Ivana KLAČKOVÁ[1], Jaromír KLARÁK[2], Ivan KURIC[3]

ZASTOSOWANIE METOD 'DEEP LEARNING' W OPROGRAMOWANIU MATLAB

Streszczenie: Głębokie uczenie (ang. deep learning) jest podkategorią uczenia maszynowego, które można w praktyce przemysłowej wykorzystać do kontroli maszyn i urządzeń lub innych obszarów produkcji. Istnieje możliwość wspomagania prac lub zastąpienia pracownika, którego działania nie mogą zapewnić stałej niezawodności i wydajności. Drugą zaletą jest redukcja kosztów, a tym samym wzrost konkurencyjności cenowej. Ze względu na te atrybuty nacisk kładziony jest na rozszerzenie możliwości zastosowania metod w praktyce. Podstawą głębokiego uczenia się są sieci neuronowe, które na podstawie dostarczonych informacji wejściowych oraz określeniu współczynników wag mogą symulować procesy produkcyjne.

Słowa kluczowe: sieci neuronowe, głębokie uczenie się, neuron, splotowa sieć neuronowa

UTILIZATION OF DEEP LEARNING METHODS IN MATLAB

Summary: Deep learning is mainly used in industry, where can provide inspection of devices, products, or other aspects of interest. In this way, the person is replaced or supplemented, and his abilities in a productive and non-productive environment. This is due to the fact that man can not ensure constant reliability and productivity. A secondary advantage is cost reduction and in this way increased price competitiveness. Because of these attributes, emphasis is placed on expanding the possibilities and applicability of the methods in practice. The core of deep learning is neural networks that provide for the assignment of specific information. Based on this creation and weight allocation, so-called learning is based on input information.

Keywords: neural networks, deep learning, neuron, convolutional neural network

1. Introduction

In-depth learning methods work primarily on the principle of neural networks. The design of the functioning of these networks is based on the way in which the neurons forming the nervous system contained in each animal also work. The basic element of the nervous system is the neuron. The role of the neuron is to receive the nerve

[1] Ing., PhD., University of Žilina, Faculty of Mechanical Engineering, Department of Automation and Production systems, researcher: ivana.klackova@fstroj.uniza.sk
[2] Ing., University of Žilina, Faculty of Mechanical Engineering, Department of Automation and Production systems, PhD. student: jaromir.klarak@fstroj.uniza.sk
[3] prof. Dr. Ing., University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science: kuric.ivan@gmail.com

impulses, process them and send them in the form of electrical impulses to the neural environment. Thus, information is disseminated by synapses to other neurons and to further processing. So, with some complexity, it is possible to create a system that performs both elementary and complex tasks. A generalization of the basic principle of the neuron can be expressed as the reception of input information, processed and forwarded.

2. The basic element of artificial neural network

As already mentioned, the basic element of the artificial neural network is perceptron (neuron). In terms of the functionality of this element, three basic activities are performed here, namely input of information, processing and output of information, where the information is taken from several similar elements and the transmitted information is also sent to several subjects.

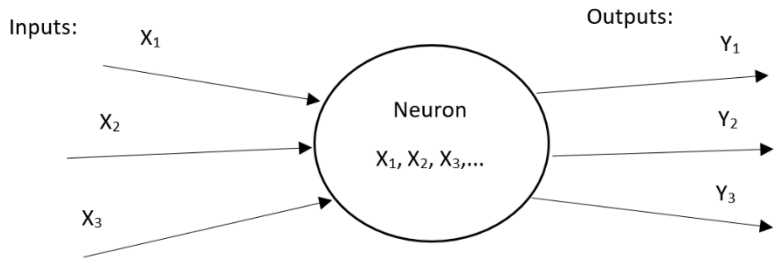

Figure 1. Representation of perceptron

In principle, this processing is described as collecting inputs and transforming this information into output. As shown in Fig. 1 an illustration of the functioning of perceptrone. As a result, this may be written as follows:

$$(X1 + X2 + X2 + Xn) \rightarrow \text{processing} \rightarrow (Y1 + Y2 + Y2 + Yn) \quad (1)$$

3. The principles of learning neural networks

In the case of the use of neurons in the learning process, the weight parameters, also referred to as **ω**, enter here. Weighing parameters are related to each input. This parameter determines the significance of each single element. For the learning process, neurons are grouped into a layer, which then connect to each other. As information passes through this system, weights are assigned to individual inputs, and at a certain stage the value of a particular weight value is specified. The goal is to assign a desired weight value that corresponds to the nature of the input information with respect to the output nature of the information. This determines the so-called value of the significance of a single parameter (Fig. 2) against input and output (Ronald J. Williams, 2008) [5].

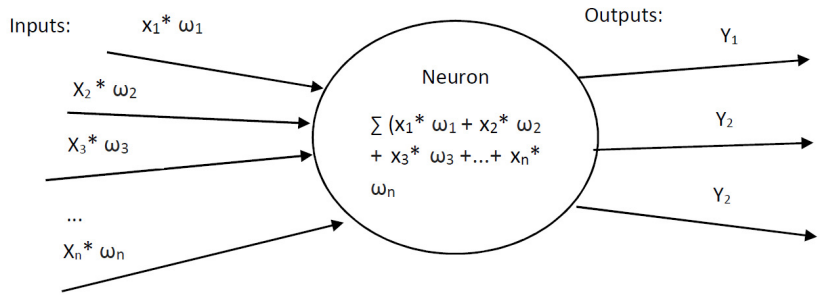

Figure 2. The principle of neuron functioning on input with weight values

4. Assigning values for scales

Values for weights are determined in different ways. In most cases, the value of weights is in the range of -1 to 1. The course of the function can be selected according to the difficulty of the task and the applicability of the given method to solve the task. An example is the use of activation functions such as the hyperbolic (tanh) tangent or the sigmoid function. The sigmoid function is described as:

$$f(x) = \frac{1}{1+e^{-x}} \tag{2}$$

and the tanh function as:

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \tag{3}$$

The most widely used activation function in deep learning methods is the rectified linear unit also known as ReLU (Fig. 3). It offers increased performance and generalization in the learning process compared to the previous two functions. The principle of the function is based on two basic factors, and that is the assignment of weight values according to whether the values are negative or positive. In the case of negative values, a zero value is assigned. In case of positive values, these values remain original after the activation function. So the function of the function in positive values is linear. (X. Jin, 2015) [7]

$$f(x) \quad = \begin{Bmatrix} X_i \;\; ak \;\; X_i \;\geq 0 \\ 0 \;\; ak \;\; X_i \;< 0 \end{Bmatrix} \tag{4}$$

A limitation in some cases may be if certain information gets low negative values, for example in values between (- 0.01, 0). In the normal activation function, they acquire zero weights. For other layers, this information may still be essential, but this degrades its materiality to zero. As a result, this information from the learning process is suppressed.

To avoid the adverse effect, similar ReLU-based activation functions have been proposed, such as LReLU, PreLU.

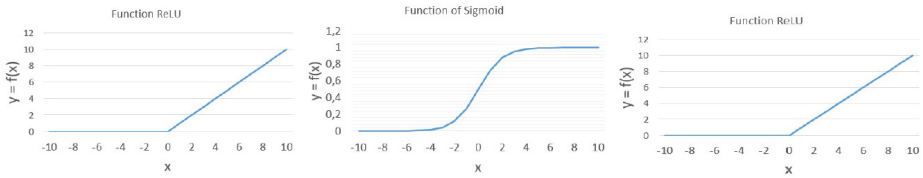

Figure 3. Examples of activation functions

5. Principle of neural network

As mentioned, perceptrons are grouped into layers, whereupon these layers are joined to other layers to form a neural network (Fig. 4). In terms of functionality, the layers are divided into 3 basic types, namely:

- Input layer
- Hidden layers
- Output layer.

The input layer represents the input parameters based on which the given task is checked. In the case of visual processing, this is the input image format. In other cases, these are parameters based on which the classification and the resulting evaluation according to the selected criteria take place. Hidden layers analyze relationships between individual attributes. The last part of the network is generally output. At this stage, the input-based information is assigned to the predefined types. Weights are also counted back in the learning process. This is based on what values are attributed to individual information. Subsequently, they are compared with setpoints to generate the optimal weight value for the information.

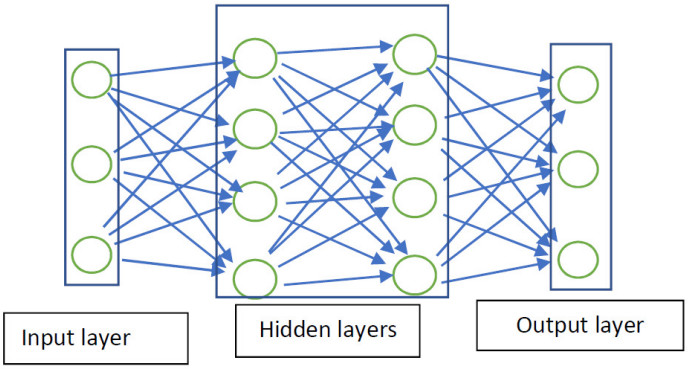

Figure 4. Neural network scheme

6. Convolutional neural network

A convolutional neural network is a system designed primarily to process visual 2D content. Generally, this type of network is known as CNN from Convolutional neural

network. The convolution is based on the mathematical principle of convolution. In neural networks, it consists in using matrices that serve as filters. Based on the given filters, elements such as characteristic feature boundaries and the like are searched for, and as a result the relationships between the properties are searched.

7. Visual input for image recognition

The input data for in-depth learning is in 2D image format. Images consist of pixels. In the digital world, everything that is visible is represented in pixels. Pixels store the color information of the pixel. The standard pixel coding is 8 bits, representing 256 options for a black and white image. The numeric expression is 0 for black and 255 for white. A color image consists of colors that are produced by a combination of three primary colors, red, green and blue (RGB - RED, GREEN, BLUE). Each color is coded with 8 bits, resulting in 2^{24} color options. In integer, it is 16777216. At the input, the images are transformed into a spatial matrix of [y x 3] shape for a color image, and for a black and white image, it is a [y x 1] matrix, ie a single-layer matrix.

8. Use of convolutional neural networks in Matlab environment

To demonstrate the usability of Matlab, a system was created to train on the basis of the input material. After training, it was validated by comparing the output in the form of a test subject classification with a real classification competence. Subsequently, the results were summarized in accuracy.

8.1 Experiment design:

The design of the experiment consists in the first step of creating a database of 928 pieces. The classification categories are 10, which corresponds to one category for each digit. Each category will contain 100 pieces of images. Subsequently, a layer structure is created, with an input layer with parameters [227 227 3] corresponding to a 227 x 227 pixel input image size for the color image. Combination of convolution layers, maxpooling layers and ReLU layers will follow. The output will be a classification and softmax layer where the test images should be classified. The goal of the experiment is to verify that it is possible to achieve a correct classification of test images with more than 98% accuracy. The number of test objects contains 232 pieces.

8.2 Experiment progress:

The first step is to load the input data as a database. This database will be divided into two parts for the purpose of training and testing at a ratio of 80:20 at random. In the next step, these layers are created:

- imageInputLayer([227 227 3]) ;
- maxPooling2dLayer(2,'Stride',[2 2]);
- convolution2dLayer(2,1);
- reluLayer();

- maxPooling2dLayer(2,'Stride',[1 1]);
- convolution2dLayer(3,2);
- reluLayer();
- fullyConnectedLayer(10) ;
- softmaxLayer() ;
- classificationLayer().

After the layers are declared, the training settings are created. After all the conditions have been prepared, the actual learning takes place. After training, the trained network is tested on a test number database created by randomly dividing the number database. The following is a comparison and summary of the classified and correct image labels. From the given summary, it is possible to determine to what extent the network is correct in classifying images of the type on which it was trained.

9. Performing the experiment

First, a database was created. The database has been divided into subfolders labeled by content. Based on the name of the folder, their affiliation, the so-called "Label", was determined. Using the function, the entire database, which consisted of 1160 objects, was split. A random 4: 1 split resulted in the creation of 2 databases, the more extensive being used for training and the second for testing. An example of database elements can be seen in Fig. 5.

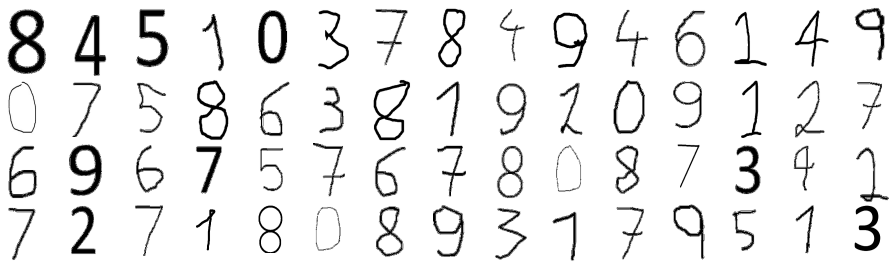

Figure 5. Some elements from the database used

This was followed by the creation of learning parameters as mentioned during the experiment. Parameters for learning were declared in the settings. All conditions were followed by learning using the Matlab function. The course of network training can be seen in Fig. 6. The blue line shows the correct learning. orange error in learning process.

The accuracy of the trained neural network was calculated according to the formula:

$$Accuracy\ of\ trained\ network = \frac{\sum legally\ classified\ object}{number\ of\ all\ objects\ tested}$$

After the quantification we got the accuracy of the trained network 99,14%.

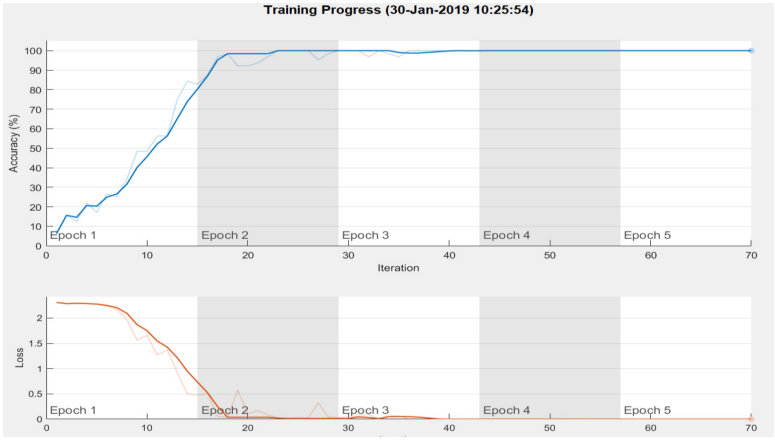

Figure 6. Course of neural network training

The illustration can be seen in Fig. 7, where there are two types of numbers. The small numbers above the images of the tested numbers are classified categories for each digit.

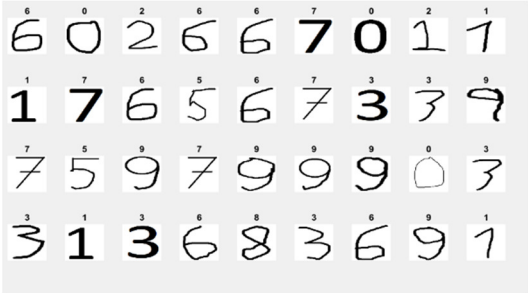

Figure 7. Representation of some elements with assigned classification categories

10. Conclusion

In this experiment, a network was designed and trained. The aim was to achieve a greater than 98% accuracy in the classification of test subjects. After quantification, an accuracy of 99.14% was achieved. In this respect, the objective has been met. It has been confirmed that even a simple network can achieve the correctness of classification with an accuracy of more than 98%. Several circumstances have appeared in the process of conducting the experiment. For learning, it is advisable to use a higher number of training elements. In this case it was only 928 pictures. As can be seen in the test object images, it is evident that it was trained on objects with strong contrast and clear boundaries.

In the case of using training material with less pronounced elements, it would be appropriate to use a higher number of objects to train such a network. In comparison of the demonstrated network and the proposed network, better ability to classify objects with less significant parameters is expected. It is therefore proposed to create

another experiment using training material with less pronounced boundaries and a higher number of different interferences.

Acknowledgement

This article was made under the support of APVV project – APVV-16-0283. Project title: Research and development of multi-criteria diagnosis of production machinery and equipment based on the implementation of artificial intelligence methods.

REFERENCES

1. BANERJEE M., MITRA S.PAL S.K.: Rough fuzzy MLP: Knowledge encoding and classification. IEEE TRANSACTIONS ON NEURAL NETWORKS 1998 Nov., 1203-1216.
2. Create Simple Deep Learning Network for Classification. (n.d.). Retrieved from *https://uk.mathworks.com: https://uk.mathworks.com/help/deeplearning/examples/create-simple-deep-learning-network-for-classification.html.*
3. NWANKPA Ch., IJOMAH W., GACHAGAN A., MARSHALL S.: Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. Retrieved from arXiv:1811.03378v1: *https://arxiv.org/pdf/1811.03.*
4. KRIZHEVSKY A., SUTSKEVER I., & HINTON G. E.: Imagenet Classification with Deep Convolutional Neural Networks. ADV NEURAL INFORM PR 2012, 1097-1105.
5. WILLIAMS R. J., D. Z.: A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation 2008, 270-280.
6. SIMARD P. Y.: Document Analysis and Recognition. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, (p. 958). Edinburgh 2003.
7. JIN X., C. X.: Deep Learning with S-shaped Rectified Linear Activation *Units.* Retrieved from *https://arxiv.org/pdf/1512.07030.pdf*
8. SÁGOVÁ, Z., LOZKIN, A., KLAČKOVÁ, I.: Calculate the trajectories of mechatronic systems and CAD/CAM. In: 10th International Technical Conference, Technological Forum 2019, 18. – 20.6.2019, ISBN 978-80-87583-30-2.
9. LENHARD, R., KADUCHOVÁ, K., PUCHOR, T., KLAČKOVÁ, I.: Simulácia distribúcie tepla v elektrotechnickej skrini pomocou CFD. In Sborník konference: TechSoft Engineering ANSYS 2019, Setkání užívatelů a konference. 25. ročník konference, 29. – 31. máj 2019, hotel Kurdějov, ISBN 978-80-907196-1-3.
10. ZAJACKO I., GAL T., SAGOVA Z., MATEICHYK V., WIECEK D.: Application of artificial intelligence principles in mechanical engineering -, MATEC Web. conference v. 244, 2018, Innovative Technologies in Engineering Production (ITEP'18), Article number: UNSP 01027.
11. TLACH, V., CISAR, M., KURIC, I., ZAJACKO, I.: Determination of the Industrial Robot Positioning Performance. In Modern Technologies in Manufacturing (MTEM 2017 – AMATUC), Cluj Napoca, 12.10 – 13.10.2017, Article number: UNSP 01004.