# ABOUT ONE APPROACH TO APPROXIMATION OF CONTINUOUS FUNCTION BY THREE-LAYERED NEURAL NETWORK

**Ramin Rzayev**
Cybernetic Institute of the National Science Academy of Azerbaijan Republic
raminrza@yahoo.com

**Aygun Alasgarova**
Khazar University, Azerbaijan Republic
a.alasgarova@mail.ru

## Abstract

*According to the accepted opinion a three-layered neural network with feedforward structure is considered as universal approximator for any continuous function. However this is rather conventional representation since trained on finite set of points the given neural network will be more likely interpolator of continuous function, than its approximator. Therefore, that retrain the neural network up to universal approximator so that approximation Vejershtrass's theorem is full carried out it is offered the combined training of neural network by Remez's algorithm with "error backpropagation" and to construct the neural approximation kernel.*

***Key words**: continuous function, Vejershtrass's theorem, feedforward three-layered neural network, neural approximation kernel, Remez's algorithm, error backpropagation algorithm.*

## 1. Introduction

Since 1986 year, after "error backpropagation" algorithm has been offered by Rummelhart, various non-recurrent neural-set constructions began to be applied widely in the decision of many applied problems such as modelling and identification of systems, generation and decomposition of signals, recognition and classification of patterns, etc. Based on the differential nature of neural networks this algorithm has allowed to reveal approximate properties of the networks using nonlinear activation functions into neurons from hidden layers. It has been established [1] that feedforward neural networks even with one hidden nonlinear layer are capable to be a universal approximator for any continuous function on the set consisting of finite number of points. Moreover, known theorems of Chebishev and Valle-Pussen [2] have shown that the problem of approximation of continuous function $f(x)$ on the closed limited set $\Omega$ is equivalent to a problem of approximation of the same function on a subset $\Omega_0 \subset \Omega$ consisting of finite number of points.

The problem of approximation of the continuous functions by neural networks was considered by many authors. In particular, it is theoretically established that the neural network with one nonlinear hidden layer can approximate any continuous function with necessary accuracy [3]. Thus the estimation order of the approximation inversely depends on number of neurons used into the hidden layer. In practice to obtain the required estimation of approximation it is necessary to involve significant amount nonlinear neurons into the hidden layer of neural network, and it, as a rule, leads to increase in time spent for neural network training. It does using of neural networks less effective, especially in the solving of management problems, where often it is necessary to make operatively the "correct-reasonable" decision in response to possible external and internal factors of influence. In the given article the approach to training the neural networks approximating continuous functions is considered somewhat different from traditional. The essence of this approach consists in the following.

For approximation any continuous function $f(x)$ defined on a segment $[a,b]$ instead of traditional neural network

$$y_{net} = \sum_{i=0}^{n} \omega_i \cdot \varphi(\xi_i \cdot x - \theta_i) \tag{1}$$

with any parameters $\omega_i, \xi_i, \theta_i$ and limited sigmoid activation function $\varphi_i(x) = \varphi(\xi_i \cdot x - \theta_i)$ of $i$-th neuron from the hidden layer, it is selected a network of the similar structure

$$y^*_{net} = \sum_{i=0}^{n} \omega_i \cdot \varphi(\xi^*_i \cdot x - \theta^*_i)\,, \tag{2}$$

where parameters $\xi^*_i, \theta^*_i$ are in advance optimized according to inequality

$$\left| \sum_{i=0}^{n} \varphi\left(\xi^*_i \cdot x - \theta^*_i\right) - 1 \right| < \varepsilon_1,\ \varepsilon_1 > 0 \tag{3}$$

by co-application of error backpropagation and Remez’s algorithm for any training set $\{x_j, 1\}_{j=0}^{n+1}$. The neural network inducing the polynomial $net_n(x) = \sum_{i=0}^{n} \varphi(\xi^*_i \cdot x - \theta^*_i)$ and creating thus “favourable conditions” for approximation of any continuous function we shall name as neural-approximation kernel. Training of the neural networks constructed on the basis of such kernel will be limited by optimization only output connections weights $\omega_i (i = \overline{0,n})$. It is obvious, that it considerably will reduce time allocated on training.

## 2. Training of neural networks by combined application error backpropagation and Remez's algorithm

It is well known that for any continuous function it is practically impossible to construct a polynomial of its best approximation $P_n^*(x)$ [2]. Therefore, the decision of approximation problems usually reduce to the finding of the generalized polynomial

$$P_n(x) = \sum_{i=0}^{n} c_i \cdot \varphi(x)\,, \tag{4}$$

which differs from $P_n^*(x)$ on infinitesimal. For the finding such approximate polynomial there are many applied methods. Among these Remez's algorithm in many respects is the most successful. Having obtained the general recognition, it is everywhere used in practice for the approximate representation of continuous functions by polynomials.

Since all multilayered neural networks with feedforward structure finally induce the generalized polynomial type of (4) that, obviously, during network training it is meaningful to use Remez's algorithm, which in contrast to traditional methods allows to optimize the neural network up to universal approximator for any continuous function on all field of its definition. Let's consider the principle of this algorithm in the combination to algorithm "error backpropagation" on an example of continuous function approximation of one variable by the non-recurrent neural network with one hidden nonlinear layer.

Let $f(x) \in C[a,b]$ and on the segment $[a,b]$ let's select the system of $n+2$ various points

$$a \le x_0^{(1)} < x_1^{(1)} < ..... < x_n^{(1)} < x_{n+1}^{(1)} \le b \tag{5}$$

by formula

$$x_j^{(1)} = a + \frac{1}{2}(b-a)(1-\cos\frac{\pi \cdot j}{n+1}),\ \mathrm{j} = \overline{0, \mathrm{n}+1}. \tag{6}$$

Thinking $y_j^{(1)} = f\left(x_j^{(1)}\right)$ let's create the training set $\left\{x_j^{(1)}, y_j^{(1)}\right\}_{j=0}^{n+1}$ on the base of system (5). Then on its basis having applied to a neural network the algorithm "error backpropagation" one can construct the neural network as the best approximator of the function $f(x)$ in the form of

$$N_n^{(1)}(x) = \sum_{i=0}^{n} \omega_i^{(1)} \cdot \varphi(\xi_i^{(1)} \cdot x - \theta_i^{(1)})\,, \tag{7}$$

where $\omega_i^{(1)}, \xi_i^{(1)}, \theta_i^{(1)}$ are net parameters optimized on the points of system (5). Taking in response that $\left| f\left(x_j^{(1)}\right) - N_n^{(1)}\left(x_j^{(1)}\right)\right| = const,\ j = \overline{0, n+1}$ let's suppose that

$$f(x) - N_n^{(1)}(x) = r_n^{(1)}(x),\ \left|\mathrm{r}_\mathrm{n}^{(1)}\left(x_j^{(1)}\right)\right| = \underline{E}_n^{(1)},\ \left\|r_n^{(1)}(x)\right\| = \max_{a \le x \le b}\left| f(x) - N_n^{(1)}(x)\right| = \overline{E}_n^{(1)}. \tag{8}$$

As is known, size $E_n(f)$ of the best approximation of the function $f(x)$ on all segment $[a,b]$ always not less the best approximation $\underline{E}_n^{(1)}$ of this function on the system (1), i.e. $\underline{E}_n^{(1)} \le E_n(f)$. On the other hand, if it has appeared that neural network would be the best function approximator, and this process by that would be finished. Therefore, let's believe that

$$\underline{E}_n^{(1)} \le E_n(f) < \overline{E}_n^{(1)} . \tag{9}$$

Further it is necessary replace system of points (5) by system

$$a \le x_0^{(2)} < x_1^{(2)} < \ldots\ldots < x_n^{(2)} < x_{n+1}^{(2)} \le b \tag{5'}$$

so that following conditions were satisfied:

$$sign\, \mathrm{r}_\mathrm{n}^{(1)}\left(x_{j+1}^{(2)}\right) = -sign\, \mathrm{r}_\mathrm{n}^{(1)}\left(x_j^{(2)}\right);\ \mathrm{r}_\mathrm{n}^{(1)}\left(x_j^{(2)}\right) \ge \underline{E}_n^{(1)};\ \max_\mathrm{j}\left|\mathrm{r}_\mathrm{n}^{(1)}\left(x_j^{(2)}\right)\right| = \overline{E}_n^{(1)} . \tag{10}$$

On the basis of points system (5') let's create training set $\left\{x_j^{(2)}, y_j^{(2)}\right\}_{j=0}^{n+1}$, where $y_j^{(2)} = f\left(x_j^{(2)}\right)$, and applying algorithm «error backpropagation» for function $r_n^{(1)}(x) = f(x) - N_n^{(1)}(x)$ one can construct the neural network $\pi_n^{(1)}(x)$ on system (5'). Then we can assume that

$$N_n^{(2)}(x) = N_n^{(1)}(x) + \pi_n^{(1)}(x);\ f(x) - N_n^{(2)}(x) = r_n^{(1)}(x) - \pi_n^{(1)}(x) = r_n^{(2)}(x);$$

$$\left|r_n^{(2)}\left(x_j^{(2)}\right)\right| = \underline{E}_n^{(2)},\ \left\|\mathrm{r}_\mathrm{n}^{(2)}(x)\right\| = \overline{E}_n^{(2)} . \tag{8'}$$

As well as in the previous case let's assume that $E_n(f) < \overline{E}_n^{(2)}$ (for in case of $E_n(f) = \overline{E}_n^{(2)}$ corresponding output $N_n^{(2)}(x)$ would be the best substitute for function $f(x)$), so

$$\underline{E}_n^{(2)} \le E_n(f) < \overline{E}_n^{(2)} . \tag{9'}$$

Similarly let's replace system (5') by system

$$a \le x_0^{(3)} < x_1^{(3)} < \ldots\ldots < x_n^{(3)} < x_{n+1}^{(3)} \le b \tag{5''}$$

so that at all $j = 0, 1, 2, \ldots, n+1$ following conditions are satisfied

$$sign\, \mathrm{r}_\mathrm{n}^{(2)}\left(x_{j+1}^{(3)}\right) = -sign\, \mathrm{r}_\mathrm{n}^{(2)}\left(x_j^{(3)}\right);\ \mathrm{r}_\mathrm{n}^{(2)}\left(x_j^{(3)}\right) \ge \underline{E}_n^{(2)};\ \max_\mathrm{j}\left|\mathrm{r}_\mathrm{n}^{(2)}\left(x_j^{(3)}\right)\right| = \overline{E}_n^{(2)} . \tag{10'}$$

After that it is necessary create the neural network inducing output $\pi_n^{(2)}(x)$ as the best approximator of the function $r_n^{(2)}(x) = f(x) - N_n^{(2)}(x)$ on points system (5'') and let's assume

$$N_n^{(3)}(x) = N_n^{(2)}(x) + \pi_n^{(2)}(x);\ f(x) - N_n^{(3)}(x) = r_n^{(2)}(x) - \pi_n^{(2)}(x) = r_n^{(3)}(x);$$

$$\left|r_n^{(3)}\left(x_j^{(3)}\right)\right| = \underline{E}_n^{(3)},\ \left\|\mathrm{r}_\mathrm{n}^{(3)}(x)\right\| = \overline{E}_n^{(3)} \tag{8''}$$

etc. while we shall not achieve for any $k$ realization of the condition $E_n(f) = \overline{E}_n^{(k)}$. All this technology of training of the chosen neural network with one hidden layer for approximation of continuous function $f(x)$ can be described by iterative ratio, which essence rather easily reveals on the scheme presented in fig. 1.

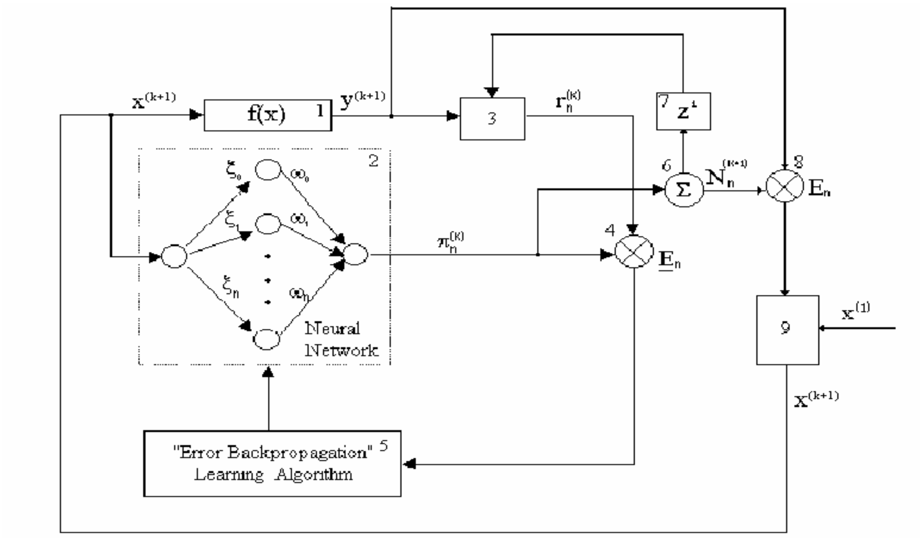


Fig. 1 The scheme of the combined training of neural network

Thus, for any $x \in [a,b]$ the optimal signal $N_n^{(k+1)}(x)$ being the best approximator for corresponding value of function $f(x)$ is formed in the output of the adder 6 (fig. 1) and is defined by

$$N_n^{(k+1)}(x) = N_n^{(k)}(x) + \pi_n^{(k)}(x) \tag{11}$$

with initial condition $N_n^{(0)}(x) = 0$. The output $\pi_n^{(k)}(x)$ is induced by the three-layer neural network 2 optimized by algorithm "error backpropagation" on the basis of the training set constructed on points of system

$$a \le x_0^{(k+1)} < x_1^{(k+1)} < \dots\dots < x_n^{(k+1)} < x_{n+1}^{(k+1)} \le b\,. \tag{12}$$

For each point of the given system should be satisfied following three conditions:

$$sign\, \mathrm{r}_\mathrm{n}^{(\mathrm{k})}\left(x_{j+1}^{(k+1)}\right) = -sign\, \mathrm{r}_\mathrm{n}^{(\mathrm{k})}\left(x_j^{(k+1)}\right);\ \mathrm{r}_\mathrm{n}^{(\mathrm{k})}\left(x_j^{(k+1)}\right) \ge \underline{E}_n^{(k)};\ \max_{\mathrm{j}} \left|\mathrm{r}_\mathrm{n}^{(\mathrm{k})}\left(x_j^{(k+1)}\right)\right| = \overline{E}_n^{(k)}, \tag{13}$$

where

$$f(x) - N_n^{(k)}(x) = r_n^{(k)}(x),\ \left|\mathrm{r}_\mathrm{n}^{(\mathrm{k})}\left(x_j^{(k)}\right)\right| = \underline{E}_n^{(k)},$$

$$\left\|r_n^{(k)}(x)\right\| = \max_{a \le x \le b} \left| f(x) - N_n^{(k)}(x) \right| = \overline{E}_n^{(k)}. \tag{14}$$

In more detail let's consider the scheme of the neural network combined training. Signal $r_n^{(k)}(x)$ is formed in the output of the block 3 and after traditional training through the comparison block 4 and the training block 5 the neural network 2 approximates it on points system $\left\{x_j^{(k+1)}\right\}_{j=0}^{n+1}$. After checking in the comparison block 8 conditions $\underline{E}_n^{(k)} \le E_n(f) < \overline{E}_n^{(k)}$

through the block 9 replacement of points system $\{x_j^{(k+1)}\}_{j=0}^{n+1}$ by system $\{x_j^{(k+2)}\}_{j=0}^{n+1}$ is carried out so that conditions (13) were satisfied. To satisfy to these conditions, it is enough to replace in system $\{x_j^{(k+1)}\}_{j=0}^{n+1}$ one point by point $x^* \in [a,b]$ in which $|\ r_n^{(k+1)}(x^*)\ | = \overline{E}_n^{(k+1)}$ (presence of such point is caused by known Veyershtrass's theorem), and all the others to leave not changed and obtained system to accept as system $\{x_j^{(k+2)}\}_{j=0}^{n+1}$.

Process of replacement can be carried out as follows. If the point $x^*$ is between two points $x_j^{(k+1)}$ and $x_{j+1}^{(k+1)}$ of system $\{x_j^{(k+1)}\}_{j=0}^{n+1}$, then by means of $x^*$ it is necessary to replace that from its in which the difference $r_n^{(k+1)}(x)$ has the same sign as in the point $x^*$. If the point $x^*$ is at the left of all points from system $\{x_j^{(k+1)}\}_{j=0}^{n+1}$ and $sign\, \mathrm{r}_\mathrm{n}^{(\mathrm{k}+1)}(x^*) = sign\, \mathrm{r}_\mathrm{n}^{(\mathrm{k}+1)}(x_0^{(k+1)})$, then it is necessary to replace the point $x_0^{(k+1)}$ by; if thus $sign\, \mathrm{r}_\mathrm{n}^{(\mathrm{k}+1)}(x^*) = -sign\, \mathrm{r}_\mathrm{n}^{(\mathrm{k}+1)}(x_0^{(k+1)})$, then as system $\{x_j^{(k+2)}\}_{j=0}^{n+1}$ it is necessary to select the points system $x^*, \mathrm{x}_0^{(\mathrm{k}+1)}, \mathrm{x}_1^{(\mathrm{k}+1)}, ..., \mathrm{x}_\mathrm{n}^{(\mathrm{k}+1)}$.

In case of the point $x^*$ is located at the right of all points of system $\{x_j^{(k+1)}\}_{j=0}^{n+1}$, it is necessary to act similarly. Notice that in practice it is desirable to replace on more points of system $\{x_j^{(k+1)}\}_{j=0}^{n+1}$ by new points (one of which is $x^*$) so that thus, first, all conditions (13) were satisfied and, secondly, that values $\left|r_n^{(k+1)}(x_j^{(k+2)})\right|$ are whenever possible greater.

## 3. Approximation of continuous functions with using of neural approximation kernel

Earlier we have established the concept of neural approximation kernel which under the characteristics creates "favourable" approximation environment for the best approximate of continuous functions by feedforward neural networks with single nonlinear hidden layer. Rationality of given approach let's consider on an example of approximation of the function depending on one variable.

For construction the neural approximation kernel let's choose a neural network with one nonlinear hidden layer (Fig. 2), where all output connections weights are equal to 1, and other parameters ($\xi_i$ $(i = \overline{0,n})$ – input connections weights, $\theta_i$ $(i = \overline{0,n})$ – thresholds nonlinear neurons from the hidden layer), are select as any real numbers. As is known, all polynomial approximation kernels by means of which, as a rule, are proved direct approximation theorems of the functions theory, at any argument are identically equal 1. As an example of such kernels one can to point to the binomial series

$$\sum_{i-0}^{n} C_n^k \cdot x^{n-k} \cdot (1-x)^k \equiv 1,\ x \in [0,1] \tag{15}$$

on the basis of which for continuous function is formulated Bernstein's approximation polynomial [4]

$$B_n(x) = \sum_{i-0}^{n} C_n^k \cdot x^{n-k} \cdot (1-x)^k \cdot f\left(\frac{i}{n}\right). \tag{16}$$

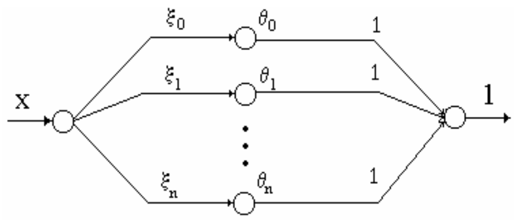

Fig. 2 Neural Network approximation kernel

Understanding that for any input signal it is practically impossible to obtain the signal 1 in the neural network output let's act as follows.

Start from the points system (5) and constructed on its basis the training set $\left\{x_j^{(1)},1\right\}_{j=0}^{n+1}$ let's train the given neural network by the scheme offered in fig. 1. As a result of training one can determine such optimum values $\xi_i^*, \theta_i^*$ $(\mathrm{i} = \overline{0,\mathrm{n}})$ for which there is an infinitesimal $\varepsilon_1 > 0$ such as for any $x \in [a,b]$ will be fulfil the inequality

$$\left|\ net_n^*(x) - 1\ \right| < \varepsilon_1, \tag{17}$$

where

$$net_n^*(x) = \sum_{i=0}^{n} \varphi(\xi_i^* \cdot x - \theta_i^*) \tag{18}$$

is the polynomial induced by neural network with sigmoid activations functions into the nonlinear neurons from the hidden layer type of

$$\varphi_i(x) = \frac{1}{1 - e^{-(\xi_i \cdot x - \theta_i)}}. \tag{19}$$

Obtained neural network with parameters $\xi_i^*, \theta_i^*$ $(i = \overline{0,n})$ we shall consider as a neural approximation kernel. On its basis and replacement the unity weights of output connections by any real numbers $\omega_i$ $(i = \overline{0,n})$ is possible to construct the neural network capable to approximate any continuous function offered in a tabular kind. Advantage of the given approach consists that training this network is limited to training only weights of output connections $\omega_i$ $(i = \overline{0,n})$ at presence of in advance optimized other parameters creating "favourable approximation environment". Really, it is visible from following reasoning.

Let continuous function $f(x)$ and neural approximation kernel (18) is set on the segment $[a,b]$. Then for the neural network type of

$$\sum_{i=0}^{n} \omega_i \cdot \varphi(\xi_i^* \cdot x - \theta_i^*), \tag{20}$$

using an inequality (13), limitation of sigmoid functions from above by 1 and believing $M = \max_{x \in [a,b]} |f(x)|$, we have:

$$\left| f(x) - \sum_{i=0}^{n} \omega_i \cdot \varphi\left(\xi_i^* \cdot x - \theta_i^*\right) \right| \le \left| f(x) - \sum_{i=0}^{n} f(x) \cdot \varphi\left(\xi_i^* \cdot x - \theta_i^*\right) \right| +$$

$$+ \left| \sum_{i=0}^{n} f(x) \cdot \varphi\left(\xi_i^* \cdot x - \theta_i^*\right) - \sum_{i=0}^{n} \omega_i \cdot \varphi\left(\xi_i^* \cdot x - \theta_i^*\right) \right| \le$$

$$\le |f(x)| \cdot \left| 1 - \sum_{i=0}^{n} \varphi\left(\xi_i^* \cdot x - \theta_i^*\right) \right| + \sum_{i=0}^{n} |f(x) - \omega_i| \cdot \left| \varphi\left(\xi_i^* \cdot x - \theta_i^*\right) \right| \le M \cdot \varepsilon_1 + \sum_{i=0}^{n} |f(x) - \omega_i| .$$

Thus, to obtain performance of Veyershtrass's approximation theorem for the neural network (20) it is enough to optimize weights $\omega_i$ $(i = \overline{0,n})$ so that $\forall x \in [a,b]$ it is carried out

$$\sum_{i=0}^{n} |f(x) - \omega_i| \to 0 . \tag{21}$$

In the conclusion let's note that optimization of weights $\omega_i$ $(i = \overline{0,n})$ can be made both traditional ways (with application "error backpropagation") and using the scheme of training as indicated above.

## References


[1] AHIEZER N.I. *Lektsije po teorii approximatsii*. Nauka, 1965 (in Russian).

[2] CHEN, H., CHEN, T. *Approximations of Continuous Functionals by Neural Networks with Application to Dynamic Systems*. IEEE Transactions on Neural Networks, vol.4, No.6, November 1993, pp. 910-918.

[3] DZJADYK V.K. *Vvedenije v teoriju ravnomernoqo priblijenija funktsiy polinomami,.* Nauka, 1977, 512 str. (in Russian).

[4] FUNAHASHI, K. *On the approximate realization of continuous mappings by neural networks.* Neural Networks, vol.2, pp. 183-192, 1989.