

Neural Time Series (ANN-TS)

Menu:	QCExpert	Predictive methods	Neural Time Series
DARWin:	NNTIMELEARN		

This module (*Artificial Neural Network Time Series – ANN-TS*) uses modeling potential of a neural network to predict and forecast future values of a univariate time series. A univariate time series is a series of n measured values y_i ($i = 1, \dots, n$) assuming that –

- values y_i are measured in regular time intervals (time can be replaced by index without loss of information), or y_i describe quantitatively time-sequential units or events without leaving out any from the series.
- values y_i are preferably real-valued (real valued predictions make sense)
- values y_i are in some way related to one or more previous values y_{i-k} ,
 $y_i = G(y_{i-1}, y_{i-2}, \dots, y_{i-r}) + \varepsilon_i$, or, shortly $G(i, r) + \varepsilon_i$, where $r < n$ is called the depth of the model.

Examples of such a series can be sampled variables in technological processes, periodical outputs from stable or unstable processes, parameters of natural processes in life sciences, geo sciences, physical or chemical sciences, series of financial or economical indices, prices, rates and so on.

If the model $G()$ is found it can be used to predict $(r+1)$ -th value from any r known subsequent values, or predict yet not measured $(n+1)$ -th value y_{n+1} from y_{n-r+1}, \dots, y_n . Prediction of an (estimated) future value $\hat{y}_{n+1} = G(n+1, r)$ is sometimes called forecast. The forecasted value can be added at the end of the series and the forecast can be repeated to estimate y_{n+2} , etc. Quality (variance) of such recursive forecast may decay fast however, depending of quality of the model $G()$ and variance of errors ε_i . Refer to Monte-Carlo forecast confidence interval in module Para-Bootstrap.

In this module, a feed forward artificial neural network model with r input variables and 1 output variable is used in place of $G()$. Two types of ANN-TS are available: (a) Autoregression model where the input variables are r measured values and output is directly y_{r+1} (model AR) and (b) Differential model where the input variables are differences $(y_i - y_{i-1})$ and the output is the next difference $(y_{r+1} - y_r)$ (model DIFF). The AR models are generally more suitable for stationary series, where the future values are expected within the interval of the previous values. The DIFF models are suitable for series with trend. Further general details concerning neural networks are found in the chapter Neural networks.

Data and parameters

Data are expected in one column in the data sheet. The computational options are set in 3 subsequent dialog windows. In the first dialog box, the column of the time series variable is selected. In the Data group you can select the required data subset: all data, or just a specified subset of rows. In *Model type* select AR or DIFF model. Define the model depth r and either length of forecast or number p of validation data. If the *Validation* option is chosen the model is computed only from the first $(n - p)$ data points. The last data are used for validation of the model forecast. Validation data are p measured points at the end of the series (y_{n-p+1} to y_n) used to check forecast capability of the computed model. If the validation data (which the model does not “see”) are forecasted precisely it is supposed that the model may be good also in forecasting future data to come. In the group *Model Architecture plot*, specify plotting options for the ANN plot. If *Visualize 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). *Rotate plot* will rotate the ANN architecture plot 90 degrees.

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 of 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 *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 k* determines exponent of the criterial function, here, the default is $k=2$, which corresponds to least squares method. Exponents between 1 and 2 may somewhat robustify the network and are recommended when the data are suspicious for outliers or possible big errors in dependent variables.

$$s(NN) = \sum |y - y_{\text{pred}}|^k$$

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. Click *Next* to get to the next window.

The *Net training* window visualizes the training process by plotting the mean absolute prediction error. Press *Train net* to start or restart learning (training). During learning process press *Pause training* to temporarily stop the learning process and *Continue training* to continue. Pressing *OK* will terminate the learning process at the current stage and create protocol and graphics output.

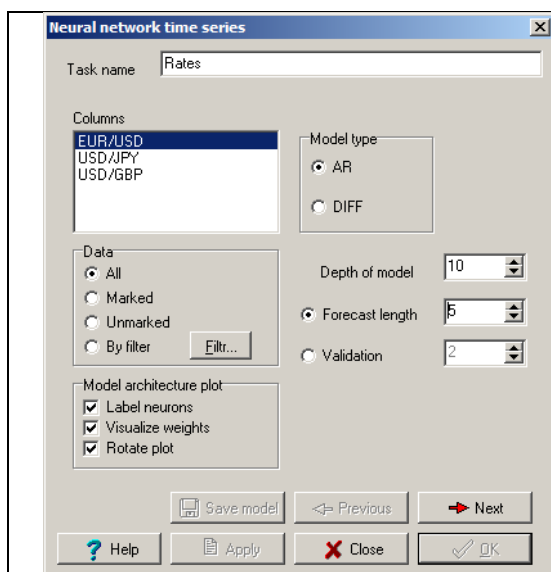


Fig. 1 Select variables and specify model

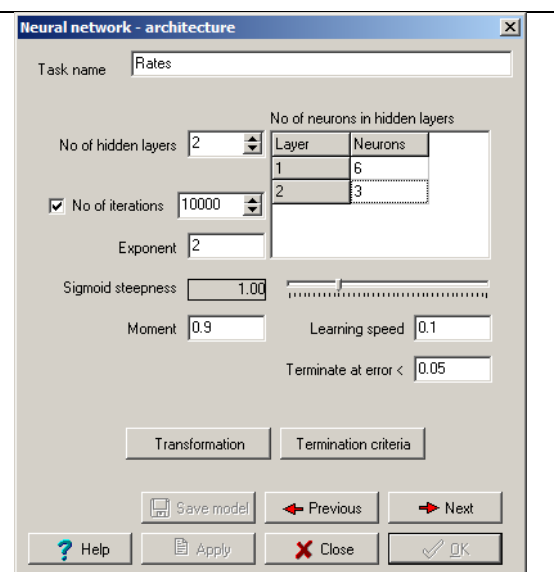


Fig. 2 Network topology and optimization strategy

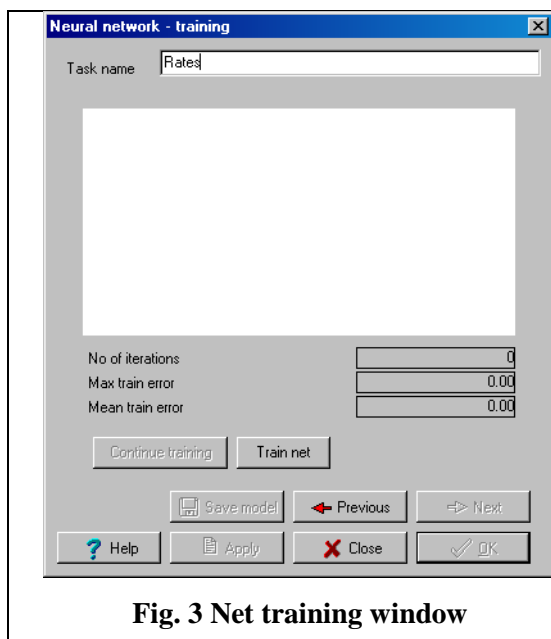


Fig. 3 Net training window

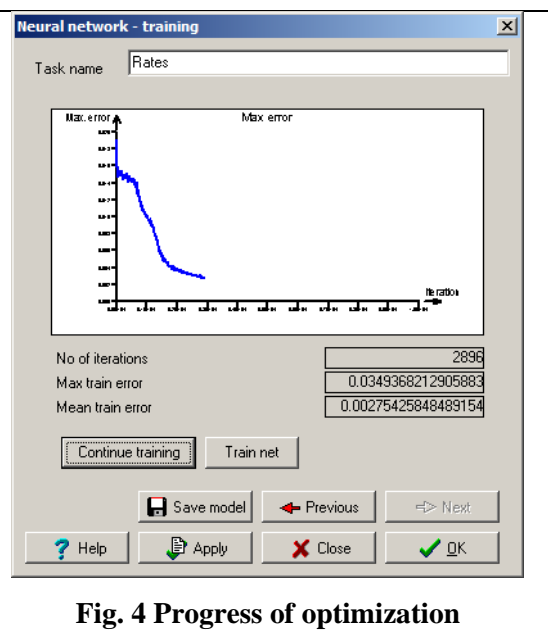


Fig. 4 Progress of optimization

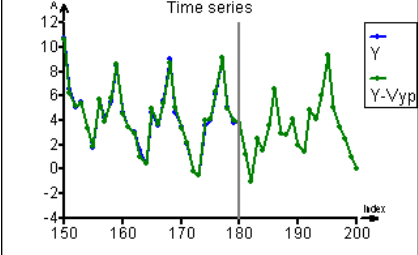
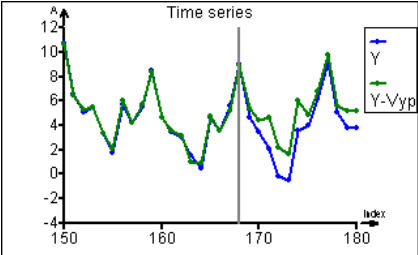
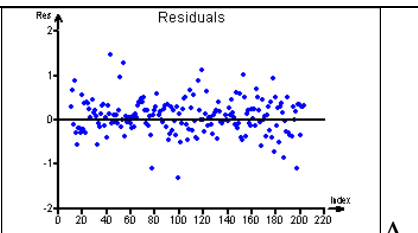
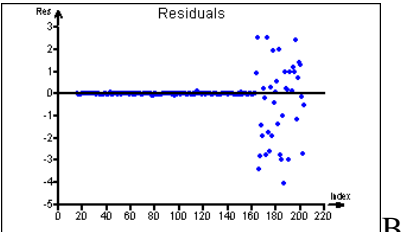
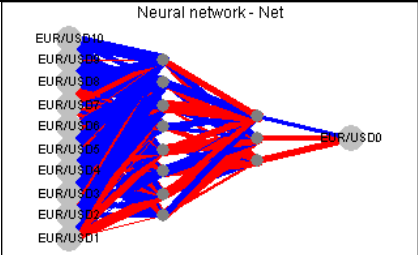
The button *Save model* will save the current model in a .QPC model file for later use in *QC.Expert™*, *DARWin*, or in the database *QCE-DataCenter®*.

Protocol

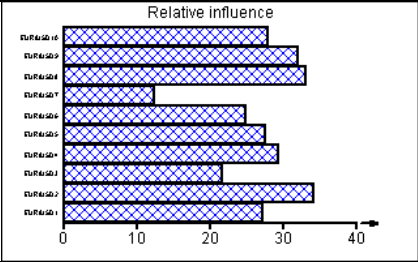
Task name	Task name from the dialog window
Data	The data subset chosen
Type of model	Selected type of model – AR or DIFF
Model depth	Number of predictors
Forecast length	Number of values to forecast
Validation	Number of last values to validate on
Independent variable	List of predictors in the time series, numbers are the backward-shift, so if the column name is X then X3 means X[i-3]
Transformation type	Type of transformation used for the predictor
Dependent variable	Column name
Transformation type	Type of transformation used for the predicted value
Layer, Neurons	Number of layer in the ANN and number of neurons in each layer
Sigmoid steepness	Selected steepness of the ANN sigmoid
Moment	Selected moment for the optimization
Training speed	Selected training speed
Terminate when error <	Terminating condition for error
Training data (%)	100% (applies only for classical multivariate ANN)
Termination conditions	Selected maximal number of iterations
Optimization report	Information about the optimization results
No of iterations	Actual number of iterations
Max training error	Actually reached maximal error of prediction
Mean training error	Actually reached mean error of prediction
Max training error :	Actually reached maximal error of validation
Mean training error :	Actually reached mean error of validation
Weights	Optimized values of the trained neural network
Layer / Neuron	Number of layer and number of neuron
Relative influence	Relative influence of X[i-j] on X[i] calculated as the sum of

	squared standardized weights in the neural network
Time series	The complete table of measured values, predicted values and residuals.

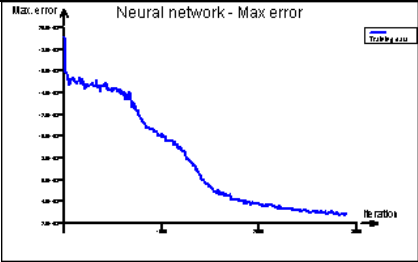
Graphs

	<p>Plot of the time series with ANN-TS prediction and validation or forecast (as selected). The measured values (blue), predicted values (green). Right of the gray vertical line is forecast (in green). If Validation was selected, the actual data are plotted along the forecast (blue). The model is based only on data left of the grey boundary line.</p>
	<p>Plot of residuals, or differences measured minus predicted including validation data, if selected. If validation residuals are much worse (bigger) than the regular residuals (B) the forecast capability is rather poor, the model is possibly overdetermined.</p>
	
	<p>Graphical representation of the network architecture. If the checkbox “<i>Display weights</i>” was checked (see Fig. 1 on page 2) 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</p>

the appropriate checkbox was checked.



Relative influence of each of the predictors on prediction computed from the absolute weights for each predictor.



Plot of the training (network optimization) process, which decrease generally the sum of squares of differences between prediction and the actual measured values, with the number of iterations on x-axis.