UDC 004.8 (045)

O. I. Chumachenko

DEEP LEARNING CLASSIFIER BASED ON NEFCLASS NEURAL NETWORK

Technical Cybernetic Department National Technical University of Ukraine
"Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
E-mail: chumachenko@tk.kpi.ua

Abstract—*It is proposed a new class of fuzzy classifiers. It is a deep learning classifier based on NEFCLASS neural network. The pre-learning is supplied with help of Restricted Boltzman Machine.*

Index Terms—Fuzzy classifiers; deep learning; NEFCLASS neural network Restricted Boltzman Machine.

I. INTRODUCTION

The neural networks are currently used for the solution of various kinds of problems, such as approximation, classification, pattern recognition, prediction. Along with the classical approach, there are a number of methods for these problems solution, to simplify and speed up the computations. One of such method is the transition to the fuzzy logic. Fuzzy logic systems transform the numerical inputs of neural networks into data of fuzzy nature using the fuzzification process. The main element of fuzzification is a membership function μ. Its value determines whether the numerical value of the input "low", "normal" or "great." The most common types of membership functions are triangular, trapezoidal, bell-shaped and Gaussian function. Further there is the aggregation of obtained results using the fuzzy rules, logical conclusion and defuzzification – transform of fuzzy values into numeric format [5].

NEFCLASS network belongs to a class of three-layer fuzzy perceptrons, NEFCLASS neural network has a three-layer serial structure shown in Fig. 1.

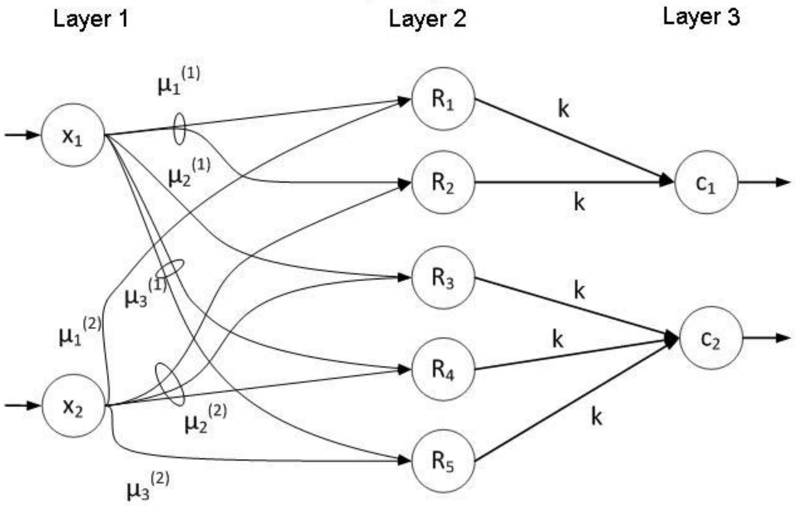

Fig. 1. Network structure NEFCLASS

1) The first layer is the input data and does not change their values.

2) The neurons of the hidden layer (Layer 2) contain fuzzy rules, for example, fuzzy rule R_j takes the form: if x_1 belongs $\mu_k^{(1)}$, x_2 belongs $\mu_l^{(2)}$, … , x_n belongs $\mu_p^{(n)}$, the output R_j will be equal to 1, where $\mu^{(1)} \dots \mu^{(n)}$ are membership functions.

In general, the membership function for the network NEFCLASS bell-shaped function of the following form

$$\mu(x) = \frac{1}{1 + \left|\dfrac{x - c}{a}\right|^{2b}}, \qquad (1)$$

where c, a, b are adjustable parameters during training process. The number of membership functions is given in an arbitrary manner, increasing the number of membership functions leads to the improved accuracy.

3) The third layer consists of output neurons, each of which corresponds to one of the classes. The output value is calculated by the equation

$$S(R)=\sum R_j\, k, \qquad (2)$$

where R_j is the output of the second layer equal to 0 or 1; k is the weight coefficient equal to 1.

II. Problem Statement

The standard approach to learning is based on the method of back propagation [1], however, a large number of adjustable parameters lead to a halt in the algorithm stop in local extremums, that influences badly which adversely affects the accuracy of the network [1].

To improve the accuracy of the network uses the concept pre-learning [2] used in training deep neural networks, for which the basic learning algorithm is a method of backpropagation. Pre-learning is done by constructing autoassociator – neural network, the output of which is to be closest to the value of the input data. Autoassociator trained teachers without the inputs identified in the training set, then the weighting coefficients autoassociator transferred to the main network and continue learning by back propagation. Thus, there is an initial setup that allows you to be as close as possible to the global extremes, which increases the accuracy of the network. The most common autoassociator are Restricted Boltzman Machine (RBM) and autoencoders [3]. In this paper we used the RBM is limited.

III. Problem Solution

Since pre-learning assumes autoassociator structure (the number of neurons and interneuronal connections, as well as activation of neuronal function) coincides with the structure of the network that pre-learning [4], the structure of the neural network must be changed NEFCLASS. In the standard model activate act as weighting coefficients. The new model is proposed to introduce a layer of neurons to calculate the membership functions. Neuron scheme of this layer is shown in Fig. 2.

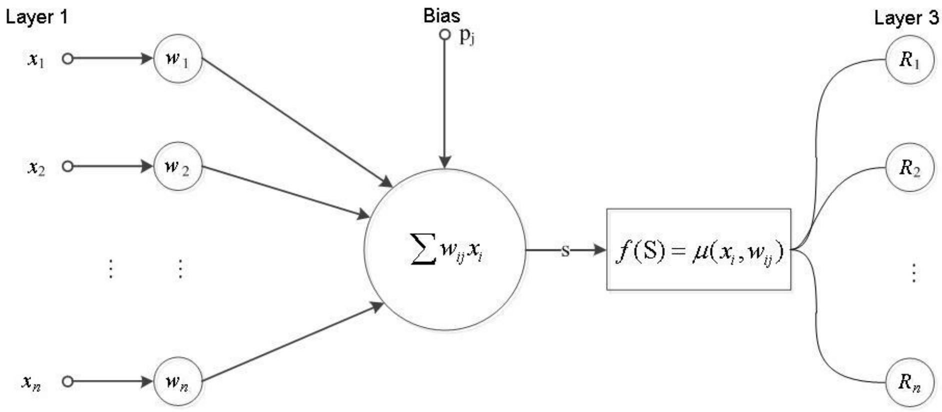

Fig. 2. Neuron scheme of the second layer

The activation function (membership function) of the neurons layer as follows.

$$f(\mathrm{S})=\mu(x_i,w_{ij}). \qquad (3)$$

To pre-learning was possible to be replaced on the sigmoidal membership function, which corresponds to the activation function in the RBM and is as follows

$$\mu(x)=\frac{1}{1+\exp\left(\sum w_{ij}x_i+p_j\right)}, \qquad (4)$$

where p is the bias; w_{ij} are weights (i is the layer number; j is the neuron number of the i layer).

The fuzzy rules in this case take the form: if under x_1, u_1 – maximum, under x_2, u_2 – maximum, … under x_n, u_n maximum, then the sample (x_1 ... x_n) belongs to the class i, where u_1 ... u_n are the outputs of the new layer. Schematically, the learning process is shown in Fig. 3.

Since the number of neurons in the hidden layer of RBM should be less than the number of visible layer neurons [6] it is introduced a limit on the number of membership functions $m < n$, where n is the number of inputs. The structure of constructed RBM follows the structure of the first and second layers of NEFCLASS neural network and the carrying of weight coefficients are realized respectively the circuit in Fig. 3.

Restricted Boltzmann machine consists of two layers: the visible, which are fed to and read data, and the invisible, which is compressed and data processing [7].

Restricted Boltzmann machine structure is shown in Fig. 3. Neurons v_i form a visible layer of neurons, their number corresponds to the number of input variables, h_i neurons form a hidden layer neurons of the hidden layer corresponds to the number of membership functions.

Boltzmann machine restricted is performed using an algorithm independent of divergence obtained weights are added to the network and NEFCLASS learning takes place by standard back-propagation.

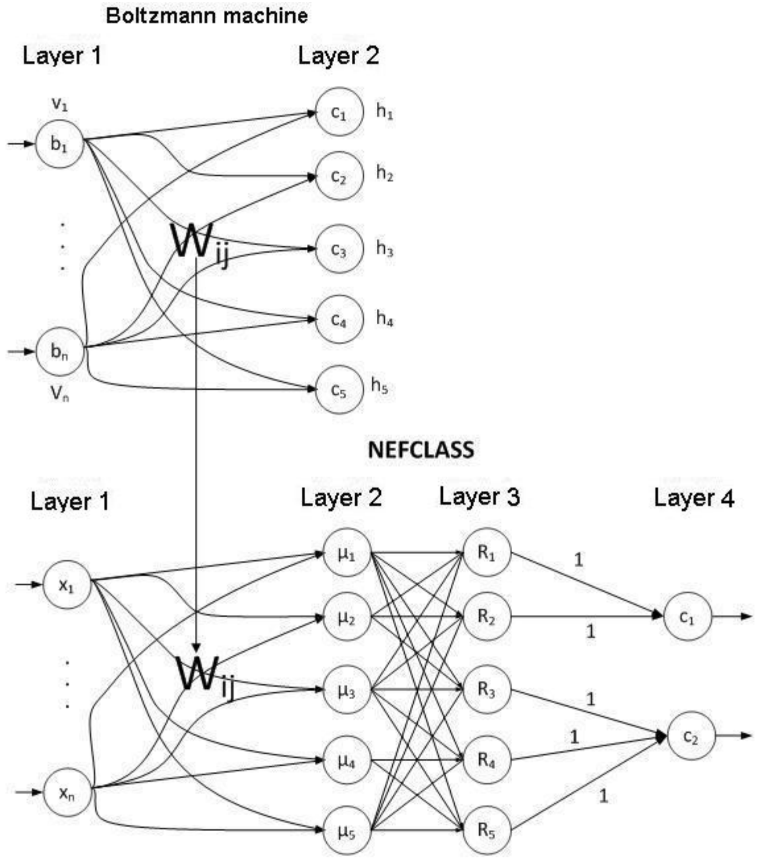

Fig. 3. Pre-learning NEFCLASS neural network

It should be noted that the network used for NEFCLASS complete set of fuzzy rules, which contains all the possible combinations of values of the membership functions.

The equation for calculating the output usually has the following form:

$$f(x) = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n, \quad (5)$$

where α_i are weight coefficients.

The third layer is composed of an adder, which calculates output NEFCLASS neural network. The output value is calculated according to the equation

$$Y = \sum R_j, \quad (6)$$

where R_j is the output of the second layer.

IV. NETWORK TRAINING ALGORITHM

Step 1. Structural adjustment network

1) For all possible combinations of values $\mu_j(x_i)$ it is created the rules R_k. During training for sigmoidal functions of the form (1) it is necessary to adjust two parameters: slope w and the bias c.

2) For all $\mu_j(x_i)$ it is set the initial values. The initial values of membership functions must correspond to the rule

$$\forall x_k \in O(x_i) : \exists \mu_j(x_i) \neq \pm 1,$$

where $O(x_i)$ is the domain of input variable definition. That is, for each value of the input variable domain of definition, there is a corresponding value of the membership functions other than ± 1. An example of the initial distribution of membership functions for the domain [-1,1] is shown in Fig. 4.

Step 2. Parametric adjustment of the RBM is made by the algorithm independent differences

1) The initial values v_0 set equal to the input data.

2) Calculate the probability that the state of the neuron of the hidden layer is equal to 1:

$$p(h_j = 1 \mid v) = \frac{1}{1 + \exp\left(\sum_{i=1}^{m} w_{ij} v_i + b_i\right)}, \quad (5)$$

where c_i is the bias of the hidden layer.

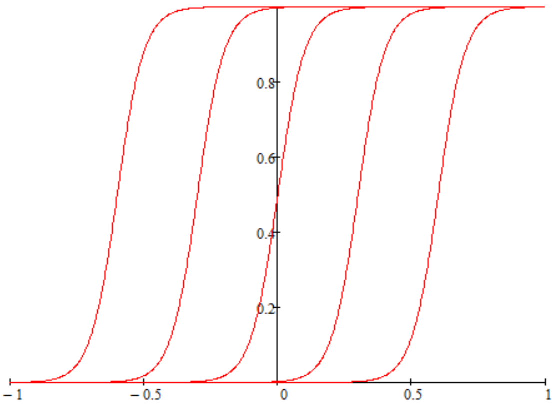

Fig. 4. The initial distribution of the membership functions

3) Assign the value of the hidden layer neuron based on the probability (if the probability of 0.9, then with probability 90% of neuron will be 1, and with a probability of 10% – 0).

4) Calculate the probability that the state of the neuron of the visible layers will be equal to 1:

$$p(v_i = 1 \mid h) = \sigma\left(\sum_{j=1}^{m} w_{ij} h_j + c_j\right), \tag{6}$$

where c_j is the bias of the visible layer; σ is the function

$$\sigma = \frac{1}{1+e^{-x}}. \tag{7}$$

5) Assign neuron layer visibility based on the probability value.

6) Repeat *step 2* if the iteration number is less than *k*.

7) Calculate the probability of the hidden layer.

A. Initial parameter values

1) The states of the hidden layer neurons are given by input data.

2) The initial value of neural connections weights are given as a small value of about 0 to Gause distribution.

3) The biases of the visible layer are given.

B. Parameters modification

1) Synaptic connections

$$w_{ij}(t+1) = w_{ij}(t) + \alpha\left[v_i(0)h_j(0) - v_i(k)h_j(k)\right].$$

2) Biases of the visible layer

$$b(t+1) = b_i(t) + \alpha\left[v_i(0) - v_i(k)\right].$$

3) Biases of the hidden layer

$$c(t+1) = c_j(t) + \alpha\left[h_j(0) - h_j(k)\right],$$

where α is the training speed.

Step 3. Parametric fuzzy network

1) The values for the coefficients w_{ij} and b_i received at the second stage are transferred from RBM to the neural network NEFCLASS as follows.

The values of a membership function p is equal to the biases of visible layer b_i. The value of the slope of the membership function is equal to the value of RBM weighting coefficients w_{ij}.

2) For each training example, a neural network output is calculated.

3) For the neural network accuracy estimate it is used the cost function of following type:

$$C(w,b) = \frac{1}{2n}\sum_x \left\| y(x) - \beta \right\|^2,$$

where n is the number of training examples in the sample, and β is the expected output vector; $y(x) - m$ is the dimensional vector output network; m is the number of possible classes.

4) Adjustment parameters are in accordance with the method of gradient descent [9]:

$$w_{ij} \to w'_{ij} = w_{ij} - \eta \cdot \mathrm{grad}(C)$$

$$b_i \to b'_i = b_i - \eta \cdot \mathrm{grad}(C).$$

5) Combined shutdown criterion.

The algorithm stops working in two cases:

– $\forall X : C(w,b) \le \varepsilon$, those for all the training examples cost function is less than the set value;

– $t < t_s$ i. e. the number of iterations is less than given ones.

V. Conclusions

The opportunity to train fuzzy neural network according to the paradigm pre-learning autoassociator, these results demonstrate improved accuracy on the test sample.

References

[1] D. Nauck, U. Nauck, and R. Kruse, "Generating classification rules with the neurofuzzy system NEFCLASS." *Biennial Conference of the North American, Fuzzy Information Processing Society*, 1996, pp. 466–470.

[2] Y. Bengio, "Learning deep architectures for AI." *Foundations and trends® in Machine Learning*, 2(1), pp. 1–127, 2009.

[3] Y. Bengio, *Deep learning of representations: Looking forward, Statistical Language and Speech Processing.* Springer Berlin Heidelberg, 2013, pp. 1–37.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview." *Neural Networks*, 61, pp. 85–117. 2015.

[5] L. A. Zadeh, "Fuzzy algorithms." *Information and Control*. 12 (2), pp. 94–102, 1968.

[6] G. Hinton, *A Practical Guide to Training Restricted Boltzmann Machines*, 2010, pp. 3–17.

[7] A. Fischer, and C. Igel, *An Introduction to Restricted Boltzmann Machines*, 2011, pp. 1–23.

[8] M. Nielsen, (viewed on September 20, 2016). Using neural nets to recognize handwritten digits [Electronic resourse] – Electronic data.– Mode of access: http://neuralnetworksanddeeplearning.com/chap1.html

Received April 28, 2016

Chumachenko Olena. Candidate of engineering. Associate Professor.
Technical Cybernetic Department, National Technical University of Ukraine “Ihor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.
Education: Georgian Politechnic Institute, Tbilisi, Georgia, (1980).
Research area: system analysis, artificial neuron nets.
Publications: more than 50 papers.
E-mail: chumachenko@tk.kpi.ua

О. І. Чумаченко. Класифікатор глибокого навчання на основі нейронної мережі NEFCLASS

Запропоновано новий клас нечітких класифікаторів. Це класифікатори глибокого навчання на основі нейронної мережі NEFCLASS. Попереднє навчання забезпечується за допомогою обмеженої машини Больцмана.

Ключові слова: нечіткі класифікатори; глибоке навчання; нейронна мережа NEFCLASS; обмежена машина Больцмана.

Чумаченко Олена Іллівна. Кандидат технічних наук. Доцент.
Кафедра технічної кібернетики, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна.
Освіта: Грузинський політехнічний інститут, Тбілісі, Грузія, (1980).
Напрям наукової діяльності: системний аналіз, штучні нейронні мережі.
Кількість публікацій: більше 50 наукових робіт.
E-mail: chumachenko@tk.kpi.ua

Е. И. Чумаченко. Классификатор глубокого обучения на основе нейронной сети NEFCLASS

Предложен новый класс нечетких классификаторов. Это классификаторы глубокого обучения на основе нейронной сети NEFCLASS. Предварительное обучение обеспечивается с помощью ограниченной машины Больцмана.

Ключевые слова: нечеткие классификаторы; глубокое обучение; нейронная сеть NEFCLASS; ограниченная машина Больцмана.

Чумаченко Елена Ильинична. Кандидат технических наук. Доцент.
Кафедра технической кибернетики, Национальный технический университет Украины «Киевский политехнический институт им. Игоря Сикорского», Киев, Украина.
Образование: Грузинский политехнический институт, Тбилиси, Грузия, (1980).
Направление научной деятельности: системный анализ, искусственные нейронные сети.
Количество публикаций: более 50 научных работ.
E-mail: chumachenko@tk.kpi.ua