Agricultural data prediction by means of neural network

Jiří ŠŤASTNÝ, Vladimír KONEČNÝ, Oldřich TRENZ

Department of Computer Science, Faculty of Business and Economics, Mendel University in Brno, Brno, Czech Republic

Abstract: The contribution deals with the prediction of crop yield levels, using an artificial intelligence approach, namely a multi-layer neural network model. Subsequently, we are contrasting this approach with several non-linear regression models, the usefulness of which has been tested and published several times in the specialized periodicals. The main stress is placed on judging the accuracy of the individual methods and of the implementation. A neural network simulation device is that which enables the user to set an adequate configuration of the neural network vis á vis the required task. The conclusions can be generalized for other tasks of a similar nature, especially for the tasks of a non-linear character, where the benefits of this method increase.

Key words: neural network, multi-layer perceptron, approximation, learning, regression

Artificial neural networks are one of many computing models used in the sphere of artificial intelligence. Like their biological equivalent, they are designed for the parallel processing of data. At the same time, they can be also used for classification tasks (Konečný et al. 2010), applicable, for example in the Knowledge-management (Svoboda 2007), for predicting, or for approximation. The neural network model is composed of artificial neurons that are interconnected and, depending on the network topology; they exchange the actuation signals – and this in the form of an activation transition function. The weights in neural networks are generated so that the outputs are dependent on the actual inputs and the internal state of the network, thus generating a model for the learning patterns. This allows using the neural networks algorithms also in economic and statistical analyses. Among the better known artificial neural networks, there belongs the multi-layer neural network (Miehie et al. 1994; Lim et al. 1999), which is being applied in the mentioned prediction task.

In the article, we are analyzing the multi-layer neural network regressive model which has been used for solving the problem of the yield of onion, the type Brown Imperial Spanish, whilst it is compared with the regressive model applied in the same task. To determine the relationship between the yield of the crop and the sowing density or the plantation density, we have suggested empirical non-linear regressive models (Meloun and Militký 1996). We may therefore perform a detailed comparison of the use of individual approaches.

MATERIAL AND METHODS

Neural networks are often employed for the data prediction and regressive purposes, while we may usefully employ them even for estimating non-linear dependencies. This option is derived from the character of the network model – i.e., its ability of general approximation. An important step towards constructing the neural network model is a selection of the network type (its topology), along with the selection of a suitable activation function and a teaching algorithm (Tang and Ishizuka 1995; Škorpil and Šťastný 2007). In the model we have constructed, a multi-layer neural perceptron is used (MLP) in the configuration (1-2-1), i.e., one neuron at the input, two in the hidden layer and one at the output, along with the non-linear activation function. For the learning itself, the Back Propagation algorithm is implemented.

For judging the usefulness of the neural network model as compared with the regressive model, we are using the teaching data contained in Table 1. Theses data have been published along with the suggested non-linear regressive models that are suitable for determining the sowing density of the selected crop (Meloun and Militký 1996). The mentioned empirical non-linear regressive models are used for determin-

Supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. MSM 6215648904/03).

Table 1. Input set of the real data of agricultural companies

No.	Density of sowing (1/m^2)	Yield of onions (g/pc of production)	MLP 1-2-1	No.	Density of sowing (1/m^2)	Yield of onions (g/pc of production)	MLP 1-2-1
1	2.0640	1.7658	1.5102	22	5.9350	0.9494	1.0149
2	2.6910	1.5907	1.4304	23	5.9720	1.1928	1.0108
3	2.6910	1.2241	1.4304	24	6.3040	0.9364	0.9745
4	2.8020	1.2832	1.4159	25	6.7090	0.8573	0.9326
5	3.2440	1.2577	1.3571	26	6.8930	0.8926	0.9144
6	3.4280	1.2681	1.3324	27	6.9300	0.8855	0.9108
7	3.6490	1.1729	1.3027	28.	7.6360	0.7631	0.8462
8	3.8710	1.3349	1.2730	29	8.0730	0.7663	0.8099
9	3.9440	1.2887	1.2632	30	8.9580	0.9053	0.7445
10	3.9810	1.1004	1.2583	31	9.5470	0.7128	0.7064
11	4.0920	1.1115	1.2435	32	9.8050	0.5661	0.6909
12	4.2760	1.3412	1.2192	33	9.8420	0.7509	0.6887
13	4.3500	0.9994	1.2094	34	10.2480	0.6526	0.6660
14	4.5340	1.2870	1.1854	35	10.5800	0.6448	0.6486
15	4.5710	1.5217	1.1806	36	10.6530	0.6184	0.6449
16	4.6820	1.0036	1.1663	37	10.8750	0.6519	0.6340
17	4.7180	1.2332	1.1617	38	11.5380	0.5710	0.6038
18	4.7920	1.1444	1.1523	39	15.0770	0.5268	0.4903
19	4.8660	1.3127	1.1429	40	15.2240	0.4701	0.4869
20	5.3450	1.1512	1.0837	41	15.5190	0.4428	0.4803
21	5.5660	0.9552	1.0573				

ing the usefulness of a neural networks prediction approach, and the options of its implementation. The accuracy of these models is the comparison criterion for the neural network.

Non-linear regression

Regression analysis is a statistical approach designated to estimate the values of a stochastic quantity on the basis of knowing other quantities. The regressive task may be divided into two parts. The first part is the selection or creation of a suitable data model, one which would correspond to the character of the empirically determined numerical data. The second part of the task is finding suitable parameters for the model, in such a fashion that the resulting function best describes the concretely measured values. For solving the second part, there are suitable methods of mathematical statistics, whereas the solution of the first part of the task is usually possible only empirically.

For comparing the usefulness of the neural network regressive model, we have used the following non-linear regressive models describing the crop yield of the onion type Brown Imperial Spanish in the location of Australia. The relationships contrast the yield of the cultivated plant and the density of sowing (Meloun and Militký 1996):

Model A (Bleasdale a Nelder, 1960)

$$y = (\beta_1 + \beta_2 x)^{-\frac{1}{\beta_3}} \quad (1)$$

Model C (Farazdaghi a Harris 1968)

$$y = \frac{1}{\beta_1 + \beta_2 + \beta_3 x^2} \quad (2)$$

Model D (asymptotic, Mead 1979)

$$y = \frac{1}{\beta_1 + \beta_2 x} \quad (3)$$

The coefficients in the equations are as follows: β_1 is the level of the plant's genetic potential, not suppressed by the influence of the competing weed, and β_2 is the level of the potential of the competing weed, described in detail (Meloun and Militký 1996).

In the solutions section, the crop sowing density is determined by the multi-layer neural network regression model, and a comparison with the non-linear empirical models is performed.

Multi-layer neural network

Neural network models are essentially simple mathematical models defining a function:

$$f: X \rightarrow Y \tag{4}$$

Every type of a model created by the artificial neural network (ANN) corresponds to a class of such functions (Miehie et al. 1994; Tang and Ishizuka 1995; Lim 1999). The neural network is configured by the number of input and output neurons, by the number of neurons in the hidden layer, the number of hidden layers and the function of neurons.

One of the most widely used algorithms for training the multi-layer neural networks is the back-propagation algorithm. The back-propagation algorithm attempts to find the minimum of the error function in the weight space using methods based on the gradient descent. The combination of weight values which minimize the value of the error function is considered to be a solution to the learning problem. Because the gradient of the error function has to be computed in each iteration, it is necessary to guarantee the continuity and differentiability of the error function.

Back propagation algorithm

Back-propagation algorithm is an iterative method where the network gets from an initial non-learned state to the full learned one (Sarle 1994; Tang 1995). It is possible to describe the algorithm in the following way:

```
random initialization of weights;
repeat
    repeat
        choose_pattern_from_training_set;
        put_chosen_pattern_in_input_of_network;
        compute_outputs_of_network;
        compare_outputs_with_required_values;
        modify_weights;
    until all_patterns_from_traning_set_are_chosen;
until total_error < criterion;
```

The learning algorithm of back-propagation is essentially an optimization method that is able to find the weight coefficients and the thresholds for the given neural network and training set. The network is assumed to be made up of neurons the behaviour of which is described by the formula:

$$y = S\left(\sum_{i=1}^{N} w_i x_i + \Theta\right) \tag{5}$$

where the output nonlinear function S is defined by the formula:

$$S(\varphi) = \frac{1}{1 + e^{-\gamma\varphi}} \tag{6}$$

where γ determines the curve steepness in the origin of coordinates. Input and output values are assumed to be in the range <0, 1>. In the following formulas, the parameter o denotes the output layer, h the hidden layer, and i, j the indexes. The index i indexes output neurons and the index j their inputs. Then y_i^h means i-th neuron output of the hidden layer and w_{ij}^0 means the weight connecting i-th neuron of the output layer and j-th neuron of the previous hidden layer.

The appurtenant back-propagation algorithm can be written in the following steps:

1. Initialization. You set at random all the weights in the network at values in the recommended range <–0.3, 0.3>.
2. Pattern submitting. You choose a pattern from the training set and put it in the network inputs. Then you compute the outputs of particular neurons by the relations (5) and (6).
3. Comparison. First you compute the neural network energy – Sum of Squared error (SSE) under the relation (7).

$$E = \frac{1}{2}\sum_{i=1}^{n} (y_i - d_i)^2 \tag{7}$$

where n is the number of network outputs, y_i is the i-th output and d_i is the i-th output of a learning pattern.

Then you compute an error for the output layer by the relation:

$$\delta_i^o = (d_i - y_i^o) y_i^o \gamma (1 - y_i^o) \tag{8}$$

4. Back-propagation of an error and weight modification. You compute for all neurons in the layer:

$$\Delta w_{ij}^l(t) = \eta \delta_i^l(t) y_j^{l-1}(t) + \alpha \Delta w_{ij}^l(t-1) \tag{9}$$

$$\Delta\Theta_i^l(t) = \eta \delta_i^l(t) + \alpha \Delta\Theta_i^l(t-1) \tag{10}$$

By the relation:

$$\delta_i^{h-1} = y_i^{h-1}(1 - y_i^{h-1}) \sum_{k=1}^{n} w_{ki}^h \delta_k^h \tag{11}$$

you back-propagate an error in the layer nearer the inputs. Then you modify the weights:

$$w_{ij}^l(t+1) = w_{ij}^l(t) + \Delta w_{ij}^l(t) \tag{12}$$

$$\Theta_i^l(t+1) = \Theta_i^l(t) + \Delta\Theta_i^l(t) \tag{13}$$

In the previous relations, the upper index l equals zero for the output layer and the value is h for the hidden layer. The coefficient δ describes the speed of learning and the coefficient α is inertia, which describes the time of the given direction before the gradient changes. The Back Propagation Algorithm is described in more detail e.g. in Škorpil and Šťastný (2008).

You then apply the step number 4 to all the layers of the network. You start with the output layer followed by the hidden layers.

5. Termination of the pattern selection from the training set. If you have submitted all patterns from the training set to the network, then continue with the step number 6, else you go back to the step number 2.
6. Termination of the learning process. If the neural network energy in the last computation has been less than the criterion selected, then terminate the learning process, else you continue with the step number 2.

RESULTS AND DISCUSSION

The task of the search for a crop yield regressive model of the Imperial Brown onion was performed by the means of a neural network. The input data was the number set (x, y), where x is the density of the nurslings per m^2 (nursling/m^2) and y is the average onion yield (g/nursling). The input values x, y are listed in Table 1. The result that has been reached has been compared with the existing empirical non-linear regressive models (1), (2) and (3) using the residual sum of squares method (RSS).

The non-linear RSS

The residual sum of squares method is among the most widely used methods for the approximation of an unknown function $f(x)$ by the means of functions $R_m(x, c_0, c_1, \ldots, c_m)$ of a known form, where $c_0, c_1, \ldots, c_m$ are the unknown coefficients. Experimentally, we obtain the values $y_i \approx f(x_i)$, $i = 0, 1, \ldots, n$, therefore also $y_i \approx R_m(x, c_0, c_1, \ldots, c_m)$. Here at holds that $n \geq m$. The task is to determine the optimum values of the coefficients $c_0, c_1, \ldots, c_m$.

For solving the optimization, we can use the modified Newton method (Nocedal and Wright 1999), the algorithm of which can be described by the steps (relations for the direction vector p_k, approximation x_k, the length of the step α_k, and they are listed in (Nocedal and Wright 1999):

$k = 0$

selection of the start approximation x_0

selection of the tolerance $\varepsilon > 0$

Repeat

Calculating of p_k from the $J_k p_k = r_k$ equations by the RSS Metod

$j = 0$

Repeat

$\alpha_k = 2^{-j}$

$j = j + 1$

Until $\phi(\alpha_k) < \phi(0)$

$x_{k+1} := x_k + \alpha_k p_k$

$k = k + 1$

Until $\alpha_k ||p_k|| < \varepsilon$

$x_{\min} \approx x_k + \alpha_k p_k$

It is possible to prove that for linear cases, it holds that $x_1 = x_0 + p_0 = x_{\min}$

Neural network model description

The created software enables us to simulate various multi-layer neural networks model configurations, to perform the network learning, to perform teaching and decision-making, i.e., in the given case, the prediction of sowing values of the selected crop.

In the mentioned software environment, we may set and interactively change the network model configuration, and this on the level of changing the number of layers, the number of neurons in each layer, modifying the transit function and changing the learning coefficients 1 and 2 of the Epsilon 1, 2 and 3 phase. For the neural network learning itself, we primarily use the Back Propagation algorithm.

The input data are read in the input text area in the form of the vector 'teaching', one that contains the input and output value for the learning algorithm (learning with the teacher). As we have mentioned, the input value is the nursling planting density per sqare meter, and the output value is the average onion yield in grams per unit of production (nursling). The character of the input vector is therefore as follows (Input area):

vyuka([*INPUT*], [*OUTPUT*])

Every line designates one input record of the learning set. The output of the neural network after learning or testing is marked in the Output Area in Figure 1. Here the input, the required and the actually computed values are shown in the form of output vectors.

Model assessment

In Figure 2, the dependency of the onion yield on the sowing density is shown. The published input data

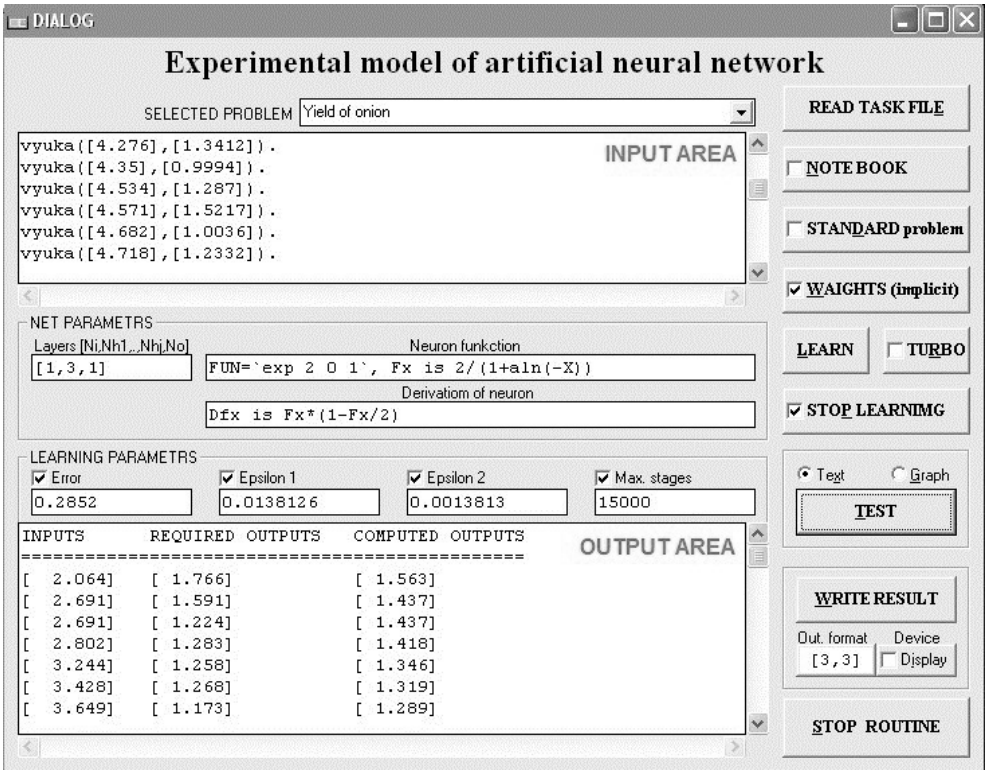

Figure 1. Introduction screen of the neural network simulator

(Meloun and Militký 1996) have been obtained by the means of agricultural practice. The predicted values obtained by using the multi-layer neural network in the given configuration are added to the chart. If we analyze the output we have reached in more detail, the network model that has been composed in the configuration (1-3-1) is, vis á vis the greater number of neurons, more exact.

If we perform a comparison of the empirical non-linear regressive model and the regression by the means of a neural network, we obtain the following values listed in Table 2.

Two configurations of the multi-layer neural network were used for the solution itself, namely (1-2-1) and (1-3-1), (1-4-1), i.e., the number of neurons in the input, the middle and output layers.

The given values show that the neural prediction model in the configuration (1-3-1) is more accurate than the described and published non-linear regressive models. A further increase in the number of neurons in the middle layer of the model has not brought a marked improvement of the attained value.

Comparing the model accuracy has been performed using the RSS. When implementing neural networks,

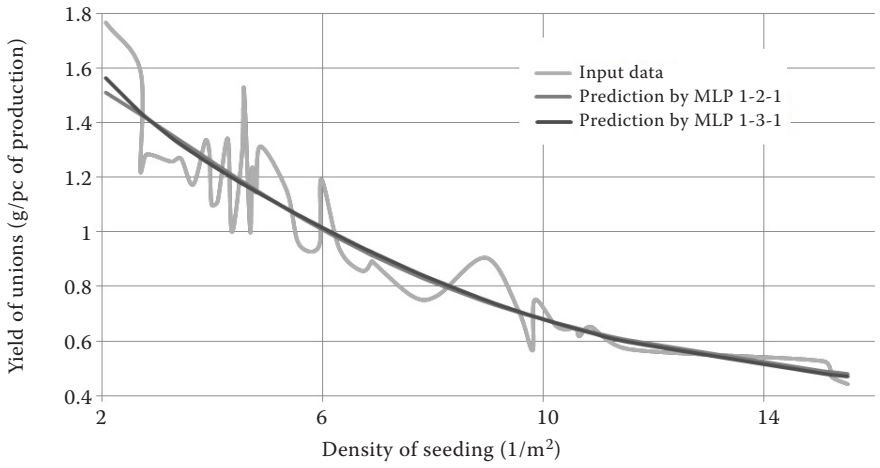

Figure 2. Output of the dependency of input data and the neural network prediction

Table 2. Comparison of the output values attained via the individual approaches

	Non-linear regression RSS	MLP (1-2-1)	MLP (1-3-1)	MLP (1-4-1)
Model A	5 896			
Model C	5 917	5 946.03	5 700.98	5 700.98
Model D	6 051			

a high model complexity is not suitable, vis á vis the practical usefulness and the length of the network learning. The models that have been chosen and that have the required accuracy and, simultaneously, have a sensibly complex transcription, are listed in Table 2.

CONCLUSION

The contribution deals with the implementation of the multi-layer neural network for the prediction of the crop yield, and the comparison of the accuracy of this approach with the accuracy of the well-known regression model designed for the prediction of empirical data. It is suitable to apply the methods from the field of artificial intelligence (e.g., neural networks) in the case of tasks that are not easily resolved using the classical deterministic or perhaps statistical methods. Among these are, in particular, non-linear regression, prediction and classification. In the present contribution, the implementation of the created neural network in a concrete case is described, whilst a neural network simulator has been used for the resolution of the task. The use of a multi-layer neural network has proved to be more accurate in the case of the given task than the published regressive model.

This research was supported by the grants MSM 6215648904/03 Research design of the Mendel University of Agriculture and Forestry in Brno.

REFERENCES

Nocedal J., Wright S.J. (1999): Numerical Optimization. Springer Series in Operations Research. Springer, Berlin.

Konečný V., Trenz O., Svobodová E. (2010): Classification of companies with assistance of self-learning neural networks. Agricultural Economics – Czech, *56*: 51–58.

Lim T., Loh W.Y., Snih Y. (1999): A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms. Available at http://www.stat.wisc.edu/~limt/mach1317.pdf

Meloun M., Militký J. (1996): Statistical processing of experimental data (in Czech). Univerzita Pardubice, Czech Republic; ISBN 80-7194-075-5.

Miehie D., Spiegelhalter D.J., Taylor C.C. (1994): Machine Learning, Neural and Statistical Classification. Ellis Horwood, NY.

Sarle W.S. (1994): Neural Networks and Statistical Models. Proceedings of the 19th Annual SAS Users Group International Conference, SAS Institute, Cary, NC, pp. 1538–1550.

Svoboda E. (2007): Knowledge-management in managerial work of business management. Agricultural Economics – Czech, *53*: 298–303.

Škorpil V., Šťastný J. (2007): Genetic Algorithm and Neural Network. In: Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications. Vouliagmeni, Greece, pp. 347–351; ISSN 1790-5117, ISBN 978-960-8457-96-6.

Škorpil V., Šťastný J. (2008): Comparison of Learning Algorithms. In: 24th Biennial Symposium on Communications. Kingston, Canada, pp. 231–234; ISBN 978-1-4244-1945-6.

Tang Z., Ishizuka O. (1995): Design and Implementations of a Learning T-Model Neural Network. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E78-A: 259–263.

Arrived on 1st June 2011

Contact address:

Jiří Šťastný, Vladimír Konečný, Oldřich Trenz, Mendel University of Brno, Zemědělská 1, 613 00 Brno, Czech Republic
e-mail: jiri.stastny@mendelu.cz, konecny@mendelu.cz, oldrich.trenz@mendelu.cz