THE RECOGNITION CHARACTERS FOR DACTYL ALPHABET WITH USE THE FEED FORWARD NEURAL NETWORK

Roman Záluský[1], Mário Krajmer[1] , Vladimír Sedlák[1] , Daniela Ďuračková[1]

1. Slovak University of Technology, Faculty of Electrical Engineering and Information Technology
E-mail: roman.zalusky@stuba.sk

Received 30 April 2011; accepted 7 June 2011.

1. Introduction

This paper deals with recognition in dependency of count of objects in the training data set. We describe the train algorithm, which is used in our project. For the capture the fingers position of the sensory glove – 5DT Data Glove Ultra is used. Next we describe the computer software "Dactyl Teacher" for recognition characters of Dactyl alphabet. This software contain a feed forward neural network, which is used for character recognition.

2. Main results

2.1. Dactyl teacher and sensory glove

We developed the software Dactyl teacher for recognizing and learning the Dactyl alphabet. Character capture is realized with the use of a commercial sensory glove - 5DT Data Glove Ultra. On this glove there are 14 optical sensors. The output of each sensor is in range from 0 to 4096. This range is redundant, therefore we adjust it to range from 0 to 127. Between each pair of fingers there is a sensor which scans the diversion of these fingers. Each finger has two sensors which scan bending of fingers.

The captured data of finger positions from the sensory glove are next processed in our software Dactyl teacher. The software contains a feed forward neural network which ensures character recognition. Output of this neural network is 6-bits character code. The information of finger positions is first processed with the neural network and then software Dactyl teacher displays the evaluated character. The program Dactyl teacher was programmed in C#.

2.2. Neural network

The neural network as is describe later was also programmed in C#. It is a part of the Dactyl teacher software. This neural network consist of four layers, input layer, two hidden layers and output layer. Input layer contains 20 input neurons, both hidden layers contains 40 neurons and output layer contains 6 neurons. Each neuron is connected with each other neuron from previous layer. Output of this neural network is the 6-bits binary character code, which represents the recognized character. The output activity of neuron is expressed in equation (1), where the summation is going through neurons which are the parents of the current neuron. We have used as the activation function the sigmoidal function. This function is expressed in equation (2). In sigmoidal function which is used by neurons in our neural network is parameter B equal to 0, parameter A is equal to 1 and the slope is equal to 0,8.

$$y = s\left(\sum_{i=1}^{n} w_i x_i + \vartheta\right) \tag{1}$$

where w_i and ϑ are the weight and threshold coefficients of the neuron, x_i are the activities of parent neurons and s(x) is the activation function of the neuron.

$$s(\xi_i) = \frac{B + Ae^{-\alpha\xi}}{1 + e^{-\alpha\xi}} \tag{2}$$

where A is the maximum of the function, B is the minimum of the function, α is the slope of the sigmoid function and ξ_i is the potential of neuron.

2.3. Learning algorithm

Each neural network must be trained by use any training algorithm. We used the back propagation algorithm to train our feed forward neural network. Back propagation is most used algorithm to train neural networks. The principle of back propagation algorithm is propagation of errors backwards through the network. The error function is expressed by equation (3). By use this error function are changing the weights and threshold coefficients by equations (4).

$$E_k(\boldsymbol{w}, \vartheta) = \frac{1}{2}(y_o - \hat{y}_o)^2 \tag{3}$$

where y_o is output activity of current neuron $\hat{y}_o$ is required activity of this neuron, ϑ is the threshold coefficient and **w** is the weights vector.

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} - \lambda \frac{\partial E}{\partial w_{ij}} + \mu \Delta w_{ij}^{(k)}$$
$$\vartheta_i^{(k+1)} = \vartheta_i^{(k)} - \lambda \frac{\partial E}{\partial \vartheta_i} + \mu \Delta \vartheta_i^{(k)} \qquad (4)$$

where λ is the learning coefficient and μ is the momentum element.

2.4. Experimental

First, we examined the number of iterations to finish the learning process of dependency of number of neurons in hidden layer and number of hidden layers (fig. 1). We choose from this dependence the neural network with two hidden layers containig 40 neurons in each hidden layer. Next we examined the success of recognition in dependency of count of objects in training data set. To learn the neural network, we have used the training data set, which contains training objects in range from 1 to 6 for each neuron. The learning process was executed with learning coefficient λ equal to 0,01 and momentum parameter equal to 0,5. Number of training objects represents a count of truthful objects for each character. Obtained dependence is showed on fig. 2. When the training data set is empty, the neural network cannot recognize any character. With increasing number of objects in training data set increases the success of recognition. For full recognition the neural network has to be learned to training data set, which contain minimal six training objects.

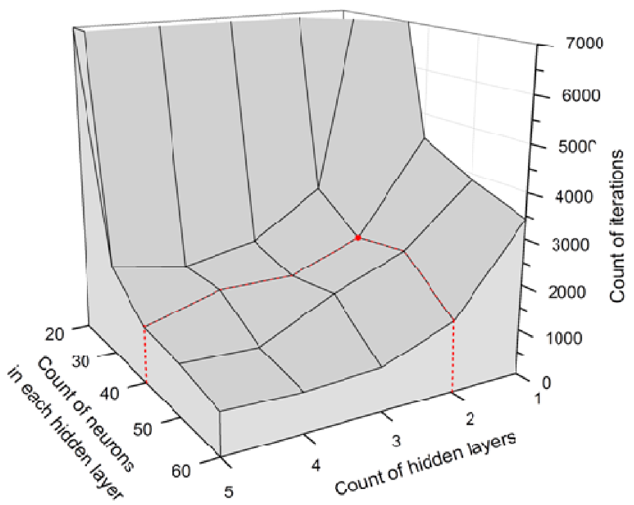

Fig.1: *Number of iterations in dependency of number of neurons in hidden layer and number of hidden layers*

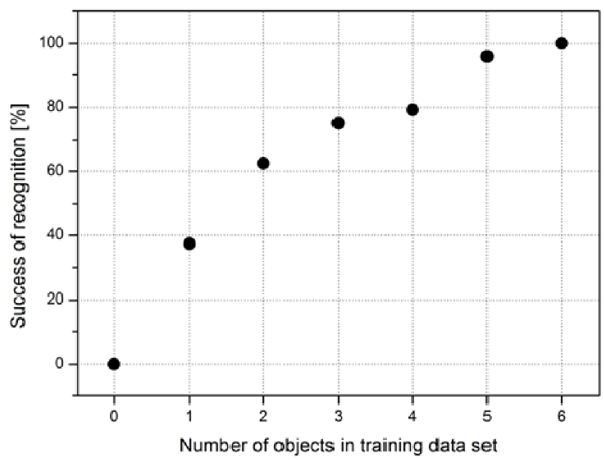

Fig.2: *Success of recognition in dependency of number of objects in training data set*

3. Conclusion

We have developed software Dactyl teacher, which is used for recognizing characters of Dactyl alphabet with use the sensory glove. The software contain also the feed forward neural network which ensures character recognition. We investigated success of recognition in dependency of count of objects in training data set. For full recognition the neural network have to learned to training data set, which contain minimal six training objects. Next we plan the neural network used in this project to implement a FPGA chip.

Acknowledgement

This contribution was supported by the Ministry of Education Slovak Republic under grant VEGA No 1/0693/08.

References:

[1] V. Kvasnička, Ľ. Beňušková, I. Farkaš, A. Kráľ, J. Pospíchal, P. Tiňo,: Introduction into the neural networks, IRIS, Bratislava, (1997).

[2] P. Sinčák, G. Andrejková, : Neural networks (The engineering metodology) part 1: Feed forward network, Elfa s.r.o., Košice (1996).

[3] P. Sinčák, G. Andrejková,: Neural networks (The engineering metodology) part 2: Recurent and modular networks (in Slovak), Elfa s.r.o., Košice (1996).