2004

THEORY AND PRACTICE OF ARTIFICIAL NEURAL NETWORK CONSTRUCTION FOR THE RECOGNITION OF HAND-WRITTEN CHARACTERS

In this paper, we describe the methods of neural networks' construction, which are used to classify images, and also the methods of input data preprocessing in order to increase the system's performance quality. In this paper, we also introduce the description of one of the possible algorithms for recognition of hand-written characters with the use of neural network set. The mathematical model we describe is notable for high consistency of operation in short time under operation conditions of neural network set.

PhD A. Lipanov
Master Degree A. Kochetov

Neural network in image processing as well as in tasks, connected with the classification of image data have been widely used. The problem of printed characters' recognition is solved everywhere with the help of strong software products having enormous databases with the information about type fonts and with a lot of additional information. Though the task of characters' recognition is (as is generally known) solved be a human immediately, regardless of the font. One of the remarkable features of neural networks is their ability to find unique relations in input data, to change their perceptibility according to the kind of its allocation [4].

In whole the process of text recognition is quite difficult and time-taking process, which is connected with a correct preliminary data processing. First we slot the object with alphanumeric data into strings and words of the text. And then we part it into characters. After this process is finished, we start with the recognition, i.e. we divide characters into different categories. The problem of segmentation of printed and hand-written characters into strings and words can be solved with the same algorithms as the methods of body structure are fixed. The process of classification is not fixed, because the way each character is written is almost unique for each person.

While developing the system of recognition of hand-written characters it is reasonable to narrow the class of recognizable characters. In connection with this fact let's take a look at the recognition task of hand-written numbers from 0 to 9. In whole the problem of character's recognition in the context of neural networks can be presented in two parts: training and recognition. Training is performed by inputting separate images indicating their belonging to one or another class. As a result of such process the recognition system gains the ability to react identically on all the objects of one class and differently on all the objects of different classes. After the process of training goes the process of recognition of new objects, that characterizes the actions of already trained system. Automation of these processes makes the problem of image recognition learning.

To recognize characters on the basis of input bit cards 8x8 points we use back propagation net with two layers. This net also has a layer of neurons that don't influence it and are not involved in the training process. This layer is an input one in the net and performs a distributive function of impulses into neurons of the next layer. All weighting coefficients of the input layer equal 1, i.e. input data do not change while they are being moved through the layer. The second layer is a Kohonen layer [3]. It contains 16 neurons. The input of each of these neurons receives 16 impulses from the output of net neurons of the distributive layer. Classical Kohonen layer works according to the scheme "the winner takes it all". Though it is more reasonable for the research to use Kohonen layers in the interpolation mode, i.e. when there are several zero impulses with values from 0 to 1 on the layer output. So the output of Kohonen layer can be interpreted as a theoretical probability of belonging of input vector to one or another class. In such case the whole group of Kohonen neurons having the biggest output can transfer its signal to the Grossberg layer [3]. The number of neurons in such a group must be selected according to the task and there is no convincing data about the optimal size if the group. All neurons beyond the group have zero output. At the same time we set some range of neurons' affinity. This value determines the maximal absolute value of the difference between outputs of two neurons in Kohonen layer, by which these neurons are considered to locate quite close to each other to get in the group with the biggest number of outputs. After the maximal output among the neurons of Kohonen layer has been chosen, we compare absolute values of output differences of a given neuron (with maximal output) and of other neurons of the layer. Outputs don't zero for those neurons, whose outputs have got in the neighborhood of maximal neuron output with radius equal to a given resemblance radius.

To allocate weight vectors in accordance with the density of input vectors, that must be divided, and placing in such a way more vectors in the neighborhood of a big number of input vectors, we have chosen a method with the use of radius adjustment. Net training begins with random weights, but at the initial stage we adjust all the weights and not only those connected with the winning Kohonen neuron. In such a way weight vectors move closer to the area of input vectors. In the process of training the weight correction starts being performed only for the closest neurons to the winning Kohonen neuron. This adjustment radius gradually becomes smaller and only those weights connected with the winning Kohonen neuron are being adjusted [1].

It is better to normalize input vectors before delivering them to their net input in the following way (1).

$$x'_i = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}} \quad (1)$$

This turns an input vector into unit vector with the same direction, i.e. into a vector with a unit length in the n-dimensional space.

There is such two-dimensional vector **V** in fig. 1 presented in coordinates **x-y**, where coordinate **x** is equal four and coordinate **y** – three. Square root of a sum of squares of these components is equal five. Division of each component of **V** by five gives vector **V** with components 4/5 and 3/5, where **V’** goes in the same direction as **V** but has unit length.

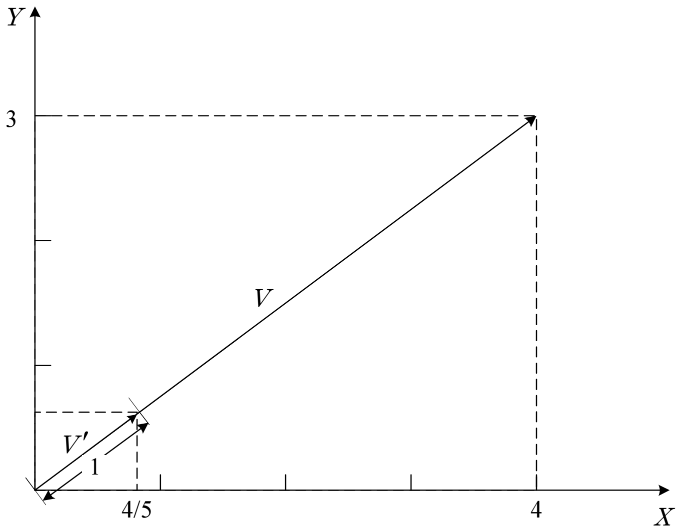

Fig. 1 - Input unit vector

There are several unit vectors on fig. 2. They end in the points of unit circumference (circumference with unit radius), which occurs only when the net has two inputs. If there were three inputs, vectors would be presented as arrows ending on a surface of unit circumference. These presentations can be transferred to nets having random number of inputs, where each input vector is an arrow ending on the surface of unit hypersphere [3].

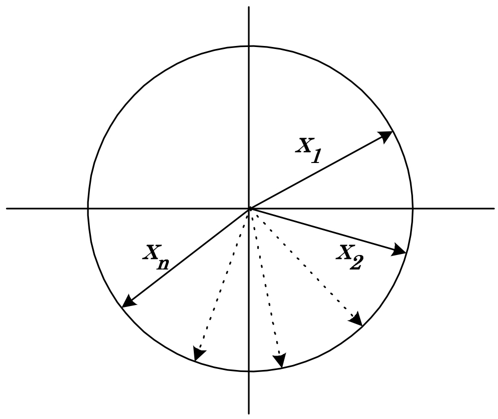

Fig. 2 Two-dimensional unit vectors on the unit circumference

At Kohonen layer training an input vector is delivered to input and we compute its dot product with weight vectors, connected with all Kohonen neurons. Neuron with the maximum value of dot product is declared the “winner” and its weights are being adjusted. Dot product used to compute values NET is a measure of convergence between input and weight vectors. Therefore the process of training consists of choosing Kohonen neuron with weight vector, which is the closest to the input vector, and of further approximation of weight vector to the input one. The following process is called self-training. The system organizes itself in such a way that the given Kohonen neuron has maximum output for input vector. The process of training has the form:

$$w_{\text{н}} = w_{\text{c}} + \alpha(x - w_{\text{c}}), \quad (2)$$

where $\mathbf{w_H}$ – new value of weight connecting input component **x** with the winning neuron; $\mathbf{w_C}$ – is a previous value of this weight; α – coefficient of training speed, that can vary in the process of training.

Each weight connected with winning Kohonen neuron changes in proportion to the difference between its value and the value of input where it is connected. The direction of change minimizes the difference between weight and its input.

On fig. 3 this process is shown geometrically in two-dimensional mode. First we find vector $\mathbf{x - w_C}$. To do this we need to make a segment from the end **w** to the end **x**. Then we shorten this vector by multiplying it by dot product α (smaller value). As a result we get a change vector **δ**. The final new weight vector $\mathbf{w_H}$ is a segment going from the beginning of the coordinates to the end of vector **δ**. As a result we can see that the effect of training is in the rotation of weight vector in the direction of input vector without any important changes in its length.

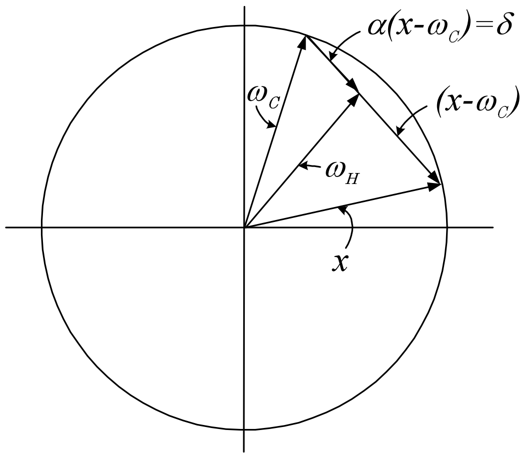

Fig. 3 - Rotation of weight vector in the process of training ($\mathbf{w_H}$ – vector of new weight coefficients, $\mathbf{w_C}$ – vector of old weight coefficients)

Variable α is a coefficient of training speed, which is first ~ 0,5 and can gradually reduce in the process of training. This gives an opportunity to make big steps at the beginning for fast rough training and smaller steps at the approach to the final value.

If each Kohonen neuron was associated with one input vector, then Kohonen layer could be trained with the help of one computing on weight. Weight of winning neuron would be equal to the components of training vector (α = 1). As a rule a training quantity includes many similar input vectors, and the net must be trained to activate one and the same Kohonen neuron for each of them. In such a case the weights of this neuron must result from averaging of input vectors, which must activate it. Gradual reduction of value α reduces the influence of each training step, so that the final value will be an average value of input vectors where the training processes. So the weights, associated with the neuron, will have their value near the "center" of input vectors, for which the given neuron is a "winner"

When the net is trained, application of an input vector leads to a required output vector. Ability of the net to generalize helps to find the correct output even by the application of input vector, which is incomplete or slightly wrong. This gives an opportunity to use the given net to recognize and restore images and to intensify signals.

To solve the problems of image processing and classification (hand-written characters recognition as well), transfer of unprocessed and preliminary data to the net input does not usually give desired results. This is not connected with the specific characters of recognizable images, but with the specific characters of possible differences among images of one or various classes. For the characters written by person such specific characters are: inclination, writing amplitude etc. It is evident, that the images of one and the same number written by different people can differ a lot. In this case we need an algorithm that is able to emphasize the common qualities of input data.

As an algorithm that helps to identify the characteristics of each symbol we can use a scheme based on the Kirsch method of boundary allocation. For each input image only those segment of image lines are chosen, which have certain direction: there are horizontal, vertical, left- and right-diagonal characteristics of symbol image. To do this, only those elements are examined in two-dimensional aperture 3x3 used in Kirsch method, which characterize the presence of a line segment in the given direction.

5	5	5
-3	0	-3
-3	-3	-3

-3	-3	-3
-3	0	-3
5	5	5

(a)

-3	5	5
-3	0	5
-3	-3	-3

-3	-3	-3
5	0	-3
5	5	-3

(b)

-3	-3	5
-3	0	5
-3	-3	5

5	-3	-3
5	0	-3
5	-3	-3

(c)

-3	-3	-3
-3	0	5
-3	5	5

5	5	-3
5	0	-3
-3	-3	-3

(d)

Fig. 4 - Image analysis on basis of Kirsch masks: a) horizontal characteristics, b) vertical characteristics, c) right-diagonal characteristics, d) left-diagonal characteristics

Formally the process can be presented with the following relations.

$$G(i,j)_H = \max(|5S_0 - 3T_0|,|5S_4 - 3T_4|) \quad (3)$$

$$G(i,j)_V = \max(|5S_2 - 3T_2|,|5S_6 - 3T_6|) \quad (4)$$

$$G(i,j)_R = \max(|5S_1 - 3T_1|,|5S_5 - 3T_5|) \quad (5)$$

$$G(i,j)_L = \max(|5S_3 - 3T_3|,|5S_7 - 3T_7|) \quad (6)$$

Values $\boldsymbol{S_i}$ and $\boldsymbol{T_i}$ are determined from the relations:

$$S_k = A_k + A_{k+1} + A_{k+2} \quad (7)$$

$$T_k = A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7} \quad (8)$$

At the same time indexes of **A** are calculated by module 8 and

$$A_k = (k = 0,1,\ldots,7) \quad (9)$$

- 8 neighboring points to the point with indexes **(i, j)**, determined as it is shown in fig. 5.

A_0	A_1	A_2
A_7	(i, j)	A_3
A_6	A_5	A_4

Fig. 5 Determination of 8 neighbors with respect to the point with indexes **(i, j)**

So on basis of the aforementioned formulae 4 images – masks are formed, which are being later applied on the input image 16x16 points. As a result we manage to get 4 more bit cards from the initial image, which characterize horizontal, vertical left- and right-diagonal characteristics o symbol image.

The performed experiments have shown, that the use of cards of symbol writing characteristics by recognition allows increasing the percentage of correct recognition of characters by artificial neural network (networks) more than on 20% in comparison with the processing of initial images.

According to the selection of training and test samplings, the described scheme of neural network organization allows getting from 40 to 60% recognition rate.

On the assumption of the fact that the input image is being processed to determine four extra images-characteristics of symbol writing, the analysis of 5 input images must be performed. In order to do this, a set of networks is used, based on the model of counterpropagation with Kohonen and Grossberg layers. On the input of one of them the so called global characteristics is being transmitted – symbol image without processing with algorithms of allocation of directed segments of image lines. On the other 4 networks the images are transmitted, which are received after allocation of horizontal, vertical, left- and right-diagonal characteristics of symbol images.

Each of the examined neural networks is trained independently and parallel. In the future the results of the work with each network can be also analyzed to determine the exact classification. There are many different methods to analyze the results of working data of neural networks of one type and architecture. For example, for the accreditation mode of Kohonen layer (only when one neuron of the layers has unit input), a voting method can be used, where the final classification is the one that was generated by most of the system networks. In the interpolation mode we can average network outputs, apply the elements of fuzzy logic, taking in account the weight of each neural network. This weight can be chosen for example on basis of statistic data about how accurate one or another neural network in the results of classification. As an alternative we can apply one more neural network to perform classification, which will be quite logical. In the process of training such network can perform the correction of parameters in such a way, that the maximal values of semantic weights correspond with the outputs of the neural network, which is more precise in evaluation. And vice versa, for neural networks, which give more mistakes in classification than others, the sensibility will be lower.

Network has a quite easy model. There is a combined network of 10 neurons constructed. Number 10 determines the number of outputs classification networks (the number of outputs corresponds with the number of classes of recognized characters – in this case there are numbers from 0 to 9). Each neuron has 4 semantic weights. Each weight determines how precise the work of the neural classification network, which corresponds with the given input (or weight). So the training process of combined network can be in some way called a process of estimation of recognition rate not only of one particular classification network but for recognition rate of each symbol by every network. There are 10 outputs in the network as well. Among them the maximal output is chosen. Input image refers to the class, which correspond with the maximal output of combined network.

The application of a set of neural classification networks, which are trained separately, together with the neural network that analyzes their outputs, gives the possibility to increase the recognition rate up to 75 % for selection of hand-written symbols MNIST (of American National Institute of Information and Technologies).

For practical experiments of the given algorithm we have developed a program, which includes the implementation of basic components. The developed hierarchy of classes includes the models of neuron, neuron layer, perceptron, counterpropagation network and recognition network with the number of input neurons 16 and with the number of neurons 10 for Grossberg layer. The application performs network training processes on basis of aforementioned selection MNIST. Conditions of network systems can be saved to study them step-by-step. The performance quality of neural network system can be tested at any time by recognizing test selections with the possibility to analyze not only work accuracy percentage wise but also the fact, at what numbers there was the biggest number of errors and for benefit of what number there was performed classification in the recognition moment. In future in order to recognize a more classes of symbols, it is possible to start with alphabet letters recognition.

We plan to continue further experiments on the problem of hand-written characters recognition with the improvement of neural classification network architecture. It is a well known fact that to chose an optimal amount of neurons in every layer of neural network and its architecture is extremely difficult. Though the quality and particular features of Kohonen layer functioning, which is included into the counterpropagation network, make it possible to perform its reorganization, i.e. to construct a model of neural network, which is able to choose an optimal amount of neurons and semantic weights as well [2]. To implement such model in a special case of counterpropagation networks' application only one extra step in neural network training is needed – to detect those neurons that should be replaced with extra elements and those that should be deleted from the architecture.

Литература

1. Grossberg S. 1974. Classical and instrumental learning by neural networks. Progress in theoretical biology, vol. 3, pp. 51–141. New York: Academic Press.

2. Kohonen T. 1984. Self–organization and associative memory. Series in Information Sciences, vol. 8. Berlin: Springer verlag

3. Ф. Уоссермен. Нейрокомпьютерная техника: Теория и практика. Перевод на русский язык, Ю. А. Зуев, В. А. Точенов, 1992.

4. Горбань А.Н. и др. Нейроинформатика. – Электронная публикация.