data perception during several hundreds of ms.

Let us compare a biological neuron vs. the most commonly used technical neuron:

•	Technical Neuron	Biological Neuron
Scheme		
Response Function		

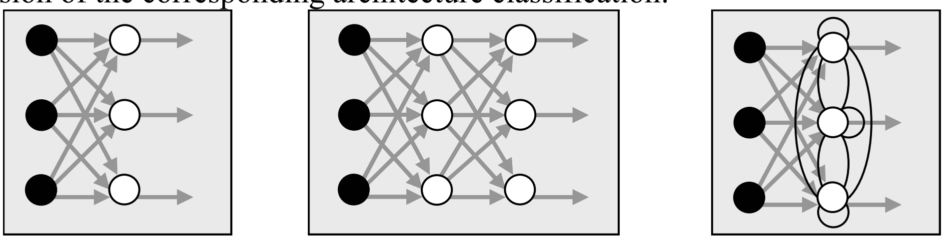
Neurons of both types respond to stimuli transmitted from many other neurons; the response level depends on the connections between the neurons.

Unlike technical neurons, the response of a biological neuron is always non-negative but the neuron doesn't respond if the signal fails to reach the critical level. Perhaps, one of the most evident differences between biological neuronetworks and modern neuroprograms has something to do with the following: the same brain may work in rather different ways depending on the currently "silent" neurons. It looks like a brain be a store of processors which are interconnected differently to solve different problems. The hardware implementing technical neurons was very diverse. At first it was relay circuits, now it is operational amplifiers but usually it is an emulation on an ordinary computer. Experts estimate capacity of the modern personal computers as capable to simulate nervous system of complex worms while the best neuronetwork specialized processors can simulate a fly.

The biological neuronetworks are separated in principal from the point of view of their inner structure: they can implement either reflex behavior or thinking. Neurophysiologically, a reflex behavior implies a relatively short activity peak in response to external signal and then relapsing into a steady state. Thinking means a long-term network processing, quite often accompanied with rather moderate but permanent brain excitation; the external signals usually bring a disturbance. Technical systems usually reproduce reflex behavior though one may compare some neuroalgorithms which solve “internally complex problems” to thinking processes.

Types of Technical neuronetworks

Some authors pay considerable attention to the architecture of technical neuronetwork. Below is a version of the corresponding architecture classification:



single-layer multi-layer fully connected

A separation of neuroalgorithms into two classes, which are Supervised (algorithm with teacher) and Unsupervised (teacher-free algorithms) ones seems to be more fundamental. A ‘learning’ is arranged as a reproduction of a set of correct samples (training textbook), in the first case; upon a completion of a training the network can respond adequately on the inputs not present in the textbook. There are no sets of correct reactions, initially, in the second case. Apparently, the supervised neuronetworks simulate a reflexive behavior rather fine. Sometimes the unsupervised neuronetworks can simulate a more interesting thing similar to thinking, while they do it much less successfully.

The supervised neuronetworks take the origin from perceptrons and could be considered as a version and modification of error back propagation networks (sometimes the supervised networks make a result of simplification of these latter). This class contains single- and multi-layer perceptrons, Boltzmann machine, the networks learning according to Hebb rule, recursive laminar and fully-connected error back propagation networks and the networks using radial basis functions. The differences in the above-mentioned systems are sometimes rather great but they always have much in common, even though the classification details differ from author to author.

The unsupervised neuronetworks seem to be more diverse though the theory behind them is mathematically more primitive. These are the Kokhonen maps, the systems with multiple local steady states (say, Hopfield networks) and the networks with resonant-adaptive adjustment. There are no apparent similarities between the classes. Finally, the combined approaches were developed though they are far from being widespread. Sometimes the methodology of such combination makes recall a slogan “Labor Made Man”. A supervised algorithm is taken as a basis which arbitrary establishes some initial, one dare say “abstract” correspondence between “raw” and “processed” data. Then the “processed data” is changed during the external “labor” with respect to "material properties" so the correspondence improves. A new correspondence between the raw and processed data is established, the processed data is changed again and so on. As a result, the neuronetwork generates on one side, an "artistic image" of the situation and on the other side it develops a skill of rapid, reflex, mapping the real data and its images. For example, connectedness of data image can be verified this way: if the "material nature" doesn't allow change of its connectedness and the approach mentioned above works, then the data images have the same connectedness. We have included this example to remind the crisis of the neuronetwork approach taken place before 80s due to the pessimism towards the possibilities of neuroalgorithms application in topology problems.

Finally, let enlist the problems which are the most peculiar for the neuronetwork approach:

– supervised learning, that includes image classification, function fitting, prediction, management, data analysis, intraclass categorization, data compression;
– unsupervised learning, that includes categorization, intraclass categorization, data analysis, data compression, associative memory.

3. Neuronetwork Algorithms: Mathematical Side.

Neuronetwork algorithms gather nowadays several approaches to data processing. The authors of these latter speculate independently a great resemblance in the organization of biological neuron systems and the patterns for data processing. No one knows exactly how the real neuron systems operate. It yields an equality of various speculations towards the principles of operation of neuron-like patterns simulating the work of that latter. These two issues bring a title for the new discipline called neuroinformatics. Such approach falls beyond the traditionally appreciated mathematical methodology of strict and precise foundation of an algorithm. High efficiency of some neuronetwork algorithms forces a student to accept the practice. Further, we shall discuss two types of neuronetwork algorithms used to be the most frequent in applications. The former is the back error propagation algorithms (BackProp), and the latter is the Kokhonen or self-organization maps (SOM).

Back Error Propagation Algorithm.

Originally the back error propagation neuroalgorithms were developed to solve the classic problem of mathematical statistics that is the problem of tabular data regression. Some simplest regression problems (such as the problem of a linear approximation) are well-known. Here a straight line must be settled and the algorithm must provide the parameters of the straight line, such as the angle of line inclination and the coordinates of a point of the line.

Neither a straight line nor a plane nor a hyperplane are used in case of a nonlinear regression. Smooth manifold must be used for an approximation of a set of points in this case. One has to increase the number of parameters to describe such smooth manifold, in comparison to the linear case; that is where the problem of multiparametric non-linear regression arises.

The complexity of multiparametric non-linear data regression algorithms was known long before the back error propagation algorithms appeared:

– the computation time increases as the number of parameters grows up;
– the regression parameters are determined ambiguously (that is so called low-conditionality of multiparametric regression problems);
– a choice of the best variant of the non-linear regression is not evident.

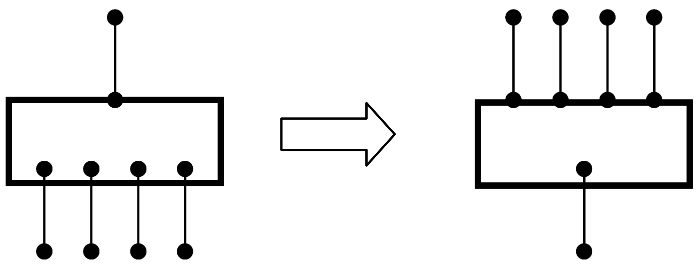
The back error propagation algorithm has successfully overcome the first problem of the mentioned ones above. Later, the experience relevant to the remaining two discrepancies has been gained. Meanwhile, there is no completed mathematical theory for these issues yet. One may say that the back error propagation neuroalgorithm solves efficiently the multiparametric non-linear regression problems. There are programs which regress efficiently tens and hundreds thousand of points and yield tens of thousand of regression parameters due to a reasonable time period.

The general multiparametric non-linear regression problem is defined as follows:

Suppose, one has a method to calculate the deviation of a regression manifold from the data points located in a space depending on the shape of the manifold. Thus, one can tell towards an estimation of deviation value H as a composite function usually defined implicitly. The function has the regression parameters β as its arguments. These latter define the manifold, in turn. One must find out the parameters β values which yield the minimum of $H(\beta)$.

One can easily give an example of algorithm solving this problem in some way: let the set of regression parameters consists of N elements. We take the first one and slightly change it. In case if H increases, we change the first element in another direction. Usually H decreases in this situation. Then we take the second regression parameter and change it. After N steps it is the turn of the first parameter again. The weak point of this approach is that the calculation time is proportionate to $N*T$ where T is the time needed to calculate H. The advantage of back error propagation method is that the total computation time is N times lower. That is a very significant decrease since sometimes N varies from hundreds to tens of thousands.

Let's illustrate the main idea of the method by a simple analogy. Suppose we have a system which consists of a tank and N input pipes of different flow capacity and only one output pipe. It is required to find out the flow capacities of all the input pipes. The slower method is to pour fluid into each pipe, one by one, and to see how it flows out. The N times faster method is that the system is inverted, and fluid is poured through the output pipe and speed of its flow from each in-pipe is noticed. Similar 'inversion' takes place in the back error propagation methods, which derive their name from this technique.



Mathematically it can be written as follows: we have an implicit function $H(\beta)$, which is, for example, defined by the following relations

$$H = H(x_i), \psi_j(x_i, \beta_k) = 0, k = 1,...N; i, j = 1,...M ; \tag{1}$$

so we have M values of x and M equations where the H function explicitly depends on x and the β parameters are a part of the equations. Once we know how to solve the general problem of finding the minimum of $H(\beta)$ function, we will also be able to solve the multparameter non-linear regression problem since this problem can usually be represented as (1); the following examples will prove it.

Let us introduce the LaGrange coefficients μ_j and the generating function

$$W(\alpha, \beta, \mu) = H(\alpha) + \sum_{j=1}^{M} \mu_j \cdot \psi_j(\alpha, \beta) k \tag{2}$$

Basing on (2) the equations for x which are a part of (1) can be rewritten as

$$\partial W / \partial \mu_j = 0, j = 1,...M . \tag{3}$$

If conditions (1) and (3) hold, as is assumed below, then $W = H$ and so

$$dH / d\beta_i = dW / d\beta_i = \partial W / \partial \beta_i + \\ + \sum_{j=1}^{M} (\partial W / \partial \alpha_j \cdot \partial \alpha_j / \partial \beta_i + \partial W / \partial \mu_j \cdot \partial \mu_j / \partial \beta). \tag{4}$$

Many algorithms require an efficient method of calculating the total derivatives of $H(\beta)$ with β parameters in order to find the extreme of $H(\beta)$. Expression (4) is a result of identical transformations, with (1) and (3) taken into account. If we simplify this expression as much as possible, it can be used to find $dH / d\beta_i$. Note that according to (3) the last terms in (4) are equal to zero and can be left out. Up to this moment the LaGrange coefficients μ were just arbitrary, purely formally introduced parameters. That is why can turn the penultimate terms in (4) into zeros by introducing the following extra conditions on the LaGrange coefficients:

$$\partial W / \partial \alpha_j = 0, j = 1,... M, \tag{5}$$

Thus (4) is reduced to a simple form

$$dH / d\beta_i = \partial W / \partial \beta_i. \tag{6}$$

(3), (5) and (6) give us the required simple way to calculate $dH / d\beta_i$. If we go back to the analogy mentioned above then we may say that additional equations (5) describe “a flow in the artificially inverted system”, assuming (3) to describe “a flow in the initial system”.

An Applied Mathematics specialist will recognize a standard LaGrange scheme in these manipulations, which is used in many applications, for example in optimal control theory. Naturally, this method proved to be efficient while solving problems of implicitly defined multi-parameter regression.

As for the back error propagation algorithm, it is enough to rewrite any regression problem in the form (1), and the formulas (3), (5) and (6) will produce the algorithm.

Let’s give an example. Suppose we have T records $(y^t, x_1^t,...x_Q^t), t = 1,...T$, and we want to find a function $y(x_1,...x_Q)$, which has a minimal sum of squared distances

$$H = \sum_{t=1}^{T} (y(x_1^t,...x_Q^t) - y^t)^2 \tag{7}$$

If the f(y) function is simple, we derive an algorithm to search for it without formulas (3), (5) and (6). However we will consider a more complex function. Suppose y is as a result of the following algorithm:

$$\alpha_i^g = f(\sum_{j=1}^{S} \beta_{ij} \cdot \alpha_i^{g-1}); g = 1,...G; y = \alpha_1^G, \tag{8}$$

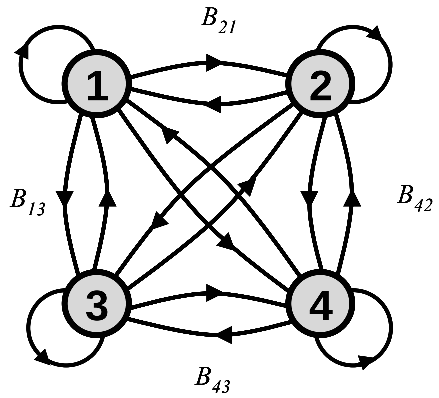
where the dependence $y(x_1,...x_Q)$ is based on the fact that during the initial iteration

$$\alpha_i^0 = x_i; i = 1,...Q;$$

where it is naturally assumed that $S \geq Q$. In (8) $f(z)$ is a certain non-linear function, for example, $f = z/(1+|z|)$, $f = arctg(z)$ or $f = \sin(z)$. Algorithm (8) defines the function $y(x_1,...x_Q)$ in G iterations as a result of a complex combination of non-linear f functions and linear summing operations. Even if f is fixed we may obtain essentially different types of functions $y(x_1,...x_Q)$ via β_{ij}, parameters variation in (8). The number of β_{ij} regression parameters is S^2, and it grows rapidly as S increases: for example, when $S = 40$, there will be 1600 regression parameters.

Before we go back to the problem of a fast search algorithm for parameters β_{ij} in order to minimize the value of (7) we will say a few words about why we define function $y(x_1,...x_Q)$ by means of (8) algorithm and not by a kind of “some expression which is more likely to be a function”. The secret is that we always assume the function to be a complex one requiring much time when calculated on a computer. That is why methods for representing a function which can be implemented on parallel working special designed processors are of interest. Suppose we have a specially designed processor able to simultaneously calculate 40 non-linear *arctangent* functions in a τ_{arctg} time and to calculate a product of multiplying a square matrix by the 40 elements vector in one operation spending the τ_{matr} time. This special processor would perform an iteration of algorithm (8) in a $\tau_{arctg} + \tau_{matr}$ time which can be a very small time-lapse. Algorithm (8) is also easy to implement if we have a local area network or an analog device for data processing. Finally, (8) illustrates general principles of work with "complexly defined" functions. So, if it turns out that for some hardware it is better to use another function definition, then it should be done.

We may draw a graphical scheme of information flows in (8) or, as one can say, a diagram of the corresponding specially designed processor (in the case S=4).



The scheme shows four non-linear transforms which correspond to *f* function, and the lines with arrows stand for 16 regression parameters. When they are non-zero, the result of non-linear operation in the previous iteration affects the next iteration. In the zero iteration the starting values are $(\alpha_1^0, \alpha_2^0, \alpha_3^0, \alpha_4^0) = (x_1, x_2, x_3, x_4)$, and in the last iteration we get a value $y = \alpha_1^G$, which finally turns out to be a function of starting state x. The scheme described here is typical for neurocomputer science, the circles are called neurons and the arrows are synapses. Thus the scheme of data flows resembles adopted in neurophysiology schemes of neuron interaction through their branchy endings – the synapses. That is why (8) represents a "neuroalgorithm". It is possible that this branching of biological neuron interlinks is also partly connected to the operation speed problem: the biological neuron 'operates' slowly, approximately for milliseconds, that is why large number of neurons connected in series (succession) would work extremely inefficiently. However, the attempts to find a complete match between the two algorithms being discussed here and biological neuronetworks haven't yet become successful.

And now, let's come back to the back error propagation algorithm. At first we consider a training demo-exercise:

Problem 0: For non-linear transform $f(z) = 1/(1+|z|)$ build an algorithm to minimize the value (7) at condition (8).

Solution:

1). We build a generating function, an analogue to (2). Since function y is met T times in (7) and, correspondingly, algorithm (8) will be also used T times, we introduce an extra index $t = 1,...T$ denoting the number of times algorithm (8) was used. As ψ we take relations (8) with the right-hand side of the equation transferred to the left $(\psi = 0)$:

$$\psi_i^{g,t} = \alpha_i^{g,t} - f(\sum_{j=1}^{S} \beta_{ij} \cdot \alpha_i^{g-1,t});$$

then $$W(\alpha,\beta,\mu)=H(\alpha)+\sum_{i=1,g=1,t=1}^{S,G,T}\mu_i^{g,t}\cdot\psi_i^{g,t}(\alpha,\beta).$$

Comment 1: the difference with (2) is that the index triplets i,g,t are used instead of indices j.

2). We obtain an equations analogous to (5).

$$\partial W/\partial\alpha_k^{r,\tau}=0;\Rightarrow\partial H(\alpha)/\partial\alpha_k^{r,\tau}+\sum_{i=1,g=1,t=1}^{S,G,T}\mu_i^{g,t}\cdot\partial\psi_i^{g,t}(\alpha,\beta)/\partial\alpha_k^{r,\tau}=0.$$

Many partial derivatives in this sum turn to zero since $\alpha_k^{r,\tau}$ are met only in some of $\psi_i^{g,t}$. There remain only the terms for which $g=r$ and $i=k$ and with $g=r+1$; and only when $t=\tau$. Differentiating ψ, we get

$$\partial H(\alpha)/\partial\alpha_k^{r,\tau}+\sum_{i=1}^{S}\mu_i^{r+1,\tau}\cdot\partial\psi_i^{r+1,\tau}(\alpha,\beta)/\partial\alpha_k^{r,\tau}=0,\Rightarrow$$

$$\partial H(\alpha)/\partial\alpha_k^{r,\tau}+\mu_k^{r,\tau}\cdot\partial\psi_k^{r,\tau}(\alpha,\beta)/\partial\alpha_k^{r,\tau}+\sum_{i=1}^{S}\mu_i^{r+1,\tau}\cdot\partial\psi_i^{r+1,\tau}(\alpha,\beta)/\partial\alpha_k^{r,\tau}=0,\Rightarrow$$

$$\partial H(\alpha)/\partial\alpha_k^{r,\tau}+\mu_k^{r,\tau}-\sum_{i=1}^{S}\mu_i^{r+1,\tau}\cdot f'(\sum_{j=1}^{S}\beta_{ij}\cdot\alpha_i^{r,\tau})\cdot\beta_{ik}=0.$$

The last expression is the equation for LaGrange coefficients μ.

Comment 2: the derivative $f'(z)=(z/(1+|z|))'=1/(1+|z|)^2$ may be expressed by function f itself.

Comment 3: The equation for μ can be solved by means of iterations. If in the equation for α_i^g the upper index indicating the iteration number changes from 0 to G, the upper index in the equation for μ_k^r changes from G + 1 to 1, while all the μ_j^{G+1} must be set to zero (it results from the fact that there are no such μ in the expression for the generating function).

Comment 4: In our case, partial derivative of the estimate $\partial H(\alpha)/\partial\alpha_k^{r,\tau}$ is different from zero only in the last iteration, where $r=G$, and only for $k=1$. In this case it is equal to $2\cdot(\alpha_1^{G,t}-y^t)$, and for all other $k>1$ and $r<G$ it is equal to zero.

3). Analogous to (5), we derive the estimation derivatives with respect to regression parameters.

$$dH/d\beta_{ij}=\partial W/\partial\beta_{ij}=-\sum_{g=1,t=1}^{G,T}\mu_i^{g,t}\cdot f'(\sum\beta_{ip}\cdot\alpha_p^{g-1,t})\cdot\alpha_j^{g-1,t}.$$

General Comment: Computational complexity of LaGrange coefficients and the structure of equations based on them correspond to the complexity of computational implementation of algorithm (8). If the LaGrange coefficients μ and functioning parameters α are known, all the derivatives with respect to regression parameters can be obtained using a simple formula which requires little computational resources. Thus the problem of quick determination of the "quickest descent direction" in the space of regression parameters is solved.

Problem 1: in order to perform all the manipulations presented above on your own, take the record number $T=5$, the number of iteration $G=6$ and the number of non-linear elements (the neurons) $S=4$ and implement the scheme mentioned above in a more explicit manner.

Problem 2: Implement the same scheme for other non-linear functions $f=\arctan(z)$ and $f=\sin(z)$.

Problem 3: For a given modification of (8),

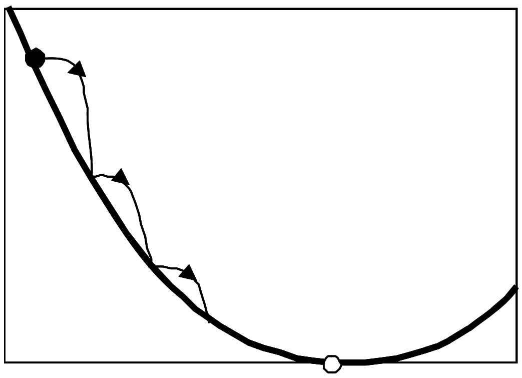
$$\alpha_i^g = f(\sum_{j=1}^{S} \beta_{ij}^S \cdot \alpha_i^{g-1}); g = 1,...G; y = \alpha_1^G ,$$

where different regression parameters are used in every iteration, implement the general scheme of obtaining the estimate derivatives and think why is this variant called a “multilayer neuronetwork”.

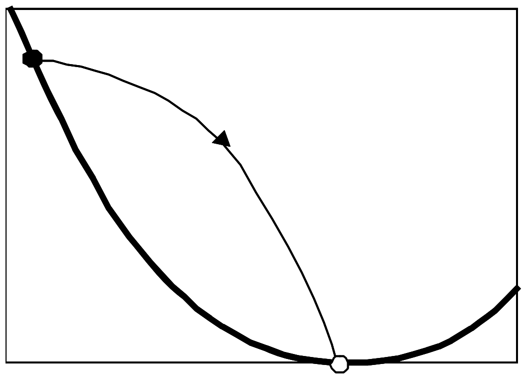
When the shift direction of the regression parameters is determined, an algorithm of reduction of the regression error is developed according to standard methods of optimization theory. The simplest one is a descent algorithm with a constant stepsize where new variable parameters of β^{new} are calculated according to the formula

$$\beta_{ij}^{new} = -\lambda \cdot dH / d\beta_{ij} ,$$

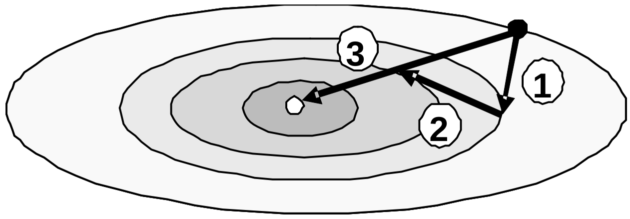
where λ is a parameter of the method. If it is small, the changes are too slow, and if it is large, there is a risk of leaving the domain where we may rely on the determined descent direction. There are various methods, such as the one where optimal value of λ is determined (quasinewtonian quickest descent method) or the one where the stepsize is determined with the ‘experience’ of the previous steps taken into account (the conjugated gradients method). These algorithms illustrate the following schemes “in the regression parameters space”; the regression estimate function shown as a “contour”.



Quickest descent with constant stepsize
(one-dimensional projection)



Quasinewtonian quickest descent
(one-dimensional projection)



Conjugated gradients method
(two-dimensional projection)

Problem 4: For the algorithm in Problem 0 write a program implementing the data regression using a method of quickest descent with constant stepsize. (Do it for example, in Mathlab environment. Since it is a higher-level language, the program will be short and easy to debug. Besides, there is a rather typical add-on for now neuronetwork in this environment. You may consult Help to see how it works. Your program is likely to work worse, to require more iterations since the Mathlab package uses better algorithms than quickest descent with constant stepsize (including such interesting ones as Levenberg-McVart algorithm). But that is only the beginning and if you read textbooks on optimization, and think how to increase the speed, the Mathlab package can be left far behind.)

Here we finish our brief review of the most complicated part of the back error propagation algorithm. These questions have been stressed because similar techniques are used in optimal control theory and in other branches of Applied Mathematics. That means that the time spent on understanding of the "quick" regression algorithms, was not spent in vain.

Data Quantizing Algorithms and Kokhonen Maps (self-organization maps, (SOM))

Kokhonen maps is a variant of data quantizing algorithms, that is representing N data points by a smaller number of sample points. Here we show a Batch SOM, which is an example of such algorithms.

1. A uniform grid of M nodes is chosen. It is oriented so that the grid approximately corresponds to the most important part of data space. As usual, M is essentially less than N.
2. Every data point is 'assigned' to the nearest node.
3. The arithmetic mean of vectors "assigned" to the group is calculated. Let it be $\vec{r}_i$ for node i.
4. The arithmetic mean of vectors "assigned" to the first neighbors of the group is calculated. Let it be $\vec{p}_i$ for node i.
5. A new node position is determined by the vector $\vec{p}_i + \lambda \cdot \vec{r}_i$ where λ is a parameter of a method (its order is about tenths of the unit).
6. Steps 2-5 are repeated several times.

As a result we obtain the grid with some features of uniformity but condensing in the source data was condensed. One may consider such grid as a compact source data set model. It may be a tool for classification of new data since these data may be assigned to the same class. The class was previously assigned to the Kokhonen map node closest to the data vector.

Other Neuronetwork Algorithms

New variants of data processing algorithms can be obtained by combination of the two previously discussed approaches. For example, if one has a lot of data, first he may build a compact Kokhonen map and then apply non-linear regression methods to this map (to say precisely, its multidimensional analogue). However, not all the neuroalgorithms can be reduced to a combination of this kind. This algorithms can be found in numerous specialized issues. Here we have introduced two extremely different variants of neuronetwork algorithms. The back error propagation algorithm, mathematically rather complicated and a relatively easy Kokhonen algorithm which nonetheless solves a number of practically significant problems.

4. Connection between GIS and Neuronetworks

(The problems of mathematical cartography and geoinformation modeling which require the use of neuronetwork algorithms to solve them.)

GIS are a good environment and tool for introducing of artificial intelligence methods and expert systems. Mathematical cartography modeling and geoinformation mapping are the bases of GIS technologies. It involved in making decision, management, expertise, forecast etc. The mathematical processing is based on identification of the uniform and isolated from each other objects. These problems are solved by the procedures of attribute classifying, zoning, evaluation and quantizing.

A number of spatial mathematical models is developed in the cartografical form. There are morphometrical maps (splitting of surfaces, inclines, gradients etc.), the maps of fields of event density and intensity, the background (trend) and residue surfaces, the spatial correlations, event anisotropy and mutual correspondence fields, synthetic maps of distribution of main factors and factor loads, integral zoning and many other. Mathematical modeling assumes an analysis more profound than simple calculation of quantitative indices. It could be production and forecast of further development of more complicated models based on space-time models of structure, dynamics and model interconnection.

A geographic approach to study the nature and social phenomena supposes their territorial variations and their study using the classification methods. The territory zoning, typology and evaluation of complexes (often including representing the obtained results on the map) are not only the methods but also the goals of research. The majority of these analytic problems can be stated as problems of geographic complexes classification. At that, a classification of factors and indices describing these complexes is preliminary performed. In these researches neuronetwork algorithms can be used to reconstruct the function from a finite set of values and to split the set of finite number of objects into classes.

Analysis of selected classes interproximity and definition of their connection to different aspects of the investigated event are required for the interpretation. Various factors and correlation analysis algorithms are used for this. The use of these methods is ineffective when dealing with great data amounts and especially with raw data with non-linear interdependencies. In this case, it is more useful to apply neuronetwork algorithms which can be interpreted as a generalization of linear statistics methods to a non-linear or locally linear case.

The time and genetic (inheritable) parameters of studied complexes are a very important basis of geographic classification. In geographic researches the complexes are usually regarded as space-time formations. The time and genetic parameters also play an important role in classifications in other social and natural sciences (biology, geology, economics, history etc.) when results are reflected on geographic maps. The multidimensional classification and the criteria for reliability of multidimensional data analysis algorithms used there could be inapplicable to real-life geographic problems and to representation of the corresponding data in GIS. Relatively more flexible neuronetwork approaches can prove to be more effective.

A choice of optimal system of base parameters according to the essence of studied events is one of the main problems of geodata analysis. A natural desire to thoroughly examine an event using the system with greatest number of parameters may result in redundancy of raw data involved in analysis. The data supplied by different sources and regarded as independent may be repeatedly duplicated or can be calculated in the base of the other (secondary data). This may distort the parameter importance and lead to mistakes in analysis results. Deeper understanding of the essence of the studied territory-distributed complex is the best way to obtain a criterion of parameter importance and to construct a parameter system adequate to the research problem. In automated mode, one could use the on-line methods for computational experiment models and techniques correction which provide data error correction and automated detection of cause-and-effect relations, lower the dimension of the multidimensional classification and geodata analyzing problems.

The basic parameters of the most classification problems of analytic geography are of different nature and can be both of quantitative and qualitative kind. That is why algorithms for geodata analyzing have to be able to deal with parameters both of numerical and non-numerical nature. This imposes certain restrictions to available mathematical analysis methods. An attributive description of analysis objects has been already formalized in GIS and represented as electronic tables. This facilitates the use of neuronetwork technologies and lessens the difficulties of analysis bound to large amounts of raw data, data gaps, heterogeneity of quantitative and qualitative object parameters. Neuronetwork approach allows data compacting and parameterizing and can create simple and illustrative data models based on the data.

In fact, the listed problems in analyzing territory-distributed processes wich are represented as geographic complexes leads to the necessity of using the methods of neuronetwork analysis for structuring multidimensional data. Detection of objects or their complexes (which are also the objects with their own complex structure), determining various spatial relations and proximity measures between them. Object clusterization and data aggregation depend on definition of the research problem and can decrease the number of analyzed objects considerably. When we turn to forecast and evaluation problems, the constructing of models for the examined processes gain importance and:

– the objects detected during the analysis become basic elements for the model
– geographic complexes parameters become numerical parameters of the object and it's properties.
– analysis of geographic complexes and their parameters leads to determination of the functional bindings and object structuring corresponding to this bindings.

Analysis of cause-and-effect event relations (changes in state of objects involved in one or another process), structure of functional bindings of selected objects, parameters of exchange processes in these bindings, determination of the field parameters are of importance when developing these models.

It is essential that labor intensity for solving the problem traditionally and using neuroalgorithms are different. In traditional cases when it is required to fully understand data structure, then compose a data processing program, then debug this program in neuroalgorithms cases it is sometimes enough to simply load "raw" data into a neuronetwork program, and obtain a result in some time (it may take about a night if database is large).

Neuronetwork is one of the efficient tools for solving loosely formalized problems using sample examples. Neuronetworks spread is due to the following features of neuronetwork approach:

1. Automated parameters adjusting in neuronetwork model for solving the problem using sample examples. No expert is required to construct a model for solving the problem.
2. Robustness. Neuronetwork solutions can solve any problem in a standard way, with no respect to its semantics, provided the problems can be represented as a set of sample examples containing input and output data.
3. Stability when used with non-reliable and noisy data.
4. Capabilities to adapt to new situations.
5. Tolerance to faults and element destructions.
6. High parallelizing abilities present in all neuronetwork models.
7. Ability to efficiently process high-dimensional and heterogeneous data.

It should be also noted that neuronetwork solutions combined with geoinformation technologies can be used to solve problems which are beyond the scope of traditional GIS applications. For example, these solutions are used in visualizing mapping of multidimensional data (which may include gaps) by means of two-dimensional manifolds embedded in data space. The feature of this technology is its ability to continuously project the data onto a map which significantly improves the accuracy in data representation. The basis for constructing such a map is a two-dimensional grid embedded in multidimensional space and approximating data. Since the grid has adjustable elastic properties (stretching and bending) it is called an elastic map. The location of grid nodes is obtained by solving a variation problem of finding the minimum of a functional whose structure and parameters depend on the research problem and determine the elastic properties of the map.

Unlike "conventional" statistics methods, neuronetworks give not a statistically reliable but a plausible solution of the problem and may be applied when empirical data are not enough for statistical analysis. Advantages of neuronetwork solutions comparing to statistical approach are its universality and automated adjustment in heavy prior uncertainty conditions. All this allows to obtain a satisfactory result quickly. Neuronetwork models imposes a short restrictions on possible variable distribution functions and allow avoid prior guesses on model structure and the form of variable distribution functions.

5. Fields of Neuronetwork GIS Application.

Neuronetworks integrated with geoinformation systems are powerful tools for solving a wide class of problems which provides an efficient decision-making support. Neuronetwork may use spatial GIS data as input and output data. Programs built on neuronetwork algorithms will dynamically modify the layers of electronic map, change the parameters of existing objects and create new ones. New map layers may be also produced as a result of processing existing data arrays and the existing layers will gain dynamic properties.

Today we can see a lot of examples illustrating the efficiency of neuronetwork GIS solutions and the range of their users is extremely wide. These solutions appear to be most advantageous when one has to deal with large amounts of data stored in large firms or institutions. This data is the basis for decision-making for the experts evaluating and forecasting the state of some field of human activity (for example, production market, real property value, territory pollution). Such problems as planning the priority of actions when developing territories and their investment attractiveness, detecting the zones of most serious ecological, social or economical situation, analyzing the properties for geological objects and many other can not be solved according to contemporary standards without using intelligent geoinformation systems.

Let us give a few easy examples:

In agriculture some GIS layers may contain data on crops sowing and others may contain data on crop capacity achieved. In this case a neuronetwork will use practical methods and technologies for crop growing and add data on particular climate, soil and other parameters of the chosen territory.

In solving forest regulation problem a neuronetwork may help to analyze the dynamics of tree growth with respect to height, diameter and volume. Processing subject GIS layers containing such data may assist in planning forest regulation activities (for example, in planting pine-trees in green zones and forest parks).

Prognostic neuronetwork models can be applied to **demography and health protection problems** where they rely upon spatial data on population density, medical statistics, environmental pollution represented as GIS levels. An expert system may determine possible lifespan, connection between different disease kinds and territory ecological conditions, forecast epidemic outbreaks.

Remote sensing data processing is a typical problem solved by geoinformation systems nowadays. Image analysis (from the viewpoint of mathematics) is based on pattern recognition theory where it is required to assign an object to one or another class according to input data. In this area neuronetwork solutions along with fuzzy logic methods have gained the widest application. Such methods turned out to be an appropriate language to describe classification rules without using exact mathematical notions (using such user-friendly terms as 'considerable', 'not large' and so on). And vice versa, to extract the rules for classifying raw data and represent them in common language. Unlike conventional statistics methods based on computations in the framework of a certain mathematical formalization, the classifiers based on neuronetwork solutions use their adapting skills while studying which doesn’t require preliminary model validation. However, it is proved that classification results obtained by these two ways may coincide, that means that the neuronetwork solution can construct the corresponding mathematical formalization on its own.

Different neuronetwork algorithms are used for classification purposes. Unsupervised neuronetworks are applied to analyzing color or black-and-white photo pixels with no connection to other map layers. The purpose is to detect homogeneous image fragments in their tone, structure or hue. Supervised learning relies on available spatial data for given territory. If it is known beforehand that the chosen photo part corresponds to forestry with a certain human imperfection degree, neuronetwork solution may use this information for classifying the image.

Now neuronetworks are used to detect spatially homogeneous image parts more frequently. This problem is very actual in theoretical and methodological bases development for a new alternative agriculture systems, ecologically safe principles of land-utilization and landscape-based land management projects. Isolated landscape elements can be detected by analyzing shape, color, interrelations and heterogeneity of image parts. Neuronetwork solution can also estimate interconnections of separate landscape elements.

In transport field neuronetwork solution may become an efficient addition to highway monitoring GIS. It means analysis of traffic density and road-bed condition, choice of optimal areas for constructing new roads and setting of construction priorities, analysis of various strategies for maintenance activities and corresponding financing. Neuronetwork solution can make an online decision on optimizing the traffic density distribution for motor roads in case of an accident which resulted in a traffic jam. Neuronetwork solution will use GIS layers for highways, places of accidents, current weather conditions and other parameters affecting traffic speed as input data. All system parameters relevant to the given accident will be corrected by neuronetwork solution and their reset to the initial condition will be possible after the stabilization. On the next stage the drivers will have online access to the cartography server from their cars via personal digital assistants (PDAs) and other mobile devices. The cartography server will contain online information on traffic situation and will give advice on optimal route prepared by the neuronetwork solution.

A problem of personal identification has become very urgent owing to the recent acts of terror. Intensive security actions are taken in airports and other places of people concentration. GIS can be used to efficiently monitor such objects and provide necessary online data to the security services. Neuronetwork solution can be used in cooperation with GIS to localize suspicious situations by on-the-fly processing of data obtained from surveillance cameras.

6. Software

Software market offers a number of various products for modeling neuronetwork solutions. Internet lets have hundreds of references to Russian and foreign sites. We can single out the following basic features implemented in all these programs such as:

– building-up a neuronetwork
– neuronetwork training
– testing the neuronetwork (imitating of functioning)

From the standpoint of computer technologies and program interfaces all these programs from the simple ones made for Unix platform with text interface to complex modular products based on the latest Microsoft technological solutions.

Integrated solutions based on GIS and neuronetworks are still not numerous despite the fact that one of the most important features of modern geoinformation systems is improving of the functional capacity of geoinformation packages by integrating special add-on modules or GIS applications. The problem of integrating neuronetworks and GIS can be solved at least in three ways:

– integrating (building in) the neuronetwork models into GIS by special means of geoinformation system (programming in built-in languages such as Avenue, MapBasic etc.)
– developing interfaces for individual neuronetwork analysis applications and GIS as for independent systems (for example, through DDE data exchange)
– development of application software for neuronetwork systems implementing GIS elements (for example, using class libraries with such classes as MapObjects, GeoConstructor, MapX etc.)

The choice of particular way depends on given requirements, set problem, available resources and work experience. The implementing peculiarities are determined by abilities of the software used both in neuronetwork and in GIS (for example, by built-in programming language, OLE and DDE means, DLL and ActiveX functional abilities). A corresponding neuronetwork GIS application is constructed basing on the chosen technology.

Two examples are given below. These are the software products that have been already constructed on the basis of neuronetworks and GIS.

A ScanEx-NeRIS Program

The NeRIS program is designed for thematic interpretation of spatial data, mainly for Earth remote sensing data. Kokhonen neuronetworks are the main tool implemented in the program; they are used for ordination, classification and thematic interpretation. As one of the methods for multidimensional data classifying, the Kokhonen neuronetworks have important supplementary properties on which is based a considerable part of algorithms used in the program.

The thematic processing package for bitmapped images implemented in ScanEx-NeRIS program has the following features:

- estimate number of classes required to describe the theme and compose a thematic map;
- estimate internal fractioning and heterogeneity of thematic objects (contours);
- estimate property distribution of expert objects in the parameter field of distant model;
- estimate presence rate of thematic objects specified in image parameter field by an expert (isolation of the image areas with different estimate levels: optimistic, realistic and pessimistic)
- generation of hierarchical classification with estimates for class interproximity (a handy interface for processing “distance maps” (“distance trees”) for experts in thematic cartography aimed at correcting and improving of the legend and geographic contents of a thematic map);
- construction of topic-oriented neuronetworks for future bitmap processing in order to detect thematic objects;
- auto tracing (vectorization) of class-by-class processing results;
- support of coordinates of the most widespread Russian and foreign cartographic projections;
- export of vector layers and bitmap surfaces in exchange formats of the most popular geoinformation systems;
- representation of classification results for neuronetworks of all types both by assigning a class index to every classified pixel and by constructing bitmap layers of 'possibility' for pixel membership in a particular class (construction of a number of such layers and their consequent visualizing allows to illustratively represent classification results, for example to detect “blank spots” (non-classified areas of space) and to provide data for final classification by conventional methods).

Arc-SDM Module for ArcView

Arc-SDM module is one of freeware add-ons for ArcView designed for modeling in GIS using fuzzy logic and neuronetworks. From the GIS users point of view modeling using this module consists of constructing a new thematic layer based on several existing ones. Arc-SDM solution uses two neuronetwork algorithms footnoted as an independent program module DataXplore. The first algorithm implements the neuronetwork solution based on radial basic functions and the second one is based on fuzzy logic clusterization. The neuronetwork solution based on radial basis function requires a learning period after which a set of parameters which determines the interconnection between input data layers and the output (result) layer is generated. The neuronetwork solution is ready for data classifying after this process.

For example in geological problem of minerals exploration the result layer contains the data on ore deposit presence or absence. The input data involved in learning process can be divided into two types: deposit locations and the areas known to have no deposits. A grid-theme is constructed basing on the input thematic vector layers of geoinformation system and then the prepared data set is transferred to the DataXplore module. The computational result is represented as a new thematic layer.

References

1. Gorban A. N.Neuronetwork learning. Moscow, SP ParaGraph, 1990, 160p.
2. Gorban A. N., Rossiev D. A. Neuronetworks on Personal Computers. Novosibirsk, Nauka, 1996, 276 p.
3. Pitenko A. A. Neuronetwork Analysis in GIS. //thesis abstract, technical science candidate 05.13.16, Krasnoyarsk, 2000.
4. Yezhov A. A., Shumsky S. A. Neurocmuting and its Applications to Econimics and Business, Moscow, 1998.
5. Proceedings of International Workshop on Dynamic and Multi-Dimensional GIS. 25-26 August 1997, The Hong Kong Polytechnic University, 252 pp. // 1997.
6. Al Gore. The Digital Earth: Understanding Our Planet in the 21st Century, 1998. http://www.regis.berkeley.edu/rhome/whatsnew/gore_digearth.html.
7. D. Dobrynin, A. Savelyev Neurotechnologies for Mathematical Deciphering RSD.//GIS-survey , №1, 1999, P.12-14. http://www.scanex.ru/.
8. Kemp, L.D., Bonham-Carter, G.F., Raines, G.L. and Looney, C.G., 2001, Arc-SDM: Arcview extension for spatial data modelling using weights of evidence, logistic regression, fuzzy logic and neural network analysis. http://ntserv.gis.nrcan.gc.ca/sdm/.
9. Mirkes Ye. M. Neurocomputer. Project of a Standard: Novosivirsk, 1998, 319 p.
10. Zinovyev A. Yu. Multidimensional Data Visualization // Krasnoyarsk, 2000, 168 p.