Neural Network

Menu:	QCExpert	Predictive methods	Neural network
DARWin:	**NNLEARN**		

Neural network (Artificial Neural Network, ANN, or NN) is very popular and powerful method, which is used for modeling the relationship between multivariate input variable **x** a multivariate output variable **y**. NN is generally considered to be non-linear regression model, which can make the network structure. The inspiration for the neural network structure of brain tissue was higher organisms, which neuron is connected with the so-called synapses to several other neurons. Electrical current (or information signal) flows through synapses, is processed by a neuron and transmitted by other synapses to other neurons.

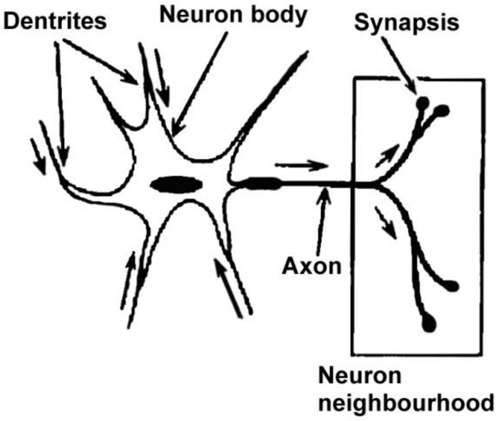


Fig. 1 Biological neuron: cell body, input dendrites, output axon, synapses-connection to other neurons

The artificial neural network tries to copy the structure and functionality of the biological neural structure and models the structure mathematically. Neuron core is represented In ANN the nodes are, by analogy, called neurons, each input variable x_i entering the j-th neuron multiplied by a weighting factor of w_{ji}. The sum of the weighted input variables $z_j = w_{0j} + \Sigma w_{ji}x_i$ is then transformed by neuron using an activation function. Activation function expresses the intensity of the neuron response to the input change. The most commonly used activation functions include logistic functions,

$$\sigma_j (z) = 1 / (1 + e^{-z}),$$

which is similar to the biological function of sensory response, for example: there is practically no difference if you touch temperatures 50K or 150K (both are too cold) or temperatures 2000K of 4000K (both too hot). But you will very precisely distinguish between 90 and 100°F (35 and 40°C), because here is the vital information. Weights w_{ji} represent the intensity of information flow between the variable and neuron or, in the case of multi-layer networks between neurons in layers, these links are sometimes called synapses by analogy to the bio-neurons and can be interpreted as significance of variables and visualized in a plot.

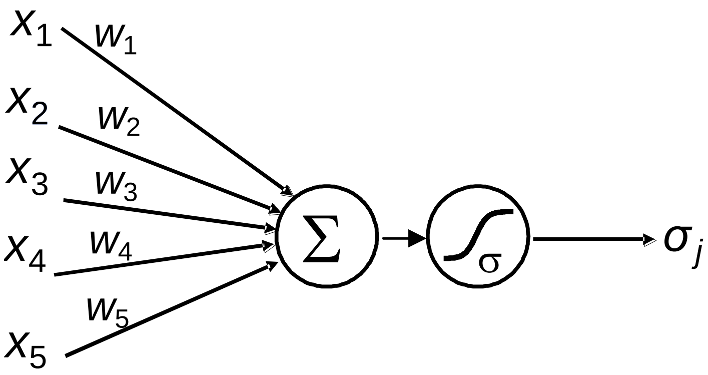
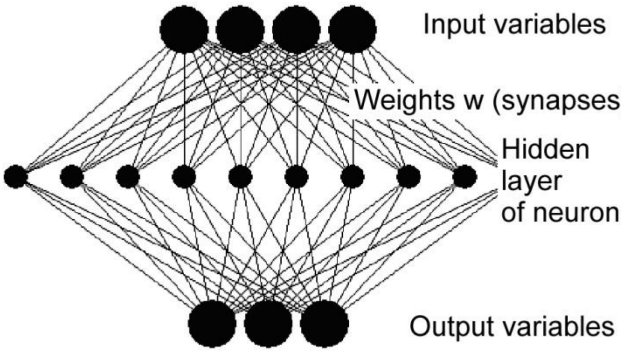


Fig. 2 Structure of an artificial neuron



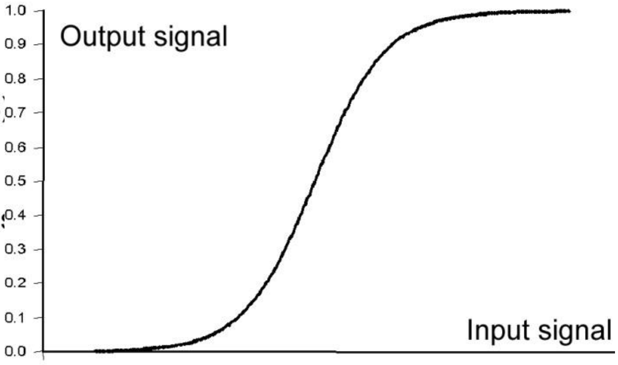


Fig. 3 Possible architecture of an ANN with 1 hidden neuron layer

Fig. 4 Commonly used activation function of a neuron σ(*z*)

Output variables are predicted as weighted linear combination of outputs from the last hidden layer neurons, $\hat{y}_i = \sum_k w_{ik}\sigma_{Lk}$. Neural network is therefore formally a special case of multiple nonlinear regression, neural network can be practically considered non-parametric regression. If the neural network did not contain any hidden layer neurons – only input and output variables, it would be a linear regression model. Neural network is optimized to satisfy least residual squares criterion. This means that the network is set so that the squares of the differences between prediction and the measured output variables value was minimal. This is the aim of iterative optimization process, which is called learning or training neural networks by finding the best values of all weights. QC.Expert™ uses an adaptive derivative Gauss-Newton algorithms to optimize the net. Trained network can then be used for prediction of output variables, for new specified input variables. Neural network model is local, that means that its prediction ability is sharply declining outside the range of the independent variables used to train it, see Fig. 5.

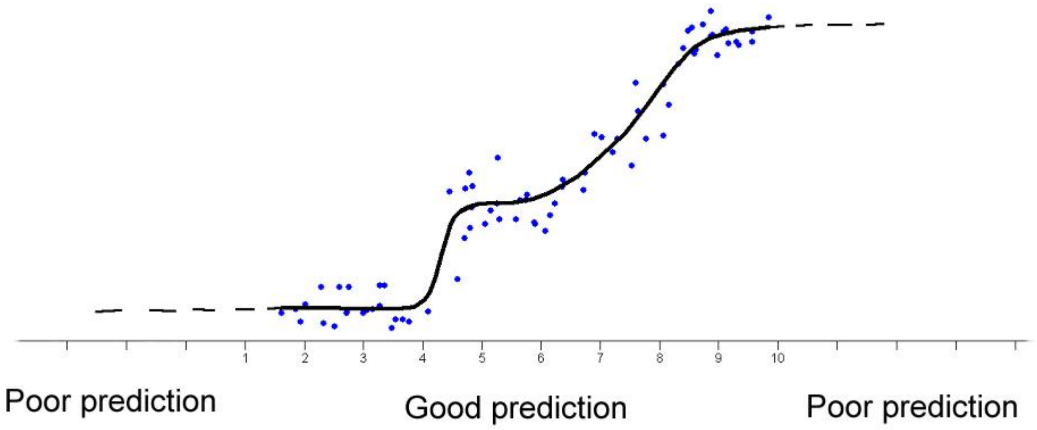


Fig. 5 Prediction capability of an ANN drops dramatically in the areas where no training data are available

A typical procedure for using neural network may be as follows.

1. Select group of predictors (independent variables, *X*) which we believe that may affect the dependent variables *Y*. Select a group of dependent variables, which should depend on the predictors. In each line must always be all values of dependent, and independent variable. Number of rows is denoted *N*.

2. Select the architecture of the neural network, the number of layers and numbers of neurons in each layer. There is no straightforward rule for the best network architecture, usually it is appropriate to use a number of neurons very roughly corresponding to the number of variables. Single hidden layer networks are recommended where we assume a linear, or slightly non-linear relationship. Two-layer network can be suitable for strongly nonlinear relationships. Using more then 3 layer networks is usually not very effective. It is necessary to keep in mind that for very complex network there is high risk of overdetermined ambiguous and unstable models or models which are difficult to optimize. Examples of possible architectures are given on Fig. 18. Number of data (lines) should be at least ten times greater than the number of neurons in the network, otherwise there is a risk of overdetermination and the ability of prediction may decrease. Usual architectures for middle to large-scale problems are networks with 2 to 20 neurons, and 1 to 3 layers.

3. Optimizing parameters of the network, or the so-called "learning" neural networks. During this process, the optimization algorithm tries to find a set of weights, so that the predicted values are in the best accordance with entered dependent variables. This consistency is measured by the sum of squares, as in other least squares regression methods. In general, it can not be guaranteed that the found solution is the best possible. Therefore, it is advisable to run optimization several times to get better residual sum of squares (RSS). Optimization starts with random values of the weights, it is therefore natural that each solution found by optimizing is completely different. Even completely different combination of the weights in the network can provide virtually identical prediction model with the same RSS.

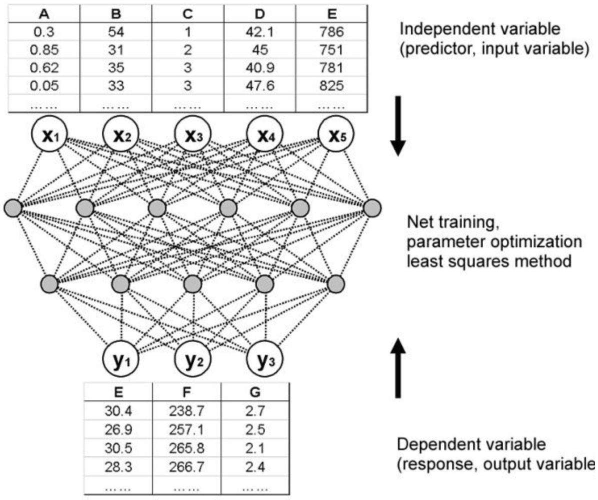
4. If an information about the reliability of prediction is required, we can use cross-validation. In this technique we select the so-called training, or learning subset of data , say *P*.*N* lines ($0 < P < 1$) to be used to train the network. The rest of the data, the remaining $(1 – P)$.*N* lines of

testing or validation data, are then used to validate network, i.e. check if the predicted values for the validation data are close to the actual dependent data.

5. The success of neural networks can be assessed according to the decrease of squares sum during the optimization process, according to fit plots of prediction and by the thickness of the lines connecting neurons (the thickness is proportional to the absolute value of the weight, which is interpreted as the intensity of information flow downward from predictors to response).

6. Prediction: A trained network may be used for predicting response variable. Put new values of the independent variables on input of the network. The structure of the variable must be the same as used to train the ANN and values should be in the same range. The network will predict the output values.

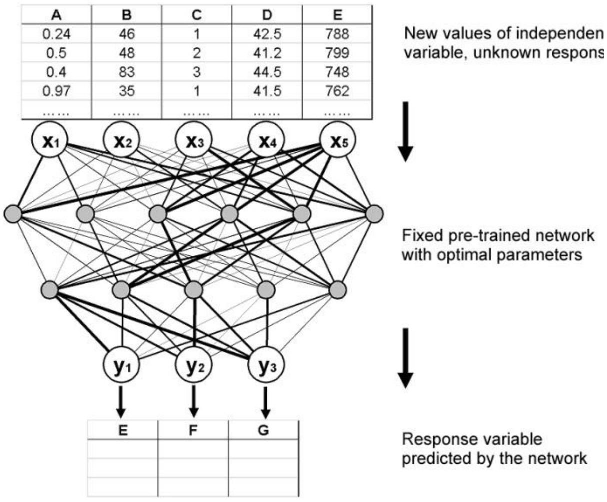
The steps are shown on Fig. 6 and Fig. 7.



A	B	C	D	E
0.3	54	1	42.1	786
0.85	31	2	45	751
0.62	35	3	40.9	781
0.05	33	3	47.6	825
……	……	……	……	……

E	F	G
30.4	238.7	2.7
26.9	257.1	2.5
30.5	265.8	2.1
28.3	266.7	2.4
……	……	……

Fig. 6 The ANN training process



A	B	C	D	E
0.24	46	1	42.5	788
0.5	48	2	41.2	799
0.4	83	3	44.5	748
0.97	35	1	41.5	762
……	……	……	……	……

E	F	G

Fig. 7 Using trained ANN for predicting unknown response

Model validation

ANN models usually do not allow the calculation of statistical parameters and diagnostics for a detailed assessment of the quality of the model (variances of the regression coefficients, F- and t- statistics for testing the relevance and significance of the model, diagnostic plots, etc.), as in the case of linear regression. It is therefore necessary to use other methods, to verify whether the model is appropriate for description of the phenomenon under study. Neural network is a very flexible instrument and can easily lead to a situation where the model will suspiciously well describe (fit) the data, but not the phenomenon (variables relationship) as a whole. This is reflected in very poor prediction of the values of the dependent variables for the new independent variables that have not yet occurred in the data, although they may be located inside the interval of training data.

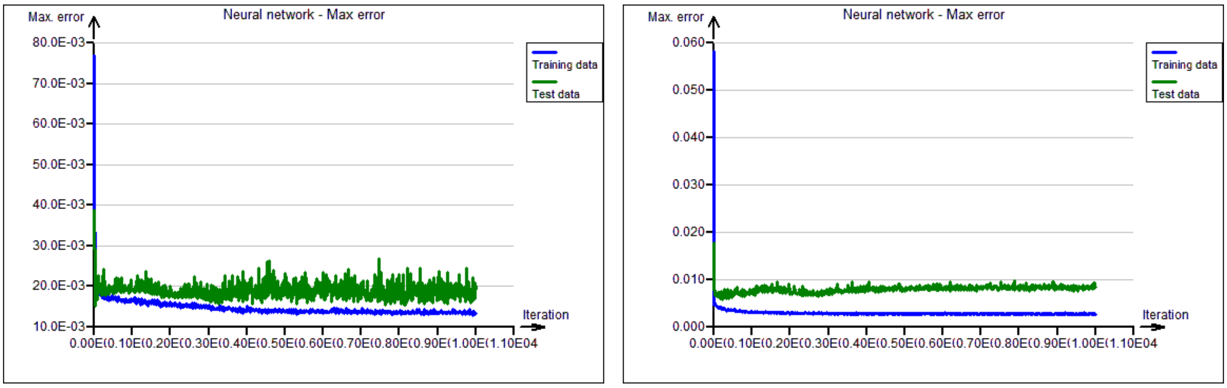


Fig. 8 ANN optimization process with cross-validation: very good prediction capability

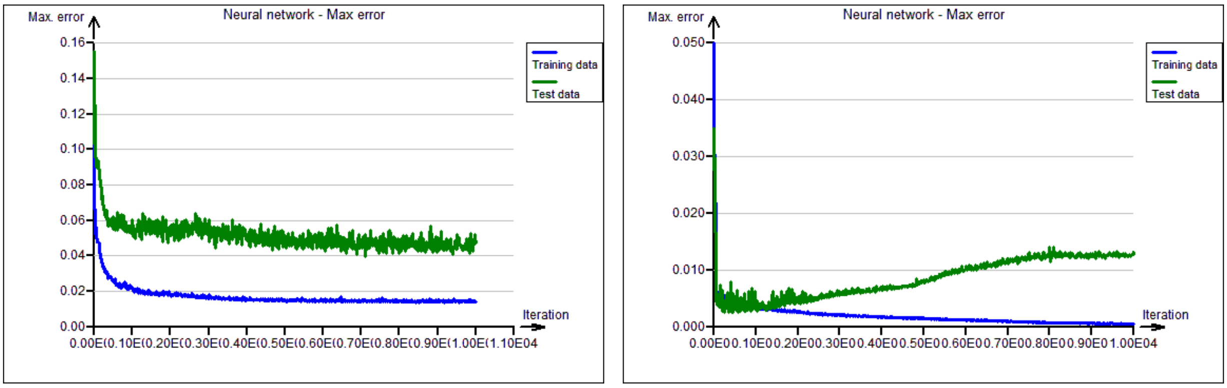


Fig. 9 ANN optimization process with cross-validation: fairly good prediction capability

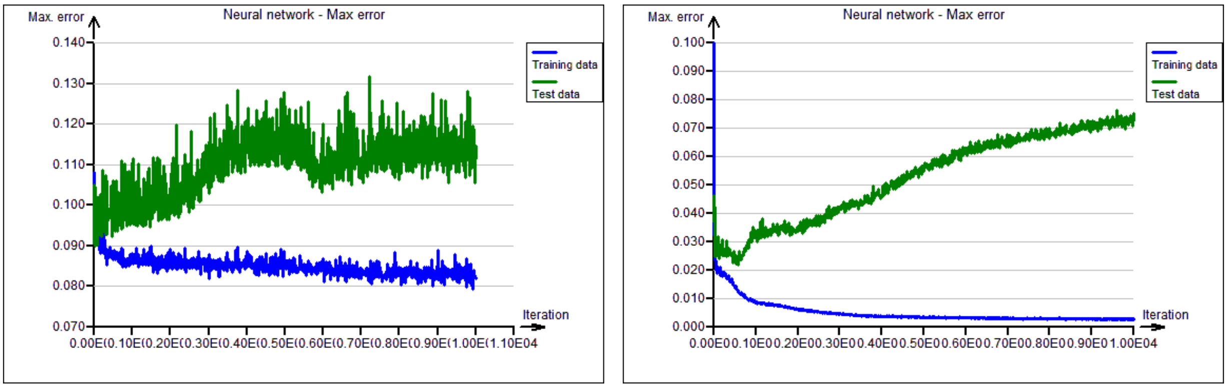


Fig. 10 ANN optimization process with cross-validation: poor or none prediction capability. The network is too complex or data size is too small, possibly there is no information in the data

To assess the prediction capabilities of the neural network validation (cross-validation) is used. Validation is based on a simple principle of training the ANN only with a certain part P of the data. This training part is chosen typically around P = 0.7 to 0.9, or 70 to 90%. The $1 - P$ rest of the data (test or validation data, not „seen“ during training the ANN) is afterwards used to calculate prediction, which which is compared to the true value of response. If this prediction agrees well with the actual response, we can confirm the ability of the ANN to correctly predict the response for data, which it has not „seen“ previously. The quality of prediction can be assessed using a graph or chart of errors during the the progress of optimization process. The Fig. 8 to Fig. 10 illustrate the use of this concept for three models, using 30% (P = 0.7) of randomly selected validation data. Fig. 11 through Fig. 14 illustrate different capabilities of NN-prediction models on the Data-prediction plot.

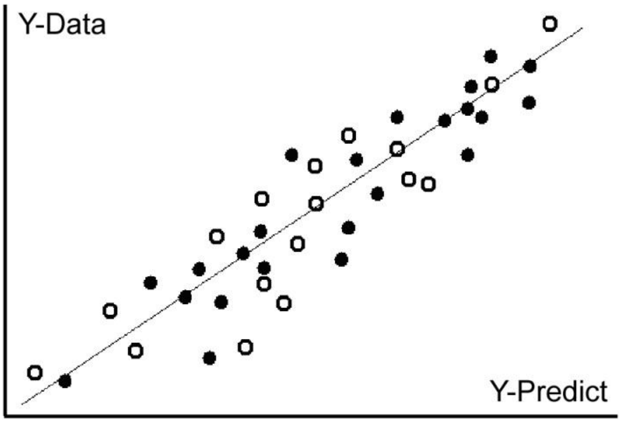


Fig. 11 Comparable quality both for training (filled) and validation (hollow) data. Good prediction capability

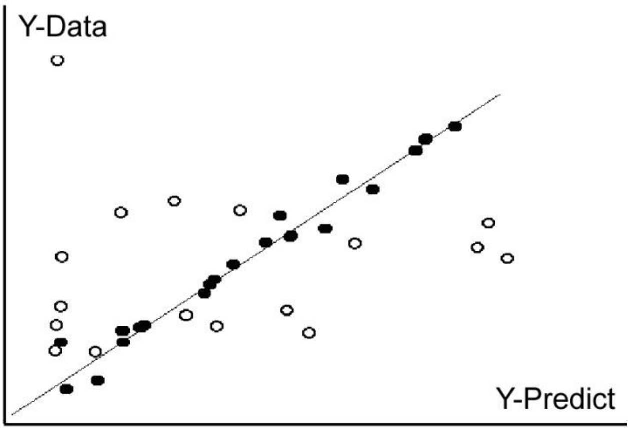


Fig. 12 Good prediction training data, but very poor for the validation data (hollow). Poor prediction capability of the network. Network probably too complex.

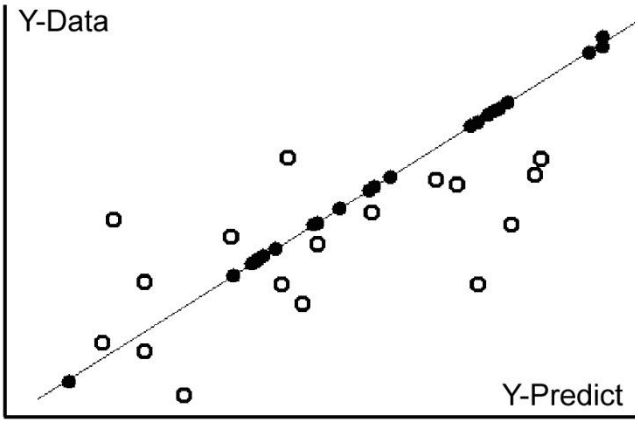


Fig. 13 Perfect prediction training data, but very poor for the validation data (hollow). Network probably too complex.

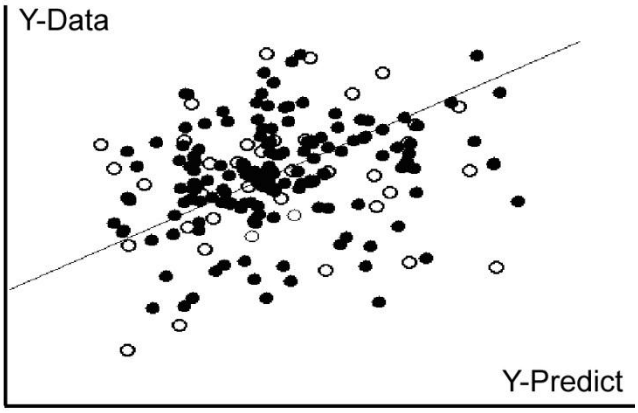


Fig. 14 Poor fit for both training and the validation (hollow) data. Try to re-run optimization, add layers or neurons, possibly there is simply no dependence between input and output.

Classification with ANN

Given the probabilistic nature of the logistic activation functions neural networks can as well be used as a modeling tool for classification when the output is a discrete variable - two-level (binary), like 0 and 1, or multi-level such as 1, 2, 3, or A, B, C. Neural network predicts the level of output variables for the given values of the independent variables, as in logistic regression. In the case of a binary response 0 – 1, prediction can be considered as the probability of occurrence of „1“. The following plots on Fig. 15 and Fig. 16 illustrate the use of ANN as a classification model. On the left there is the measured response (bright point corresponds to the value of 0, dark point value 1). On the right is a shaded map of the prediction obtained by neural network. Plots were obtained by the module Graphs - 3D-Spline.

Table 1 An example of classification data: Technological parameters X and Y (independent variable) presumably influence the result of an operation (response = OK/Fail, or 0/1). For the ANN, numerical form of the response (like 0/1) is required. The response may have more levels denoted e.g. 0, 1, 2, … or 100, 200, 300.

Parameter X	Parameter Y	Result	Result - Binary
1.5	3.1	Pass	0
1.9	2.2	Pass	0
3.5	2.8	Non-conforming	1
2.9	4.3	Non-conforming	1
2.4	2.7	Pass	0
.....	.....	.....	.....

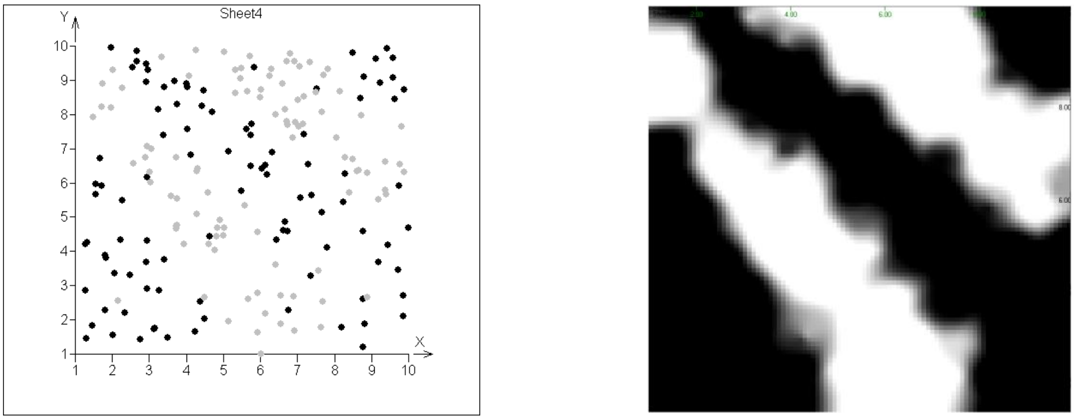


Fig. 15 Linearly separable data – One hidden layer NN with 6 neurons used as a classification model.

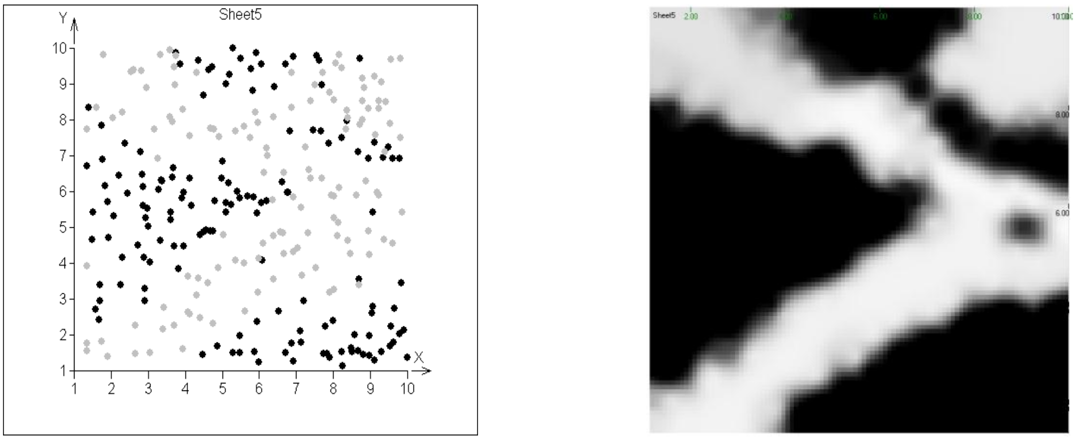


Fig. 16 Linearly inseparable data – Two hidden layers NN with 5+5 neurons used as a classification model.

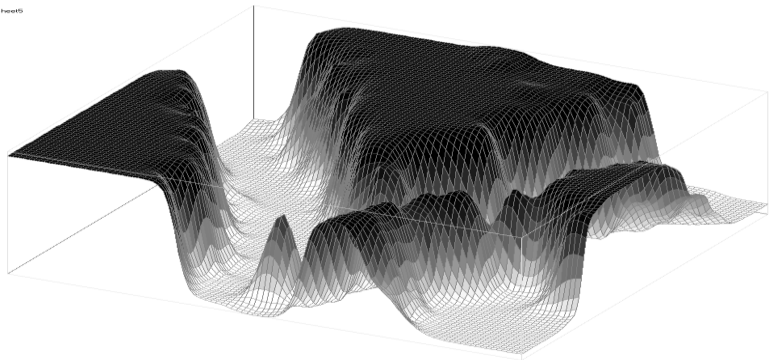


Fig. 17 3D representation of the previous example using NN prediction and 3D-Spline from the Graphs module

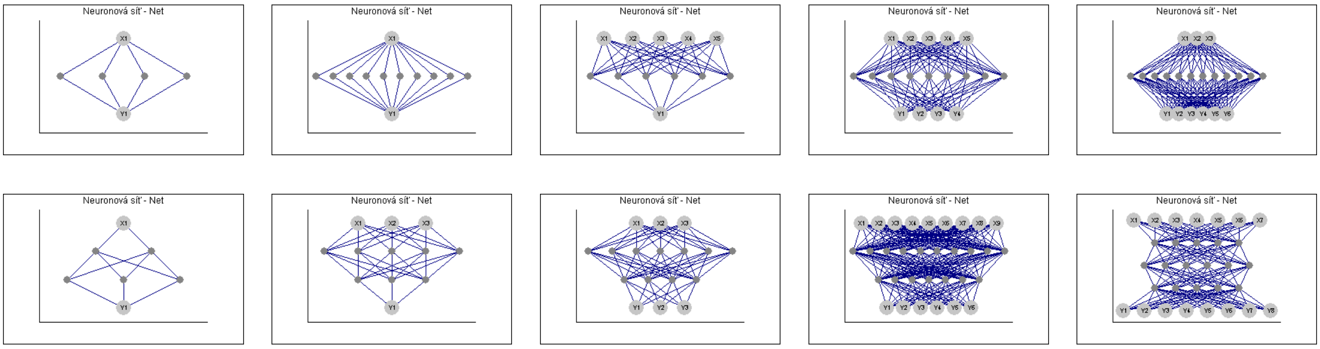


Fig. 18 Illustrative examples of suitable NN architectures. An ANN must be designed with respect to the data size and nature

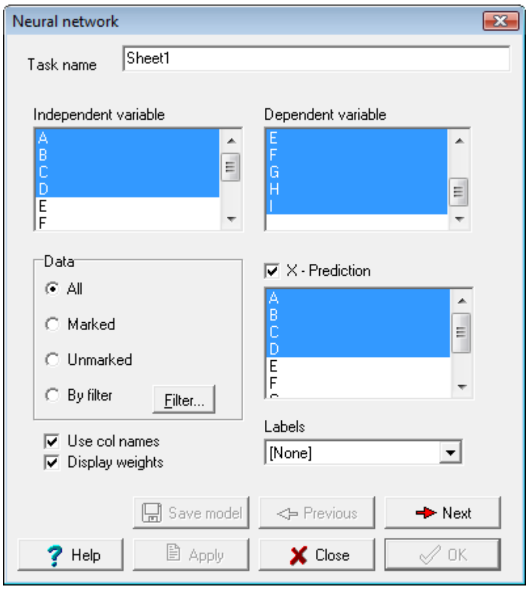
Data and parameters

Module Neural Networks has several consecutive dialog boxes, which can set the parameters of computation. In the first dialog box, the columns of independent and dependent variables are selected. In the Data group you can select the required data subset: all data, or just a specified subset of rows. Checking the box *Prediction* to calculate the value of prediction of the dependent variable for selected independent variable columns. We have to select the same number of columns, as we have selected in the field *Independent variables*, the values for prediction must also have similar values as the independent variable for reliable prediction.

The same columns can be selected as in *Independent variable* field. If the field *Use Col names* is checked, the names of the columns are used to describe the input and output neurons in the graph. If *Display weights* is checked (recommended), the absolute value of weights are visualized as the thickness of connecting lines between neurons. The sign of the weight is represented by color (blue = positive weight, red = negative weight). Click *Next* to get to the next window. In the *Neural network architecture* dialog window we will define the network architecture – number of layers and number of neurons in the layers. Typical number if layers is 1, 2 or 3 layers. More than 3 layers may be useful only in some specific cases. The *Number of neurons in the hidden layers* field determines how many neurons to include in individual layers. The problem of how to choose a suitable architecture is discussed below. The *Number of iterations* field determines the length of the calculation in terms of number of iterations of the network optimization process, recommended default value is 10000. The *Exponent* determines exponent of the criterial function, here, the default is 2, which corresponds to least squares method. Exponents between 1 and 2 will somewhat robustify the network and are recommended when the data are suspicious for outliers or possible big errors in dependent variables.

$$s(NN) = \Sigma\,|y - y_{\mathrm{pred}}|^{k}$$

Parameter *Sigmoid steepness* indicates speed (sensitivity) with which the neurons respond to change of the independent variables. The recommended value is 1. The parameters *Moment* and *Learning speed* affect the optimization algorithm. Recommended values are 0.9 and 0.1. The field *Part of training data (%)* determines what part *P* of the data is to be used for training the network. The rest (1 – *P*) of data is then used for cross-validation. Cross-validation is a technique that will check stability and prediction capability of the chosen neural network model by using only part (typically 70-90% of the original rows) of the data to train the network. Then, the dependent variable of the rest of the rows is predicted from independent variable values and the predicted values are compared to the actual, measured values (never seen before by the network). To use the cross-validation, we usually choose the ratio between 60 and 90%. Choosing 100% will disable cross-validation and all the data will be used for training the network. The data for cross-validation is chosen by random number generator. Alternatively, the user may want to select the training data manually by marking rows for cross-validation. When some rows are marked (red) check *Use unmarked data* to tell the NN to use only unmarked rows for training the network. The marked rows will then be used for cross-validation.



Neural network

Task name: Sheet1

Independent variable: A, B, C, D (selected); E, F

Dependent variable: E, F, G, H (selected); I

Data: ● All ○ Marked ○ Unmarked ○ By filter

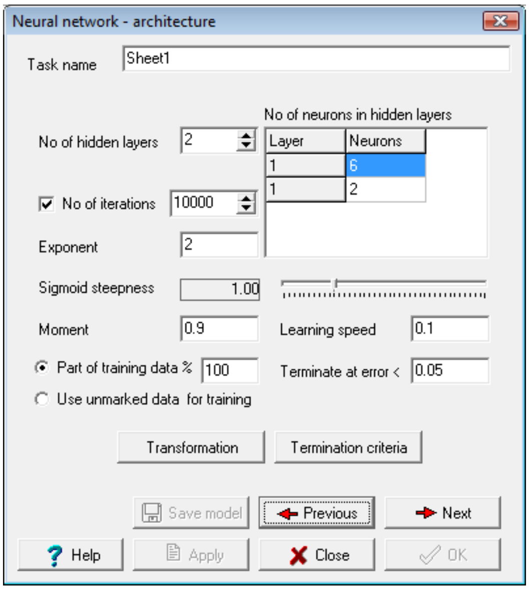
☑ X - Prediction: A, B, C, D (selected); E, F

Labels: [None]

☑ Use col names

☑ Display weights

Fig. 19 Step 1: Select predictors and responses, optionally choose the data for prediction. Check „Display weights“ do visualize importance of predictors and predictability of response



Neural network - architecture

Task name: Sheet1

No of hidden layers: 2

No of neurons in hidden layers:

Layer	Neurons
1	6
1	2

☑ No of iterations: 10000

Exponent: 2

Sigmoid steepness: 1.00

Moment: 0.9

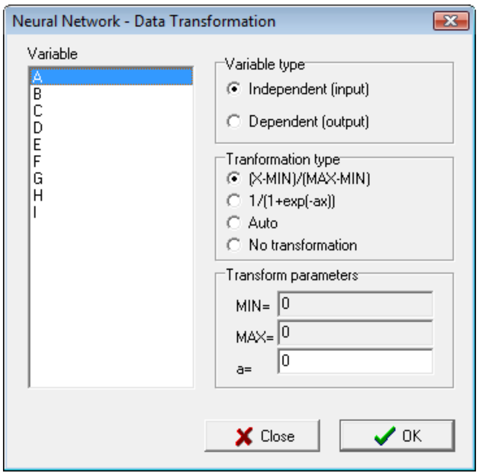
Learning speed: 0.1

● Part of training data %: 100

Terminate at error <: 0.05

○ Use unmarked data for training

Fig. 20 Step 2: Design the ANN architecture, choose number of hidden layers and number of neurons in each layer. Optionally, specify the part P of training data (rest will be used for cross-validation



Neural Network - Data Transformation

Variable: A, B, C, D, E, F, G, H, I (A selected)

Variable type
- ● Independent (input)
- ○ Dependent (output)

Tranformation type
- ● (X-MIN)/(MAX-MIN)
- ○ 1/(1+exp(-ax))
- ○ Auto
- ○ No transformation

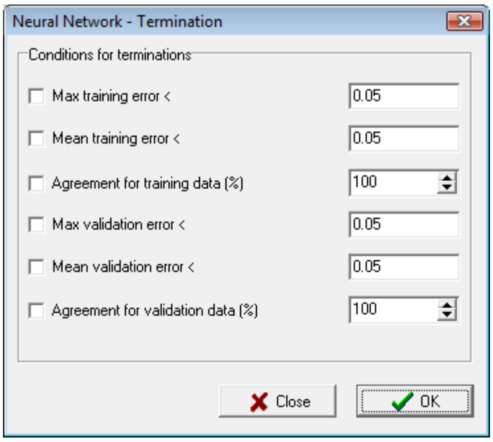
Transform parameters

MIN= 0

MAX= 0

a= 0

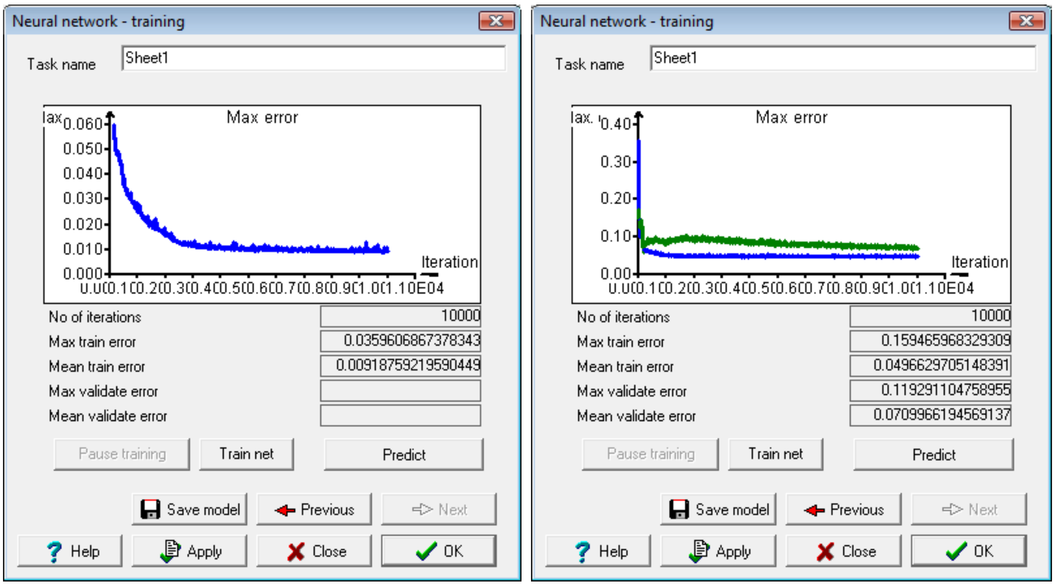
Fig. 21 Optional: Define special predictor transformation



Neural Network - Termination

Conditions for terminations
- □ Max training error < 0.05
- □ Mean training error < 0.05
- □ Agreement for training data (%) 100
- □ Max validation error < 0.05
- □ Mean validation error < 0.05
- □ Agreement for validation data (%) 100

Fig. 22 Optional: Modify the termination conditions



Neural network - training (left)

Task name: Sheet1

Max error
0.060
0.050
0.040
0.030
0.020
0.010
0.000
Iteration
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 0E04

No of iterations	10000
Max train error	0.0359606867378343
Mean train error	0.00918759219590449
Max validate error	
Mean validate error	

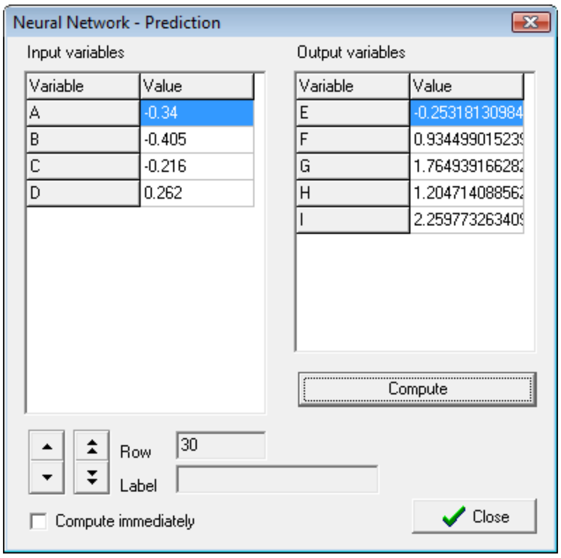
Neural network - training (right)

Task name: Sheet1

Max error
0.40
0.30
0.20
0.10
0.00
Iteration
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 0E04

No of iterations	10000
Max train error	0.159465968329309
Mean train error	0.0496629705148391
Max validate error	0.119291104758955
Mean validate error	0.0709966194569137

Fig. 23 Step 3: Run the optimization (training) process and observe how the ANN is successful. Left: no cross-validation, Right: with cross-validation. Validation data errors in green. Afterwards, you may save the model for later use, train the net again with different initial weight set, run interactive Prediction panel or press OK to get the results.



Neural Network - Prediction

Input variables

Variable	Value
A	-0.34
B	-0.405
C	-0.216
D	0.262

Output variables

Variable	Value
E	-0.25318130984
F	0.93449901523
G	1.76493916628
H	1.20471408856
I	2.25977326340

Row: 30

Label:

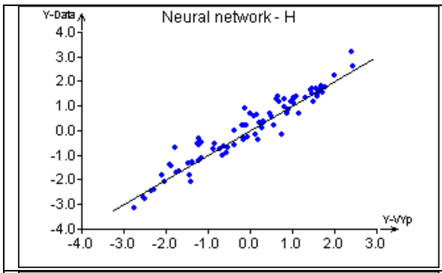
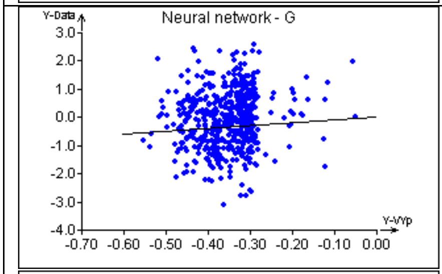
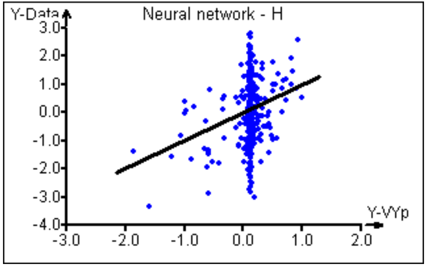
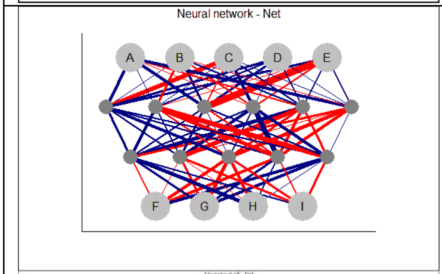
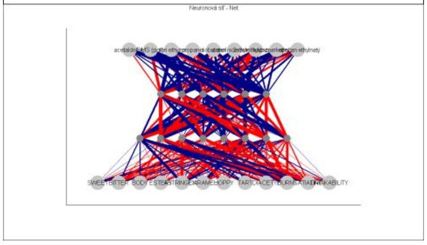
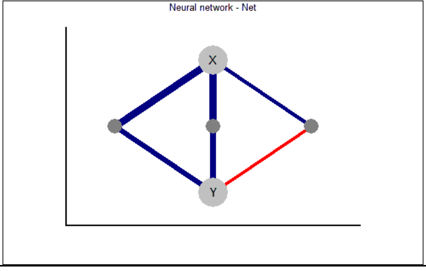
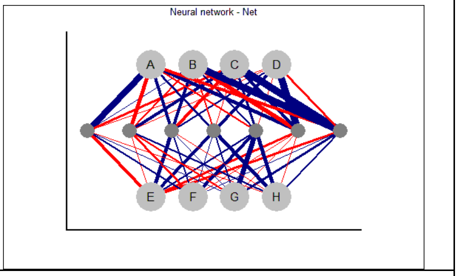
□ Compute immediately

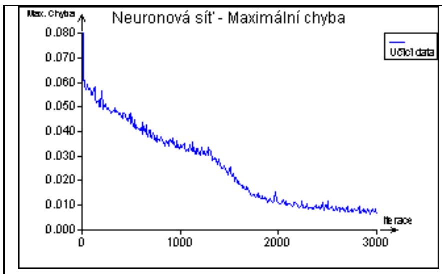
Fig. 24 Optional: Interactive prediction panel. Type new predictor values, or select a row of original data by „^“ or „v“, alter the predictor values and observe the changes in predicted response values.

Protocol

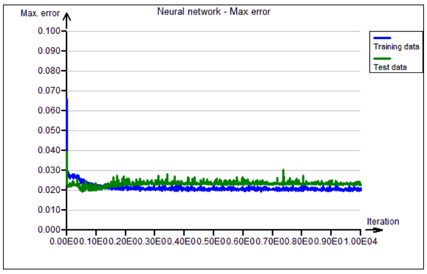
Task name	Task name
Data	Selected subset of the data
Independent variable	List of independent variables
Transformation type	Type of used transformation of the independent variables
Dependent variable	List of dependent variables
Transformation type	Type of used transformation of the dependent variables
Layer, Neurons	Number of layer and number of neurons in layers
Sigmoid steepness	User defined sigmoid steepness
Moment	User defined moment parameter for optimization process
Training speed	User defined training speed
Terminate when error <	Condition for terminating optimization process
Training data (%)	Percentage or randomly selected data for training in case of cross-validation, else 100%
Termination conditions	Condition for terminating optimization process
No of iterations	Number of optimizing iterations from dialog box
Optimization report	
No of iterations	Actual number of optimizing iterations until end or user interrupt
Max training error	Minimal max training error value reached
Mean training error	Mean training error value reached
Weights	Table of the optimized weights of the ANN
Layer / Neuron	Number or layer and number of neuron
Prediction	Table of predicted values, if chosen by the user

Graphs

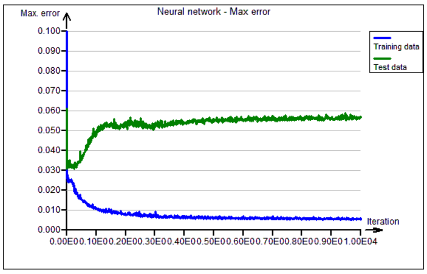
	Y-prediction plot. Plot of agreement between measured response and prediction for each of the response variables. The closer the points are scattered to the line the better the prediction of this variable. Quality of prediction usually vary from variable to variable. This plot is an overall assessment of success of the ANN model. The plot on the left shows good quality of fit.
 	If the data do not show a clear trend (like the two plots on the left) then this response variable cannot be described well with this ANN model. This can be either due to the premature termination of the optimization before reaching the optimum weights, or too simple network that is unable to identify possible more complex dependence, or sadly (and most probably) this variable simply does not depend on the selected predictors.
 	Graphical representation of the network architecture. If the checkbox “*Display weights*” was checked (see Fig. 19 on page 9) the thickness of synapses (connection lines) represent the absolute value of the corresponding weight and thus the amount of information that flows down between two neurons. From the thickness of the synapses going from the predictors we can assess their significance (the thicker lines the more significant variable). Greater weight values on the input to response nodes (thick lines going to the predictor nodes) suggest the quality of prediction of each dependent variable. Color of synapses shows only sign of the weight (red = negative weight, blue = positive weight), which is of little practical interest in complicated nets, but may be of use in simple ones. Variable nodes are labeled by the column names, if the appropriate checkbox was checked. Examples of typical architectures are shown on the left and below.  
A	Plot of the training (network optimization) process, which decrease generally the sum of squares of differences between prediction and the actual values of dependent variable, with the number of iterations on x-axis as described above. If the cross-validation is



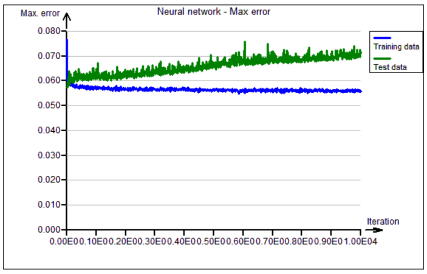
B



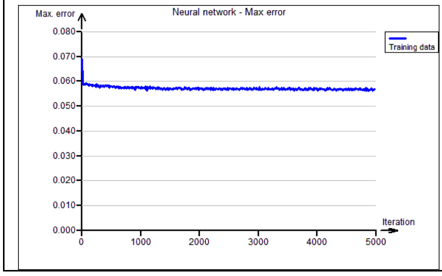
C



D



E

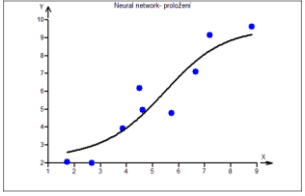


chosen the prediction error is plotted as well (green). According to the development of the maximum error of prediction curve we can assess the quality of the model and data.

Figure A This plot shows a typical successful training process, which gradually improves model for the specified data without validation.

Figure B The curve dropped steeply from 0.1 to about 0.02 which shows good quality of the model both in fitting the training data and in predicting the validation data, both curves are roughly on the same level. This is an optimal result if we intend to use ANN for predicting response from new data.

Figure C Plot with the same data as on Fig. B but with more complex ANN. The fit of the training data is considerably better than in the previous case (error about 0.006) but this is at the price of much worse prediction capability of the model (error 0.055). Such overdetermined model only fits the given data but provides poor prediction ability for any new data. It would be suitable just to interpolate existing data without cross validation. A simple example of a correct (left) and overdetermined (right) models is given below.



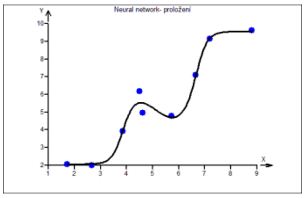


Figure D The curve shows little improvement of error (from 0.08 to 0.06). This may be due to the lack of any dependence in the data, or too simple model that is unable to explain the data, or too complex model, which failed to optimize.

Figure E A similar example shows a similar situation as Fig. D without validation.