Computer Assisted Mechanics and Engineering Sciences, **18**: 161–173, 2011.

# Neural networks and neuro-fuzzy systems applied to the analysis of selected problems of geodesy

Maria Mrówczyńska
*University of Zielona Góra*
*Z. Szafrana 1, 65-516 Zielona Góra, Poland*
*e-mail: M.Mrowczynska@ib.uz.zgora.pl*

The article presents possibilities of using different artificial neural networks and neuro-fuzzy systems to solve certain engineering geodesy tasks. Special attention is paid to tasks connected with the construction of a numerical terrain model, transformation of coordinates from the "1965" system into the "2000" system, and prediction of a time series on the basis of results of GPS measurements. The paper also includes a short description of those neural networks and neuro-fuzzy systems that provided good quality solutions of the tasks undertaken. The goal of the article is to review the papers published in the years 2005–2010.

**Keywords:** neural networks, neuro-fuzzy systems, geodesy.

## 1. Introduction

Neural networks and neuro-fuzzy systems have been included in the arsenal of strong adaptation procedures because of the possibility of applying them in specialised forms of approximation of functions (classification, auto-association, prediction of time series). Both types of networks have one common feature, which is parallel information processing. Neuro-fuzzy systems are neural networks characterised by processing of fuzzy sets. An advantage of neuro-fuzzy networks is the possibility of interpreting knowledge contained in the weights of neural connections. Classic neural networks process a numerical operation but they lack the so-called explaining module, because knowledge represented by the values of weights is dispersed and does not have a physical representation in a form understandable to the user [11, 13]. In geodesy neural networks were applied, for example, to transformation of coordinate property valuation, analysis of GPS signals, navigation and object recognition of marine, creating digital elevation model and identification of land development elements in aerial photographs [1, 4, 12, 20].

## 2. Algorithms used

In order to solve engineering geodesy problems the following have been used: neural networks (multi-layer perceptron) taught by means of the method of back propagation of errors with the use of gradient optimisation methods, networks with radial base functions, recurrent cascade networks, and neuro-fuzzy systems with the use of the Wang-Mendel model and the Takagi-Sugeno-Kang model.

### 2.1. Neural networks

The feedforward neural networks (NNs) are sometimes called *universal approximators* since they fit perfectly the analysis of regression problems [6]. In the problem investigated two basic types of

NNs, i.e. multilayer perceptron (Fig. 1) and radial basis NNs are discussed. They can be applied to approximate multivariable functions, considered as mapping of input vector $\mathbf{x} \in \mathcal{R}^N$ onto output function $\mathbf{z}(\mathbf{x}, \mathbf{w}) \in \mathcal{R}^M$. The weight vector $\mathbf{w}$ is computed by means of minimisation of objective function $E(\mathbf{w})$, which for $P$ output neurons can be written by means the Euclidean metric as:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{p=1}^{P} \|\mathbf{d}_p - \mathbf{z}_p\|. \tag{1}$$

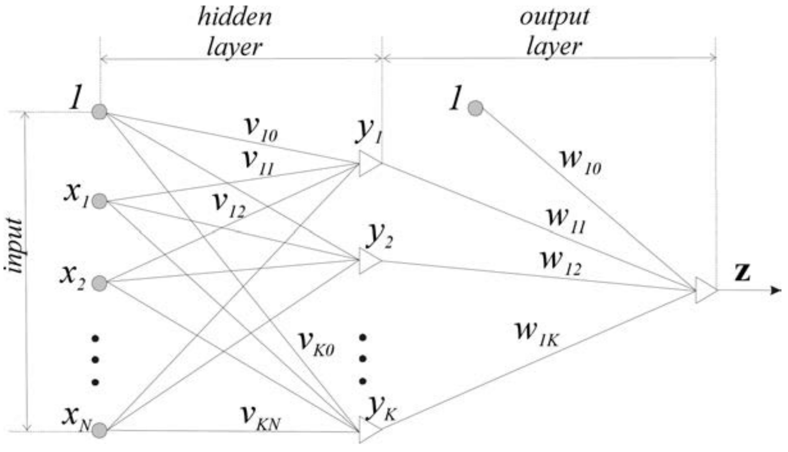

**Fig. 1.** Architecture of feedforward neural network.

In the paper norm $L_2$ is used, called the Root Mean Square Error (RMSE). In the case of a single output, i.e. for $M = 1$ this error has the form:

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

where $d_p$, $z_p$ – desired and computed values of outputs for patterns $p = 1, \ldots, P$. The evaluation method of the accuracy of information processing by means of the RMSE index is most often mentioned in literature and preferred in practice.

### 2.2. 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 $\mathbf{x} \in \mathcal{R}^N$ into the output vector $\mathbf{z} = \mathcal{R}^M$ is a result in the form of local transformations achieved by means of networks with radial base functions, consisting of neurons which carry out the transformation in the hidden layer [18]:

$$\varphi(\mathbf{x}) = (\|\mathbf{x} - \mathbf{c}\|), \tag{3}$$

where $\mathbf{c}$ is the set of centres, and $\mathbf{x}$ is the input vector. The radial network has a specific structure (Fig. 2) with one hidden layer and linear output neurons. The first layer consists of non-linear

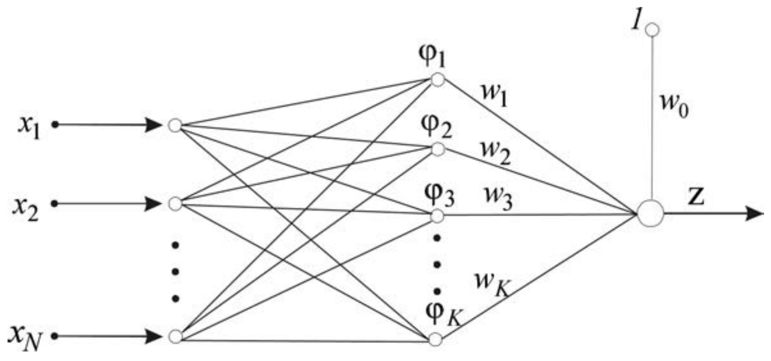

**Fig. 2.** Architecture of Radial Basis Network.

radial functions, the parameters of which (the centres $\mathbf{c}$ and ratios $\sigma$ which determine the width of the radial function) undergo adaptation in the learning process. In the case when the number of radial base functions $K$ is equal to the number of learning patterns $N$, and the condition $\mathbf{x}_1 \neq \mathbf{x}_2 \neq \ldots \neq \mathbf{x}_N$ is satisfied, the set of linear equations in terms of weights

$$\mathbf{\Phi w} = \mathbf{d} \tag{4}$$

can be solved as follows:

$$\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{d}, \tag{5}$$

where square matrix $\mathbf{\Phi}$ $(\varphi_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{c}_j\|))$ is a positive definite matrix.

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

$$\varphi(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right). \tag{6}$$

The problem of selection of parameters of radial functions ($\mathbf{c}_i$ and $\sigma_i$) is solved as minimisation of the objective function, which can be written with the use of the Euclid norm in the form:

$$E = \frac{1}{2}\sum_{i=1}^{N}\left[\sum_{j=1}^{K} w_j \varphi(\|\mathbf{x}_i - \mathbf{c}_j\| - d_i)\right]^2. \tag{7}$$

If the parameters are known, the solution of minimisation is achieved by means of the Morre-Penrose pseudo-inversion of the matrix [9, 18]

$$\mathbf{w} = \mathbf{G}^{+}\mathbf{d}. \tag{8}$$

Restriction of the number of radial base functions to $K$, a smaller number than the number of learning patterns $N$, ($K \ll N$), reduces the problem of learning networks with generalisation characteristics to two interlacing stages:

- choice of linear parameters of the network (i.e. weights of the output layer $\mathbf{w} = [w_0, w_1, \ldots, w_K]^T$) with the use of the pseudo-inversion technique,
- adaptation of non-linear parameters of radial functions ($\mathbf{c}_i$ and $\sigma_i$), which is carried out with the use of the gradient method of the greatest fall according to the formulae:

$$c_{ij}(t+1) = c_{ij}(t) - \eta \frac{\partial E}{\partial c_{ij}}, \tag{9}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta \frac{\partial E}{\partial \sigma_{ij}}. \tag{10}$$

Adaptation of non-linear parameters of radial functions concludes one learning cycle.

In order to speed up the process of learning a network it is necessary to adopt initial values of parameters of radial functions close to the optimum values. For this purpose, it is also possible to use several adaptation cycles of non-linear parameters for one cycle of selecting linear parameters of the network.

The so-called regulatory factor can be added to the criterion (7), which results from the use of Tichonov regulatory method, then the criterion (7) assumes the form [19]

$$\widetilde{E} = \frac{1}{2}\sum_{i=1}^{N}\left[\sum_{j=1}^{K} w_j \varphi\left(\|\mathbf{x}_i - \mathbf{c}_j\| - d_i\right)\right]^2 + \lambda\,\|\mathbf{P}f\|^2 = \|\mathbf{G}\mathbf{w} - \mathbf{d}\|^2 + \lambda\,\|\mathbf{P}f\|^2\,, \tag{11}$$

where $\mathbf{P}$ is a certain linear operator. Introduction of this operator presumes smoothness of the function, which is supposed to approximate an unknown solution , then it is possible to write:

$$\|\mathbf{P}f\|^2 = \mathbf{w}^T\mathbf{G}_0\mathbf{w}, \tag{12}$$

where matrix $\mathbf{G}_0$ is a square matrix $K \times K$ including the values of radial base functions. With these assumptions, after minimisation of the dependence (11), we will obtain the vector of weights $\mathbf{w}$ we are looking for:

$$\mathbf{w} = \left(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{G}_0\right)^{-1}\mathbf{G}^T\mathbf{d}. \tag{13}$$

### 2.3. Recurrent cascade multi-layer perceptron

Recurrent Cascade Multilayer Perceptron has been used to solve the task of coordinates transformation. 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. 3a [10]. 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 input layer receives stimulation from the input layer in the form of 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 $\boldsymbol{\omega}$, which includes both the coordinates $\boldsymbol{\omega} = (x, y)$ in the original system and the coordinate $x'_1$ obtained from the output, i.e. $\boldsymbol{\omega} + \mathbf{x}' = [x, y, x'_1]$. The application of this vector in the input starts another learning cycle with the expected output signal in the form of coordinate $y'_1$ in the secondary system. As a result of this course of action we obtain vector $\mathbf{y}' = [x, y, x'_1, y'_1]$.

In the second stage a recurrent cascade neural network is built (Fig. 3b), 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. 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 iteration $k$. Algorithms for learning a recurrent network make use of the above mentioned gradient optimisation methods, and, as in the case of a one direction neural network, the gradient of the objective function in relation to each weight is calculated.

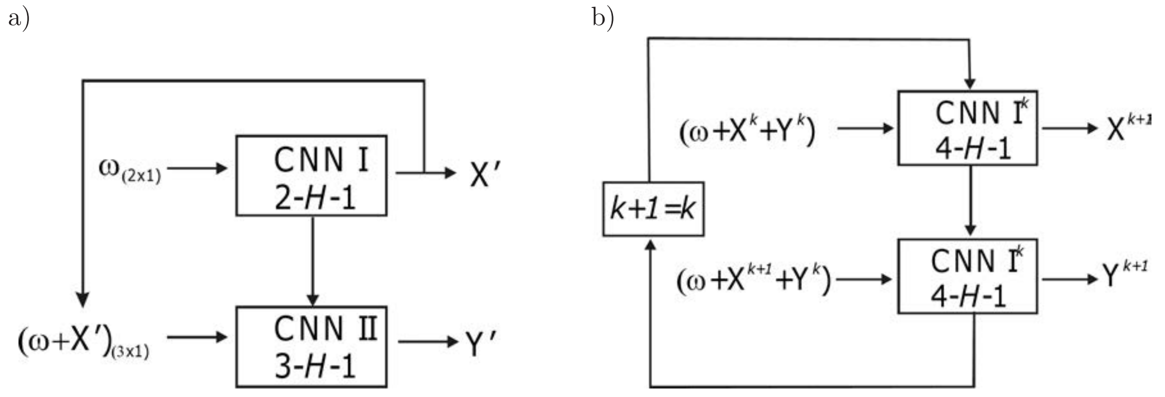

**Fig. 3.** a) Cascade NNs for the cycle $k = 1$, b) recurrent cascade NNs for cycles $k > 1$.

### 2.4. Neuro-fuzzy system

Neuro-fuzzy systems are neural networks that have the ability to transform fuzzy sets. Neuro-fuzzy systems make it possible to interpret the 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 modelled [7].

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 [13]

$$M^{(k)} = \text{ if } \bigwedge_{1\le j\le N}^{N} x_j \text{ is } A_j^{(k)} \text{ then } y = f_k(\mathbf{x}) \text{ for } k = 1, 2, \ldots, M \tag{14}$$

and the linear function

$$f_k(\mathbf{x}) = p_{k0} + \sum_{j=1}^{N} p_{kj} x_j, \tag{15}$$

where $p_k$ denotes $(N + 1)$ – a dimensional vector of parameters. A set of simple linear functions $f_k(\mathbf{x})$ makes it possible to model complicated dependencies between the input and output of the system.

One of the most widely used membership functions (MFs) in fuzzy representation of numbers is the Gauss function, defined for variable $x$, centre $c$ and variance $\sigma$, determined for set $A$ in the form (general form) [18]

$$\mu_A(x_i) = \exp\left[-\left(\frac{x_i - c_i}{\sigma_i}\right)^2\right]. \tag{16}$$

Aggregation of information included in the premises for the implication constitutes the resultant of $\mu_A(\mathbf{x})$. 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. $\mathbf{x} = \oplus(x_1, x_2, \ldots, x_N)$.

According to fuzzy procedure, 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

$$\mu_A^{(k)}(\mathbf{x}) = \prod_{j=1}^{N} \left[\left(\frac{x_j - c_j^{(k)}}{\sigma_j^{(k)}}\right)\right]. \tag{17}$$

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

$$y(\mathbf{x}) = \frac{\sum_{k=1}^{M} \mu_A^{(k)}(\mathbf{x}) f_k(\mathbf{x})}{\sum_{k=1}^{M} \mu_A^{(k)}(\mathbf{x})}. \tag{18}$$

Two neuro-fuzzy systems have been used in the paper: the Takaga-Sugeno-Kang (TSK) system and the Wang-Mendel system (WM). The task of both systems is to represent learning data pairs $(\mathbf{x}, d)$ in such a way that the set value $d$, which corresponds to the input value $\mathbf{x}$, is represented by the output function $y(\mathbf{x})$. The structures of both systems are identical in the part of the predecessor, including inference rules "if ... ", but they are different in the part of the successor "then ...". In the TSK system the conclusion assumes the form of a first rank polynomial, and in the WM network the conclusion corresponds to the centre of the function of the membership of the successor. Therefore, it is possible to say that the WM network is an exceptional case of the TSK network, in which the polynomial describing the conclusion is of zero rank [18]. The architecture of a TSK neuro-fuzzy network is presented in Fig. 4.

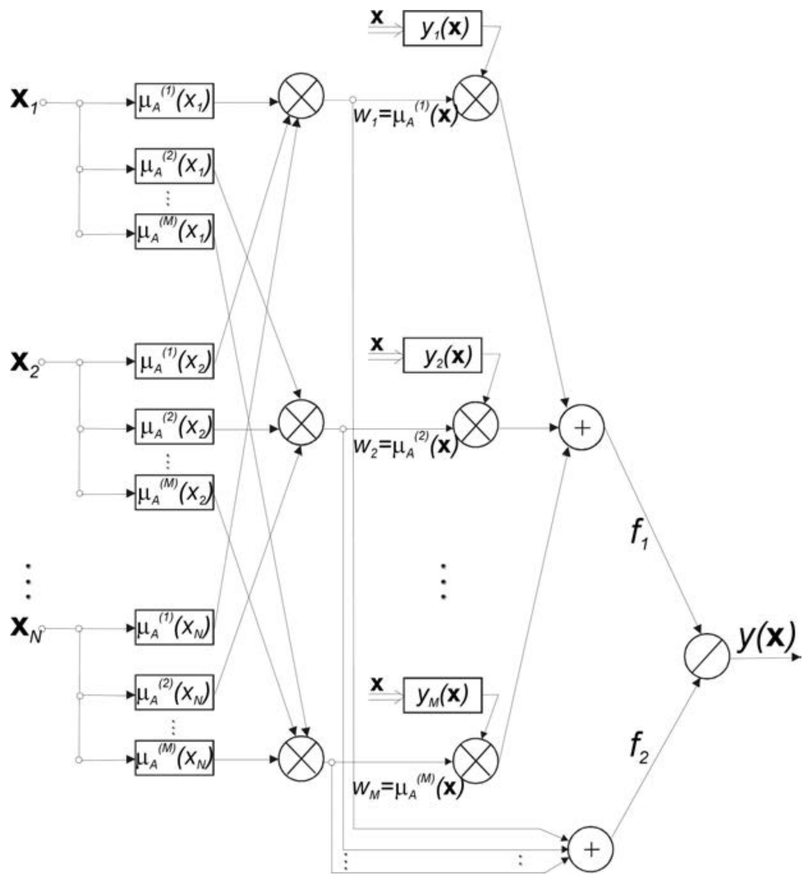

**Fig. 4.** Structure of a TSK neuro-fuzzy network.

## 3. Numerical examples

The working quality of neural networks and neuro-fuzzy systems in a particular branch of geodesy has been illustrated by the examples below.

### 3.1. Example 1. Approximation of a non-linear function of several variables

In this case the problem consists in identifying a model of reality. In order to show the approximation abilities of both systems approximation of a complex non-linear function with two variables $\mathbf{x} = [x_1, x_2]$ of the form

$$f_1(\mathbf{x}) = 0.1 + (1.0 + \sin(2x_1 + 3x_2))/(3.5 + \sin(x_1 - x_2)) \tag{19}$$

is graphically illustrated in Fig. 5.

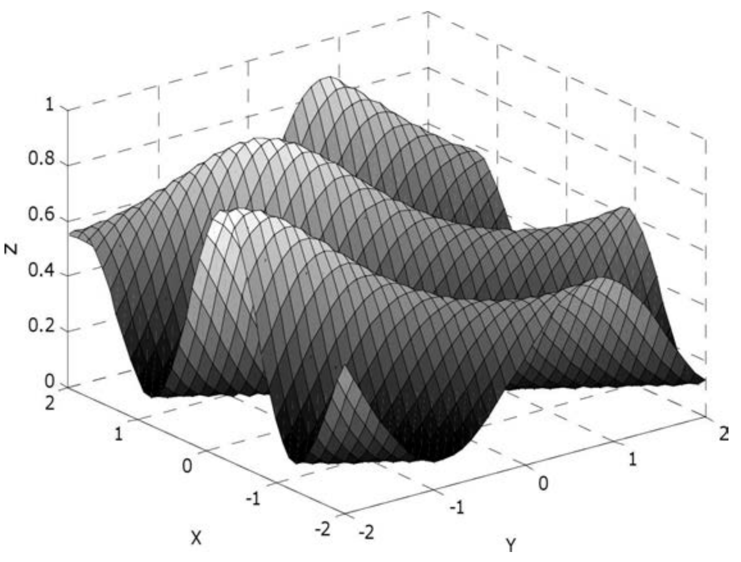

**Fig. 5.** The form of function (19) tested.

The accuracy of approximation for the test set with 1600 points was carried out by means of the multi-layered perceptron (MLP), radial basis function neural network (RBFNN), TSK system and the WM system. In the case of learning artificial neural networks the key factor in terms of the quality of results is the choice of parameters of the network and their architecture. In the process of learning, neural networks of 2_12_1 architecture were used, which were taught by means of the error back propagation method with the use of gradient optimisation methods. For the TSK system a sub-optimum number of 25 inference rules was specified. The same number of radial base functions in the RBFNN network was adopted, which made it possible to build a network of 2_25_1 architecture. The data set for all the algorithms presented was randomly divided in half into the learning set and the test set. In Table 1, errors RMSE(L) and RMSE(T) are listed for all the ANNs applied and the networks are listed according to the value of testing errors RMSE(T).

**Table 1.** Errors of neural approximations where MLP – Multi-Layered Perceptron, RBFNN – Radial Basis Function Neural Network.

| Network models and (applied algorithms) | Root Mean Square Errors (RMSE) of | |
|---|---|---|
| | learning L [m] | testing T [m] |
| Takagi-Sugeno-Kang fuzzy system | 0.0073 | 0.0074 |
| MLP (Levenberg-Marquardt learning rule) | 0.0071 | 0.0075 |
| MLP (classical gradient descent algorithm) | 0.0469 | 0.0541 |
| RBFNN | 0.1142 | 0.1728 |
| third – degree spline function | 0.0199 | |

While analysing the results, it is possible to notice that the most favourite approximation results were obtained by means of the TSK and MLP (Levenberg-Marquardt learning rule) networks, and they were much more favourable than the results obtained by means of a third rank spline function. Much worse results were obtained by means of the RBFNN network, which might have been caused by the fact that the function (19) is difficult to approximate.

### 3.2. Example 2. Digital terrain model

Real data in the form of spatial coordinates $(x, y, z)$ of 483 points situated in an area of 3 km$^2$ were used for identifying the terrain model by means of the systems discussed. The size of the learning set was 248 points, and the size of the test set was 234 points, the division into two sets was random. The efficiency of the approximation was evaluated on the basis of testing errors classification in the form of a divergent multi-level series. The classification of the absolute values of testing errors was carried out on the basis of an optimum choice of the length of class ranges containing maximum information, according to formula [2, 3]:

$$s = \left[ t + \ln \frac{tT}{s(s-1)} \right], \tag{20}$$

where $T$ – the range of the feature investigated, $t$ – the length of class range, $s$ – the value of the error classified. The solution of this equation is the number $k \approx t/s$, whose value expresses an optimum numerical relation between the length of class range and the value of the variable classified.

In order to show approximation possibilities of the algorithms presented, a RBFNN network of 2_65_1 architecture, and the TSK system, for which 32 inference rules were adopted, were taught by means of the MLP network of 2_10_1 architecture with the use of gradient optimisation methods. The results were compared with those obtained by means of approximation with a second and third degree polynomial, and approximation by means of the spline method and kriging method.

Evaluation of approximation accuracy by means of the above mentioned procedures was carried out on the basis of the root of the mean square error *RMSE*. The results of the representation of the terrain model have been illustrated in Fig. 6 [17]. The working quality of the MLP, RBFNN, TSK systems in the case of the problem of approximating a terrain model expressed in the form of a mean square error has been presented in Table 2.

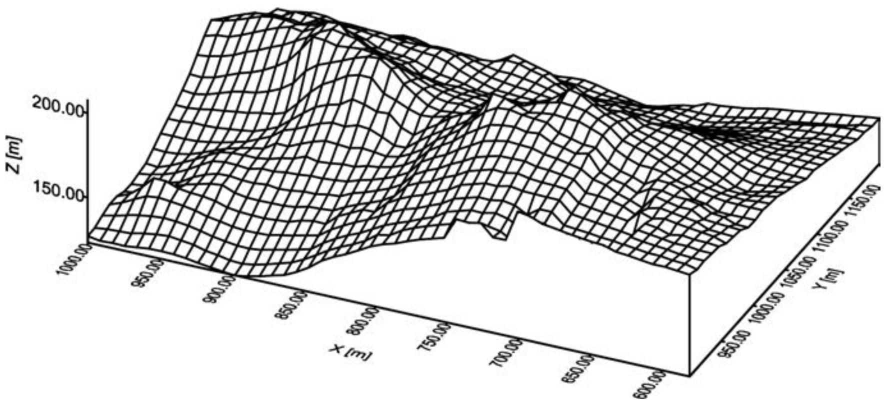

**Fig. 6.** Result of representation of a terrain model by means of TSK system.

While analysing the results, it is possible to notice that as far as quality is concerned the polynomial approximation of the model of a real terrain fragment differs from the other methods. In terms of quality neural networks and the TSK system are on the same level as the results

**Table 2.** Errors of neural approximations.

| Network models and (applied algorithms) | Root Mean Square Errors (RMSE) of | |
|---|---|---|
| | learning L [m] | testing T [m] |
| TSK fuzzy system | 0.11 | 0.19 |
| MLP (conjugate gradient rule with Fletcher-Powell algorithm) | 0.18 | 0.20 |
| RBFNN | 0.07 | 0.28 |
| MLP (Levenberg – Marquardt learning rule) | 0.14 | 0.30 |
| spline | 0.06 | 0.16 |
| kriging | 0.07 | 0.17 |
| third degree polynomial | 0.45 | 0.44 |
| second degree polynomial | 0.48 | 0.46 |

of approximation by means of the spline method and kriging method, and achieve the precision satisfying the requirements of the spatial information system.

It results from the above data concerning approximation under way that the accuracy of the task carried out by means of both systems is almost identical.

## 3.3. Example 3. Transformation of coordinates

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 $f : R^2 \to R^2$ i.e. $\mathbf{X} = f_1(\mathbf{x}, \mathbf{y})$, $\mathbf{Y} = f_2(\mathbf{x}, \mathbf{y})$ [1, 12]. The method most frequently used in the numerical realisation of this problem is the Helmert transformation (imperfection of the method – lack of immunity to thick errors). 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:

$$\mathbf{X} = f_1(\mathbf{x}, \mathbf{y}) = X_0 + C\overline{\mathbf{x}} + S\overline{\mathbf{y}}, \tag{21}$$

$$\mathbf{Y} = f_2(\mathbf{x}, \mathbf{y}) = Y_0 + C\overline{\mathbf{y}} - S\overline{\mathbf{x}}, \tag{22}$$

where

$$\overline{\mathbf{x}} = \mathbf{x} - x_0, \qquad \overline{\mathbf{y}} = \mathbf{y} - y_0,$$

$\mathbf{x}$, $\mathbf{y}$ – 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.

The transformation ratios $C$ and $S$ in sets (21) and (22) are determined from the dependence [22]:

$$C = m\cos\alpha, \qquad S = m\sin\alpha, \tag{23}$$

where $m$ – ratio of scale change, $\alpha$ – twist 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.

The numerical experiment of coordinates transformation from the system “1965” into the system “2000” that is an official spatial reference system in Poland [8]. Within an area of 50 km$^2$ 3200 points were located, which had coordinates in both systems, the points were divided at random into the learning set and the test set being two equal parts of 50%. The task of the transformation technically understood as the transformation of a system of coordinates has been solved by means of

MLP, RBFNN, RCNN and TSK. In order to solve the task the MPL network of 2_5_2_2 architecture, with non-linear neurons and hidden neurons as well as a linear output, and the RBFNN network of 2_55_1 architecture were used. The TSK system was also used, and a sub-optimum number of 35 inference rules was specified. In order to solve the task the RCNN network was also used, the architecture of which had one layer of hidden neurons $H = 8$ for both the NNs cascade (the cycle $k = 1$) and the NNs recurrent cascade (cycles $k > 1$).

The most favourable results were obtained by means of the TSK system and RCNN, for which the random mean square error (RMSE) was 0.008 m [14]. Detailed results of the transformation are presented in Table 3.

**Table 3.** Errors of transformation.

| Network models and (applied algorithms) | Root Mean Square Errors (RMSE) of | |
|---|---|---|
| | learning L [m] | testing T [m] |
| RCNN (Levenberg-Marquardt learning rule) | 0.007 | 0.008 |
| TSK fuzzy system | 0.007 | 0.008 |
| MLP (Levenberg-Marquardt learning rule) | 0.011 | 0.012 |
| RBFNN (Gaussian function) | 0.013 | 0.014 |
| Helmert transformation | 0.004 | |

As a result of the use of the RCNN network and the TSK system the RMSE for the test set was 0.008 m, i.e. was on the level of the accuracy of results obtained by means of professional software (C-GEO, WinKalk). While analysing the results, the algorithm invented by Prof. Kadaj should also be appreciated, which includes post-transformation corrections and enables reaching the level of accuracy of 0.004 m.

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.

## 3.4. Example 4. Prediction of a time series

A time series is a series of specified values, registered at fixed time intervals [15, 21]. The problem of prediction consists in estimating future values of a time series on the basis of future values of the components of the series. From information available about variable $\mathbf{x}$ at past moments as the set $\{x(k-1), x(k-2), ..., x(k-p)\}$, a neuro-fuzzy system determines the value $y(k)$ at moment $k$.

Research into the application of a neuro-fuzzy system for predicting a time series was carried out on GPS-RTK data, representing changes in the module of the "zero" vector between the base station and a "moving" receiver [21]. The GPS-RTK technique was used to monitor changes of shapes as well as monitoring engineering objects undergoing continuous changes (deformations). Measurements intended for registration of displacements of the controlled points on the level 5–10 mm were taken at points of the control-measurement network of the upper reservoir of the pumped-storage power station in Żarnowiec. The satellite observations collected make it possible to automate the process of monitoring changes taking place in the object, and they can also be used for periodical statistics. Thus, it is possible to determine displacements of points in the object as well as predict GPS signals, which can be a basis for the assessment of future changes in the position of points of the control-measurement network.

The number of changes in the module of vector $\mathbf{x}(k)$ being predicted was reduced to $k = 2500$, from which the first 1250 were adopted as a learning part of the data set. The time series, the prediction carried out with a TSK system and the prediction error have been presented in Figs. 7, 8 and 9. A vertical line separates the learning part and the testing part, and the errors for both parts are also presented.

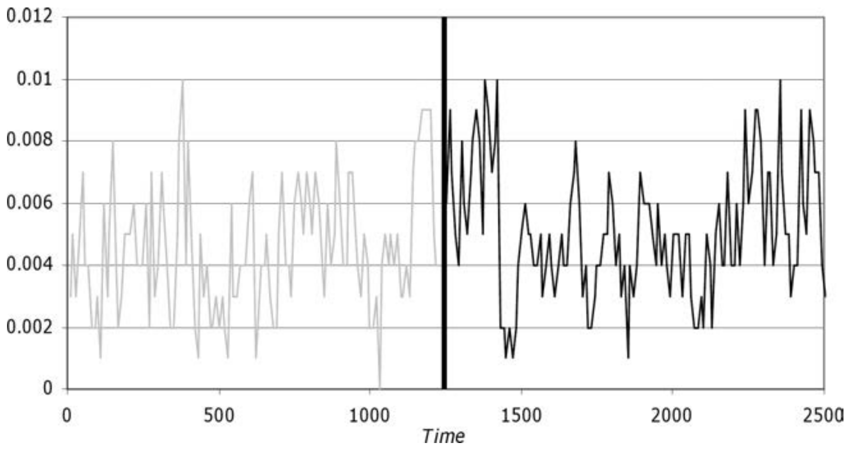

**Fig. 7.** Time series.

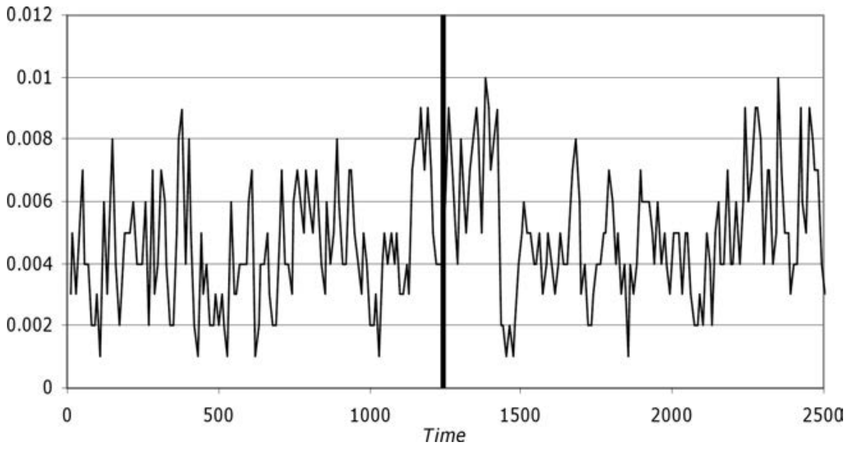

**Fig. 8.** Prediction by means of a TSK neuro-fuzzy system.

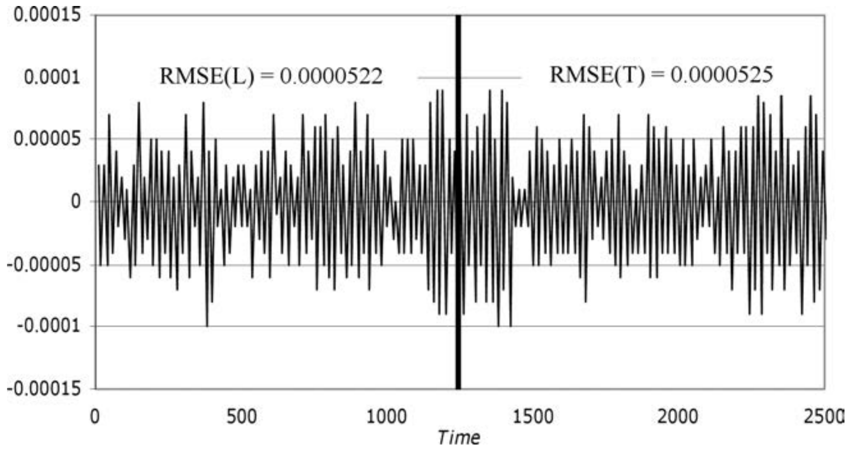

**Fig. 9.** Prediction error.

The values of prediction estimated are presented as a function called the predicator of series $p$. The predicator which results from the application of a neuro-fuzzy system is a non-linear predicator.

## 4. Conclusions

Apart from the examples of the application of neural networks and neuro-fuzzy systems in the field of geodesy discussed above, the author has also attempted to use the systems for assessing the state of deformation of an object, resulting from mining and designate the displacement of buildings situated on expansive soils. Neural networks were also used for equalising levelling nets, horizontal linear-angular nets and predicting vertical deviations of a steel stack [4]. In particular, neural networks and neuro-fuzzy networks were also used for:

- estimating errors of a surface model described with shape functions,
- determining displacements of a building situated on expansive soils [5],
- determining deviations of the structure of a steel stack ventilating underground gas tanks,
- solving selected tasks from the field of linear algebra by means of neural networks,
- building a geological model of the Głogów-Barudz proglacial stream valley near Nowa Sól,
- determining vertical displacements of points controlled in the Legnica-Głogów Copper District area in the years 1967–2008 [16],
- determining vertical displacements of controlled points, which were destroyed or damaged during the research (the Hecht-Nielsen neural network),
- determining deformations of a bridge under test loads.

The analysis of the problems presented in the article, and those discussed in other papers, leads to a conclusion that while solving some tasks engineering geodesy tasks the most favourable results were obtained with the use of MLP networks taught by means of the error back propagation method. The most effective gradient optimisation method was the Levenberg-Marquardt method. Good results were also obtained when the TSK system was used. However, it is necessary to observe that in the case of the latter system in order to obtain good results it is necessary to specify the right number of inference rules. In order to solve tasks of transformation of coordinates the RCNN network was also used, which made it possible to carry out transformation from the "1965" system into the "2000" system with the accuracy comparable with the accuracy obtained by means of commercial software.

The numerical procedures presented in the article resulted from the implementation carried out by the author with the use of files available in the MATLAB environment. In particular, MATLAB Neural Network Toolbox and Fuzzy Logic Toolbox simulators were applied. Some of the posed problems were further resolved with the use of their own programs written in MATLAB and FORTRAN program.

## References

[1] A. Barsi. *Performing coordinate transformation by artificial neural network*, Allgemeine Vermessungs-Nachrichten 4/2001.

[2] L. Brillouin. *Science and information theory* [in Polish], PWN, Warszawa, 1969.

[3] W. Feller. *An introduction to probability theory and its applications* [in Polish], PWN, Warszawa, 1980.

[4] J. Gil. *Examples of applications of neural networks in geodesy* [in Polish], Publishing House of the University of Zielona Góra, Zielona Góra, 2006.

[5] J. Gil, M. Mrówczyńska. *Applying a neural network of Hopfield type for recognizing the dynamics of unequal displacements of buildings located on the expansive grounds* [in Polish], Publisher of the Silesian University of Technology, Gliwice 2008.

[6] J. Hertz, A. Krogh, R.G. Palmer, *Introduction to the theory of neural computation* [in Polish], WNT, Warszawa, 1993.

[7] J.S. Jang, C.T. Sun, E. Mitzutani. *Neuro-Fuzzy and Soft Computing*, Prentice Hall, New York, 1997.

[8] R. Kadaj. *Problems of geodesic and cartographic resources transformation to "2000" system* [in Polish], 1th Technical and Scientific National Conference "Numerical cartography and geodetic science", Rzeszów 2005.

[9] A. Kiełbasiński, H. Schwetlick. *Numerical linear algebra* [in Polish], WNT, Warszawa 1992.

[10] M. Kłos, Z. Waszczyszyn. *Application of cascade neural networks for identification of elastic circular arch parameters*, Computers &Structures, **89**: 581–589, 2011.

[11] R.A. Kosiński. *Artificial neural networks. Nonlinear dynamics and chaos* [in Polish], WNT, Warszawa, 2004.

[12] L.S. Lin. *Application of neural network and least squares collocation to GPS height Transformation*, Asian Association on Remote Sensing, Beijing, 2009.

[13] J. Łęski. *Neuro fuzzy systems* [in Polish], WNT, Warszawa, 2008.

[14] M. Mrówczyńska. *A numerical experiment of transformation of coordinate system by means of neural networks and neuro fuzzy systems* [in Polish], Review of the Geodetic, No. 12, Warszawa, 2009.

[15] M. Mrówczyńska. *Approximation abilities of neuro-fuzzy networks*, Geodesy and Cartography, Warszawa, **59**(1): 2010.

[16] M. Mrówczyńska. *Identification reference system of levelling network on the area of Legnica-Głogów copper district* [in Polish], Acta Geodesia et Descriptio Terrarum, Wrocław 2010.

[17] M. Mrówczyńska. *The accuracy and efficiency of the relief features using neural networks* [in Polish], Ph.D. Thesis, Warsaw University of Technology, 2005.

[18] S. Osowski. *Neural networks for information processing* [in Polish], Publishing House of the Warsaw University of Technology, Warszawa, 2006.

[19] L. Rutkowski. *Methods and techniques of artificial intelligence* [in Polish], Polish Scientific Publishers PWN, Warszawa 2006.

[20] A. Stateczny, T. Praczyk. *Artifical neural networks in the diagnosis of marine facilities* [in Polish], Gdansk Scientific Society, Gdynia 2002.

[21] R. Szpunar, J. Walo, A. Pachuta, T. Olszak. *Study of the use OFT kinematic solution for monitoring of engineering objects* [in Polish], 6th Scientific and Technical Conference, Warszawa-Białobrzegi, 2003.

[22] Technical Guidelines G-1.10, *Formulas and parameters mapping coordinate systems* [in Polish], Head Office of Geodesy and Cartography, Warszawa 2001.