LABORATORY WORK №2
MULTILAYER PERCEPTRON FOR TIME SERIES FORECASTING

Objective: To study the training and functioning of a MLP for time series prediction.

Theoretical materials

The ability of neural networks (after training) to generalize and prolong the learning outcomes permits to building various predictive systems on their basis.

Let the time series $x(t)$ be given in the interval $t = \overline{1,m}$. Then the task of forecasting is to find the continuation of the time series in an unknown interval, that is, it is necessary to determine $x(m+1)$, $x(m+2)$ etc. (Fig. 1.).

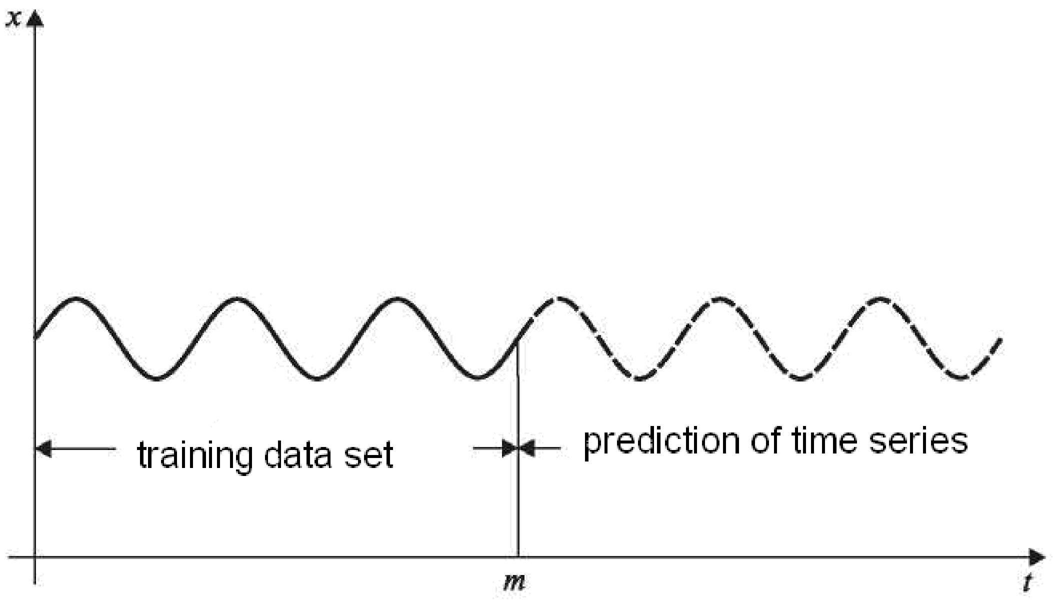

Fig. 1. Time series predicting

The "sliding window" approach is used for forecasting. It is characterized by a window length p equal to the number of elements in a time series simultaneously fed to the neural network. This determines the structure of the neural network, which consists of p input and one output neurons. The set of known values of the time series forms a training data set. The dimension of training data (number of training samples) is defined by the following way:

$$L = m - p$$

This model corresponds to the linear autoregression model and is described by the expression

$$\overline{x(t)} = \sum_{k=1}^{p} \omega_k x(t - p + k - 1) - T,$$

where $\omega_k, k=\overline{1,p}$ – weights; T- threshold; $\overline{x(t)}$ – assessment of the value of time series $x(t)$ at the time t.

The prediction error is defined as

$$e(t)=\overline{x(t)}-x(t).$$

The training patterns of a neural network can be represented as a matrix, the rows of which characterize the vectors supplied to the input of the network:

$$X=\begin{bmatrix} x(1) & x(2) & \dots & x(p) \\ x(2) & x(3) & \dots & x(p+1) \\ \dots & \dots & \dots & \dots \\ x(m-p) & x(m-p+1) & \dots & x(m-1) \end{bmatrix}.$$

This is equivalent to moving the window along a time series with step equaled one.

Thus, a set of known elements of the time series is used to train the neural network for forecasting.

Generalized delta rule

Consider a multilayer perceptron, which consists of one hidden layer with a sigmoid activation function of neurons and an output linear neuron (Fig. 2). Such a network is often used to go beyond the range of values that is limited by the non-linear activation function. Let’s derive the generalized delta rule for such kind of neural network.

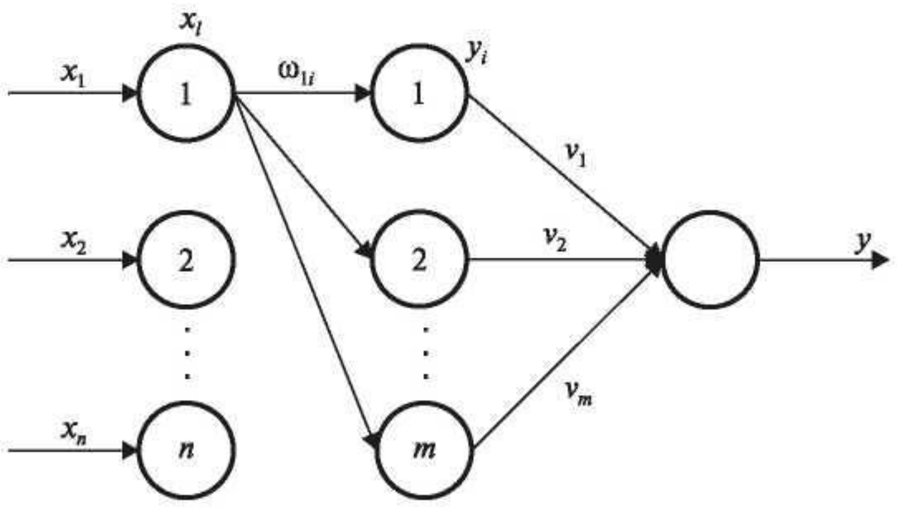

Fig. 2. MLP

The output of the neural network is calculated as:

$$y = \sum_i v_i y_i - T,$$

where v_i – i-th weight of the output unit.

The output values of the hidden neurons are defined as

$$y_i = F(S_i) = F\left(\sum_l \omega_{li} x_l - T_i\right) = \frac{1}{1+e^{-S_j}}.$$

Consider a generalized delta rule for such a network. Let's find the errors of the neurons of different layers of the neural network. So, for the output neuron

$$\gamma \;= (y - e).$$

The errors of the hidden neurons are calculated as follows:

$$\gamma_i = (y-e)v_i.$$

The learning rule for output unit is defined as

$$v_i(t+1) = v_i(t) - \alpha(y-e)y_i,$$

$$T(t+1) = T(t) + \alpha(y-e),$$

The learning rule for hidden units is defined as

$$\omega_{li}(t+1) = \omega_{lj}(t) - \alpha\gamma_i F'(S_i)x_l,$$

$$T_i(t+1) = T_i(t) + \alpha\gamma_i F'(S_i),$$

where $\gamma_i = (y-e)v_i$, $F^{'}(S_i) = y_i(1-y_i)$

Training Algorithm:

1. The weight coefficients and threshold values of the neural network are randomly initialized in a narrow range of values, for example [−1 1].

2. The learning step $\alpha(0<\alpha<1)$ and the desired mean square error E_e of the neural network are selected.

3. Data from the training set are sequentially fed to the input of the neural network, and the following actions are performed for each input training example:

a) phase of forward propagation of the input data is performed and the output values of all neural elements of the network is calculated:

$$y_i = F(S_i) = F\left(\sum_l \omega_{li} x_l - T_i\right) = \frac{1}{1+e^{-S_j}}.$$

$$y = \sum_i v_i y_i - T,$$

b) phase of back propagation of the signal is performed. As a result, the error of the neurons is determined for all layers of the network. For the output and hidden layers, the error is computed by the following way:

$$\gamma \ = y \ - e \ ,$$

$$\gamma_i = (y - e) v_i$$

c) for each layer of the neural network, the weight coefficients and thresholds of the neurons change in accordance with the generalized delta rule: The learning rule for output unit is defined as

$$v_i(t+1) = v_i(t) - \alpha(y-e)y_i,$$

$$T(t+1) = T(t) + \alpha(y-e), \qquad)$$

The learning rule for hidden units is defined as

$$\omega_{li}(t+1) = \omega_{lj}(t) - \alpha\gamma_i F'(S_i) x_l,$$

$$T_i(t+1) = T_i(t) + \alpha\gamma_i F'(S_i),$$

where $\gamma_i = (y-e)v_i$, $F^{'}(S_i) = y_i(1-y_i)$

4. Data from the training set are sequentially fed again to the input of the neural network and the total squared error of the neural network is calculated:

$$E_s = \frac{1}{2}\sum_{k=1}^{L}\sum_{j}(y_j^k - e_j^k)^2,$$

5. Repeat by going to 3.

6. Algorithm is continued until weights will cease to change or total mean square error will be less than desired error $E_s \le E_e$, or the total mean square error E_s does not decrease or decrease slightly.

Remark:

For better neural network training, the input data can be transformed into a smaller range of values.

Let the input be in the range $[x_{\min}, x_{\max}]$. It is necessary to map them into a range $[a,b]$, i.e.

$$x \in [x_{\min}, x_{\max}] \to [a,b].$$

In this case, each component of the input sample is transformed as follows:

$$\overline{x_i^k} = \frac{(x_i^k - x_{\min})(b-a)}{(x_{\max} - x_{\min})} + a.$$

For instance if $[x_{\min}, x_{\max}] \to [0,1]$ then

$$\overline{x_i^k} = \frac{x_i^k - x_{\min}}{x_{\max} - x_{\min}}.$$

TASK:

DEVELOP MLP WITH ONE HIDDEN LAYER USING ANY PROGRAMMING LANGUAGE FOR PREDICTION OF THE FOLLOWING FUNCTION

$$y = a\sin(bx) + d \quad .$$

The structure of MLP consists of sigmoid neurons in the hidden layer and one output linear unit. The appropriate number of units in the hidden layer is determined by trial and error.

Training should be performed using 30, 50 and 100 points, respectively. Testing should be performed using 20 points. X changes in increments of 0.1. The learning rate is chosen by the student independently. It is necessary also to use adaptive learning rate and to compare the results.

Report content:

1. Learning outcomes: graph of time series, a table with the following columns: reference value, real value, deviation; the graph of the error change depending on the epoch.
2. Prediction results: table with the following columns: reference value, real value, deviation.
3. Comparison of results

Variants

№ of variant	a	B	d	Number of input units
1	9	1	0.2	5
2	2	6	0.3	3
3	4	5	0.5	5
4	1	10	0.6	7
5	2	9	0.1	5
6	3	8	0.2	5
7	4	7	0.8	7
8	5	6	0.3	7
9	6	5	0.4	9
10	1	3	0.1	5
11	2	12	0	7
12	3	14	0	5
13	4	2	0.5	3
14	7	3	0.6	4
15	6	4	0.7	6
16	5	5	0.8	7

17	2	6	0.9	5