Application of Neural Networks for Object Recognition in Video Surveillance Systems of Industrial IoT

Andrew M. Tolkachev[1], Natalia V. Toutova
Department of Intelligent Systems in Management and Automation
Moscow Technical University of Communication and Informatics
Moscow, Russia
[1] dezirtir96@gmail.com, [2]e-natasha@mail.ru

Abstract. **In this paper the use of neural networks for object recognition in video surveillance systems of the industrial Internet of things is discussed, as well as a comparison of two types of neural networks: a perceptron and a convolutional neural network with regard to the problem of automatic recognition of the queue of people in a supermarket or a bank. The principle of operation of neural networks, their structure and realization are described. Two types of neural network are compared. It is shown that the convolutional neural network surpasses the perceptron in recognition accuracy and it is expedient to use it in the automatic queue recognition system.**

Key words: **neural network, recognition, live queue, perceptron, video surveillance systems.**

I. INTRODUCTION

People recognition is the key component of video surveillance systems, which are widely used for security purposes. Another application is a queue control systems in banks and supermarkets to control and improve quality of service as well as automates goods delivery points.

It is designed using industrial Internet of things (IIoT) concept. Among its components are cameras along with microcomputers, which are running recognition program. They analyze video stream and count the length of the queue in the bank or the supermarket. Then it periodically sends data to the cloud database. The data include time, the queue length and proving pictures. Then the bank or supermarket manager can analyze the data offline and make decisions to improve quality of service.

The recognition is very important part of such systems. Mot recognition algorithms are based on neural networks. Neural networks are models of biological neural networks of the brain, in which neurons are simulated with relatively simple, often identical, elements (artificial neurons) [1; 2].

They are also used in automation of processes of pattern recognition, adaptive control, functional approximation, prediction, the creation of expert systems, the organization of associative memory and many other applications. A wide range of tasks solved by neural networks does not allow to create universal, powerful networks at the present time, forcing to develop specialized networks functioning according to various algorithms.

For years, long queues have become a very serious problem in the life of society. Every person a day at least once meets this problem: in a supermarket, a snack bar, a ticket office. The constant standing in the queues takes away a huge amount of precious time, which is constantly lacking. Recognizing the live queue in the supermarket will help automate the goods delivery points. Automation of points of delivery of goods is a current direction [4].

To develop such a system, it is necessary to choose the structure of a neural network. In this paper, two types of neural networks are compared: a perceptron and a convolutional neural network as applied to the problem of automatic recognition of the queue of people in a supermarket or a bank.

II. MODELS OF THE NEURAL NETWORK

The perceptron is one of the first models of neural networks. The perceptron consists of three types of elements, namely: the signals coming from the sensors are transmitted to the associative elements, and then to the reacting elements. Thus, perceptrons allow us to create a set of “associations” between input stimuli and the necessary reaction at the output. Biologically, it corresponds to the transformation, for example, of visual information into a physiological response from motor neurons. The perceptron is based on a mathematical model of information perception by the brain. Different researchers define it differently. In its most general form, it represents a system of elements of three different types: sensors, associative elements and reactive elements.

The first to work include S-elements. They can be at rest (signal is 0) or in the excitation state (the signal is 1). Then the signals from the S-elements are transmitted to the A-elements along the S-A bonds. These links can have weights that are only –1, 0, or 1.

Then the signals from the sensor elements passed through the S-A bonds fall into the A-elements, which are also called associative elements. A single element can correspond to several elements. If the signals arriving at the A-element collectively exceed a certain threshold θ, the A-element is excited and gives a signal equal to 1. Otherwise (signal from the S-elements did not exceed the threshold of the A-element), a zero signal is generated.

A-elements are called associative, because A-elements are aggregators of signals from sensory elements. For example, we have a group of sensors, each of which recognizes a piece of the letter “D” on the picture under study. However, only their totality (i.e., when several sensors output a signal equal to 1) can excite the A-element entirely. The A-element does not react to other letters, only to the letter “D”. That is, it is associated with the letter “D”. Perceptron is used for pattern recognition, weather forecasting, etc.

The neural network is a special architecture of artificial neural networks, proposed by Jan Lekun in 1988 and aimed at effective image recognition, is part of the technology of in-depth training. It uses some features of the visual cortex, in which so-called simple cells that react to straight lines from different angles were opened, and complex cells whose reaction is associated with the activation of a certain set of simple cells. Thus, the idea of convolutional neural networks is to alternate convolutional layers and subsampling layers. The network structure is unidirectional, essentially multilayered. For training, standard methods are used, most often the method of back propagation of the error. The function of activation of neurons is at the choice of the researcher.

Differences:

- Application of the principle of shared weighting coefficients, where the S-type neurons divide the weight coefficients with other neurons of the same layer; this approach allows to significantly reduce the number of free parameters of the neural network and reduce its resource intensity in relation to the required RAM size.
- The multi-layer perceptron architecture contributes to the possibility of reducing the computational complexity of the network by reducing the necessary amount of synaptic connections.
- Perceptron is needed for light tasks (by type to recognize the letter), and the convolutional network allows to optimize the neural network, i.e. first, for example, classifies the object (square), then details the object (road sign), and at the end concretizes the object (the sign “Entry prohibited”). And deep training itself determines how many layers to make, what will be the identifiers. Thus, it determines how to find the road sign and specify it.
- A convolutional layer in a neural network is just a layer that allows you to reduce the dimension of the feature card (features are called features in English literature and lectures). Convolutions are not the opposite of deep neural networks, deep neural networks are simply neural networks with a large number of layers, compared to the perceptron, and that's all.

As a model of a neural network, a perceptron was selected (Fig. 1). In the perceptron there is no feedback and, consequently, memory, all elements of the previous layer are associated with all elements of the next layer using weights calculated using the training set. Hidden layers are involved only in calculations, their number and dimensions affect the accuracy of the neural network and the power required for the calculation [5].

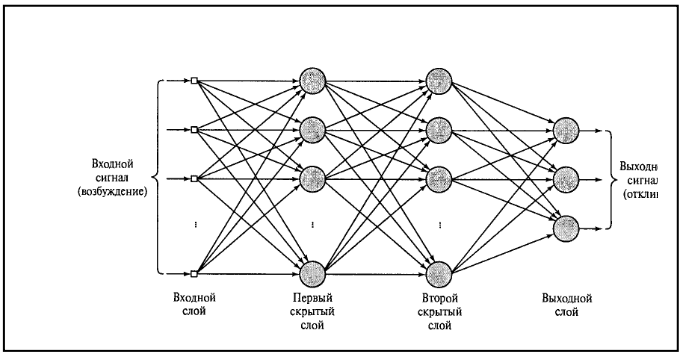

Fig. 1. General view of perceptron

The experiments were carried out in the mathematical package Octave 4.4.1. For realization, a training set consisting of black and white images in the size 20x20 was collected. Each pixel is encoded with one of the shades of gray, taking a value from 0 (black) to 255 (white). An example of an image and a matrix representation of its fragment are shown in Fig. 2.

Fig. 2. An example of a queue image and a matrix representation of its fragment

It is necessary to select extremely carefully images for the training set. We should take into account a number of conditions: different illumination of the room; the camera lens, foreign objects, for example racks or a shopping trolley, may fall; a malfunction of the camera itself, as a result of which noise can be observed in the image (Fig. 3) or “broken pixels”.

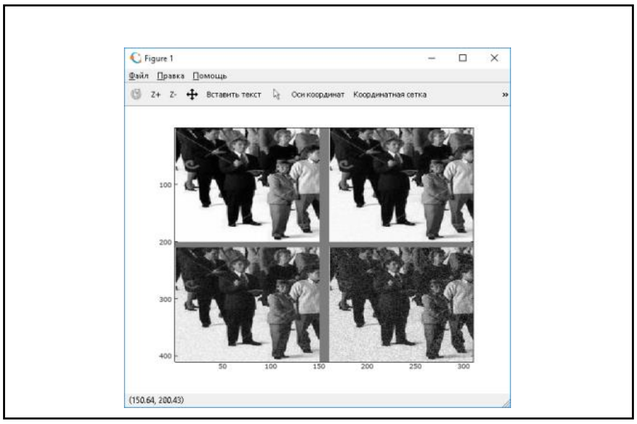

Fig. 3. An example of distortions in the image of a live queue, there is noise

III. TRAINING AND ESTIMATION OF ACCURACY OF NEURON NETWORK

The training set consists of 200 images, 150 of which are distorted copies of the first 50, obtained programmatically, by changing the matrix parameters of the image, and is divided into classes 120 with a live queue, 80 without a queue. Each example includes 400 parameters (the product of width and height), as well as the value of the output parameter that corresponds to the presence of a live queue, “1” is a live queue, “0” is no live queue. To further test the operation of the neural network, the training set is divided into a training set and a test set in 70% to 30% ratios. As a result, a training set of 140 examples was obtained (90 with a live queue, 50 without) and a test set of 60 examples (30 for each class). As a neural network architecture, a perceptron with one hidden layer was chosen. Experimentally, it was decided to use 5 elements of a hidden layer, the results of experiments in the form of a comparative graph are shown in Fig. 4. Training was conducted by the method of back propagation of the error.

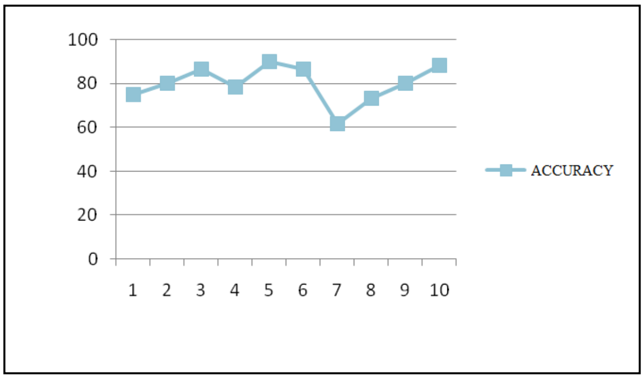

Fig. 4. Changing the accuracy of a neural network with changing the hidden layer

Before learning the neural network, it is also necessary to specify the initial values of weight coefficients that determine the weight of each element of the previous layer with respect to the element of the next layer, and the errors necessary for the back propagation method of the regularization coefficient, which affects the speed and correctness of the algorithm operation, and the termination conditions of the method. Although the weight coefficients are due to the method of back propagation of the error, in order to avoid the symmetry of the coefficients, it is necessary to specify certain initial values that are different from each other. To do this, we perform a “random initialization” assigning weight coefficients random values from the range [–ε; ε]. Thus, the weighting coefficients will take random values: $-\varepsilon \le \Theta ij\ (l) \le \varepsilon$. In this problem, ε was chosen to be 0.005. The regularization factor (λ) influences the work of the training method: in its absence, the neural network can be trained to work correctly only with these identical training; If the coefficient is too small, the method will require much more iterations to find the weighting coefficients; if the coefficient is too large, the method may not find the optimal result, since at each iteration it will “step over” through the solution. The regularization factor (λ) can be dynamic or static, at the initial stages of training the neural network is used static, since this simplifies the possibility of manual verification of the results. The initial value of the coefficient, obtained experimentally (Fig. 5), λ = 0.5.

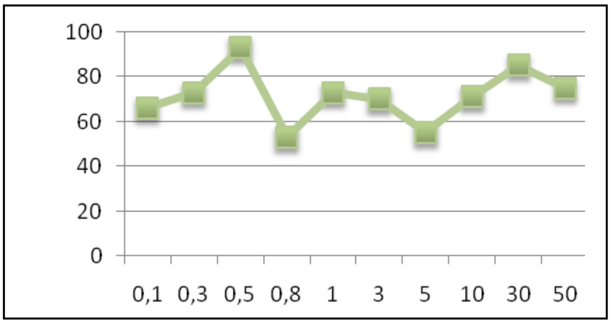

Fig. 5. The accuracy of the neural network from the regularization coefficient

A condition for stopping the work of a learning method can be a given finite number of iterations or conditions for achieving an error, when as a result of the following iterations there is a slight change in the values. In this problem, as a stopping condition, it was decided to use 100 iterations of the method of back propagation of the error. The accuracy of the neural network was calculated on the number of correctly predicted classes with respect to all classes.

IV. TESTING DEVELOPED NEURAL NETWORK AND COMPARISON WITH ANOTHER MODEL OF NEURON NETWORK

For testing, we chose an image (Fig. 6), which does not apply to either the training set or the test set, and is a queue.

Fig. 6. Image for testing the developed program

The function “imread” was loaded with an image, and also loaded the optimal values of the weights obtained earlier. For classification, the function “predict” was used, which was fed to the input image matrix and the matrix of the weights. At the output of the neural network we get the probability that the image belongs to this class.

The test result is shown in Fig. 7.

Fig. 7. The result of the developed program

The program recognized this image as a live queue which it follows that the network is working correctly.

We will conduct such an experiment using the open library of computer vision Yolo, written in C.

Yolo uses a different approach. A single neural network is applied to the full image. This network divides the image into regions and predicts the limiting fields and probabilities for each region. These limiting fields are weighted by the projected probabilities.

The abbreviation YOLO stands for You Only Look Once. This model imposes a grid on the image, dividing it into cells. Each cell attempts to predict the coordinates of the detection zone with a confidence estimate for these fields and the probability of the classes. Then, the confidence estimate for each detection zone is multiplied by the probability of the class to obtain a final rating [6].

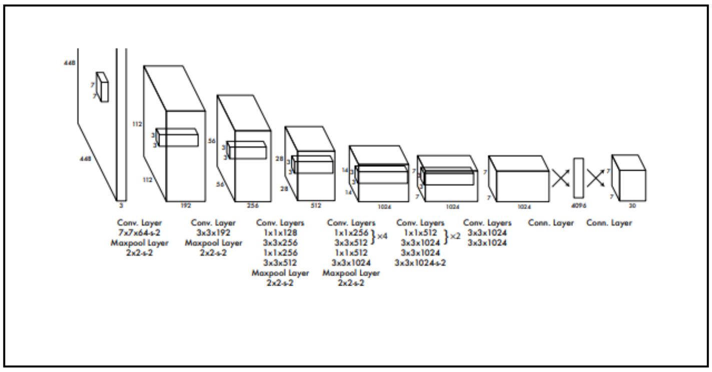

Fig. 8. The architecture of the Darknet YOLOv3 neural network

The testing was carried out on a computer with the operating system Ubuntu Linux 16.04 with updated drivers installed library OpenCV 3.4.3.

In the linux console, we enter the commands for installing the neural network Darknet:

```
git clone https://github.com/pjreddie/darknet
cd darknet
make
```

Next, we load the weights YOLOv3 command:

wget https://pjreddie.com/media/files/yolov3.weights

We load the image with the name WH7GpHNnzpg.jpg into the ./darknet/data/ directory and start the processing process with the command:

```
./darknet detect cfg/yolov3.cfg yolov3.weights data/WH7GpHNnzpg.jpg
```

After image processing, we get the following result:

Fig. 9. Result of image processing in YOLOv3

V. CONCLUSIONS

A neural network in the form of a perceptron with reasonable accuracy can only give out the fact of having a queue, so this type of neural network is not suitable for use in queue control systems.

The Darknet YOLO convolutional neural network surpasses the perceptron in recognition accuracy and it is advisable to use it in the queue control systems.

VI. REFERENCES

[1] Maas A. L., Hannun A. Y., Ng A. Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models // Proceedings of the 30 th International Conference on Machine Learning. Atlanta, 2013. Vol. 28.

[2] Воронов В.И., Воронова Л.И. О повышении результативности магистерских программ в условиях инновационной экономики // Инновационные подходы в науке и образовании: теория, методология практика / Под общ. ред. Г.Ю. Гуляева. Пенза, 2017. С. 35–44. [Voronov V. I., Voronova L. I. Increasing productivity master's program in the innovation economy. In: Innovative approaches in science and education: theory, methodology, practice / Ed. G. Yu. Gulyaev. Pensa, 2017. P. 35–44].

[3] Воронова Л.И., Воронов В.И.. Machine Learning: регрессионные методы интеллектуального анализа данных: Учеб. пос. / МТУСИ. М., 2018. [Voronova L. I., Voronov V. I. Machine Learning: Regression methods data mining: A tutorial / MTUCI. Moscow, 2017].

[4] Marmanis H., Babenko D. Algorithms of the Intelligent Web. Greenwich: Manning Publications Co., 2009.

[5] Brink H., Richards J., Fetherolf M. 2016. Real-World Machine Learning. Greenwich: Manning Publications Co., 2016.

[6] Agrawal P., Girshick R., Malik J. Analyzing the performance of multilayer neural networks for object recognition // ECCV. 2014. P. 329–344.