

UDK 004.89:528.236
Stručni članak

Methods of Artificial Intelligence used for Transforming a System of Coordinates

Józef GIL, Maria MRÓWCZYŃSKA – Zielona Góra¹

ABSTRACT. The article discusses the problem of transformation in the form of the function $f: R^2 \rightarrow R^2$ as 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 the feed-forward 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.

Keywords: feed-forward neural networks, fuzzy networks, coordinate transformation, geodesy.

1. 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 (Bryś and Zielina 2005). 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:

$$\mathbf{v}^T \mathbf{P} \mathbf{v} \rightarrow \min. \quad (1)$$

¹ Prof. Dr. Sc. Józef Gil, Dr. Sc. Maria Mrówczyńska, Institute of Building Engineering, University of Zielona Góra, 1 Szafrana St., PL-65-516 Zielona Góra, Poland, e-mail: m.mrowczynska@ib.uz.zgora.pl.

where: \mathbf{v} – the vector of corrections

\mathbf{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 the feed-forward 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.

2. Selected Methods of Transforming Coordinates

2.1. Gradient algorithms for learning the feed-forward neural networks

Artificial neural networks (ANNs) have strong theoretical foundations and a wide practical use. Any problem that can be solved by means of classic modelling or statistical methods can be solved by means of neural networks (Jang et al. 1997). Definitely, in most cases feed-forward 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 $\mathbf{x} \in R^N$ into the set of output variables $\mathbf{z} \in R^M$ (Osowski 2006).

A feed-forward neural network, the general outline of which is presented in Fig. 1, has been used in order to carry out calculations connected to the transfor-

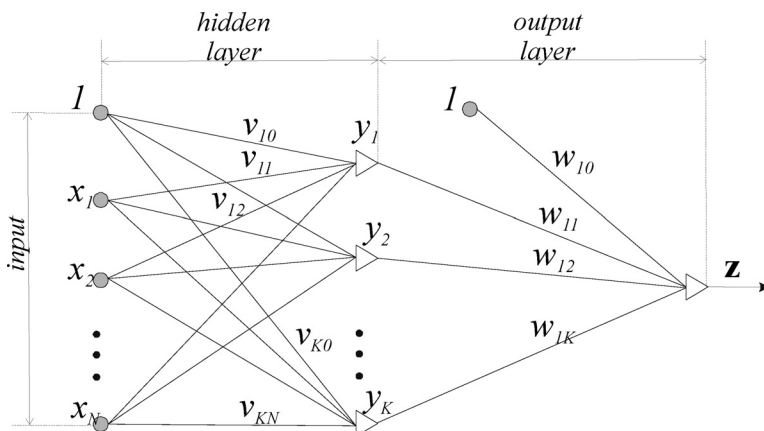


Fig. 1. Feed-forward Neural Network.

mation of coordinates from a primary system into a secondary system (Kavzoglu and Saka 2005).

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:

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

where:

p – the number of input vectors,

M – the number of coordinates of the output vector,

z_j^i – the coordinate of the output vector ($i = 1, 2, \dots, p$), ($j = 1, 2, \dots, M$),

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

In the case of solving the issue of coordinate transformation coordinate output vector number is $M=2$.

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 $\nabla E(\mathbf{w})$ according to the relation:

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

in which $-\nabla E(\mathbf{w})$ denotes the general direction of the minimization, and η the learning ratio (the ratio of the iterative step). 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 (Bishop 2006):

- the greatest descent (linear approximation of the function $E(\mathbf{w})$),
- quasi-Newtonian methods: the method of variable metrics, the Levenberg-Marquardt method of conjugate gradients, and the Resilient Back-Propagation algorithm.

2.2. Radial neural networks (RBFNN)

The stochastic approximation of a multi-variable function achieved by means of feed-forward 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 (Barsi 2001). 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 $\mathbf{x} \in R^n$ into the output vector $\mathbf{z} \in R^M$ 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 (Jang et al. 1997):

$$\mathbf{x} \rightarrow \varphi(\|\mathbf{x} - \mathbf{c}\|), \quad \mathbf{x} \in R^n \quad (4)$$

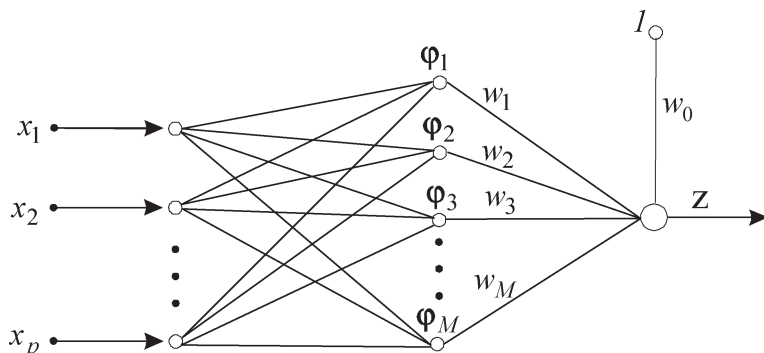


Fig. 2. Radial Basis Network.

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

$$F(\mathbf{x}_i) = d_i. \quad (5)$$

The transformation function which is being estimated has the form:

$$F(\mathbf{x}) = \sum_{i=1}^M w_i \varphi(\|\mathbf{x} - \mathbf{c}_i\|) \quad (6)$$

The symbols included in the formula (6) denote:

w_1, w_2, \dots, w_M – weights of the radial basis functions,

$\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_M \in R^d$ – centres of the radial basis functions, functions φ are located above centres

$\mathbf{x} \in R^d$ – the input vector.

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

$$\varphi(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right). \quad (7)$$

2.3. Recurrent cascade neural networks (RCNN)

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 feed-forward neural network because of the way the output signal is created.

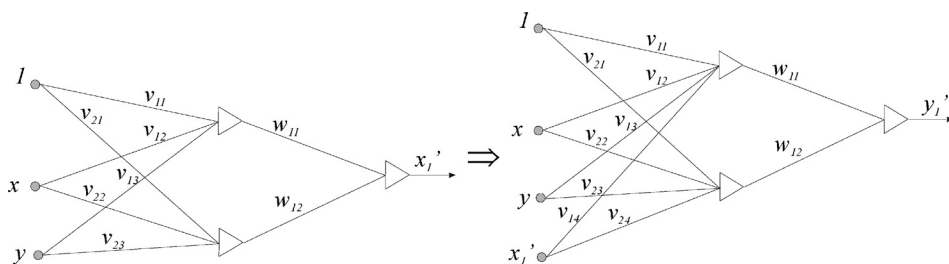


Fig. 3. Non-recurrent cascade neural network.

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 \mathbf{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. $\mathbf{x}' = [x, y, x'_1]$. 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 $\mathbf{y}' = [x, y, x'_1, y'_1]$. The number of learning cycles corresponds to the number of members of the learning set.

In the second stage a recurrent cascade neural network is built, in which the input vector is created by the coordinates of the points in the original system and the secondary system $\mathbf{X} = (x, y, x', y')$ and there are feedback connections between the output layer and the input layer. 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 (Osowski 2006).

3. 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 modelled.

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 k^{th} inference rule, written in the relation (Heine 1999):

$$M^{(k)} = \text{if } \bigwedge_{1 \leq j \leq N} x_j \text{ is } A_j^{(k)}, \text{ then } y = f_k(\mathbf{x}) \text{ for } k = 1, 2, \dots, M \quad (8)$$

and the linear function:

$$f_k(\mathbf{x}) = p_{k0} + \sum_{j=1}^N p_{kj} x_j, \quad (9)$$

where p_k denotes $(N + 1)$ – a dimensional vector of parameters.

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 centre c and the variance σ , determined for the set A in the form (general form) (Jang et al. 1997):

$$\mu_A(x) = \exp \left[- \left(\frac{x - c}{\sigma} \right)^2 \right]. \quad (10)$$

An aggregation of information included in the premises for the implication constitutes the resultant of the membership function $\mu_A(\mathbf{x})$. 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. $\mathbf{x} = \oplus(x_1, x_2, \dots, x_N)$.

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 k^{th} inference rule:

$$\mu_A^{(k)} = \prod_{j=1}^N \left[\frac{1}{1 + \left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}} \right)^{2b_j^{(k)}}} \right]. \quad (11)$$

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

$$z(\mathbf{x}) = \frac{\sum_{k=1}^M \mu_A^{(k)}(\mathbf{x}) f_k(\mathbf{x})}{\sum_{k=1}^M \mu_A^{(k)}(\mathbf{x})} \tag{12}$$

The architecture of a neuro-fuzzy network is presented in Fig. 4.

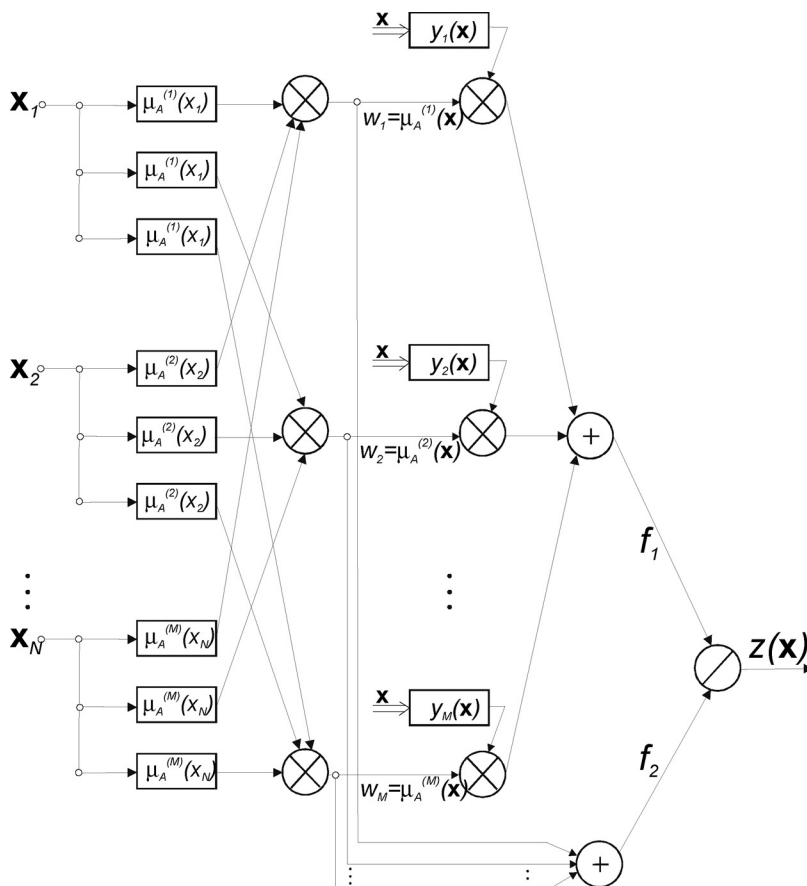


Fig. 4. Structure of a TSK neuro-fuzzy network.

It is also possible to notice that the Takagi-Sugeno-Kang fuzzy system and normalized radial neural networks are equivalent when certain conditions are satisfied.

4. Numerical Example

Technical transformation understood as re-calculating coordinates from the primary system into the secondary system in the case of a two dimensional task consists in the realisation of the function $f: R^2 \rightarrow R^2$ i.e. $\mathbf{X} = f_1(\mathbf{x}, \mathbf{y})$, $\mathbf{Y} = f_2(\mathbf{x}, \mathbf{y})$ (Mrówczyńska 2011). The method most frequently used in the numerical realisation of this problem is the Helmert transformation (imperfection of the method – lack of immunity of outliers). For example, the Helmert transformation (transformation of coordinates from the primary system into the secondary system) is carried out by means of the following formulae (Kadaj 2005):

$$\mathbf{X} = f_1(\mathbf{x}, \mathbf{y}) = X_0 + C\bar{\mathbf{x}} + S\bar{\mathbf{y}} \quad (13)$$

$$\mathbf{Y} = f_2(\mathbf{x}, \mathbf{y}) = Y_0 + C\bar{\mathbf{y}} - S\bar{\mathbf{x}}, \quad (14)$$

where:

$$\bar{\mathbf{x}} = \mathbf{x} - x_0, \quad \bar{\mathbf{y}} = \mathbf{y} - y_0,$$

\mathbf{x}, \mathbf{y} – coordinates of points in the primary system;

\mathbf{X}, \mathbf{Y} – coordinates of points in the secondary system;

x_0, y_0, X_0, Y_0 – coordinates of the centres of weights of sets in both systems.

For all the adaptation points we calculate corrections necessary for data (catalogue) coordinates in the secondary system:

$$v_{xi} = X_i - X'_i \quad (15)$$

$$v_{yi} = Y_i - Y'_i. \quad (16)$$

(i – the adaptation point indicator), and on their basis we calculate a transformation error (Kadaj 2005):

$$m_t = \sqrt{\frac{\sum_{i=1}^n (v_{xi}^2 + v_{yi}^2)}{n - 2}}, \quad (17)$$

where n – the number of adaptation points.

Deviations and the transformation error are the basis for the assessment of the correctness of the coordinates of the adaptation points. If these coordinates are not loaded with significant systematic errors or thick errors, then the transformation error should not exceed the value of the acceptable position of the point m_p in a particular network class (for example: if the adaptation points are points of state class II (divisions in Poland), then the transformation error should not exceed the value $m_p = 0.05 m$, and the values of deviations at the adaptation points should not exceed the value $k \cdot m_p$, where k assumes values between 2 and 3 depending on the number of adaptation points or particular restrictions included in technical conditions.

The transformation ratios C and S in sets (13) and (14) are determined from the dependence:

$$C = m \cos \alpha, \quad S = m \sin \alpha, \tag{18}$$

where:

m – ratio of scale change;

α – rotation angle of coordinate system axis.

When the Helmert transformation is used, in addition, the Hausbrant post-transformation correction is advisable, which leads to equalisation of the deviations of coordinates at adaptation points and the right correction of coordinates of all the points transformed. In order to further explain it is worth adding that the introduction of the Hausbrant post transformation correction causes the coordinates of the adaptation points in the secondary system to remain in the form in which they were adopted for transformation, and all the other transformation points are assigned corrections determined by means of interpolation sets:

$$V_{xj} = \frac{\sum_{i=1}^n \left[v_{xi} \left(\frac{1}{d_{ij}} \right)^2 \right]}{\sum_{i=1}^n \left(\frac{1}{d_{ij}} \right)^2}, \quad V_{yj} = \frac{\sum_{i=1}^n \left[v_{yi} \left(\frac{1}{d_{ij}} \right)^2 \right]}{\sum_{i=1}^n \left(\frac{1}{d_{ij}} \right)^2} \tag{19}$$

Summing after $i = 1, 2, \dots, n; j$ – transformation point indicator.

A graph of the task for the Hausbranta correction has been presented in Fig. 5.

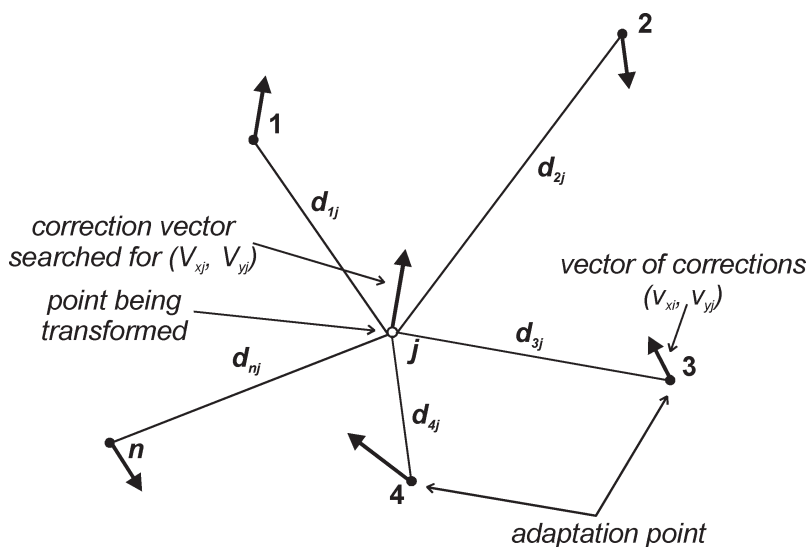


Fig. 5. A graph of the task for the Hausbrant correction.

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 small error in the test data.

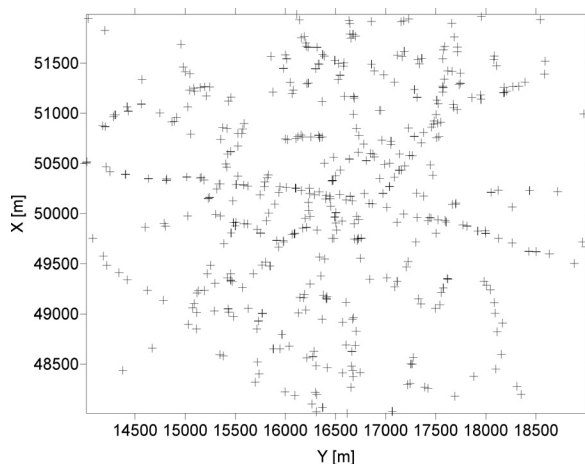


Fig. 6. *The learning set.*

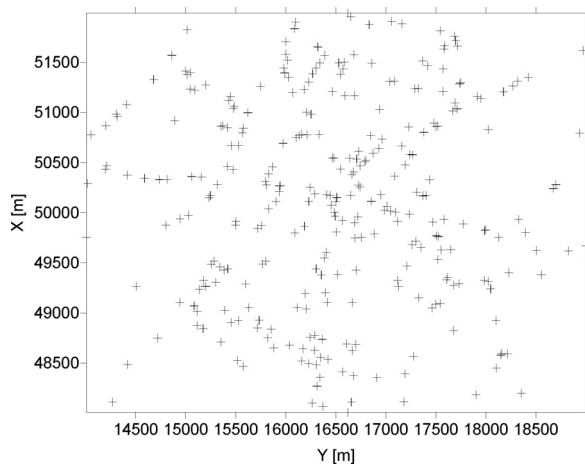


Fig. 7. *The validation set.*

The numerical experiment of coordinates transformation from the system “1965” (Krasowski’s ellipsoid, quasi-stereographic projection (four zones) and Gauss-Kruger projection (one zone)) into the system “2000” (WGS84 ellipsoid and Gauss-Kruger projection) that is an official spatial reference system in Po-

land (Kadaj 2005). Within an area of 20 km² 764 points were located, which had coordinates in both systems. The solution of the task of transformation has been achieved on the basis of a learning set has 458 points (Fig. 6) and a validation set has 306 points (Fig. 7). The points of the learning set represent the adaptation points, and the result of the transformation of the coordinates of the points of the validation 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 root of mean square error *RMSE* (21), has been determined for a particular minimization method. A change in the value of the gradient in two consecutive

Table 1. *RMSE for different gradient optimization methods (feed-forward neural networks).*

The gradient methods	The network architecture	The transformation error RMSE [m]	
		the learning set	the validation set
the Levenberg-Marquardt method	2_3_2_1	0.010	0.011
the quasi-Newtonian method		0.012	0.014
the RPROP algorithm		0.013	0.015
the conjugate gradients method		0.094	0.105
the greatest descent		0.527	0.641
Helmert transformation		0.006	0.007

Table 2. *RMSE for a different number of radial basis functions.*

Number of radial basis functions	The transformation error RMSE [m]	
	the learning set	the validation set
10	0.028	0.026
15	0.011	0.011
20	0.042	0.061

Table 3. *RMSE for different radial basis functions.*

The radial basis functions	The transformation error RMSE [m]	
	the learning set	the validation set
the Gauss function	0.011	0.015
the spline function of fourth degree	0.014	0.024
the spline function of third degree	0.012	0.027
the central function	0.010	1.005
the Hardy's function	0.037	147.812
the linear function	0.107	192.137
the squared function	0.032	203.814
Helmert transformation	0.006	0.007

Table 4. *RMSE for different gradient optimisation methods (recurrent cascade neural networks).*

The gradient methods	The network architecture	The transformation error <i>RMSE</i> [m]	
		the learning set	the validation set
the Levenberg-Marquardt method	2_5_1	0.006	0.007
the quasi-Newtonian method	2_8_1	0.007	0.010
the conjugate gradients method	2_7_1	0.018	0.026
the greatest descent	2_9_1	0.378	0.546
Helmert transformation		0.006	0.007

Table 5. *RMSE for a fuzzy neural system TSK.*

The gradient methods	The transformation error <i>RMSE</i> [m] (the validation set)	
	the neural networks	the Takagi-Sugeno-Kang fuzzy system
the Levenberg-Marquardt method	0.010	0.007
the quasi-Newtonian method	0.012	0.008
the <i>RPROP</i> algorithm	0.014	0.011
the conjugate gradients method	0.091	0.074
the greatest descent	0.822	0.608
Helmert transformation	0.006	0.007

iterations on the level $1e-10$ has been adopted as a criterion for stopping the iteration process. The solution of the task by means of the Helmert method has been assessed on the basis of a transformation error calculated both for points of the teaching set and the test set according to the form given by the formula (17).

In order to provide complete information about the process of learning networks it is necessary to add that for feed-forward 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 basis 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 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:

$$\alpha = \alpha_1 V_h - \alpha_2 D_A - \alpha_3 D_w + \alpha_4 t_A. \quad (20)$$

The local minimum of this function, described in the paper (Babuska and Verbruggen 1997), makes it possible to determine a sub-optimum number of clusters for a particular data set. The ratios of the scale $\alpha_i (i = 1, 2, 3, 4)$ 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 root of mean square error (*RMSE*) calculated as (for $M=1$):

$$RMSE = \sqrt{\frac{1}{P} \sum_{p=1}^P (d_p - z_p)^2}. \quad (21)$$

As a result of the algorithm giving the best results, the algorithm is based on neuro fuzzy system (TSK), we can distinguish three groups of transformation errors had been made on a validation set of size 306 points, namely:

- 62% of the points for the transformation error was less than 0.005 m,
- 37% of the apparent transition point error ranged from 0.005 m to 0.010 m,
- and only 1% of the points have been affected by an error of transformation more than 0.010 m.

Graphical representation of the transformation error values for a set of test points is illustrated in Fig. 8. It can be seen that the largest errors occur at the edges of transformation of the area, which is where the density of points making up the validation set was the smallest. However, part of the zone where the validation set points are located in high-density, transformation errors are possible.

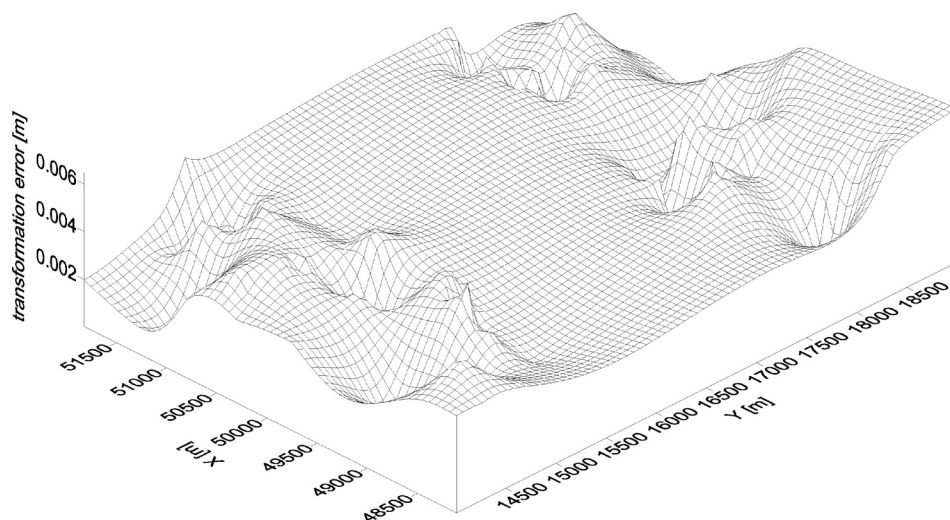


Fig. 8. Transformation error.

5. 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 ANNs which make use of gradient learning methods, the most favourable optimization have been obtained by means of the Levenberg – Marquardt method, the quasi – Newtonian method 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 basis 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.

As a result of the use of the RCNN network and the TSK system the RMSE for the validation set was 0.007 m, i.e. was on the level of the accuracy of results obtained by means of professional software (C-GEO, WinKalk).

The transformation suggested in the paper, which makes use of neural networks and neuro-fuzzy systems can be merely a favourable alternative, in terms of the quality of results, to the Helmert method (including the Hausbrandt corrections), which is characterised by a clear and simple procedure. However, it is necessary to notice that when ANNs are used, we do not have to know the function and transformation ratios, or introduce post-transformation corrections.

References

- Babuska, R., Verbruggen, H. B. (1997): Constructing fuzzy models by product space clustering, in H. Hellendoorn, D. Driankov (Eds.): "Fuzzy model identification", Springer, Berlin.
- Barsi, A. (2001): Performing coordinate transformation by artificial neural network, *Allgemeine Vermessungs-Nachrichten*, 108, 134–137.
- Bishop, C. M. (2006): *Pattern Recognition and Machine Learning*, Springer, New York.
- Bryś, H., Zielina, L. (2005): Application of global system of rectangular coordinates UTM in the maps in coordinate system 1992 for engineering for the environment and water (in Polish), *Przegląd Geodezyjny*, 2, Warszawa.
- Faulkner, J. (2001): *Einführung in Neuronale Netze*, Universität Tübingen.
- Heine, K. (1999): *Beschreibung von Deformationsprozessen durch Volterra- und Fuzzy-Modelle sowie Neuronale Netze*, Deutsche Geodätische Kommission, Reihe C, 516, München.
- Jang, J. S. R., Sun, C. T., Mitzutani, E. (1997): *Neuro-Fuzzy and Soft Computing*, Prentice Hall, New York.
- Kadaj, R. (2005): Problems of geodesic and cartographic resources transformation to "2000" system (in Polish), I Ogólnopolska Konferencja Naukowo – Techniczna "Kartografia numeryczna i informatyka geodezyjna", Rzeszów.
- Kavzoglu, T., Saka, M. H. (2005): Modelling local GPS/levelling geoid undulations using artificial neural networks, *Journal of Geodesy*, 78(9), 520–527.
- Leandro, R. F., Santos, M. C. (2007): A neural network approach for regional vertical total electron content modelling, *Studia Geophysica et Geodaetica*, 51(2), 279–292.
- Lin, L. S. (2009): Application of neural network and least squares collocation to GPS height Transformation, *Proceedings of ARCS, Remote Sensing*, Beijing.
- Łêski, J. (2008): *Neural fuzzy systems* (in Polish), Wydawnictwo Naukowo-Techniczne, Warszawa.
- Mrówczyńska, M. (2011): Neural networks and neuro-fuzzy systems applied to the analysis of selected problems of geodesy, *CAMES*, 18(3), 161–173.
- Osowski, S. (2006): *Neural networks for information processing* (in Polish), Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa.
- Rivas, I., Personnaz, L. (2003): MLPs (Mono – Layer Polynomials and Multi – Layer Perceptrons) for Nonlinear Modeling, *Journal of Machine Learning Research*, 3, 1383–1398.
- Rojas, R. (1996): *Neural Networks, A Systematic Introduction*, Springer – Verlag, Berlin.
- Tadeusiewicz, R. (1993): *Neural Networks* (in Polish), Akademicka Oficyna Wydawnicza, Warszawa.
- Turgut, B. (2010): A back-propagation artificial neural network approach for three-dimension coordinate transformation, *Scientific Research and Essays*, 5(21), 3330–3335.
- Zadeh, L. A. (1965): Fuzzy sets, *Information and Control*, 8, 338–353.
- Zell, A. (2003): *Simulation neuronaler Netze*, Oldenbourg, München.

Metode umjetne inteligencije korištene za transformaciju koordinatnog sustava

SAŽETAK. Članak razmatra problem transformacije u obliku funkcije $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ kao ovisnosti između koordinata prvobitnog sustava $[x, y]$ i koordinata sekundarnog sustava $[X, Y]$. Zadatak transformacije $[X, Y] = f(x, y)$, koja tehnički znači transformaciju koordinatnog sustava, riješen je uz pomoć feed-forward neuronskih mreža sigmoidalne vrste, radijalnih neuronskih mreža, povratnih ukomponiranih neuronskih mreža i neuro-fuzzy sustava koristeći model Takagi-Sugeno-Kang. Primijenjeni numerički postupci omogućavaju dobivanje razine točnosti zadatka koja je jednaka kartografskoj točnosti snimaka Prostornih informacijskih sustava.

Ključne riječi: feed-forward neuronske mreže, fuzzy mreže, transformacija koordinata, geodezija.

Primljeno: 2012-09-20

Prihvaćeno: 2012-12-05