Source: Mrówczyńska M. Primeneniye metodov iskusstvennogo intellekta dlya transformatsii sistemy koordinat [Methods of Artificial Intelligence Used for Transforming a System of Coordinates]. In: *Contemporary Issues of Concrete and Reinforced Concrete: Collected Research Papers*. Minsk. Institute BelNIIS. Vol. 4. 2012. pp. 6–19. (rus)

Mrówczyńska M., PhD., Institute of Building Enginieering, University of Zielona Gόra , Poland

METHODS OF ARTIFICIAL INTELLIGENCE USED FOR TRANSFORMING A SYSTEM OF COORDINATES

ПРИМЕНЕНИЕ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ТРАНСФОРМАЦИИ СИСТЕМЫ КООРДИНАТ

Abstract

The article discusses the problem of transformation in the form of the function $f: R^2 \to R^2$ as a dependence between the coordinates $[x, y]$ of the original system and the coordinates $[X, Y]$ of the secondary system. The task of the transformation of $[X, Y] = f(x, y)$ technically understood as the transformation of a system of coordinates has been solved by means of one direction two layer neural networks of the sigmoidal type, radial neural networks, recurrent cascade neural networks, and neuro-fuzzy systems with the use of the Takagi-Sugeno-Kang model. The numerical procedures applied make it possible to obtain a level of accuracy of the task equivalent to the cartographic accuracy of pictures in the Spatial Information Systems.

Аннотация

В статье обсуждается проблема трансформации в виде функции $f: R^2 \to R^2$, представляющей собой зависимость между координатами [x,y] первичной системы и координатами [X,Y] вторичной системы. Задача трансформации [X,Y] = f(x,y), технически понимаемая как трансформация системы координат, решена посредством однонаправленных двухслойных нейтральных сетей сигмоидального типа, радиальных нейтральных сетей, рекуррентных каскадных нейтральных сетей, нейро-нечетких каскадных нейтральных сетей с применением модели Такаги-Сугено-Канга. Применяемые численные методы при постановке задачи позволяют достичь уровня точности, эквивалентного картографической точности изображений в Пространственных Информационных Системах.

INTRODUCTION

The transformation of coordinates is an important task from the field of geodesy because of the introduction of a unified system of coordinates complying with the standard at the time of European integration. The transformation of the coordinates of cartographic systems operating in various reference systems is possible when the formula is known for specifying the relationship between the coordinates of the connecting points with known coordinates in both systems [4]. The most frequently used method for calculations of this kind is the Helmert transformation, in which the estimation of parameters is achieved on the assumption of the minimum weighed length of the vector of corrections for the adaptation points

$$v^T Pv = \min, \tag{1}$$

where: v — the vector of corrections

P — the weight matrix of the coordinates.

This transformation is used in the tasks of:

— the transformation of geodesic and photogrammetric coordinates,

— affine calibration,

— specifying parameters for the displacement of a completed engineering object.

Unfortunately, in the process of specifying transformation parameters the Helmert method has a considerable inaccuracy since it is not immune to gross errors.

As far as the abovementioned method is concerned, the author suggests an alternative approach to the transformation of coordinates — by means of one direction two layer neural networks, radial neural networks, recurrent cascade neural networks, and neuro-fuzzy systems. The results of the numerical realization of the task of transforming the coordinates of points from the original system into the secondary system have been compared in terms of accuracy with the results obtained by means of professional algorithms.

SELECTED METHODS OF TRANSFORMING COORDINATES

Gradient algorithms for learning one direction two layer neural networks

Artificial neural networks have strong theoretical foundations and a wide practical use. Any problem that can be solved by means of classic modeling or statistical methods can be solved by means of neural networks [8]. Definitely, in most cases one direction multi-layer neural networks of the sigmoidal type are used in practice. From the mathematical point of view neural networks of this type play the part of the stochastic approximation of a multi-variable function, which transforms the set of input variables $x \in R^N$ into the set of output variables $z \in R^M$ [11].

A two layer neural network (two layer perceptron), the general outline of which is presented in Fig. 1, has been used in order to carry out calculations connected to the transformation of coordinates from a primary system into a secondary system [9], [5].

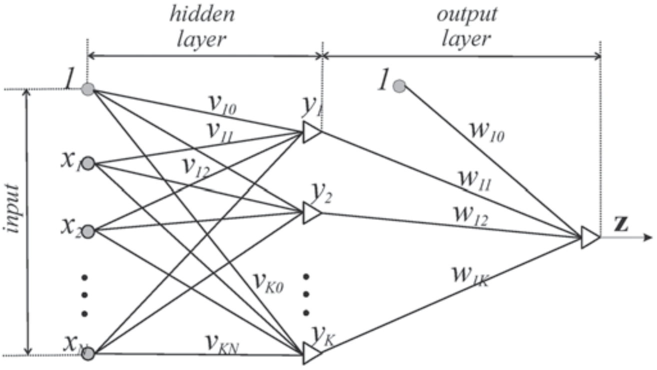

Fig. 1. Multilayer Network

Defining the relation as a cause and effect connection between the input and the output of the network with a priori determined topology is the process of learning the network, which consists in adapting parameters of the network called weights.

The technique of learning neural networks uses gradient optimization methods. The basis for the algorithms used to learn the network is an objective function (energy function), defined by means of Euclides metrics as a sum of the squares of the differences between the values of the input signals of the network and the assigned values in the form:

$$E = \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{M}\left(z_j^{(i)} - d_j^{(i)}\right)^2, \qquad (2)$$

where:

p — the number of input vectors,

M — the number of coordinates of the input vector,

z_j^i — the coordinate of the output vector $(i = 1,2,...,p)$, $(j = 1,2,...,M)$,

$d_j^{(i)}$ — the coordinate of the assigned vector $\left(x^{(i)}, y^{(i)}\right)$, $(i = 1,2,...,p)$, $(j = 1,2,...,M)$.

The minimization of the objective function (2) taking into account the sigmoidal activation function, consists in the correction of the weights v_{ij} and w_{ij} (Fig. 1) on the basis of the information included in the gradient of the objective function $\nabla E(w)$, according to the relation

$$\Delta w = -\eta \nabla E(w), \tag{3}$$

in which $-\nabla E(w)$ denotes the general direction of the minimization, and η the learning ratio (the ratio of the iterative step).

Most gradient optimization methods use the square model of the function in the vicinity of a particular solution point $w(t)$ (extension into the Taylor series) in order to obtain information about the curvature of the function included in the Hessian in the following activation step $k+1$. In order to obtain convergence towards an optimum solution, gradient methods of learning networks widely known from the theory of optimization have been used in this paper, namely [3]:

— the greatest decline method (linear approximation of the function $E(w)$,

— quasi-Newtonian methods: the method of variable metrics, the Levenberg-Marquardt method of conjugate gradients, and the Resilient Back-Propagation algorithm .

Radial neural networks

The stochastic approximation of a multi-variable function achieved by means of multi-layer neural networks is global in character, because the transformation of the function estimated into any point in space is achieved as a result of simultaneous stimulation of a number of neurons. A complementary method of transforming the input set into the output set is the adaptation of a number of single approximation functions to the members of the set of assigned values within a limited area of multi-dimensional space. The transformation is local in character, and the transformation of a full input vector $x \in R^n$ into the output vector $z \in R^M$ is a result in the form of local transformations achieved by means of networks with radial base functions (Fig. 2), consisting of neurons which carry out the transformation in the hidden layer [8]:

$$x \to \varphi\left(\left\|x - c\right\|\right),\ x \in R^n. \tag{4}$$

If in the input we have p input vectors $x_i (i = 1,2,\ldots,p)$, which are going to be transformed into a set of real numbers $d_i (i = 1,2,\ldots,p)$, then the problem consists in searching for an estimator of the transformation function.

$$F(x_i) = d_i. \tag{5}$$

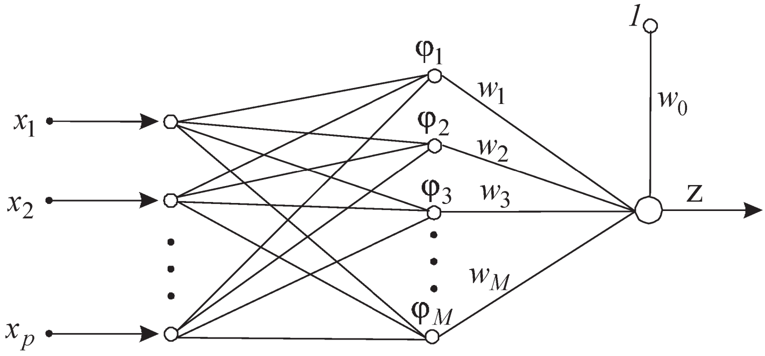

Fig. 2. Radial Basis Network

If the number of neurons M is introduced as much smaller than the number of learning patterns $p(M << p)$, , then the transformation function which is being estimated has the form

$$F(x) = \sum_{i=1}^{M} w_i \varphi\left(\left\| x - c_i \right\|\right). \qquad (6)$$

The symbols included in the formula (6) denote:

w_0, w_1,... w_M — scalar ratios called weights,

$c_1, c_2, \ldots, c_M \in R^d$ — vectors called centers, functions φ are located above centers,

$x \in R^d$ — the input vector.

The most widely used radial function φ (apart from a number of others, some of them being imperfect) is the Gauss function (a simplified form)

$$\varphi(x) = \varphi\left(\left\| x - c_i \right\|\right) = \exp\left(-\frac{\left\| x - c_i \right\|^2}{2 y_i^2} \right), \qquad (7)$$

which has nothing in common with the assumptions of the normal distribution, but satisfies the conditions of the Parzen method and brings good results in practice. On the basis of a suitable procedure for the minimization of the objective function

$$E = \sum_{i=1}^{p} \left[\sum_{j=1}^{M} w_j \varphi\left(\left\| x_i - c_j \right\| - d_i \right) \right]^2 \qquad (8)$$

the coordinates of the vector of weights w are determined as well as the values of the parameters c and σ of the radial functions. If the parameters are known, the solution of the minimization is achieved by means of the Green pseudo-inversion of the matrix

$$w = G^{+}d. \tag{9}$$

Whether good results are obtained from the transformation of coordinates by means of neural networks with radial base functions depends on the correct architecture of the network, the number o radial base functions and their widths as well as the value of the learning ratios [2], [6].

Recurrent cascade neural networks

Recurrent cascade multi-layer perceptron neural networks are created by adding suitable feedbacks to one direction networks. The feedbacks are obtained from the output layer of neurons and directed to neurons of the input layer. Therefore, it is a dynamic system, which works as a one direction multi-layer network because of the way the output signal is created.

The operation of a cascade neural network is divided into two stages. In the first stage a non-recurrent cascade network is used according to the structure presented in Fig. 3. The architecture of a non-recurrent cascade network as a one direction network is constituted by a one step increase of the dimension of the input vector and the output vector. In the initial stage the first layer receives stimulation from the input layer in the form of the vector *x* with the coordinates (*x*,*y*) of the point in the original system, and the expected output signal is the coordinate x'_1 of the point in the secondary system. After the learning process is completed there is an increase in the dimension of the input vector, which includes both the coordinates (x,y) in the original system and the coordinate x'_1 obtained from the output, i.e. $x' = [x, y, x'_1]$. The application of this vector in the input starts another learning cycle with the expected output signal in the form of the coordinate y'_1 in the secondary system. As a result of this course of action we obtain the vector $y' = [x, y, x'_1, y'_1]$. The number of learning cycles corresponds to the number of members of the learning set.

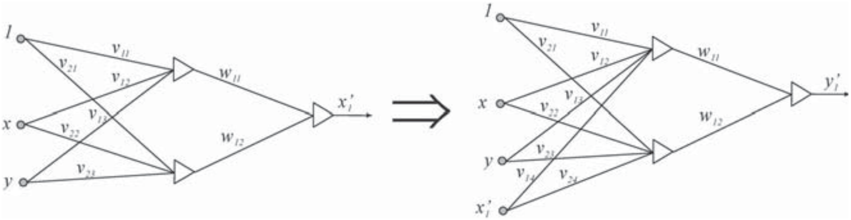

Fig. 3. Non-recurrent cascade neural network

In the second stage the course of action concentrates on the choice of a structure for a recurrent cascade network (Fig. 4).

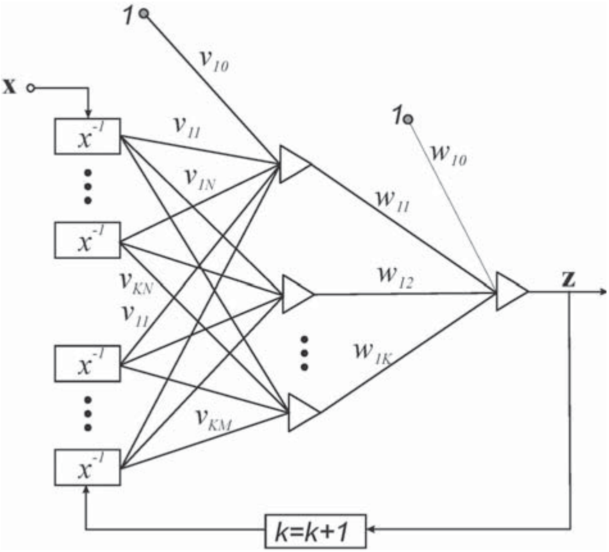

Fig. 4. Recurrent cascade neural network

In the second stage a recurrent cascade neural network is built (Fig. 4), in which the input vector is created by the coordinates of the points in the original system and the secondary system $X = (x, y, x', y')$ and there are feedback connections between the output layer and the input layer. It should be noticed that during the process of learning the input vector is updated. For the iteration $k+1$ in the input vector, there are coordinates of points in the secondary system obtained from the iteration k. Algorithms for learning a recurrent network make use of the abovementioned gradient optimization methods, and the same as in the case of a one direction neural network we calculate the gradient of the objective function (2) in relation to each weight. A detailed form of the recurrence formula, which makes it possible to calculate the gradient at any moment k on the basis of its value at the previous moments, is included in the paper [8].

NEURO–FUZZY SYSTEM OF THE ADAPTIVE TYPE

Neuro-fuzzy systems are neural networks which have the ability to transform fuzzy sets. Neuro-fuzzy systems make it possible to interpret knowledge accumulated in the weights of neural bonds, which is the basis for formulating sets of fuzzy conditional rules «if — then». One of the basic methods of obtaining bases of knowledge consisting of rules «if — then» consists in extracting rules on the basis of numerical data about the inputs and outputs of the phenomenon which is being modeled.

In this case the Takagi — Sugeno — Kang (TSK) system is usually used, whose advantage is a small number of calculations necessary to determine the output value of the system. The knowledge basis of the TSK system is *M* inference rules «if — then» together with a linear function (prime polynomial) in the conclusion of the kth inference rule, written in the relation [7]

$$M^{(k)} = \text{if} \overset{N}{\underset{1\le j\le N}{\wedge}} x_j \text{ is } A_j^{(\mathrm{k})}, \text{ then } y = f_k(x) \text{ for } k = 1,2,\dots,M \tag{10}$$

and the linear function

$$f_k(\mathrm{x}) = p_{k0} + \sum_{j=1}^{N} p_{kj} x_j \tag{11}$$

where pk denotes (*N*+1) — a dimensional vector of parameters. A set of simple linear functions $f_k(x)$ makes it possible to model complicated dependences between the input and the output of the system.

Fuzzy sets as a generalization of ordinary sets are characterized by a partial membership of members in a particular set. Similarly to classic sets, which are described by means of characteristic functions, fuzzy sets are described by means of membership functions with values from the range [0,1] [10]. One of the most widely used functions of membership in a fuzzy representation of numbers is the Gauss function, defined for the variable *x*, the center c and the variance σ, determined for the set *A* in the form (general form) [8]

$$\mu_A(x) = \exp\left[-\left(\frac{x-c}{\sigma}\right)^2\right]. \tag{12}$$

In the paper the function (12) has been replaced with a rational Gauss function (bell function) described by the formula

$$\mu_A(x) = \frac{1}{1+\left(\frac{x-c}{\sigma}\right)^{2b}}, \tag{13}$$

where the parameter b influences the form of the generalized function. Depending on what value of the parameter σ is chosen, a Gauss function, a triangular function or a trapezium function can be described. An aggregation of information included in the premises for the implication constitutes the resultant of the membership function $\mu_A(x)$. The aggregation operator is represented by the transformation $\oplus:[0,1]^N$ carried out in order to obtain the value $x \in [0,1]$, i.e. $x = \oplus(x_1, x_2, \ldots, x_N)$.

According to a fuzzy procedure, the aggregation of the premises for the implication will be interpreted as an algebraic product, which is expressed by the formula for the kth inference rule

$$\mu_A^{(k)}(x) = \prod_{j=1}^{N} \left[\frac{1}{1 + \left(\frac{x_j - c_j^{(k)}}{y_j^{(k)}} \right)^{2b_j^{(k)}}} \right]. \tag{14}$$

The output value of the system is obtained as a weighed mean of the output values of particular rules

$$y(x) = \frac{\sum_{k=1}^{M} \mu_A^{(\kappa)}(x) f_k(x)}{\sum_{k=1}^{M} \mu_A^{(k)}(x)}. \tag{15}$$

The architecture of a neuro-fuzzy network is presented in Fig. 5.

There are five layers in the network. In the first layer the input values x_j become fuzzy and the parameters $c_j^{(k)}, \sigma_j^{(k)}, b_j^{(k)}$ (centers, widths and shapes) are determined by means of gradient learning methods. In the second layer implication premises are aggregated. In the third layer the values of the function fk(x) (TSK function) are generated on the basis of the adaptation of the parameters pkj (k=1,2,..., M, j=1,2,..., N). Two summing neurons constitute the fourth layer, one of them calculates the weighed sum of the signals yk(x), and the other the sum of the weights (the weights are interpreted as an aggregated value $\mu_A^{(k)}(x)$). The last — fifth layer contains only one neuron which generates the output signal of the network y(x) according to the dependence (15).

It is also possible to notice that the Takaga — Sugeno — Kang fuzzy system and normalized radial neural networks are equivalent when certain conditions are satisfied [7].

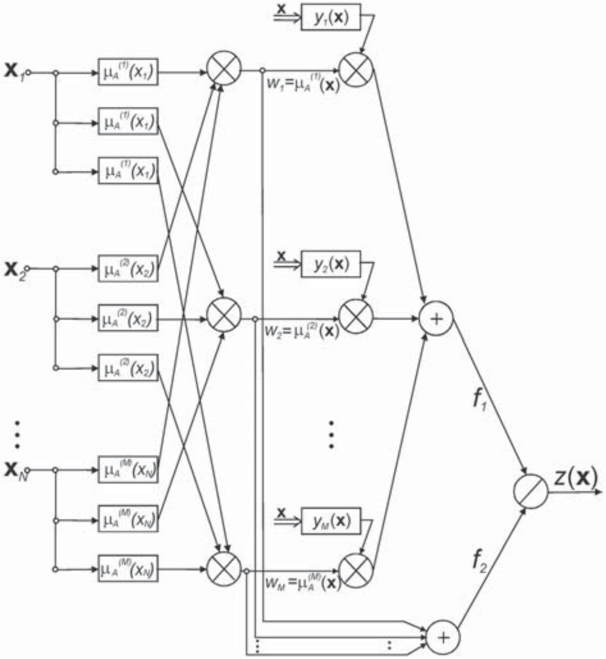

Fig. 5. Structure of a TSK neuro-fuzzy network

NUMERICAL EXAMPLE

The transformation of coordinates from the original system into the secondary system has been carried out with the use of a suitable structure of neural networks matched to a particular task. The author of the paper attempts to choose the structure and parameters of the network so as to approximate the assigned values in a statistically optimum way, i.e. to obtain an acceptably low extent of error in the test data.

The solution of the task of transformation has been achieved on the basis of a learning set and a test set, and each set has 1600 points. The points of the learning set represent the adaptation points, and the result of the transformation of the coordinates of the points of the test set from the original system into the secondary system is a result of the influence of the network on data not participating in the process. For each of the abovementioned and implemented network structures presented in tables 1÷5 an optimum value of the objective function, whose value is expressed by the root of the random mean square error (RMSE), has been determined for a particular minimization method. A change in the value of the gradient in two consecutive iterations on the level 1e-10 has been adopted as a criterion for stopping the iteration process.

In order to provide complete information about the process of learning networks it is necessary to add that for multi-layer perceptron networks a variable number of hidden layers has been used as well as a variable number of neurons in particular layers. The training of radial networks takes into account a variable number of radial base functions and variable values of the parameter σ while choosing their shapes.

An effective method of obtaining highly accurate results of transformation consists in the application of the results of the transformation $[X, Y] = f(x,y)$ by means of neural networks, which are created by the output variables of a Takaga — Sang — Kang (TSK) neuro-fuzzy network. An important problem while building the structure of a TSK neuro-fuzzy network is to determine the number of fuzzy inference rules, which is determined on the basis of the minimization of the value of the global statistic measure

$$\alpha = a_1 V_h - a_2 D_A - a_3 D_w + a_4 t_A . \tag{16}$$

The local minimum of this function, described in the paper [1], makes it possible to determine a sub-optimum number of clusters for a particular data set. The ratios of the scale a_i=(1,2,3,4) have been determined by means of a genetic algorithm with the use of a tournament selection of chromosomes, for the likelihood of one-point crossing equal 0,77 and the likelihood of mutation equal 0,0077.

The effectiveness of the use of particular algorithms for solving the task of transformation of coordinates is shown by the results included in tables 1÷5, in the form of the random mean square error (RMSE) calculated as

$$RMSE = \sqrt{\frac{1}{P}\sum_{p=1}^{P}(d_p - z_p)^2} . \tag{17}$$

Table 1

Characteristics of the accuracy of the transformation $[X, Y] = f(x,y)$ by means of neural networks

The gradient methods	The network architecture	The transformation error RMSE [m]	
		the learning set	the testing set
the Levenberg-Marquardt method	2_5_2_1	0,011	0,012
the RPROP algorithm		0,012	0,013
the quasi-Newtonian method		0,012	0,013
the conjugate gradients method		0,132	0,158
the greatest decline method		0,794	0,975

Table 2

Characteristics of the accuracy of the transformation $[X, Y] = f(x,y)$ by means of nural networks depending on the number of radial functions

Number of radial basis functions	The transformation error RMSE [m]	
	the learning set	the testing set
60	0,034	0,029
90	0,013	0,014
100	0,057	0,092

Table 3

Characteristics of the accuracy of the transformation $[X, Y] = f(x,y)$ by means of neural networks depending on the radial base functions used

The radial basis functions	The transformation error RMSE [m]	
	the learning set	the testing set
the Gauss function	0,013	0,014
the spline function of fourth degree	0,015	0,022
the spline function of third degree	0,013	0,024
the central function	0,008	1,124
the Hardy's function	0,045	189,271
the linear function	0,099	220,404
the aquared function	0,020	227,510

Table 4

Characteristics of the accuracy of the transformation $[X, Y] = f(x,y)$ by means of recurrent cascade neural networks

The gradient methods	The network architecture	The transformation error RMSE [m]	
		the learning set	the testing set
the Levenberg-Marquardt method	2_10_1	0,007	0,008
the conjugate gradients method	2_15_1	0,009	0,011
the quasi-Newtonian method	2_10_1	0,022	0,034
the greatest decline method	2_15_1	0,594	0,684

Table 5

Characteristics of the accuracy of the transformation $[X, Y] = f(x',y')$ obtained with the use of the TSK system on the basis of results determined by means of neural networks

The gradient methods	The transformation error RMSE [m] (the testing set)	
	the neural networks	the Takaga — Sugeno — Kang fuzzy system
the Levenberg-Marquardt method	0,012	0,008

end of Table 5

The gradient methods	The transformation error RMSE [m] (the testing set)	
	the neural networks	the Takaga — Sugeno — Kang fuzzy system
the RPROP algorithm	0,013	0,008
the quasi-Newtonian method	0,013	0,010
the conjugate gradients method	0,158	0,092
the greatest decline method	0,975	0,729

CONCLUSIONS

The methods applied to transform coordinates of the points from the primary system into the secondary system make it possible to better use the calculating potential of artificial intelligence. An optimized measure of the quality of the neural networks and the algorithms applied for the transformation of coordinates from one system into the other is the generally preferred mean square error in the input, regarded as the basic measure of error purely mathematical in structure.

While looking at the data included in tables 1÷5 it is possible to compare the effectiveness of the algorithms learning neural networks with a particular structure on the basis of the tests which have been carried out. While using neural networks which make use of gradient learning methods, the most favourable optimization have been obtained by means of the Levenberg — Marquardt method, the method of variable metrics, and the RPROP algorithm. The other gradient methods have proved to be ineffective for solving the assigned task (table 1). Another procedure used for the transformation of coordinates is a radial network as a natural complement of sigmoidal networks. Satisfactory results of the transformation of coordinates can be obtained when the right architecture of the network is chosen as well as the right number and type of radial base functions and their width, and as usual, the right value of the learning ratio (tables 2 and 3).

A considerable improvement of the quality of the adaptation of systems of coordinates has been obtained as a result of the use of cascade neural networks and the TSK neuro-fuzzy system operating on the basis of the results of learning neural networks with gradient methods. The results of an optimum activation of output neurons included in tables 4 and 5 prove that strong non-linear systems consisting of a large number of variables should be optimized by creating intermittent results, which, when processed later, will make it possible to assess very accurately the results of the transformation in the whole space in question.

References

1. *Babuska R., Verbruggen H.B.*: Constructing fuzzy models by product space clustering, in «Fuzzy model identification». H. Hellendoorn, D. Driankov, Eds., Springer, Berlin, 1998.
2. *Barsi A.*: Performing coordinate transformation by artificial neural network. Allgemeine Vermessungs-Nachrichten 4/2001.
3. *Bishop Ch.M.*: Pattern Recognition and Machine Learning. Springer, New York, 2006.
4. *Bryś H., Zielina L.*: Aplikacja globalnego systemu współrzędnych prostokątnych UTM na mapach w układzie «1992» na potrzeby inżynierii środowiska i gospodarki wodnej. Przegląd Geodezyjny Nr 2, Warszawa 2005.
5. *Faulkner J.*: Einführung in Neuronale Netze. Universität Tübingen, 2001.
6. *Gil J.*: Przykłady zastosowań sieci neuronowych w geodezji. Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra 2006.
7. *Łęski J.*: Systemy neuronowo — rozmyte. Wydawnictwo Naukowo-Techniczne, Warszawa, 2008.
8. *Osowski S.*: Sieci neuronowe do przetwarzania informacji. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2006.
9. *Rojas R.*: Neural Networks. A Systematic Introduction. Springer — Verlagen, Berlin, 2003.
10. *Zadeh L.A.*: Fuzzy sets. Information and Control, 1965.
11. *Zell A.*: Simulation neulonaler Netze. Oldenbourg. Munchen, 2003.

Получено 20.09.12 г.