The Second International Conference "Problems of Cybernetics and Informatics"
September 10-12, 2008, Baku, Azerbaijan. Section #2 "Intellectual Systems"
www.pci2008.science.az/2/14.pdf

DECISION OF APPLIED TASKS WITH USE NEURAL NETWORKS

Sabit Kerimov[1], Elena Ragimova[2]

Azerbaijan State Oil Academy, Baku, Azerbaijan
[1]*kerimov.sabit@gmail.com*, [2]*elena1409@yahoo.com*

With use neural networks it is possible to decide such tasks as: classification of images (recognition of the letters, recognition of speech), approximation of function (at the decision of tasks of modeling), forecasting (acceptance of the decisions in business, a science, engineering), categorization (is based on similarity of images), optimization (finding of the decisions satisfying to system of restrictions), management (account of such entrance influence, at which the system goes on a desirable trajectory). Practically any of set forth above tasks can be reduced to a task solved with use neural networks [1]. And consequently always there is a question « how to build a network? ».

In a package MATLAB there are opportunities of development neural networks by a program way and with use of a program environment *NNTool*. Let's consider a program way of creation neural networks. For the decision of each of concrete tasks there are 4 basic stages of realization of the neural -network approach.

Example 1. A task of forecasting. Let there is a sequence of the prices on valuable papers. Using neural networks, it is necessary to determine with known probability, when the prices for valuable papers will grow (it is necessary them to sell) and when the prices for valuable papers will fall (it is necessary to buy).

1. Definition of inputs and target parameters of a network. Let on an input there are following meanings of argument:

P = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20];

On an output there are following meanings of the prices:

T = [56 56.5 57 59.5 61 61 62 62.5 62 61 61 61.5 64 62.5 65.5 67 67 66.5 67.5 68 70];

plot (P, T, 'o')

In result the following diagram of entrance given (fig. 1) is received.

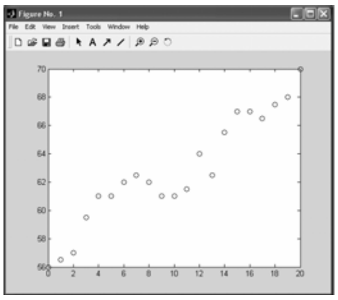


Figure 1. The diagram of the price on a paper

It is necessary to analyses an opportunity of forecasting of the further behavior of the price on valuable papers, using the mechanism neural networks.

2. Creation of a network. For creation of the unidirectional network we shall use function *newff* [2], having syntax:

net = newff (PR [, S1 S2 ... SN], {TF1 TF2 ... TFN1},
BTF1, BLF, PF)

Here:

PR- Rx2- a matrix of the minimal and maximal meanings for R of input elements;

[S1 S2 ... SN]- the size of the latent layer, for *N1* of layers ;

{TF1 TF2 ... TFN1} - function of activation of neurons of the *k*- layer, by default *'tansig'*;

BTF1- function of training of a network, by default *'traingd'*;

BLF- function of adjustment of a weights and displacement, by default *'learngdm'*;

PF - function of a mistake, by default *'mse'*.

For our task the function *NEWFF* is used to create two layers of a neural network. In the first layer of a network will settle down 4 neurons with function of activation *TANSIG*, and in the second layer there will be one neuron with function of activation *PURELIN*. So network will have 4 inputs and one output with restrictions on an input from 0 up to 20.

net = newff ([0 20], [5 1], {'tansig' purelin '});

3. Training a network. We set criterion of the ending of training - meaning of a rejection, at which the training will be considered completed:

net.trainParam.goal = 0.01;

We set a maximum quantity of cycles of training. After this quantity of cycles will be executed, the training will be completed:

net.trainParam.epochs = 50;

Also we shall train a network by algorithm of return distribution

net = train (net, P, T);

Let's receive a network with the updated weights

y = sim (net, P)

Let's construct the diagrams initial price and diagram, which is under construction on an output received neural network (fig. 2):

plot (P, T, P, y, 'o')

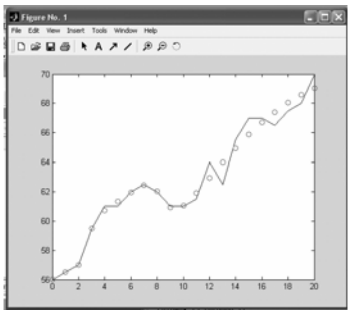


Figure 2 . The diagrams: initial price (continuous line), on an output neural network (line from symbols 'o')).

4. Testing a network. Let's set arguments, for which it is required to receive the forecast for meaning of the price, for example:

P = [45 46 47 48]

For check we shall generate a file of the correct answers:

T = [83 83 83 84]

Let's submit the given file of arguments on a network:

y = sim (net, P)

To compare results it is possible if to deduce the diagram of the answers from a network on the test task, and also correct variant.

plot (P, T, P, y, 'o')

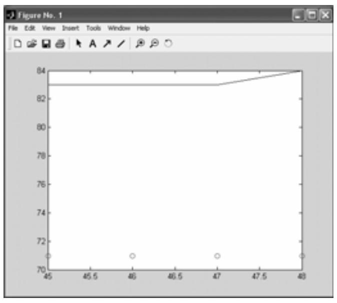
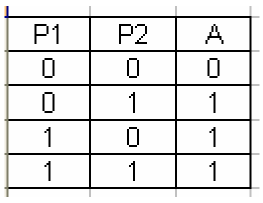


Figure 3. Testing of a network for forecasting of the price. A continuous line the forecast of a network, line from symbols 'o' - initial variant

Proceeding from the diagram in a fig. 3, it is possible to make a conclusion, that the network in this case reflects the tendency of the prices on valuable papers and for exacter account it is necessary much more data.

Example 2. A task of classification. Let's formulate in the terms of a neural network a task of performance of operation logic "OR".

1. Definition of input and target parameters of a network. Before creation of a network it is necessary to prepare a set of the training and target data. Let's make the table of the validity for logic function "OR", where P1 and P2 - inputs, and And - desirable output (tab. 1).



P1	P2	A
0	0	0
0	1	1
1	0	1
1	1	1

The table 1. The table of the validity of logic function "OR"

According to the table 1, it is possible to write, that on an input there are meanings,
P = [0 0 1 1; 0 1 0 1];
On an output there should be meanings
T = [0 1 1 1];
plot (P, T, 'o')

2. Creation of a network The following stage is the creation of a neural network. As perceptron is used for the decision of simple classification tasks, we shall choose a network consisting of one perceptron with two inputs. So at us in network 2 inputs *(P1, P2),* 1 output *(A),* 1 layer and 1 neuron in a layer. As function of creation of a new network we shall use function *newp* (creation perceptron), having the following syntax:

NET = NEWP (PR, S, TF, LF),

Here:
PR- a matrix of the size Rx2, consisting from the minimal and maximal meanings of input elements *R*;
S – quantity of neurons; *TF*- function of activation of neurons, by default *'hardlim'*; *LF* - function of adjustment of weights and displacement, by default *'learnp'*.

For our case it is possible to write:

net = newp ([0 1; -2 2], 1);

3. Training a network. Now network should be trained, since it is impossible to expect that at once after a stage of creation of a network last will carry out correctly function logic "OR". We set criterion of the ending of training:

net.trainParam.goal = 0;

We set a maximum quantity of cycles of training:
net.trainParam.epochs = 20;
Also we shall train a network by algorithm of return distribution (fig. 4):

net = train (net, P, T);

From figure 4 it is visible, that the training was stopped, when the function of the purpose has reached the established size (goal = 0). Let's receive a network with the updated weights *y = sim (net, P)*
Let's construct the diagrams the initial diagram and diagram, which is under construction on the basis of an output of a neural network.

plot (P, T, P, y, 'o')

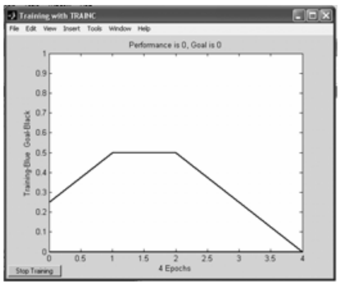


Figure 4. Curve of training logic "OR "

4. Testing a network. So, the algorithm of training has found the exact decision of a task. The correctness of work of a network can check up by run of the trained network. In the given task it is natural to use the same data set, as at training. Contents of a window of viewing coincide with meaning of a vector of the purposes - network works correctly.

So it is possible to make the following conclusion, that at a choice the neural networks for the decision of a concrete task are necessary to make a series of experiments with various networks, before will be received suitable. Thus to not be entered in error local minima of function of a mistake, it is necessary some times to train each network.

Literature

1. Demuth H., Beale M. Neural Network Toolbox. For Use with MATLAB. The MathWorks Inc. 1992-2000.
2. В.Дьяконов, В.Круглов. Математические пакеты расширения MATLAB