Application of a Multi-layer Perceptronfor Mass Valuation of Real Estates

Tomasz BUDZYŃSKI, Poland

Key words: mass valuation of real estates, multi-layer perceptron, teaching algorithms.

SUMMARY

In the process of mass valuation of real estates models of regression, in particular multiple regression are applied. At the same time new methods of determination of real estate values are searched for, which will allow to ensure the higher accuracy of estimation. One of these methods are artificial neural networks (ANN). The most used type of ANN is a multi-layer perceptron. It ensures the highest accuracy of estimation among all types of artificial neural networks.

The paper presents general rules of operations of a multi-layer perceptron. Four teaching algorithms which may be used in the process of teaching the multi-layer perceptron are described: back propagation of errors, conjugate gradient descent, quasi-Newton and Levenberg-Marquardt.

In the practical part of the paper results of experiments concerning utilisation of a multi-layer perceptron for mass valuation of real estates are presented using an example of non-built-up areas planned for one-family houses, located in the city of Otwock close to Warszawa. Accuracy of determination of real estate values using the multi-layer perceptron taught by means of four mentioned above teaching algorithms are compared.

At the end conclusions resulting from performed experiments are presented.

Application of a Multi-layer Perceptronfor Mass Valuation of Real Estates

Tomasz BUDZYŃSKI, Poland

1. INTRODUCTION

In the process of real estate valuation, and in particular, in the case of mass valuation, where statistical analysis methods are applied, new methods of determination of a real estate value are searched for, which would allow to achieve higher accuracy of results.
Artificial neural networks (ANN) represent one of methods which might be an alternative for the commonly applied method of multiple regression.

ANN are the highly sophisticated modelling technique, which allows to project functions of a very high level of complexity (Statsoft Polska Ltd., 2000). Their name originates from a network of brain nervous cells (McCluskey 1996). ANN have been created as a result of multiyear investigations performed in the field of artificial intelligence, which concerned, inter alia, construction of models of the basic structures which occur in a brain (Statsoft Polska Ltd. 2000).

A simplified neuron model has been applied in the ANN. It has been defined in the following way: Input signals (values) reach the neuron. They are the primary data values, which enter the network from outside or they are intermediate signals which originate from outputs of other networks, being the elements of the network. Every signal is introduced to the neuron through a connection of a specified strength (weight), which corresponds to the synapse ffectiveness in a biological neuron. Every neuron has one threshold value, which specifies the intensity of stimulation required for ignition.

Every neuron calculates the weighted total of its entries, and the level of stimulation, which is thus determined, becomes the argument of the activation function, which calculates the output value of the neuron.
Similarly to construction of a human's brain, ANN consist of many processing points (neurons), arranged in layers and of many mutual connections (synapses) between those points.

Many types of neural networks exist. Multi - layer perceptrons (MLP) with one or two hidden layers are mostly used.

Architecture of a multi-layer perceptron with one hidden layer is presented in Fig.1.

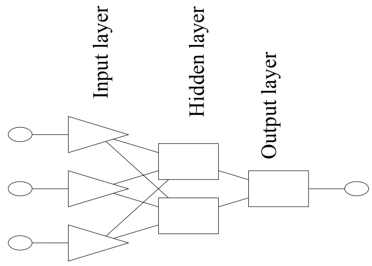

Fig. 1. Architecture of a multi-layer perceptron

Basing on investigations concerning utilisation of ANN for real estate valuation, it has been stated that – among investigated ANN models - the highest accuracy of determination of a real estate value may be obtained using the multi-layer perceptron (Wilkowski, Budzyński 2006).

Results obtained encouraged to continue investigations related to utilisation of the multi-layer perceptron for the needs of real estate valuation.

2. TEACHING THE MULTI-LAYER PERCEPTRON USING VARIOUS TEACHING ALGORITHMS

Although ANN are software routines which simulate activities of neural networks, they are not "programmed"; instead, they are rather „taught"(trained) using various examples. In the case of utilisation of ANN for real estate valuations, such examples are: real estate prices including selling prices or the rent level.

Similarly to other types of neural networks, the multi-layer perceptron utilises the parallel structure of data processing. Training of the network is performed by means of processing of input data (examples). Successive passes of series of data through the multi-layer perceptron result in adjustment of weights of particular connections and threshold values of the neurone, in such a way, that differences between the results of work of the network (the real estate value) and the expected result (the real estate price) are minimised.

Not every multi-layer perceptron. i.e. the perceptron of an arbitrary number of neurones (including hidden neurones) may achieve the result satisfactory for the ANN user. The appropriate network architecture should be selected. Searching for the appropriate network is performed through teaching and testing successive variants. In order to accelerate the process, some routines, which simulate operations performed by the multi-layer perceptron, allow to automate it by means of utilisation of an automated designer, which controls networks of various numbers of neurones, in order to find the best network. The best network is usually considered such a network, which is characterised by the lowest value of the validation error.

If the user designs the multi-layer perceptron independently, the number of hidden layers, and then, the number of neurones in particular layers must be selected and then the training of the network must be performed. This procedure has been applied in reported investigations.

One of the most important components of the process of designing the multi-layer perceptron is the decision concerning the number of hidden layers and the number of neurones hidden in each layer. Their number depends on the level of complexity of the task which is to be solved by neural networks. If the number of neurone is too high, this may result in the network "over-training"; in turn, if their number is too small, the network will recognise properties existing in the data set with insufficient accuracy. The total number of neurones should several times exceed the number of weights in the network. In the case of valuation, the number of neurones in the input layer may be limited by means of appropriate selection of properties of real estates.

Similarly to other neural networks, training of the multi-layer perceptron consists of three stages: the stage of teaching, the stage of testing and the stage of analysis of results. It requires that the file with examples is divided into at least 2 subsets: the teaching subset and the testing subset. Each of distinguished subsets should contain examples which are representative for the entire dataset.

In the stage of teaching the network is taught using the teaching subset. The network recognises general relations which occur in the dataset. In the process of teaching the network, the, so-called, teaching algorithms are used. Several teaching algorithms exist for the most popular ANN architecture, the multi-layer perceptron, such as: the method of back propagation of errors, the conjugate gradient method, the quasi-Newton method and the Levenberg-Marquardt method.

Then, in the stage of testing, operations of the network are tested. Those tests are performed for the validation subset, which contains data, which does not occur in the training subset. This allows to estimate, whether the network was taught the general data structure or was just adjusted to specific values existing in the teaching subset. In the first case the error in the teaching subset is of the same order as the error in the validation subset; in the second case the error in the teaching subset is much smaller than the error in the validation subset. Values of errors in both subsets may be controlled in the process of teaching. When the error in the teaching subset decreases and the error in the validation subset increases, teaching of the network should be terminated, otherwise the network could become "over-trained".

When a lot of examples are available, a separate testing subset may be distinguished from the dataset. The testing subset contains examples, which are not used in the process of teaching and validation. Testing of the network using this testing subset, if the error is small, ensures the user that the network will generate reliable results.

Analysis of results is the final stage of teaching the network. In the case of regression tasks, such as real estate valuation, it covers determination of data for the above subsets, among others:

- the mean error (considered as the modulus of the difference of the required value and the obtained output value) for the output variable (real estate value),
- the quotient of standard deviations for errors and data – being the basic factor of the quality of the regression model constructed by the network.

The above training algorithms are described below for the multi-layer perceptron.

2.1. Back Propagation of Errors Algorithm

The algorithm of back propagation of errors determines, in the course of teaching, the local value of the gradient versus each weight for every presented case. Weights are modified after presentation of every teaching case.

The method of determination of the signal of error depends on location of the modified neuron. This value is calculated in different way for neurons of the output layer and for neurons of the hidden layers.

The signal value of the error in the output layer is the product of the derivative of the error function of the network and the derivative of the neuron activation function.

Values of the signal of the error for neurons located in hidden layers is equal to the product of the derivative of the neuron activation function and the weighted total of signals of errors determined for neurons of the next layer. In the course of "weighting" of errors, which undergo the process of back propagation (the origin of the name of the discussed method), weights associated with connections leading from the modified neuron to neurons located in the next layer, are applied.

2.2. Conjugate Gradient Descent Algorithm

The conjugate gradient descent algorithm modifies weights in a cumulated way. This means, that weights are modified once in the course of implementation of one epoch, i.e. in different way than in the case of the back propagation method, where modification of weights is performed after every presentation of the successive case, contained by each epoch. In the case of application of the conjugate gradient descent algorithm the mean value of the gradient

for each epoch, for all cases, is determined on the surface of the error; this gradient value is the basis for modification of weights, which is performed in the final stage of each epoch.
In the course of its operations, the conjugate gradient descent algorithm performs the series of linear searching along selected directions on the surface of the error. Initially, it determines the directions of the maximum slope, so it behaves similarly to the algorithm of back propagation of errors. However, instead of making the steps proportional to the teaching factor in specified directions, the step is performed in a well defined manner, after determination of the movement direction, a point is searched for along the selected line (following the determined direction), which corresponds to the minimal error value. Effective implementation of this process is possible, since it requires the relatively short time, as searching is performed only in one dimension, along the specified line. After shifting the teaching process to the minimal point, located along the selected direction, successive linear searches are performed, one for each epoch. They are performed along lines, which create associated directions with the direction, which has been determined earlier. Those associated directions are determined in a way, which ensures that the minimum values, determined along directions considered in previous steps, are maintained (none of steps of the algorithm cannot deteriorate the previous results).

In the course of determining the associated directions it is assumed that the surface of errors is locally paraboloidal; it means that within surroundings of the point, where the teaching process is located, it is described, with satisfactory accuracy, by a multi-dimensional square function. Usually this assumption is not met in the ideal way; however, solutions obtained under this assumption, concerning the direction where the minimum should be searched for once again, are practically satisfactory. Applicability of the square surface model is controlled in practice; if the algorithm states that the current direction of searching, resulting from the square model and technique of conjugate gradients does not allow to decrease the error value, the new direction of the maximum slope is determined in the given point and the next sequence of searching for the minimum in that direction is started. It should be noticed that – as it results from the general properties of the error function surfaces – after achieving the point located close to the real minimum of a multi-dimensional function, its nature is modified in such a way that the assumption concerning the square form of the error surface becomes true. This allows for fast achievement of the point, which is characterized by the lowest error value, in particular, in the final stage of searching – it means, when it is very difficult to perform the simple gradient methods (as the back propagation method).

2.3. Quasi-Newton Algorithm

The quasi-Newton algorithm uses the fact, that the direction to the minimum can be found on the square (parabolic) error function, using the Hesse matrix, i.e. the matrix of partial derivatives of the second order. Sufficiently close to the minimum, each surface of the error may be considered as a square surface. However, since calculation of the Hesse matrix is difficult and time consuming, and steps of Newton may lead to erroneous results on a non-parabolic surface, approximation of the inverse Hesse matrix is created in an iterative process.

In its first step, approximation follows the maximum slope line and later its compatibility with the estimated Hesse matrix becomes higher.

The quasi-Newton algorithm modifies weights as a complex. While in the case of back propagation weights are corrected after every case, in the quasi-Newton algorithm the mean gradient of the error surface is calculated, for all cases, before all weights ware updated at the end of the epoch.

The quasi-Newton algorithm is the most popular algorithm of non-linear optimization and it is considered to be the fast concurrent algorithm. However, it has some disadvantages: its numerical stability is lower than the method of conjugate gradients, it may have the tendency of concurrence to local minima and it requires more memory.

2.4. Levenberg-Marquardt Algorithm

Operations of the Levenberg-Marquardt algorithm is based on the assumption, that the real function, which is modelled by the network and which combines input signals with one output signal is of linear nature. This assumptions allows for precise determination of the minimum of the error function, what may be performed in one step, without the necessity to apply any iterations. After determination of the hypothetic minimum of the error function, the algorithm performs its testing. If the error value is lower than the error in the starting point, coordinates of the new point determine the new starting point, and the set of weights, corresponding to that point is considered as the new approximation of optimum parameters of the network, which is being taught. If the assumption concerning the linear nature of the approximated function is not met (it is usually not met, but the high divergence is considered here), then the error in the determined point may be much greater than the error in the starting point. Then the algorithm returns to the starting point and attempts to improve the situation using the technique based on the maximum slope method. This process is repeated in successive epochs. The main advantage of the Levenberg-Marquardt algorithm is the ability to determine new solutions in a way, which is always the compromise between the solution achieved by the highest slope method and the above mentioned algorithm of the hypothetic linear approximation. If the solution achieved by the algorithm of the hypothetic linear approximation led to decrease of the error value, its strengthens the assumptions concerning the linearity. If the hypothesis concerning the linearity did not work, steps, which led to solutions resulting in increase of error values, are rejected and successive attempts to modify the weights according to the hypothetic linear approximation are performed with greater care. During its operations, the algorithm switches between two described approaches, using the hypothetic linear approximation in most possible cases and coming back to the maximum slope method only when the evident failure occurs; this results in high speed of operations performed by this algorithm.

3. INVESTIGATIONS ON THE ACCURACY OF DETERMINATION OF A REAL ESTATE VALUE USING THE MULTI-LAYER PERCEPTRON, TRAINED BY MEANS OF VARIOUS TEACHING ALGORITHMS

Investigations concerning the accuracy of determination of a real estate value using the multi-layer perceptron, trained by means of various teaching algorithms, have been performed basing on the example of the market of non-built-up areas, to be used for one-family houses located in Otwock close to Warsaw.

Data of 114 transactions (land, non-built-up parcels, planned for one-family houses, located in Otwock) performed in the period 2000-2001 were investigated. Prices of 1 sq. m. of lands have been updated for January 2002.

Features, which may influence the land prices in Otwock, were specified. They are: location, neighbourhood, access to public transport, technical infrastructure, state of developing, parcel size, shape of a parcel.

In order to specify features which should be considered in the process of construction of a multi-layer perceptron model, the genetic algorithm and the backward step method were applied independently. The same results were obtained by those methods, i.e. except the feature „shape of a parcel" all other features are useful in the process of creation of the multi-layer perceptron model.

Then, all transactions were divided into the teaching subset – 71 cases, the validation subset – 29 cases and the testing subset – 14 cases. Those subsets have similar statistical characteristics – the mean value 60.00 zł/m^2, 60.44 zł/m^2 and 59,77 zł/m^2, respectively, and the standard deviation values: 31.27 zł/m^2, 33.31 zł/m^2, 30.35 zł/m^2, respectively.

Using the Statistica Neural Networks software package multi-layer perceptrons were created with three, four and five hidden neurons, respectively. Each of those multi-layer perceptrons was taught by four teaching algorithms, describe above. For each model of the multi-layer perceptron, taught by one of the mentioned algorithms, 100 neural networks were created. In the phase of experiments the total of 1200 multi-layer perceptrons were created. The constructed models, taught by particular teaching algorithms, were recorded in four, separate files of the network.

Table 1 below presents errors (the root of totals of squares of errors of particular cases, determined by the function of the error of the network) in the teaching subset (the teaching error), the validation subset (the validation error) and in the testing subset (the test error) for the 10 best neural networks out of 100 constructed neural networks, which have 3, 4 and 5 hidden neurons, respectively, and which were taught by the following teaching algorithms: back propagation of errors (BP), conjugate gradients (CG), quasi-Newton (QN) and Levenberg-Marquardt (LM). Besides the type of the teaching algorithm, Table 1 presents the number of the teaching epoch, in which the given network was characterised by the smallest value of the validation error, i.e. when it was the best. The error in the validation subset for the best network, taught by the given teaching algorithm, is presented in bold.

Table.1. Characteristics of selected multi-layer perceptrons.

Number of hidden neurones	Error in teaching file zł/m^2	Error in validation file zł/m^2	Error in test file zł/m^2	Network number in a file	Teaching algorithm, number of teaching epoch
3	9.696718	10.49435	11.76747	67	BP 97
3	9.988878	10.57217	10.97158	33	BP 99
3	10.00016	10.63275	11.94458	24	BP 86
3	10.27986	10.71907	11.67362	38	BP 85
3	10.08341	10.79088	12.40815	46	BP 97
3	10.20296	10.82443	11.39018	87	BP 92
3	9.755157	10.85956	11.33153	14	BP 97
3	10.19758	10.92883	12.62590	55	BP 89
3	9.911646	11.01226	11.66084	97	BP 97
3	9.975479	11.07343	11.80595	50	BP 97
4	9.643689	10.59871	11.31541	163	BP 99
4	9.800481	10.60128	11.73295	138	BP 99
4	10.25254	10.83268	11.75028	134	BP 90
4	10.14432	10.85078	12.12913	150	BP 95
4	9.77638	10.91646	11.90574	185	BP 98
4	9.915918	10.94451	11.00020	146	BP 96
4	10.05354	11.15980	11.38728	142	BP 96
4	10.28936	11.24815	12.18099	181	BP 97
4	10.10934	11.27656	11.89238	139	BP 93
4	10.15821	11.28541	12.07704	166	BP 98
5	9.584256	**10.36612**	11.35798	242	BP 92
5	9.462977	10.41314	10.86535	237	BP 98
5	9.76110	10.44890	11.36997	267	BP 99
5	10.93719	10.50233	12.59375	249	BP 12
5	9.665446	10.55590	10.33745	209	BP 93
5	9.958743	10.63591	11.85453	245	BP 90
5	9.796437	10.66357	11.14308	244	BP 96
5	10.06525	10.72067	12.17929	260	BP 84
5	9.686867	10.76743	11.61175	208	BP 97
5	10.25147	10.78526	11.50730	277	BP 99
3	8.553579	9.317702	9.744248	15	CG 99

Number of hidden neurones	Error in teaching file zł/m^2	Error in validation file zł/m^2	Error in test file zł/m^2	Network number in a file	Teaching algorithm, number of teaching epoch
3	9.562352	9.360718	11.14156	28	CG 61
3	8.456292	9.408476	9.856787	93	CG 65
3	7.918314	9.455911	9.865039	39	CG 88
3	8.470978	9.518941	9.801648	1	CG 99
3	9.219148	9.543596	10.68496	48	CG 52
3	8.628581	9.553536	9.925418	81	CG 99
3	8.951097	9.559634	10.88147	37	CG 70
3	9.075138	9.574868	11.15096	77	CG 55
3	8.699602	9.659920	12.28238	71	CG 76
4	8.896472	9.180777	11.35737	184	CG 96
4	8.805346	9.365056	9.458848	146	CG 73
4	8.836677	9.381079	10.80636	106	CG 95
4	9.082006	9.554819	11.09459	152	CG 89
4	8.597639	9.578359	10.32522	113	CG 94
4	8.472374	9.642568	9.984596	111	CG 99
4	8.307498	9.650772	9.826212	147	CG 79
4	8.755473	9.682064	10.45972	145	CG 60
4	9.337041	9.745168	10.53199	141	CG 65
4	8.681951	9.791487	9.883537	107	CG 83
5	8.047045	**9.097722**	10.67246	272	CG 99
5	8.425230	9.204469	10.86818	228	CG 94
5	8.378257	9.225664	10.0433	274	CG 86
5	8.697652	9.369480	10.24015	250	CG 87
5	9.372861	9.459508	10.92464	282	CG 36
5	7.985135	9.472550	10.86984	270	CG 90
5	8.563875	9.513753	9.634813	210	CG 95
5	8.743315	9.603307	11.21027	243	CG 98
5	8.879027	9.712059	9.970314	208	CG 90
5	9.008045	9.772772	10.31649	218	CG 71
3	8.577321	9.251484	10.44241	64	QN 92
3	8.854031	9.414275	10.59096	67	QN 69
3	9.028775	9.487705	10.86711	15	QN 78
3	8.875493	9.631400	10.00006	53	QN 83
3	8.684761	9.654127	9.549309	16	QN 86
3	9.326501	9.659567	12.54091	85	QN 56
3	9.379531	9.679789	11.1352	44	QN 64

Number of hidden neurones	Error in teaching file zł/m^2	Error in validation file zł/m^2	Error in test file zł/m^2	Network number in a file	Teaching algorithm, number of teaching epoch
3	9.055143	9.683311	12.46885	25	QN 90
3	9.183319	9.690530	11.28474	83	QN 58
3	8.672523	9.699506	10.91776	49	QN 98
4	8.996874	**9.058687**	10.02242	113	QN 86
4	9.294679	9.168142	11.21536	189	QN 62
4	8.843727	9.447860	11.13248	112	QN 98
4	8.922932	9.554887	11.22612	173	QN 74
4	8.958952	9.652721	10.37332	152	QN 56
4	8.809630	9.685871	11.74442	181	QN 74
4	8.981136	9.685885	10.32785	196	QN 85
4	9.110983	9.750262	10.59101	138	QN 72
4	8.054635	9.757134	10.12670	104	QN 96
4	8.729833	9.781310	10.76395	136	QN 99
5	8.054216	9.437988	10.88689	254	QN 80
5	8.105234	9.438128	12.02157	247	QN 81
5	9.282457	9.556997	10.56921	259	QN 57
5	8.605539	9.633930	9.003491	251	QN 78
5	8.094070	9.717563	10.25081	226	QN 98
5	8.700970	9.738841	12.49768	275	QN 98
5	9.129708	9.763016	10.42838	277	QN 55
5	8.736699	9.789002	10.24300	268	QN 94
5	9.236421	9.821799	11.95663	297	QN 99
5	9.099674	9.841247	11.44189	208	QN 72

3	7.973481	8.838693	11.05852	21	LM 95
3	7.853191	9.096793	10.47364	50	LM 66
3	7.929702	9.370268	9.804918	38	LM 87
3	8.113187	9.431512	9.591531	20	LM 68
3	7.649477	9.435407	9.113512	41	LM 97
3	8.436505	9.444197	10.21510	65	LM 57
3	7.698186	9.492765	10.86492	28	LM 97
3	7.715733	9.617524	9.552825	46	LM 67
3	8.158360	9.669183	10.04896	44	LM 27
3	8.191848	9.702894	9.77632	40	LM 99
4	7.460221	**8.534605**	10.97577	170	LM 94
4	8.412120	8.796665	10.95999	161	LM 71
4	7.825629	9.076036	11.11574	154	LM 99

Number of hidden neurones	Error in teaching file zł/m^2	Error in validation file zł/m^2	Error in test file zł/m^2	Network number in a file	Teaching algorithm, number of teaching epoch
4	7.755382	9.099799	11.27052	118	LM 56
4	7.499825	9.106124	10.54094	114	LM 97
4	7.661975	9.201407	9.707891	146	LM 57
4	8.191708	9.285498	10.46798	134	LM 97
4	7.979015	9.355980	10.15521	133	LM 88
4	8.294124	9.375969	10.19638	139	LM 94
4	7.211140	9.411457	10.34621	169	LM 87
5	7.550017	8.587820	9.388863	282	LM 28
5	7.307299	8.788293	10.74277	269	LM 62
5	7.761809	9.011996	11.19687	228	LM 83
5	8.237223	9.055224	11.94212	243	LM 98
5	7.888611	9.128721	11.30042	203	LM 99
5	8.383007	9.210142	10.33717	219	LM 55
5	8.048870	9.280541	10.09614	278	LM 67
5	7.859905	9.289233	10.97187	232	LM 98
5	7.311426	9.307494	10.88693	271	LM 96
5	7.953575	9.308376	9.673532	281	LM 97

4. CONCLUSIONS

Out of investigated teaching algorithms, the Levenberg-Marquardt algorithm allows to construct the best multi-layer perceptron, i.e. the perceptron characterised by the smallest value of the error in the validation subset. The conjugate gradient (CG) and the quasi-Newton algorithms allow to achieve lower accuracy of determination of real estate values. The back propagation of errors algorithms (BP) turned to be characterised by the lowest efficiency, comparing to other investigated algorithms. Those conclusions concern the network of architecture of low complexity level, which consist of several input neurons, several hidden neurons and one output neuron, i.e. the networks constructed for the needs of determination of real estate values.

Differences in the intensity of teaching the multi-layer perceptron by means of various teaching algorithms for the network of low architectural complexity using about 100 examples become unimportant.

Increase of the number of hidden neurons in the multi-layer perceptron not always results in decrease of the value of the error of determination of a real estate value. In the case of teaching the neural network using the Levenberg-Marquardt (LM) and the quasi-Newton (QN) algorithm the error in the validation subset for the network with 5 hidden neurons was greater than the same error in the network with four neurons in the hidden layer.

5. REFERENCES

McCluskey W.J. 1996. *Zastosowanie sztucznej inteligencji w wycenie masowej dla potrzeb taksacji nieruchomości.* (Utilisation of artificial intelligence for real estate valuation).Wycena nr 5(28). Olsztyn.

Statsoft Polska sp. z o.o. 2000. *podręcznik elektroniczny STATISTICA Neural Networks* (Electronic Manual).

Wilkowski W., Budzyński T. 2006. Application of Artificial Neural Networks for Real Estate Valuation. Shaping the Change XXIII FIG Congress Munich. Germany. October 8-13. 2006

BIOGRAPHICAL NOTE

Tomasz Budzyński Ph.D.

Born in Warsaw in 1974. Studies of Geodesy and Cartography at the Warsaw University of Technology. Graduated his (M.Sc.) in Geodesy in 1998. Obtained his Ph.D. with a dissertation "Research on an effective method of determination of cadastral value" at the Warsaw University of Technology in 2005. Current position: full-time research worker at the Warsaw University of Technology (Faculty of Geodesy and Cartography). Member of the Polish Society of Surveyors.

CONTACTS

Tomasz Budzyński
Warsaw University of Technology
Plac Politechniki 1
PL 00-661 Warsaw
POLAND
Tel. (48-22) 625-15-27
(48-22) 234-75-89
Fax: (48-22) 625-15-27
Mobile: +48-603-196-198
E-mail: tbudzynski@o2.pl