

**USER'S
MANUAL
Part 2:
Basics About Modeling with
aiNet**

Table of Contents - Part 2: User's Manual: Basics About Modeling with aiNet

1. Introduction	_____
1. Modeling a phenomena	_____
1.1 General	_____
1.2 Describing a Phenomena	_____
1.3 Noisy Data	_____
1.4 Selecting the Variables of the Phenomenon - General	_____
1.5 Efficiency of the Model	_____
1.6 Brief Review of Basic Concepts	_____
2. Variable Types and Preparation of Model Vectors	_____
2.1 Basic concepts	_____
2.2 Selecting variable types	_____
2.3 Preparation of model vectors in some other cases	_____
2.4 Conclusion	_____
3. Measures for the Estimation of Prediction Error and Preparation of Efficient Model	_____
3.1 General	_____
3.2 Measures for the Estimation of Prediction Error	_____
3.3 Penalty coefficient	_____
3.4 Modeling tools	_____
3.5 Relations between modeling tools	_____
3.6 Assistant tools	_____
4. Conclusion	_____
5. References	_____

Chapter 0

1. Introduction

This part of User's Manual explains some basic features about modeling in general and how to use this with aiNet

The theoretical basis will not be described in detail. The basic idea of aiNet comes from artificial intelligence, or more precisely, from its special area - neural networks. While such tools work as "black boxes", some assistant tools were added to improve the efficiency of aiNet. These tools give, amongst others, mathematical functions, which correlate with some basic statistical measures. Therefore the efficiency of the model may be defined mathematically, the efficiency of different models may be compared; reliability and/or the quality of the solution may be compared with existing solutions. Finally, there are some ideas (not included in the program), which allow a simple explanation of the predicted results.

aiNet should be able to solve most problems, where there is sufficient data and/or sufficiently strict descriptions (not to "soft knowledge") of phenomena. Using aiNet, problems may be solved by selecting different coefficients, which define the learning of usual artificial neural networks and also the problems of local minima (see, for example [1], [2], [3], [4]). aiNet may be used as a filter in the data preparation for other neural networks. Further, it may be used for the analysis of problems, and for many other purposes

NOTE: All examples shown in the manual are calculated with aiNet. They are not fictitious, moreover, they represent a real capability and the efficiency of aiNet!

¹ Ramirez, M., R. & Arghya, D., **A faster learning algorithm for back-propagation neural networks in NDE applications**, Proceedings of the 2nd International Conference on AI, pp. 275-283, 1991.

² Samaad, T., **Backpropagation Improvements based on Heuristic Arguments**, Theory Track, Neural and Cognitive Sciences Track, International Joint Conference on Neural Networks, Vol. 1, Washington, D.C., 1990.

³ Johansson, E., M., Dowla, F., U. & Goodman, D., M., **Backpropagation learning for Multi-layer Feed-forward Neural Nets Using the Conjugate Gradient Method**, Lawrence Livermore National Laboratory, 1990.

⁴ Shanno, D., F., **Conjugate Gradient Methods with Inexact Searches**, Mathematics of Operations Research, Vol. 3, pp. 244-256, 1978.

Chapter 1

2. Modeling a phenomena

2.1 General

A tendency to master the events in every-day-life demands foresight - the prediction of natural phenomena. Phenomena are modeled with mathematical models. Modern science enables such models with an analytical approach. Reliable models of different phenomena are strongly connected with the events in the developed world: a competitive struggle on the market and material progress. Let us try to analyze the money market and insurance on the one hand and the health service and optimal solutions in technical disciplines on the other hand ...

A reliable treatment of natural phenomena is based on measurements and the description based on relations between the observed results. From the theoretical point of view, the relations are most appropriately specified in terms of abstract mathematical models representing mathematical laws. But from the practical point of view, simulated analog models based on electronic devices are sometimes more convenient. The aim of this part of manual is therefore to introduce a generalized interpretation of a description of phenomena which may be realized in various information processing systems. aiNet simulates one of such information processing systems.

It should be noted that in the process of modeling phenomena we want to **describe extremely complex reality** on the one hand with **simple**, usually **very simplified**, abstract mathematical models on the other hand (see Figure 1.1).

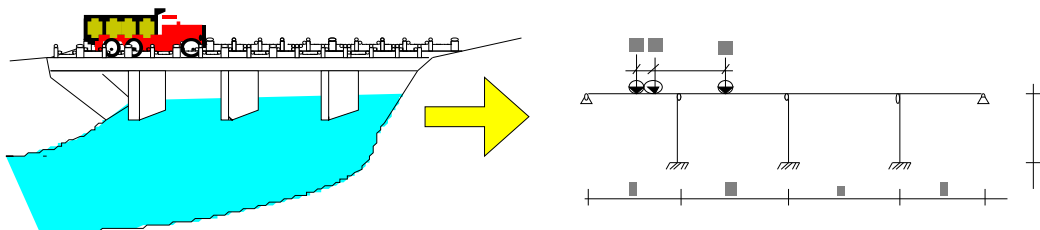


Figure 1.1: Description of the reality with the abstract mathematical model.

2.2 Describing a Phenomena

Take a simple example of a natural phenomenon, which is graphically presented in Figure 1.2. These are two points in Cartezius coordinate system. They represent results of the measurements - air temperature as a function of the day hour (e.g. temperature at 8:00 a.m. was 5.3°C, at 12:00 was 13.1°C).

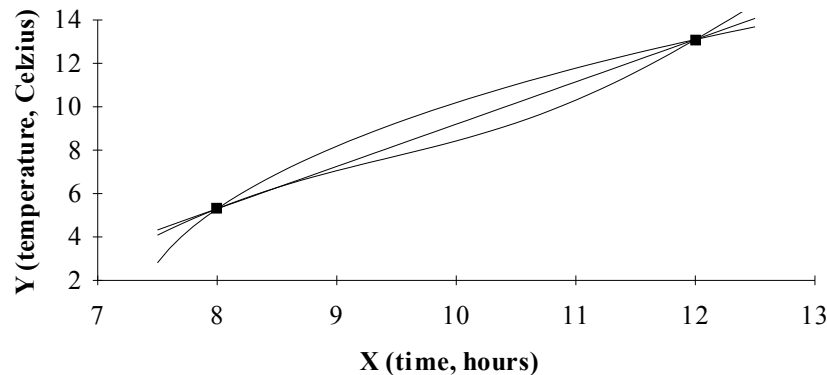


Figure 1.2: Graphical representation of a phenomenon: changing of the day-air temperature.

While we have only two measurements - two points uniquely define only straight line - the model could be linear. But, besides the two values there is supplementary, qualitative information about a phenomenon (the temperature has usually a low value in the morning, the highest value approximately at noon, and then it slowly decreases until the following morning). Given this information, it may be concluded that the linear model is insufficient for the description of the phenomenon (which is basically non-linear). Therefore, the most important rule in the modeling process is:

ALWAYS TRY TO COLLECT ENOUGH DATA IN ORDER TO DESCRIBE THE PHENOMENON PERFECTLY (SUFFICIENTLY)!

The present example of a natural phenomenon is only two-dimensional in the mathematical sense. The rule sounds trivial in this case, but practice shows that there are many problems, where it is required to model something with insufficient data. This is not a problem of modeling itself, but the problem of acquisition of knowledge about phenomenon.

The problem is solved with acquisition of additional information about the phenomenon: the results of new measurements or additional expert's knowledge. New measurements in our particular case give the improved situation, presented in Figure 1.3. The day-temperature changing is shown at proportional time steps from the morning until the evening. Obviously, the phenomena is non-linear. As Figure 1.3 shows, the new data is now sufficient for the satisfactory description of the phenomenon.

Sometimes, in rare cases, the situation arises where there is too much data. Such cases do not have an influence on the modeling of the phenomenon, only upon the computing time that may be consumed. The model becomes less efficient for this reason, and in some cases, such as real time applications, it may become unusable.

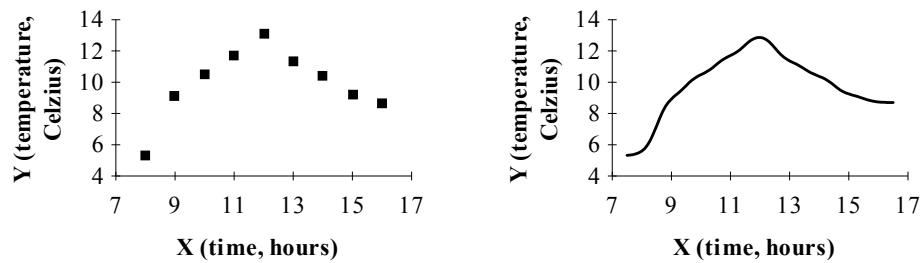


Figure 1.3: Graphical presentation of the day-temperature changing at proportional time steps and its mathematical model.

Determining whether there is too much data is a complex function. This is not included in aiNet and it is unimportant for better understanding. It should be mentioned, however, that the method corresponds to the well known self-organization process of neurons [5]. The number of data samples compresses to a smaller number of data samples, which are representative of the original data. This may be done in an optimal way where the new data is not just the extracted data from the original data, but is generated as new, formed, abstract data [6].

2.3 Noisy Data

There are many other problems in practice. If day-temperature is measured with three different instruments in random time steps (such data reflects real life cases), the situation shown Figure 1.4 is found. The phenomenon where simultaneously measured values (with different instruments) gain different values, is called noise. This is not a characteristic of the measured phenomenon only.

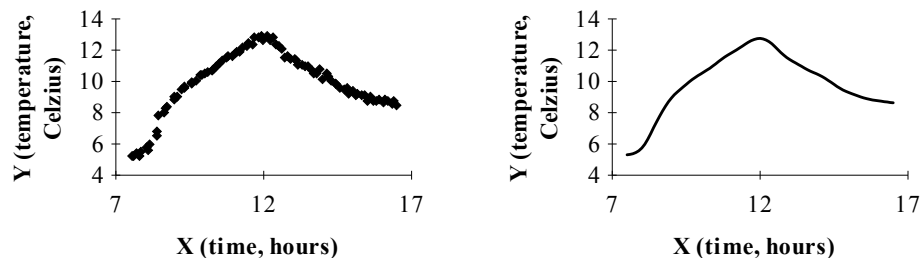


Figure 1.4: Graphical presentation of a phenomenon, where characteristics are measured in random time steps with different instruments and its mathematical model.

Noise may also appear in cases where it is required to evaluate a phenomenon observed in many different ways, for example the political situation in a country - the problem is the influence of the subjectivity of different observers. Obviously, all real-life problems involve dealing with noise.

⁵ Kohonen, T., **Self-Organization and Associative Memory**, Second Edition, Springer-Verlag, Berlin, 1988.

⁶ Grabec, I., **Self-Organization of Neurons Described by the Maximum-Entropy Principle**, Biol. Cybern., **63**, pp. 403-409, 1990.

Noise also depends, besides the errors which may result from measurement, on the way of modeling a phenomenon. The description of a phenomenon may consider too small a number of measurements, or a too large a number of measurements with small differences. Therefore, at least a basic knowledge about the phenomenon is of great importance. But with new phenomenon, this is often impossible. Solving this dichotomy involves the use of different statistical tools (e.g. parametric analysis), which make it possible to determine the significance and/or the influence of a single parameter. Unfortunately, such tools are usually very sophisticated and divert users from the application. In comparison to those tools, aiNet allows simplified, physically clear estimation of the noise in the data and assessment of the importance of single parameters. More about this topic is found in Chapter 3.

2.4 Selecting the Variables of the Phenomenon - General

Consider the example of the day-temperature changing and assume that a tool (e.g. aiNet), is able to describe the phenomenon perfectly. A used mathematical model demands only two variables for the description (hour of the day and the temperature at that time). Everything works fine, until instead of the mapping (see Figure 1.5)

$$\text{day_hour} \rightarrow \text{temperature},$$

one demands an inverse mapping. In this case, it is obvious that the reverse function is incorrectly defined.

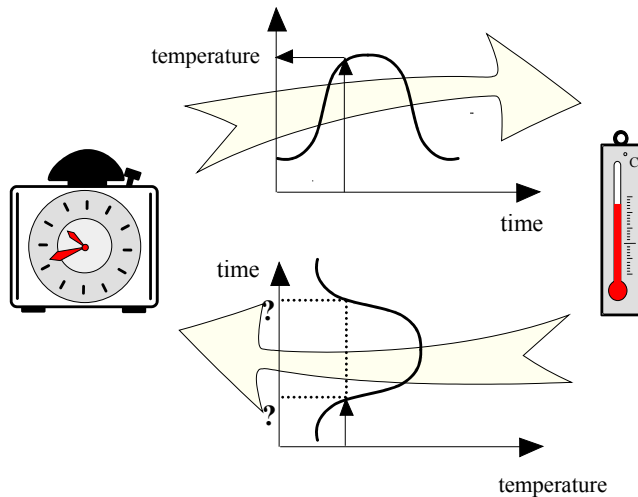


Figure 1.5: Mathematical relations in modeling the phenomenon of the day-temperature function.

Look at the simplest solution. The exact mathematical mapping

$$\text{temperature} \rightarrow \text{day_hour}$$

has a practical meaning. In order to obtain the correct answer, the relevant questions need to be asked first, which means modeling of the phenomenon in such cases. In this case, a new, additional variable must be introduced (see Chapter 2). The variable needs to describe an ascendant and a descendent part of the curve. In other words: the morning and afternoon, assuming that the highest temperature is reached at noon. The description of the mathematical model now has a new

dimension and the solutions, given by that model, lie on two parallel plains (see Figure 1.6). As it may be seen, the additional variable gives a simple solution, with the solution lying on corresponding plain.

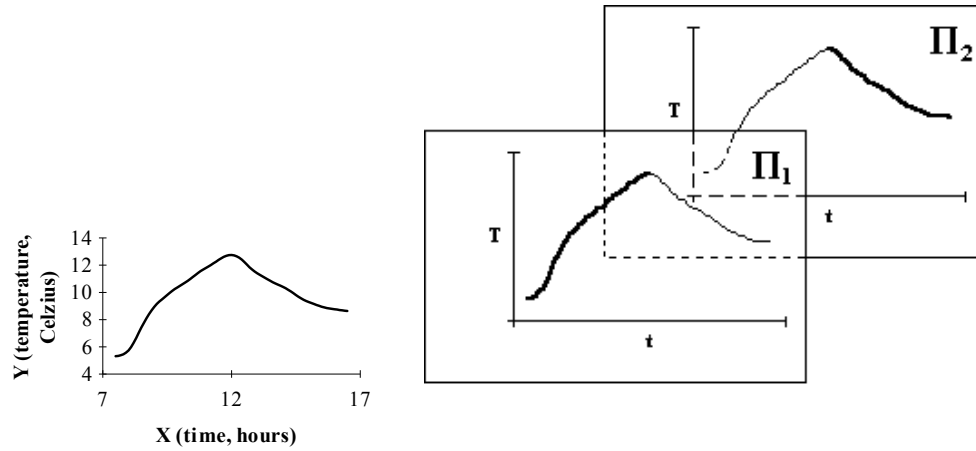


Figure 1.6: Graphical presentation of the mathematical model of the day-temperature changing in case of additional variable.

The efficient preparation of the model demands logical variable selection; the best choice is the case where variable has a practical, physical meaning. The above example shows the introduction of additional variable due to the inverse reversion, which divides the description of the phenomenon into two parts.

2.5 Efficiency of the Model

The problem has been analyzed with a small data sample at the beginning of this chapter. It should be clear that the reliability of the solution, and the further efficiency of the model, strongly depend on the quality of the data base (reliability of the measurements and/or collected facts), and representativity of the samples. Representativity of the samples means an approximately uniform distribution of the samples in the problem space and, of course, sufficient number of those samples.

This is illustrated in a simple example. Figure 1.7 shows the results of measurements and two solutions which may be obtained in cases of different reliability levels, based on the measured data.

It may be required to produce a model that gives, exact, or almost exact results in comparison with the given data. A general solution of such a model is shown in Figure 1.7./b/. A very rough curve may be observed, which seems unnatural, but in fact goes through the measured points (as demanded). As little is known about the phenomenon (small number of measurements of an unreliable source), it would be unreasonable to demand great precision of the model. Therefore, greater deviations in solutions from the available data should be allowed. The consequence of such modeling is shown in Figure 1.7./c/. The solution curve seems much more natural than before. Moreover, such model has very good generalization capabilities.

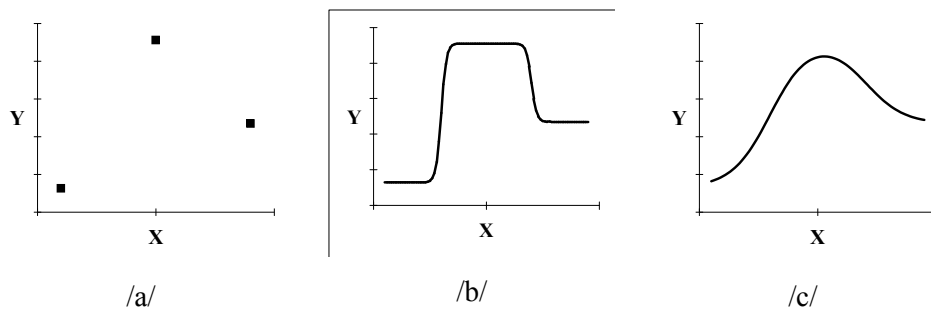


Figure 1.7: Two models of the phenomenon of the day-temperature changing for different reliability levels.

In order to obtain the best precision, aiNet provides tools to assist the determination of the optimal solution for the available dataset. The problem is reduced to the classical optimization problem, which, for example, may be solved in the iteration process by a procedure, called a bisection (see Figure 1.8).

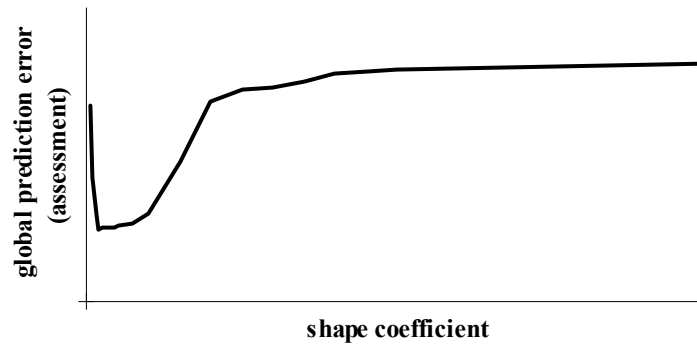


Figure 1.8: Minimum (prediction) error for the available dataset.

In aiNet, it should be mentioned that the reliability of the model for the available dataset is regulated by one parameter only. This is termed ‘shape coefficient’, for the time being, whilst, as may be seen from Figure 1.7, influences the shape of the solution curve (solution hyperplane).

2.6 Brief Review of Basic Concepts

Modeling of the phenomenon means the selection of the *right attributes* of a phenomenon and the suitable coding of the *right number of measurements and/or facts* in **the mathematical form**. aiNet is provided with tools, which allow accurate modeling in a defined sense, and they are presented in later chapters.

A model is a dataset or a data base itself. A data base consists of model vectors (see next chapter) which describe the phenomenon. It’s components represent the values of the variables of the phenomena.

Chapter 2

3. Variable Types and Preparation of Model Vectors

3.1 Basic concepts

Basic concepts will be explained through the example of phenomenon of the day-example. Temperature was measured every hour; measurement was started at 8:00 in the morning, and ended at 4:00 p.m. The phenomenon is modeled with a group of 9 vectors, so called **model vectors**:

model vector	=	time	temp.
mv ¹	=	8.00	5.30
mv ²	=	9.00	9.10
mv ³	=	10.00	10.50
mv ⁴	=	11.00	11.70
mv ⁵	=	12.00	13.10
mv ⁶	=	13.00	11.30
mv ⁷	=	14.00	10.40
mv ⁸	=	15.00	9.20
mv ⁹	=	16.00	8.70

Figure 2.1: Model vectors for the description of the phenomenon of the day-temperature changing.

Each model vector has two components - the phenomenon is described with two **variables**: time, when measurement was done, and the temperature at that time. General formulation of model vector may be written as:

$$\{\mathbf{mv}\}^T = \{\text{var_1}, \text{var_2}\}^T = \{time, temperature\}^T.$$

And finally, each phenomena may be written in a general form as follows:

model vector	=	var_1	var_2	...	var_M
mv ₁	=	value ₁₁	value ₁₂	...	value _{1M}
mv ₂	=	value ₂₁	value ₂₂	...	value _{2M}
.
.
mv _N	=	value _{N1}	value _{N2}	...	value _{NM}

Figure 2.2: General description of the phenomenon in mathematical form.

In defining the characteristics of a phenomenon, it is necessary to introduce the concept of **input variable** and **output variable**. For example, when we want to know *what temperature* there was *at an appointed hour* the input variable is the time (hour) of measurement, and the output variable is the temperature at that time (hour).

Generally, input and output variables are complementary. If the input variables are written as one partial vector and output variable as another partial vector, concatenation of both vectors gives a new vector (the model vector), which completely describes the phenomenon. The model of the phenomenon, consisting of model vectors with known values of input and output variables, is able to predict unknown values of output variables, so called prediction vectors. Naturally, the values of input variables of the prediction vectors are known.

Take an example of a phenomenon, which is completely described with N model vectors. Each model vector (**mv**) has M input variables and one output variable (M+1 st variable of the model vector). We want to use the model for the prediction of missing values - values of output variable of the same phenomena for three new prediction vectors (**pv**), which have known values of input variables only.

model vector	=	input var 1	input var 2	...	input var M	output var 1
mv₁	=	m_value ₁₁	m_value ₁₂	...	m_value _{1M}	m_value _{1,M+1}
mv₂	=	m_value ₂₁	m_value ₂₂	...	m_value _{2M}	m_value _{2,M+1}
.
.
mv_N	=	m_value _{N1}	m_value _{N2}	...	m_value _{NM}	m_value _{N,M+1}
pv₁	=	m_value _{N+1,1}	m_value _{N+1,2}	...	m_value _{N+1,M}	p_value _{1,M+1}
pv₂	=	m_value _{N+2,1}	m_value _{N+2,2}	...	m_value _{N+2,M}	p_value _{2,M+1}
pv₃	=	m_value _{N+3,1}	m_value _{N+3,2}	...	m_value _{N+3,M}	p_value _{3,M+1}

Figure 2.3: Mathematical description of the phenomenon in the case of prediction of unknown values of output variables.

model vector	=	time	temperature
mv_1	=	8.00	5.30
mv_2	=	9.00	9.10
mv_3	=	10.00	10.50
mv_4	=	11.00	11.70
mv_5	=	12.00	13.10
mv_6	=	13.00	11.30
mv_7	=	14.00	10.40
mv_8	=	15.00	9.20
mv_9	=	16.00	8.70

pv_1	=	10.50	11.15
pv_2	=	11.25	12.75
pv_3	=	15.75	8.95

Figure 2.4: Predicted values of the temperature for the selected time points.

The common scheme from Figure 2.3, has, for the case shown in Figure 2.1, the form shown in Figure 2.4 (the interest are the unknown temperatures at 10:30 a.m., at 11:15 a.m. and at 15:45 p.m.). The predicted values are shown in shadowed boxes. As shown in the example, the model predicts the unknown values through interpolation and/or extrapolation. In the procedure, it uses a non-linear function, whose shape is defined by the penalty coefficient (see Chapter 3).

3.2 Selecting variable types

We can always choose between more options in the modeling process. Basically, we distinguish between two types of variables:

- uniform type ($-\infty, +\infty$) and
- discrete type (integers only, e.g. 1, 2, 3, ...).

Discrete type of variable is just a special case of uniform type. It was introduced with the intention of speeding up the computing time, and for the sake of the simplifying the modeling in some cases. Its written form is more comprehensive and more convenient for use in some special cases. Unfortunately, the discrete type of variable may not be used as an output variable. Nevertheless, if we want to predict the value, already described by the discrete variable, it is necessary to change the description of the phenomenon. This method is shown in the following subsections.

As an example, take the phenomenon of the day-temperature model. By measuring the time and the corresponding temperature, the most logical selection seems to be the **uniform variable type** for both variables of the phenomenon. Variables can have theoretically all the values between 0 and 24 (if the time is measured in hours) or between -273°C and $+\infty^{\circ}\text{C}$ (if the temperature is measured in degrees Celsius). Model vectors have a very simple form and may be written as shown in Figure 2.1.

The phenomenon is non-linear. To allow inverse mapping, new additional variable must be introduced. Part of the curve, either ascendant or descendant, can be defined with a new **discrete variable type** (see Figure 1.5, Chapter 1). Model vectors can be written as follows:

model vector	=	time	temperature	period
mv ₁	=	8.00	5.30	1
mv ₂	=	9.00	9.10	1
mv ₃	=	10.00	10.50	1
mv ₄	=	11.00	11.70	1
mv ₅	=	12.00	13.10	1
mv ₆	=	13.00	11.30	2
mv ₇	=	14.00	10.40	2
mv ₈	=	15.00	9.20	2
mv ₉	=	16.00	8.70	2

Figure 2.5: The new form of the phenomenon of the day-temperature changing including the third discrete variable type.

All model vectors are transformed to the new, abstract hyperspace before the computation (prediction; see *Normalize Model Vectors* in User's Manual, Part 1). For this reason, the value size of the discrete variable type has no influence on the final prediction results or modeling of the phenomenon. For the sake of simplicity, it is recommended to use integer numbers in ascendant order (e.g. 1 and 2 in above example, Figure 2.5).

3.3 Preparation of model vectors in some other cases

Let us take a look at an interesting case, when in the evening the temperature suddenly increases due to temperature inversion (see Figure 2.6). We get a second, ascendant part of the curve, with the same sign of the derivative. The same coding may be used for the complete description of the phenomenon as in the previous case, only now, the third part of the curve has its own discrete value.

model vector	=	time	temperature	period
mv ₁	=	8.00	5.30	1
mv ₂	=	9.00	9.10	1
mv ₃	=	10.00	10.50	1
mv ₄	=	11.00	11.70	1
mv ₅	=	12.00	13.10	1
mv ₆	=	13.00	11.30	2
mv ₇	=	14.00	10.40	2
mv ₈	=	15.00	9.20	2
mv ₉	=	16.00	8.70	2
mv ₁₀	=	17.00	9.30	3
mv ₁₁	=	18.00	9.70	3

Figure 2.6: Coding of the phenomenon in the case of an increased afternoon temperature.

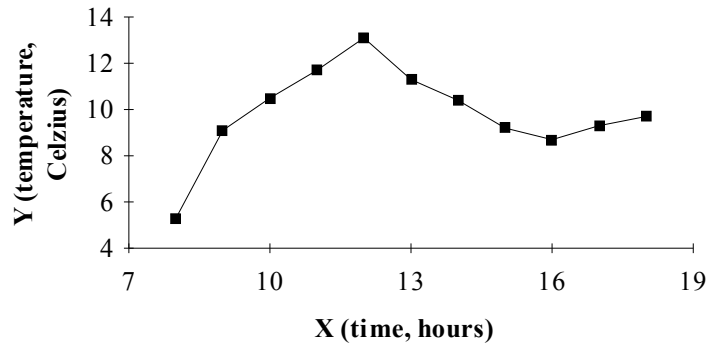


Figure 2.7. Graphical presentation of the phenomenon in the case of an increased afternoon temperature.

Generally, the description of the phenomena is independent of the selection of input and output variables (what is logical). For example, the model based on the description from Figure 2.6 simply gives the answers to the questions; what was the temperature at certain hour in the morning, in the afternoon or in the evening.

Given the hour and the temperature at that hour aiNet may then be used to determine the period of the event. With discrete variable types, the answer cannot be obtained at once, as mentioned earlier. But we can simulate discrete variable type with the uniform type as follows:

model vector	=	time	temperature	morning	afternoon	evening
mv ₁	=	8.00	5.30	1.0	0.0	0.0
mv ₂	=	9.00	9.10	1.0	0.0	0.0
mv ₃	=	10.00	10.50	1.0	0.0	0.0
mv ₄	=	11.00	11.70	1.0	0.0	0.0
mv ₅	=	12.00	13.10	1.0	0.0	0.0
mv ₆	=	13.00	11.30	0.0	1.0	0.0
mv ₇	=	14.00	10.40	0.0	1.0	0.0
mv ₈	=	15.00	9.20	0.0	1.0	0.0
mv ₉	=	16.00	8.70	0.0	1.0	0.0
mv ₁₀	=	17.00	9.30	0.0	0.0	1.0
mv ₁₁	=	18.00	9.70	0.0	0.0	1.0

pv ₁	=	10.50	11.15	0.99	0.01	0.00
pv ₂	=	11.25	12.75	0.99	0.01	0.00
pv ₃	=	15.75	8.95	0.01	0.99	0.00

Figure 2.8: Description of the phenomenon of the day-temperature changing in case of coding it with the uniform variable types only, and prediction for three cases.

The values may be interpreted as the certainties of each event. These events in above cases are certainties, that the temperature at the appointed hour was reached in the morning (first output variable), in the afternoon (second output variable), and in the evening (third output variable).

The final solution in Figure 2.8 is the event with the highest value. The results show that the model was successfully predicting all of the time periods. The correct answers have values higher than 99%.

It should be mentioned that both descriptions are identical in the case when time period arises as input variable. The computation is faster, than with the description using discrete types of variables.

3.4 Conclusion

The descriptions of the basic concepts introduced in Chapter 2 will be repeated.

Model vectors represent a way of coding the phenomenon in the mathematical form as vectors whose components are important variables of the phenomena. Variables of the phenomenon are divided into *input variables* and *output variables*. Basically, the variables may be described in two types: *a uniform type* and *a discrete type*.

General advice.

- During the preparation of the knowledge base (or data base) the assurance of representativity of model vectors is of great importance. It is well known that neural networks give very good results in cases of interpolation in problem space, but possibly very poor, or even useless results in cases of extrapolation. We need, therefore, to assure, that the model vectors "cover" all, or at least as many combinations of extreme values (minimum and maximum values) of the input variables as possible. In the case where all combinations of input variables of model vectors are possible, the necessary number of model vectors with extreme values is at least 2^N . N represents the number of input variables in a model vector (input variables of the phenomenon).
- In some practical examples, when the number of input variables is large, the above requirement may be too excessive. Luckily, as practical experience shows, such examples are very rare. Although this might happen, we will model the phenomenon with a large number of variables and the above requirement will not be satisfied. The results may be useful or even good, but we must bear in mind, that some of predictions were obtained by extrapolation, which is physically questionable.
- The preparation of an efficient model depends on the appropriate selection of variables of the phenomenon on one side, and the sufficient number of descriptors of the phenomenon (model vectors) on the other. In practice, we are usually limited by the number of model vectors. There is a relationship between the number of variables of the phenomenon and the number of model vectors, which can not be determined analytically. It is usual, that a higher number of variables demands a higher number of model vectors for sufficient modeling.

Chapter 3

4. Measures for the Estimation of Prediction Error and Preparation of Efficient Model

4.1 General

The use of classical neural networks (e.g. back propagation neural network - BP NN) introduces the problem of training (learning). Instead of concentrating on the problem itself, a lot of time must be spent determining the learning coefficients. Here is the point where science comes very close to art while, except in rare cases, there are no general valid rules or mathematical expressions, which define procedures or ways to selection of learning coefficients so far. Literature, and experience also, shows, that the selection of learning coefficients in many cases strongly depends on concrete phenomenon. The selection of learning coefficients in the best cases, may be generalized to very similar phenomena. There is often one unavoidable, and difficult problem - noise in the data. The user (researcher) never knows, if the neural network is stuck in some local minima, if the data is very noisy, or if a better result is possible, and so on.

aiNet, compared to other similar tools for dealing with neural networks, allows creative work. The researcher can concentrate most of her/his time on the modeling of the phenomenon. For this reason, she/he has at a disposal a few, very simple tools. We will not describe the theoretical backgrounds in detail here; it should be mentioned only, that all the methods are based on multidimensional non-parametric regression (see [7],[8]).

Taking into account the assumption that the preparation of model vectors is independent of the tool, it may be determined that the final aiNet model depends on one coefficient only (see subsection *Penalty coefficient*). With regard to the BP NN, which demands the selection of at least three learning coefficients, the selection the number of hidden layers and the number of neurons in these hidden layers, aiNet's advantages are evident. aiNet does not demand classical learning in most practical cases and the order of presentation of model vectors is also unimportant (commonly, we can choose among at least four presentations, see [9]). The procedure is insensitive to noise in the

⁷ Grabec, I. & Sachse, W., **Automatic Modeling of Physical Phenomena: Application to Ultrasonic Data**, J. Appl. Phys., 69 (9), 1991.

⁸ Grabec, I., **Modeling of Natural Phenomena by a Self-Organizing System**, Proc. ECPD NEUROCOMPUTING, Vol. 1, No. 1, 1990.

⁹ Ramirez, M., R. & Arghya, D., **A faster learning algorithm for back-propagation neural networks in NDE applications**, Proceedings of the 2nd International Conference on AI, pp. 275-283, 1991.

data. Therefore, the model can give good results even in cases of very high noise level. The final judgment of validation of the above, and the comparison of the efficiency of various neural network models (aiNet models and others) is left to the users of the program.

4.2 Measures for the Estimation of Prediction Error

Modeling phenomena urgently demands mathematical assessments of the prediction error or reliability of the results*. The definition of the global or local prediction error is independent of the used method (filtration, verification, prediction; see the explanation in next subsections). It may be simply written as:

$$E = x_m - x_p,$$

where lower indexes m and p present the measured value of the variable and the predicted value of the variable, respectively. In a more complicated description

$$E_{ij} = x_{m_{ij}} - x_{p_{ij}}$$

the lower index i describes the i -th model vector and lower index j the j -th variable of the model vector. Such a defined error is called the **signed local error**.

The Signed local error has sense in cases only, where there is no noise in the data, or the noise is very small. In all other cases, we determine the **absolute local error**. As the name tells, the description is similar to the description of a signed local error. The Local error is determined as the absolute value:

$$\overline{E}_{ij} = |x_{m_{ij}} - x_{p_{ij}}|$$

RMS (prediction) error on the variable is then defined as square root of sum squares (error power square) divided by the number of model vectors (N):

$$G_j = \sqrt{\frac{1}{N} \sum_{i=1}^N (E_{ij})^2}$$

Total RMS (prediction) error is defined similarly, the expression under the square root is divided with the number of all output variables (M):

$$G = \sqrt{\frac{1}{N * M} \sum_{j=1}^M \sum_{i=1}^N (E_{ij})^2}$$

It should be mentioned that when there is only one output variable, the RMS prediction error on the variable is the same as the total RMS prediction error.

Two different options may be used for normalizing model vectors with aiNet. The RMS error on variable and total RMS error are therefore not directly compared with similarly defined measures in other tools, which use the transformation in the unit hyperspace.

* aiNet gives assessment of the prediction error on the whole data base; the procedure is described in details in the relevant subsections.

4.3 Penalty coefficient

Different statistical methods demand the selection of the shape (type) of the function, which best suits the description of the phenomenon. The coefficients of empirical (regression) equations are determined then with the method of the least squares. In such a way we try to describe the phenomenon with some advanced, presumed empirical law. While the data is usually incomplete, the selected law is fitted closely to the available data and usually fails only when more data is acquired. The available databases are not representative in most of the practical cases and therefore the automatic modeling of the phenomenon is very appropriate when new data is obtained.

Compared to the above mentioned parametric methods, the non-parametric** model, prepared by aiNet, does not need any advanced, presumed law. Also, when the data is changed, or new data is added to the data base, aiNet adapts the model automatically. Only one coefficient, the ***penalty coefficient***, must be defined. It has an indirect relation to the learning error (and/or learning threshold) in classical neural networks - that both influence the final solution. The influence of the penalty coefficient on the solution is illustrated using the concrete example in the third part of the User's Manual.

The Penalty coefficient determines the shape of the curve in two-dimensional problems, and the shape of the hyperplane in three or more multi dimensional problems, respectively. While the penalty coefficient influences the accuracy and/or the efficiency of the model, the determination of its optimal value represents the key point in the modeling process. Tools which help users find the right value of penalty coefficient and help better understand the phenomenon, are described in the next subsections. It should be mentioned at this point, that there are few opportunities for changing the penalty coefficient in problem space. The most simple variant is the one, where the penalty coefficient has a constant value. The other possibilities are mostly dynamical adaptations, where its value is additionally modified for each point in the hyperspace by different built-in methods.

4.4 Modeling tools

Modeling tools will be explained using the example of the day-temperature changing. The temperature at the beginning and at the end of the experiment was measured in two, relatively short time steps using the same instrument. Somewhere in the middle of the experiment, the temperature was measured in two random time points using two different instruments (see Figure 3.1).

** The term *parameter* in this particular case relates to the coefficients of regression equations which are usually used for the description of phenomena!

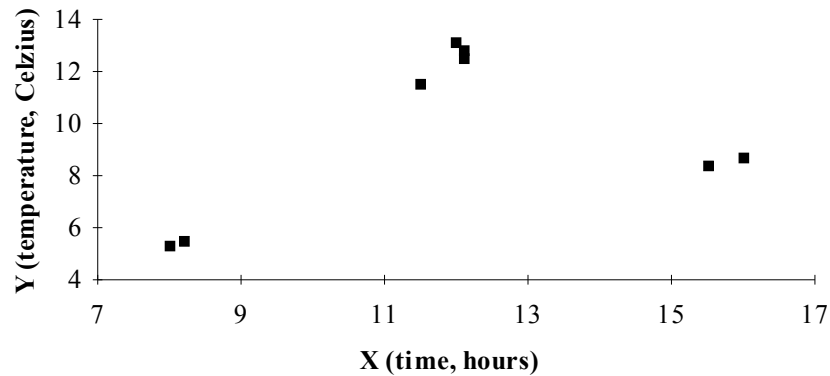


Figure 3.1. Phenomenon of the day-temperature changing - example of one of many possible descriptions.

Filtration tool

The most basic tool in the modeling process is filtration. It gives the most straight forward estimation of the noise in the data. The value of *each output variable for each model vector* from the data base is predicted on the basis of *all model vectors* from the data base. A small value of penalty coefficient allows the estimation of the noise in the data with one of the RMS global errors (choice depends on the user and its knowledge about the phenomenon).

It should be noted, that the noise is not only the function of errors in measurements, or function of subjectivity in preparation of model vectors. It may happen that one of the important influencing variables is excluded (is not taken into account) from the modeling process. Even a theoretically perfect data base will give in such cases an error in the prediction which cannot be avoided. The error is equal to the influence of the excluded variable. High noise (estimation of one of the RMS errors) may be a good indicator of exclusion of one or more influencing variables of the phenomenon. The problem is not entirely trivial, while the phenomenon may be chaotic, too.

Very noisy data might significantly slow down or even block the training of usual neural networks in some cases. Filtration is a very useful tool in these cases. It can be used as a filter for the preparation of the new data base, which is free (or almost free) of noise. The data can then be normally used for the training of other neural networks, for example, BP NN.

Verification tool

Verification of models in case of using classical neural networks is based on a very simple principle: two thirds of the available data is used for the preparation of the model, and the last third is used for the verification (the selection of model vectors in both groups is random). A basic problem in such procedures is the size of available data bases in reality - in most cases we have only a limited number of model vectors. If from already limited number of model vectors one third has been cut, a very truncated data base is left. A model, prepared on a truncated data base may give poor predictions, which may lead to the conclusion that the model is poor.

Verification in aiNet works on a very simple principle. The value of *each output variable for each model vector* from the data base is predicted on the basis of *all other model vectors* from the data base, which means that the model vector under consideration is temporarily removed from the data

base. Such a procedure gives the global error estimation which is a **better** global error estimation, as it is determined using the whole data base. We might say, that this is the prediction error. The last statement is more representative using a good data base, and many model vectors.

Prediction tool

The key tool in the sense of the use of efficient non-parametric model is prediction. When the model is prepared (using the tools above, using the input variables of the phenomenon and using the value of the penalty coefficient), the prediction may be used for the determination of output variables of the phenomenon for model vectors given only input variables. We want to predict the unknown values of the variables, which completely describe the phenomenon in one, special case, for example:

- we want to know the temperature at 01:32 p.m. in case of the day-temperature changing,
- what is the diagnosis in case of back pain, if the patient is up to 60 years, and the pain appears just on one side of his back, and so on.

Prediction is a tool, intended for the end user.

4.5 Relations between modeling tools

Qualitative relations between the measured and the predicted values, which are given by different modeling tools, are shown in Figure 3.2. The case from Figure 3.1 is used as a test example.

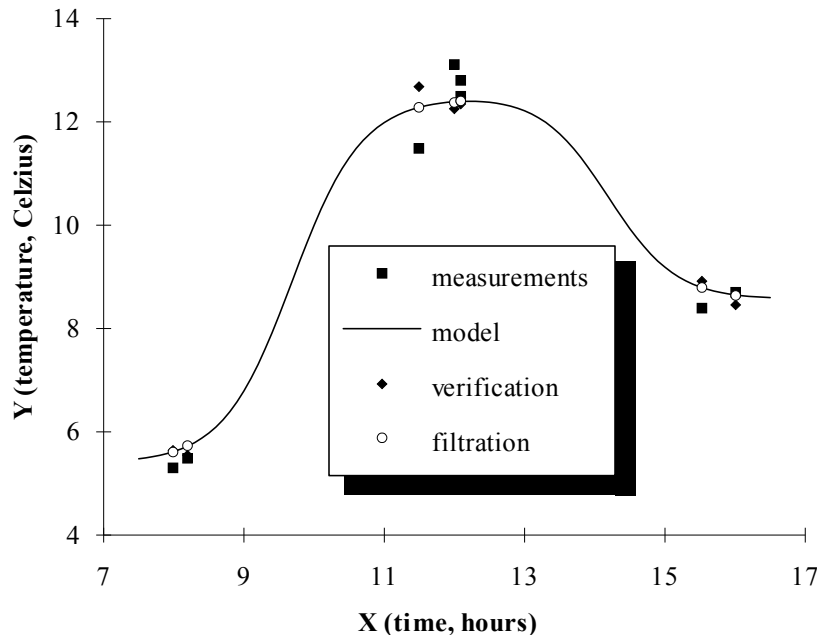


Figure 3.2. Relations between measured and predicted values for different modeling tools.

Solid squares show the measured data. The dashed line shows the curve, which describes the phenomenon on the basis of all measurements (model vectors from the data base), and it has been

calculated using a prediction tool. A filtration tool determines the points on that curve (labeled with a circle). Verification determines points (labeled with solid diamond); in general, these points have the greatest distances to the measured points. The Figure also shows that the filtration tool gives the lowest value for the global error estimation, and the verification tool the highest value. In the case of a perfect data base and a very large number of model vectors, both estimations converge in a single value which, in reality, will almost never happen.

Figure 3.2 shows another interesting fact. We have more data in an area (approximately at noon), which seems to be noisy. Both absolute local errors, obtained either by a filtration tool or by a verification tool are approximately the same in some points in this area. This does not only prove the above statement, but it also shows that aiNet accepts noisy data without difficulty.

The illustration of the influence of the penalty parameter on the solution will be shown quantitatively. We chose the case with static (constant) value of the penalty coefficient in the problem domain. Figure 3.3 shows qualitative relations between estimations of global errors for both modeling tools. The curve of verification tool has a shape known from the optimization problems. The optimal solution is obtained at one single value of penalty coefficient. This is not the optimal solution, but it is the optimal solution of the available data base. As may be seen from Figure 3.3, the optimal value of penalty coefficient can be simply determined by different methods, e.g. bisection.

Error estimation of the filtration tool always converges in some value. If this value is zero and the data base is representative, we can state that the noise in the data is very low. The higher the value, the more noisy the data and the less efficient the model.

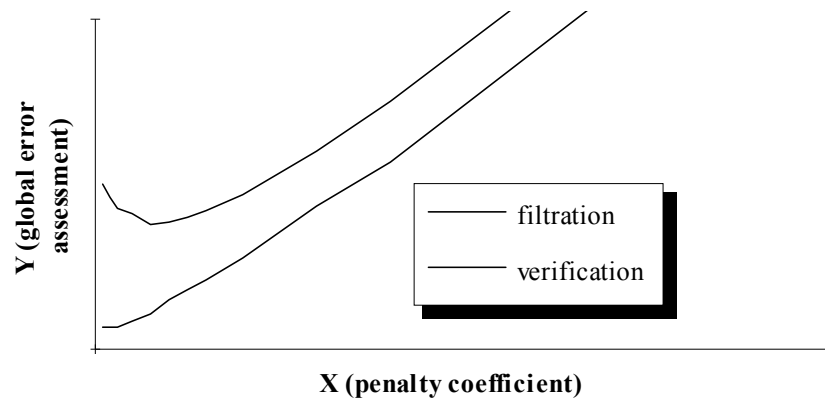


Figure 3.3. Relations between global error estimations for filtration and verification for variation of penalty coefficient.

4.6 Assistant tools

aiNet contains some assistant tools which allow simple control and visualization of the predicted results. These are different graphical tools which show comparison between the predicted and the measured results of the phenomenon, and the estimations of absolute local errors for model vectors from the data base. Assistant tools may be used in combination with all other modeling tools (filtration, verification, ...).

For the end user, the estimation of either signed or absolute prediction local error is of great importance. For this reason, the prediction tool gives an estimation of local error, too. While it strongly depends on local characteristics of the phenomenon, we suggest the use of local error estimation in prediction together with one of global error estimations, obtained with the verification or the filtration tool. Experienced users can prepare their own criteria for the reliability of the model, based on these two error estimations.

Figure 3.4 shows the estimation of absolute local error for the case from Figure 3.1. The example is changed in a way that higher noise was added to some measured data in the middle area (approximately at noon), and additional measured data in equidistant time points were added.

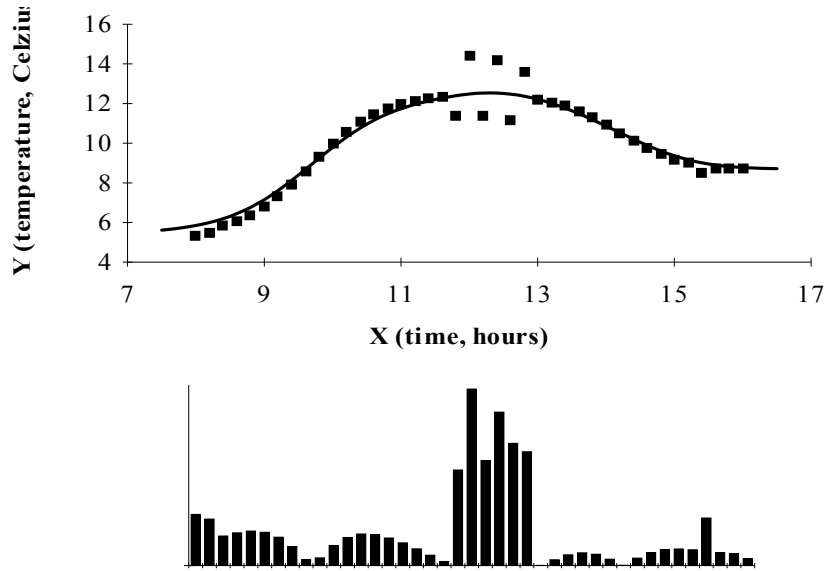


Figure 3.4. Estimation of absolute local error.

Additional testing tools, mainly with graphical output, are being designed and tested for future inclusion into aiNet.

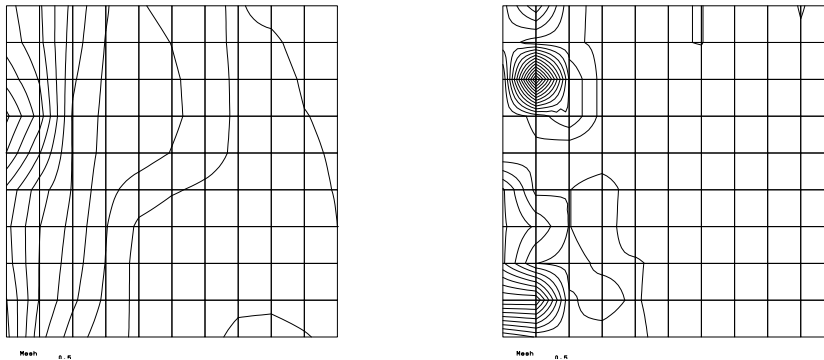


Figure 3.5: Graphical presentations of prediction, and distribution of local prediction error.

It will show the isolines (lines of equal value) of the output variable at variation of two input variables and fixed or excluded values of other input variables. One of the important results might be the distribution of the estimation of local prediction error, which may be shown in the same manner in the same or in another picture (see Figure 3.5). A theoretical basis of these tools along with main ideas will be presented when the tools have been included into aiNet.

Chapter 4

5. Conclusion

This part of User's Manual briefly describes basic concepts about modeling with aiNet, and the available tools in the program. The authors expect that the brief description will offer appropriate grounding for working with the program. Additional assistant tools will be prepared in the future.

Authors gratefully acknowledge all comments and warnings, with regard to either the user's manual or the program.

Chapter 5

6. References

1. Ramirez, M., R. & Arghya, D., **A faster learning algorithm for back-propagation neural networks in NDE applications**, Proceedings of the 2nd International Conference on AI, pp. 275-283, 1991.
2. Samaad, T., **Backpropagation Improvements based on Heuristic Arguments**, Theory Track, Neural and Cognitive Sciences Track, International Joint Conference on Neural Networks, Vol. 1, Washington, D.C., 1990.
3. Johansson, E., M., Dowla, F., U. & Goodman, D., M., **Backpropagation learning for Multi-layer Feed-forward Neural Nets Using the Conjugate Gradient Method**, Lawrence Livermore National Laboratory, 1990.
4. Shanno, D., F., **Conjugate Gradient Methods with Inexact Searches**, Mathematics of Operations Research, Vol. 3, pp. 244-256, 1978.
5. Kohonen, T., **Self-Organization and Associative Memory**, Second Edition, Springer-Verlag, Berlin, 1988.
6. Grabec, I., **Self-Organization of Neurons Described by the Maximum-Entropy Principle**, Biol. Cybern., **63**, pp. 403-409, 1990.
7. Grabec, I. & Sachse, W., **Automatic Modeling of Physical Phenomena: Application to Ultrasonic Data**, J. Appl. Phys., 69 (9), 1991.
8. Grabec, I., **Modeling of Natural Phenomena by a Self-Organizing System**, Proc. ECPD NEUROCOMPUTING, Vol. 1, No. 1, 1990.
9. Casciati, F. & Peruš, I., **Neural Networks: an Alternative Answer to Structural Engineering Needs**, ECC Worskhop on ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS TRAINING IN STRUCTURES, Athens, January, 1994.