***Journal of Intelligent Learning Systems and Applications*, 2011, 3, 17-25**
doi:10.4236/jilsa.2011.31003 Published Online February 2011 (http://www.SciRP.org/journal/jilsa)

Function Approximation Using Robust Radial Basis Function Networks

Oleg Rudenko, Oleksandr Bezsonov

The Department of Computer Engineering and Control, Kharkov National University of Radio Electronics, Kharkiv, Ukraine.
Email: o.bezsonov@gmail.com

Received April 9th, 2010; revised July 20th, 2010; accepted August 4th, 2010

ABSTRACT

Resistant training in radial basis function (RBF) networks is the topic of this paper. In this paper, one modification of Gauss-Newton training algorithm based on the theory of robust regression for dealing with outliers in the framework of function approximation, system identification and control is proposed. This modification combines the numerical robustness of a particular class of non-quadratic estimators known as M-estimators in Statistics and dead-zone. The algorithms is tested on some examples, and the results show that the proposed algorithm not only eliminates the influence of the outliers but has better convergence rate then the standard Gauss-Newton algorithm.

Keywords: *Neural Network, Robust Training, Basis Function, Dead Zone*

1. Introduction

Function approximation involves estimating (approximating) the underlying relationship from a given finite input-output data set

$$y(x) = f(x) + \xi \tag{1}$$

where $x \in R^{M\times 1}$ is an input vector; $f(\cdot)$ is the arbitrary nonlinear function, unknown in the general case; ξ is the unobserved disturbance with unknown characteristics; has been the fundamental problem for a variety of applications in system identification, pattern classification, data mining and signal reconstruction [1-4].

Feedforward neural networks such as multilayer perceptrons (MLP) have been widely used as an approach to function approximation since they provide a generic black-box functional representation and have been shown to be capable of approximating any continuous function defined on a compact set in R^N with arbitrary accuracy [5-7].

It has been proved that a radial basis function network (RBF) can approximate arbitrarily well any multivariate continuous function on a compact domain if a sufficient number of radial basis function units are given [8].

In contrast to MLPs, RBF networks use a localized representation of information. The RBF network requires less computation time for the learning and more compact topology than MLP. The network can be configured with one radial basis function centre at each training data point. Thus, the complexity of the network is of the same order as the dimensionality of the training data and the network has a poor generalization capability. The RBF decomposition of $f(x)$ is

$$\hat{f}(x) = \sum_{i=0}^{N} w_i \varphi_i(x,r) = w^T \varphi(x,r) \tag{2}$$

where $w \in R^{N\times 1}$ is a vector of linear weights, $\varphi \in R^{N\times 1}$ is a vector of RBFs and r is a distance.

An important advantage of RBFN from viewpoint of practitioners is, therefore, clear and understandable interpretation of the functionality of basis functions.

The traditional RBF basis function is defined by Euclidian distance $r_E = \|x_i - t_j\|$ and Gaussian activation function by $\varphi_j(x_i) = \exp\{-0.5 r_E^2 \sigma^{-2}\}$, where x_i is the input sample number *i*, t_j is the center of *j*-th radial basis function (radii), σ is the standard deviation. If we use the Mahalanobis distance $r_M = (x_i - t_j)^T R^{-1} (x_i - t_j)$ where $R^{-1} = [r_{ij}^k]$ is weight matrix, *M* is the dimension of input vector x_i, *N* is the number of neurons, for the RBF activation function we have

$$\Phi_j(x) = \exp\left[-(x - t_j)^T R_j^{-1} (x - t_j)\right] \tag{3}$$

where R_j is the covariance matrix. Geometrically t_j represents the center and R_j the shape of the *j*-th basis function. A hidden unit function can be represented as a hyper-ellipsoid in the N-dimensional space.

All the network parameters (weights, centers and radii) may be determined using various learning algorithms have been used in order to find the most appropriate parameters for the RBF decomposition.

A network iteratively adjusts parameters of each node by minimizing some cost function which can be defined as an ensemble average errors.

$$F\left[e(k,\theta)\right] = \frac{1}{k}\sum_{i=1}^{k}\rho\left(e(i,\theta)\right) \qquad (4)$$

where $\rho\left(e(i,\theta)\right)$ is a scalar loss function;

$e(i,\theta) = y(i) - \hat{f}(i)$ represents the residual error between the desired $y(i)$, and the actual network outputs, $\hat{f}(i)$; *i*—indicates the index of the series; θ comprises all the unknown parameters of the network,

$$\theta(k) = \left[c^0, c^1, t_1^1, \cdots, t_M^1, r_{1,1}^1, \cdots, r_{M,M}^1, \cdots, c^N, t_1^N, \cdots, t_M^N, r_{1,1}^N, \cdots, r_{M,M}^N\right]^T.$$

The problem of neural network training (estimating θ) which approximates the function (1) "well", has essentially been tackled, based on the following two different assumptions [9]:

(A1) The noise has some probabilistic and/or statistical properties.

(A2) Regardless of the disturbance nature, a noise bound is available, *i.e.* $\xi_k^2 \le \delta_k^2$.

Assumption (A1) leads to different stochastic training methods that are based on minimization of some loss function. Different choices of loss functions arise from various assumptions about the distribution of the noise in measurement. The most common loss function is the quadratic function corresponding to a Gaussian noise model with zero mean, and a standard deviation that does not depend on the inputs. The Gaussian loss function is used popularly as it has nice analytical properties. However, one of the potential difficulties of the standard quadratic loss function is that it receives the large contributions from outliers that have particularly large errors. The problems in the neural network training are that when the training data sets contain outliers, traditional supervised learning algorithms usually cannot come up acceptable performance. Since traditional training algorithms also adopt the least-square cost function (4), those algorithms are very sensitive to outliers.

Techniques that attempt to solve these problems are referred to robust statistics [10,11]. In recent years, various robust learning algorithms based on M-estimation have been proposed to overcome the outlier's problems [12-17].

The basic idea of M-estimators is to replace the quadratic function in the cost function (4) by the loss function so that effect of those outliers may be degraded.

Traditional approaches of solving such a problem are to introduce a robust cost function (4), and then, a steepest descent approach is applied. The idea of such an approach is to identify outliers and then to reduce the effect of outliers directly.

Alternative approaches have been formulated in a deterministic framework based on Assumption (A2). In this context the training problem is then to find a θ belonging to the class of models (2) for which the absolute value of the difference between the function (1) and model is smaller than $|\delta_k|$ for all times k.

Three different types of solutions to this problem have mainly been explored in literature. The first method is to formulate the estimation problem in a geometrical setting. Different proposals result from this approach but Fogel and Huang [18] proposed a minimal volume recursive algorithm (FHMV) which minimizes the size of an ellipsoid and was attractive for on-line estimation.

The second alternative is to derive estimation algorithm for stability consideration together with the geometrical (ellipsoidal-outer-bounding algorithm by Lozano-Leal and Ortega) [19]. The third approach is to obtain estimation (training) algorithm from modifying the exponentially weighted recursive least squares algorithm (EW-RLS) [9].

All these algorithms have a dead zone. The dead zone scheme guarantees convergence of the neural network training algorithm in the present of noise $\xi_k^2 \le \delta_k^2$.

It should be noted that this dead zone may serve as value that limits the accuracy of the obtained solutions, *i.e.* determines its acceptable inaccuracy.

The proposed method combines the numerical robustness of a particular class of non-quadratic M-estimators and dead-zone.

2. Robust Gauss-Newton Training Algorithm

The estimation $\hat{\theta}$ is the solution of the following set of equations

$$\frac{\partial F(e)}{\partial \theta_j} = \frac{1}{k}\sum_{i=1}^{k}\psi\left(e(i,\theta)\right)\frac{\partial e(i,\theta)}{\partial \theta_j} = 0 \qquad (5)$$

where $\psi\left(e(i,\theta)\right) = \dfrac{\partial \rho\left(e(i,\theta)\right)}{\partial e(i,\theta)} = \omega\left(e(i),\theta\right)e(i,\theta)$ is the influence function and $\omega\left(e(i),\theta\right)$ is the weight function.

For quadratic function $\rho\left(e(i),\theta\right)$ in the maximum likehood estimation case (5) has a closed form solution, the sample mean. The sample mean is substantially affected by the presence of outliers.

For many non-quadratic loss functions, the Equation (5) does not have closed form solution, but can be solved by the some iterative or recursive methods.

The minimization of the criterion (4) can therefore be performed using Gauss-Newton algorithm.

$$\hat{\theta}(k)=\hat{\theta}(k-1)+\frac{P(k-1)\nabla\hat{f}(k)\psi(e(k))}{1+\psi'(e(k))\nabla^T\hat{f}(k)P(k-1)\nabla\hat{f}(k)} \quad (6)$$

$$P(k)=P(k-1)- -\frac{P(k-1)\nabla\hat{f}(k)\nabla^T\hat{f}(k)P(k-1)}{1+\psi'(e(k))\nabla^T\hat{f}(k)P(k-1)\nabla\hat{f}(k)}\psi'(e(k)) \quad (7)$$

where

$$\nabla\hat{f}(k)=\left[\frac{\partial\hat{f}(k)}{\partial c^0},\frac{\partial\hat{f}(k)}{\partial c^1},\frac{\partial\hat{f}(k)}{\partial t^1},\frac{\partial\hat{f}(k)}{\partial r^1_{1,1}},\frac{\partial\hat{f}(k)}{\partial r^1_{1,2}},\cdots,\right.$$
$$\left.\frac{\partial\hat{f}(k)}{\partial r^1_{1,M}},\cdots,\frac{\partial\hat{f}(k)}{\partial r^1_{M,M}},\cdots,\frac{\partial\hat{f}(k)}{\partial c^N},\frac{\partial\hat{f}(k)}{\partial t^N},\frac{\partial\hat{f}(k)}{\partial r^{1N}_{1,1}},\cdots,\frac{\partial\hat{f}(k)}{\partial r^N_{M,M}}\right]^T$$

$$\frac{\partial\hat{f}(k)}{\partial c^0}=1 \quad \frac{\partial\hat{f}(k)}{\partial c_i}=\Phi_i(x,t_i,R_i)$$

$$\frac{\partial\hat{f}(k)}{\partial t_i}=-c_i\frac{\partial A}{\partial t_i}e^{-A} \quad \frac{\partial\hat{f}(k)}{\partial r^{ij}_m}=-c_i\frac{\partial A}{\partial r^{ij}_m}e^{-A}$$

with $A=(x-t_j)^T R^{-1}(x-t_j)$.

The initial value of the matrix $P(0)$ is chosen as in the recursive MLS (RMLS), *i.e.* $P(0)=\lambda I$, where $\lambda>>1$. and the initial dimension of the identity matrix I is given as $S\times S$, where $S=1+(M^2+M+1)$ is number of adjustable parameters of a network containing 1 neuron. Because after the introduction into the network a new *n*-th neuron the dimension of the $P(k)$ increases, the values of elements in matrix $P(k)$ are reset and initialized again, then S becomes equal to $S=1+n(M^2+M+1)$, where n—the current number of neurons in the network.

The influence function $\psi(e)$ measures the influence of a datum on the value of the parameter estimate. For example, for the least-squares with $\rho(e)=0.5e^2$, the influence function is $\psi(e)=e$, that is, the influence of a datum on the estimate increases linearly with the size of its error and without bound, which confirms the non-robustness of the least-squares estimate.

Huber proposed a robust estimator so-called an M-estimator, M for maximum likelihood. M-estimator is the solution of (5) where different non-quadratic loss function $\rho(e(i),\theta)$ are used.

Following Huber [10], a distribution of the noise contaminated by outliers expressed by a mixture of two probability density functions

$$p(x)=(1-\varepsilon)p_0(x)+\varepsilon q(x) \quad (8)$$

where $p_0(x)$ is the density of basic distribution of a measurement noise; $q(x)$ is the density of a distribution of outliers; $\varepsilon\in[0,1]$ is the probability of occurring a large error.

Even if the basic $p_0(x)$ and contaminating $q(x)$ distributions are Gaussian with zero mean and variances σ_1^2 and σ_2^2, $\sigma_1^2<<\sigma_2^2$ hence, than optimal for the Gaussian distribution estimations (6)-(7), obtained by choosing $\rho(e)=0.5e^2$, will be unstable.

The density distribution p^* for ε—contaminated probability distributions (8), which gives the minimum Fisher information, contains a central region $p=(1-\varepsilon)p_0(\xi)$ and tails with exponentially decreasing density $p_0(\xi)=ce^{-\lambda|x|}$. Usage of these distributions allowed to obtain nonlinear robust Maximum likelihood estimates, that are workable for almost all the noise distributions. This algorithm combines the conventional least mean square (LMS) if $|e(k)|\le 3\sigma_1^2$ and least absolute deviation (LAD) if $|e(k)|>3\sigma_1^2$ stochastic gradient algorithms and called the mixed-norm LMS algorithm [10,20,21].

On the other hand, the choice of loss function, different from the quadratic, ensures the robustness of estimates, *i.e.* their workability for almost all distributions of noises. Currently, there are many such functions $\rho(e)$, however, keeping in mind that $\rho''(e(k))=\psi'(e(k))$ is used in the learning algorithm (6)-(7) it is advisable to choose such functions $\rho(e(k))$, which have nonzero second derivatives. As these functions can be taken, for example [22,23],

$$F_1[e(k)]=\left(\frac{e^2(k)}{2c}\right) \quad (9)$$

$$F_2[e(k)]=c\ln\cosh\left(\frac{e(k)}{c}\right) \quad (10)$$

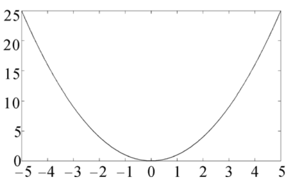
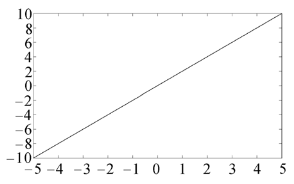
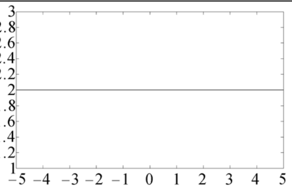
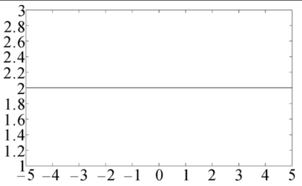
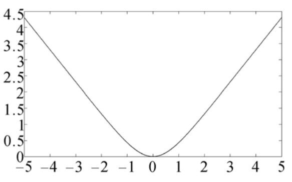
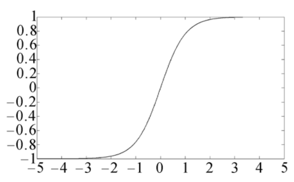
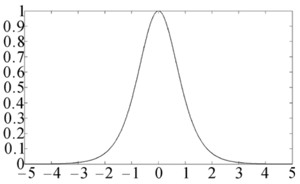
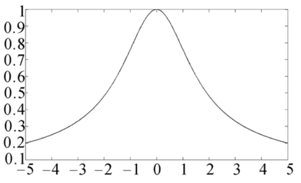
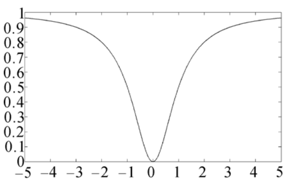
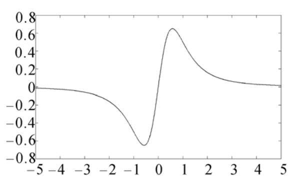
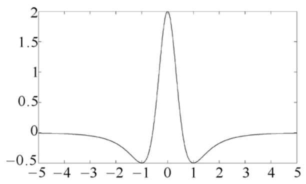
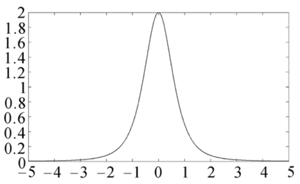
$$F_3[e(k)]=\frac{e^2(k)}{c^2+e^2(k)} \quad (11)$$

graphs of which are shown in **Table 1**.

It should be noted that in case of using functionals as (9) and (10) a problem of the selection (evaluation) of parameter *c* (in **Table 1** shapes of the functionals with *c* = 5 are shown) arises.

The standard deviations σ_1^2 and σ_2^2 in (8) are usually unknown and must be estimated and they can be

Table 1. Graphs of functions (9)-(11), their first and second derivatives and weight functions.

$F\left[e(k)\right]$	$\psi\left(e(k)\right)$	$\psi'\left(e(k)\right)$	$\omega\left(e(k)\right)$
			
$F_1\left[e(k)\right]$			
			
$F_2\left[e(k)\right]$			
			
$F_3\left[e(k)\right]$			

taken into account in the learning algorithm. If σ_1^2 and σ_2^2 do not change over time, this evaluation can be carried out by stochastic approximation:

$$\hat{\sigma}_1^2(k)=\begin{cases}\hat{\sigma}_1^2(k-1)+\dfrac{1}{l_1(k)}\left(e^2(k)-\hat{\sigma}_1^2(k-1)\right)\\ \text{for}\left|e(k)\right|\le 3\hat{\sigma}_1(k-1)\\ \hat{\sigma}_1^2(k-1)\ \text{otherwise}\end{cases} \tag{12}$$

$$\hat{\sigma}_2^2(k)=\begin{cases}\hat{\sigma}_2^2(k-1)+\dfrac{1}{l_2(k)}\left(e^2(k)-\hat{\sigma}_2^2(k-1)\right)\\ \text{for}\left|e(k)\right|> 3\hat{\sigma}_1(k-1)\\ \hat{\sigma}_2^2(k-1)\ \text{otherwise}\end{cases}$$

where

$$l_1(k)=k-l_2(k)$$

$$l_2(k)=\begin{cases}0\ for\left|e(k)\right|\le 3\hat{\sigma}_1(k-1)\\ l_2(k-1)+1\ \text{otherwise}\end{cases}$$

The total variance of noise, calculated as

$$\sigma^2(k)=\begin{cases}\hat{\sigma}_1^2(k)\ \text{for}\left|e(k)\right|\le 3\hat{\sigma}_1(k-1)\\ \hat{\sigma}_2^2(k)\ \text{otherwise}\end{cases} \tag{13}$$

can be used for normalizing the selected functional

$$\rho\left(e^*\left(k,\theta,\sigma^2\right)\right)=\rho\left(\frac{e(k,\theta)}{\sigma^2}\right) \tag{14}$$

It should be noted that as estimation of the parameter *c* in the functionals (9) and (10) 3σ can be used.

3. Modification of Robust Gauss-Newton Algorithm with Dead Zone

Dead zone, which determines the degree of permissible errors, can be set as follows:

$$\psi_1'\left(e^*(k),\delta\right)=\begin{cases}\psi'\left(e^*(k)\right)\ \text{for}\ \left|e^*(k)\right|>\delta\\ 0\ \text{for}\ \left|e^*(k)\right|\le\delta\end{cases} \tag{15}$$

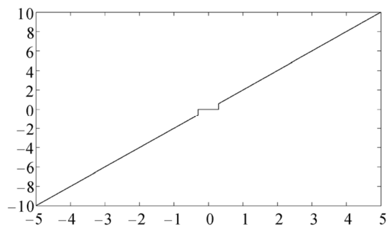
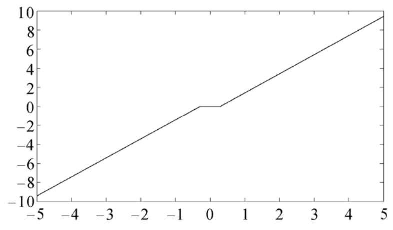
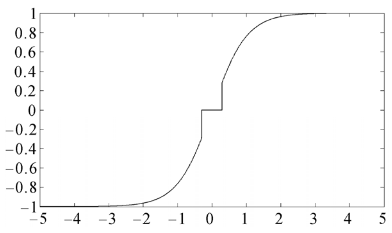
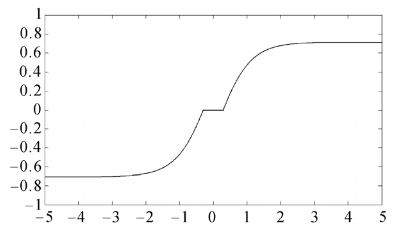
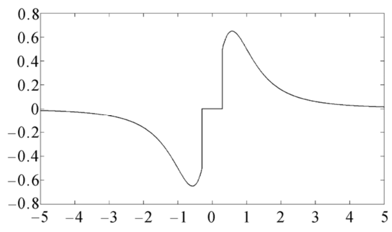
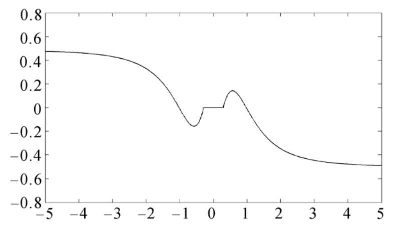
and

$$\psi_2'\left(e^*(k),\delta\right)=\begin{cases}\psi'\left(e^*(k)\right)-\delta\ \text{for}\ e^*(k)>\delta\\ 0\ \text{for}\ \left|e^*(k)\right|\le\delta\\ \psi'\left(e^*(k)\right)+\delta\ \text{for}\ e^*(k)<-\delta.\end{cases} \tag{16}$$

The forms of functions (12) and (13) are shown in the **Table 2** (columns 2 and 3, respectively).

In this case, the robust Gauss-Newton algorithm takes the form

Table 2. Graphs of (9)-(11) functions derivatives with dead zones.

$F[e(k)]$	$\psi_1(e^*(k),\delta)$	$\psi_2(e^*(k),\delta)$
$F_1[e(k)]$		
$F_2[e(k)]$		
$F_3[e(k)]$		

$$\hat\theta(k)=\hat\theta(k-1)+\frac{\alpha P(k-1)\nabla\hat f(k)\psi\left(e^*(k),\delta\right)}{1+\psi'\left(e^*(k),\delta\right)\nabla^T\hat f(k)P(k-1)\nabla\hat f(k)} \tag{17}$$

$$P(k)=P(k-1)-\frac{\alpha P(k-1)\nabla\hat f(k)\nabla^T\hat f(k)P(k-1)}{1+\psi'\left(e^*(k),\delta\right)\nabla^T\hat f(k)P(k-1)\nabla\hat f(k)}\psi'\left(e^*(k),\delta\right) \tag{18}$$

where

$$\psi\left(e^*(k),\delta\right)=\begin{cases}\psi\left(e^*(k)\right)-\delta \text{ for } e^*(k)>\delta\\ 0 \text{ for } \left|e^*(k)\right|\le\delta\\ \psi\left(e^*(k)\right)+\delta \text{ for } e^*(k)<-\delta\end{cases}$$

$$\alpha=\begin{cases}1 \text{ for } \left|e^*(k)\right|>\delta\\ 0 \text{ otherwise}\end{cases}$$

Table 1 (column 3) shows that for the functional (11) there are areas where $\psi'<0$. This can lead to instability of estimates $\hat\theta$. In this case, in the algorithm (17), (18) instead of $\psi'\left(e^*(k),\delta\right)$ the weighting function $\omega\left(e^*(k),\delta\right)$ should be used, which, as seen from the **Table 1** (column 4) is always greater than zero. In this case, algorithm (17, 18) takes the form

$$\hat\theta(k)=\hat\theta(k-1)+\frac{\alpha P(k-1)\nabla\hat f(k)\psi\left(e^*(k),\delta\right)}{1+\omega\left(e^*(k),\delta\right)\nabla^T\hat f(k)P(k-1)\nabla\hat f(k)} \tag{19}$$

$$P(k)=P(k-1)-\frac{\alpha P(k-1)\nabla\hat f(k)\nabla^T\hat f(k)P(k-1)}{1+\omega\left(e^*(k),\delta\right)\nabla^T\hat f(k)P(k-1)\nabla\hat f(k)}\omega\left(e^*(k),\delta\right) \tag{20}$$

4. Experimental Results

Consider using an RBF network to approximate the function [24]

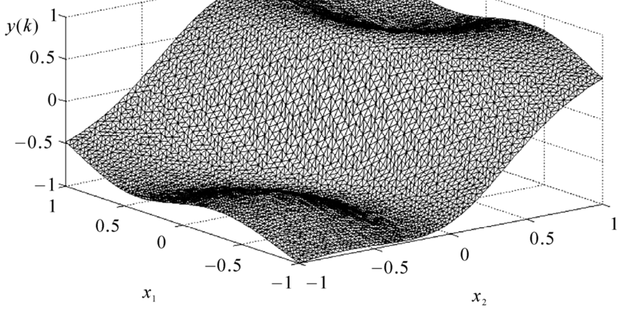
$$y(k)=0.725\sin\left(\frac{16x_1+8x_2}{3+4x_1^2+4x_2^2}\right)+0.2x_1+0.2x_2+\xi(k) \tag{21}$$

where $x=\left[x_1,x_2\right]^T$ is an input signal that was generated using uniformly distributed random data in range [–1, 1]. The additive noise $\xi(k)$ is a Gaussian mixture that smixes two types of noises, a large portion of normal noise with smaller variance and a smaller portion of noise with higher variance, *i.e.* $\xi(k)=(1-\varepsilon)q_1(k)+\varepsilon q_2(k)$, where $0<\varepsilon<0.2$ is a small number to denote the contamination ratio and ($q_1(k), q_2(k)$—normally distributed noises with variances σ_1^2 and σ_2^2 respectively). 50000 training data points were used for investigation of the given function. A surface described by function (21) without noise is shown on the **Figure 1(a)**, on the **Figure 1(b)** the same surface with noise $\xi(k)$ ($\sigma_1=0.6$ and $\sigma_2=12$) is shown. On the **Figure 2** the cross-sections of the function (21) are given (dashed line denotes the reconstructed function). The results of approximation of the function (21) with different values of ε, σ_1^2 and σ_2^2 are given in the **Table 3**. Here are the values of the RMS error, calculated after training the network for 2500 reference values using the formula

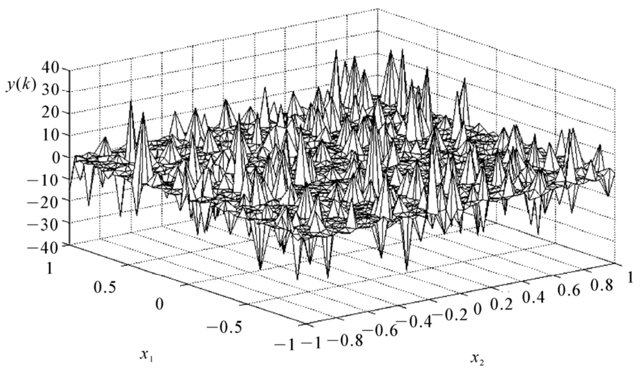
$$\varsigma=\frac{1}{2500}\sqrt{\sum_{i=1}^{2500}\left(y^*(i)-\hat{y}(i)\right)^2}$$

where y^*—the reference value of the output signal in the absence of interference measurements; $\hat{y}$—real networks output.

Graphs of the adjustment σ_1 and σ_2 estimations at



(a)



(b)

Figure 1. A surface described by function (21), (a) without noise $\xi(k)$; (b) with noise $\xi(k)$.

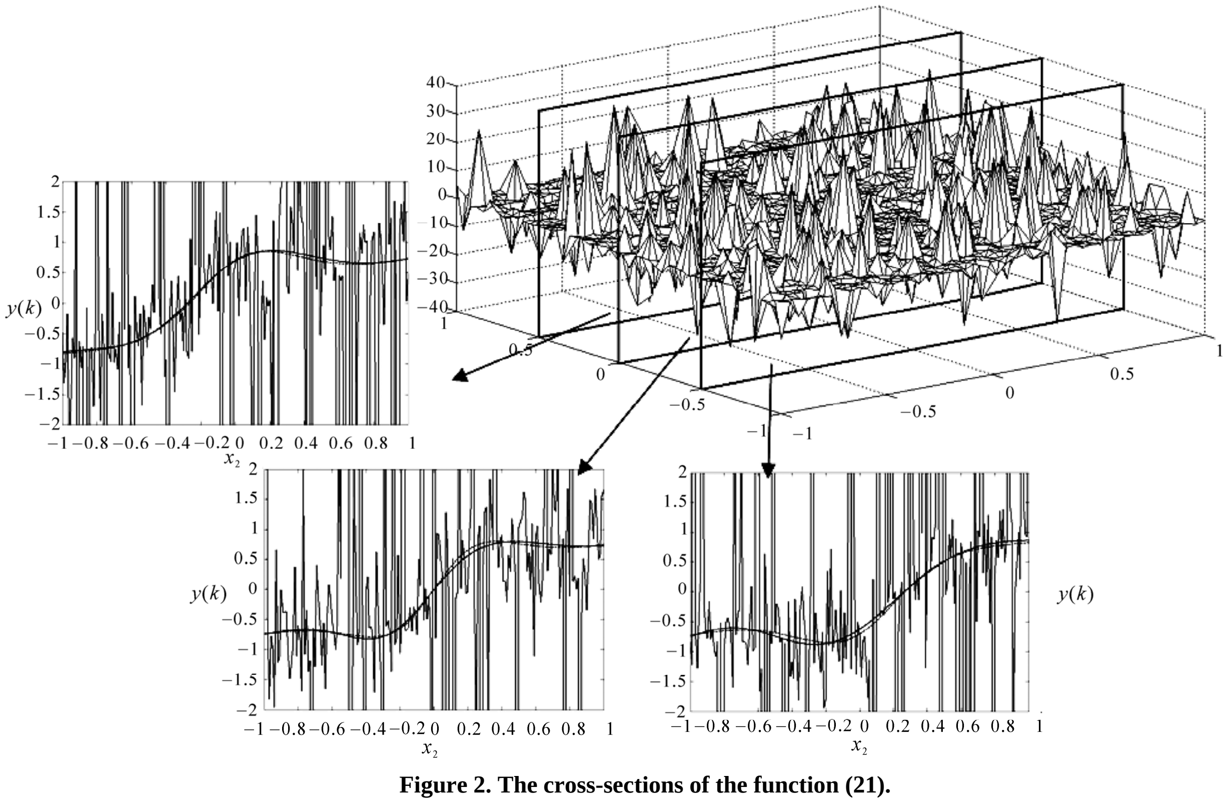


Figure 2. The cross-sections of the function (21).

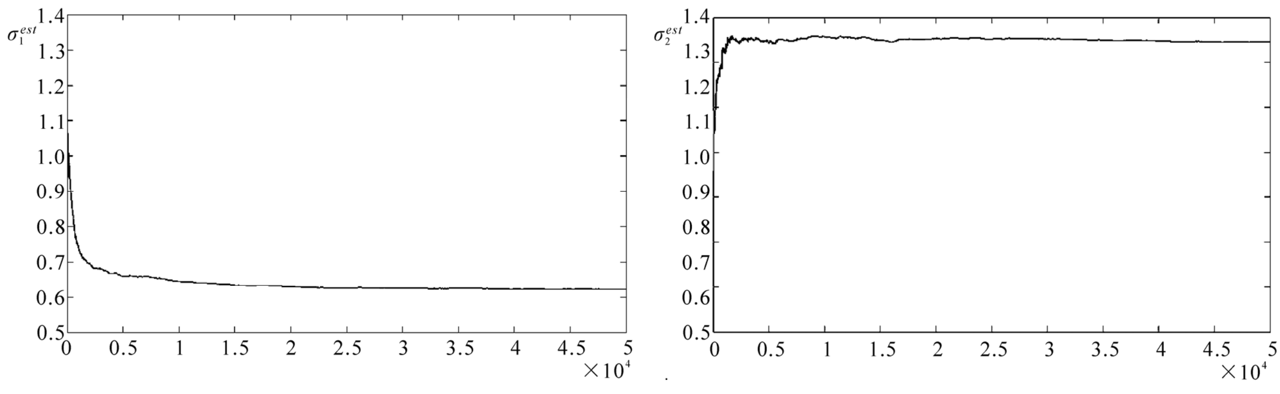


Figure 3. Results of the estimation $\sigma_1 = 0.6$ and $\sigma_2 = 12$ with $\varepsilon = 0.2$.

Table 3. The results of function (21) approximation.

Given parameters				$F[e(k)] = \left(\frac{e^2(k)}{2c}\right)$			$F[e(k)] = c\ln\cosh\left(\frac{e(k)}{c}\right)$			$F_3[e(k)] = \frac{e^2(k)}{1+e^2(k)}$ (with a weight function)		
ε	σ_1^{ref}	σ_2^{ref}	Number of outliers	ς without dead zone	ς With dead zone (15)	ς with dead zone (16)	ς without dead zone	ς with dead zone (15)	ς with dead zone (16)	ς without dead zone	ς with dead zone (15)	ς with dead zone (16)
0.0	0	0	0	0.6286	-	-	-	-	-	-	-	-
		3	5061	1.5252	2.7339	2.6047	1.5556	2.4468	2.3937	2.0836	2.8747	2.8137
0.1	0.6	6	5008	1.6415	2.4909	2.4697	1.6553	2.2052	2.2047	1.8936	2.7882	2.7199
		12	4991	1.9389	1.9634	1.9491	1.7256	1.7386	1.7379	1.6365	2.3665	2.3088
		3	10013	1.6497	2.1061	2.0698	2.3438	3.0111	2.9940	2.9080	2.9365	2.9198
0.2	0.6	6	10020	2.0402	2.1209	2.0813	2.2875	2.4361	2.4113	2.2054	2.7103	2.5998
		12	10111	1.9863	2.2117	2.1887	2.3682	2.7750	2.7217	2.5152	2.7012	2.6260

Table 4. Estimations of σ_1, σ_2 and N.

Given parameters				Estimations		
ε	σ_1^{ref}	σ_2^{ref}	Real number of outliers	σ_1^{est}	σ_2^{est}	Estimated number of outliers N
0.0	0	0	0	-	-	-
		3	5061	0.6369	4.0902	4758
0.1	0.6	6	5008	0.6166	6.7468	4984
		12	4991	0.6073	12.5611	4969
		3	10013	0.7351	4.3658	9957
0.2	0.6	6	10020	0.6151	6.8815	9897
		12	10111	0.6220	12.8381	10005

each step of training the network are shown in **Figure 3**. Estimations of σ_1, σ_2 and number of outliers are given in **Table 4**.

As seen from the simulation results, the algorithm (12) gives reasonably accurate estimates of σ_1^2 and σ_2^2 (assuming $\sigma_1^2 << \sigma_2^2$) that are used in the normalization of the loss function, which ensures high accuracy of approximation of very noisy nonlinear functions. Also it should be noted that the usage of dead zones has reduced training time by about 20%.

5. Conclusions

This paper proposes a resistant radial function network on-line training algorithm based on the theory of robust regression for dealing with outliers in the framework of function approximation.

The proposed algorithm minimizes an M-estimate cost functions instead of the conventional mean square error and represents one modification of recursive Gauss-Newton algorithm with dead-zone. These dead zone may serve as value that limits the accuracy of the obtained solutions.

Utilization of dead zones can decrease training time of the network.

If the distribution of the noise contaminated by outliers expressed by a mixture of two Gaussian distributions with unknown standard deviations σ_1^2 and σ_2^2, $\sigma_1^2 << \sigma_2^2$ these can be estimated and taken into account in the training algorithm.

It is an efficient algorithm for practical using in investigation of real nonlinear systems. It is expedient to develop this approach further and to investigate other robust cost functions and training algorithms such as Levenberg-Marquardt algorithm.

REFERENCES

[1] B. Kosko, "Neural Network for Signal Processing," Prentice-Hall Inc., New York, 1992.

[2] S. Haykin, "Neural Networks. A Comprehensive Foundation," 2nd Edition, Prentice Hall Inc., New York, 1999.

[3] C. M. Bishop, "Neural Network for Pattern Recognition," Clarendon Press, Oxford, 1995.

[4] H. Wang, G. P. Liu, C. J. Harris and M. Brown, "Advanced Adaptive Control," Pergamon, Oxford, 1995.

[5] R. Hecht-Nielsen, "Kolmogorov's Mapping Neural Networks Existence Theorem," *First IEEE International Conference on Neural Networks*, San Diego, Vol. 3, 1987, pp. 11-14.

[6] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, Vol. 2, No. 4, 1989, pp. 303-314. doi:10.1007/BF02551274

[7] T. Poggio and F. Girosi, "Networks for Approximation and Learning," *Proceeding of the IEEE*, Vol. 78, No. 9, 1990, pp. 1481-1497. doi:10.1109/5.58326

[8] J. Park and I. W. Sandberg, "Universal Approximation Using Radial-Basis-Function Network," *Neural Computation*, Vol. 3, No. 2, 1991, pp. 246-257. doi:10.1162/neco.1991.3.2.246

[9] C. C. de Wit and J. Carrillo, "A Modified EW-RLS Algorithms for Systems with Bounded Disturbances," *Automatica*, Vol. 26, No. 3, 1990, pp. 599-606. doi:10.1016/0005-1098(90)90032-D

[10] P. J. Huber, "Robust Statistics," John Wiley, New York, 1981. doi:10.1002/0471725250

[11] R. E. Frank, M. Hampel, M. Rohchetti and W. A. Stanel, "Robust Statistics: The Approach Based on Influence Functions," John Wiley & Sons Inc., Hoboken, 1986.

[12] C. C. Chang, J. T. Jeng and P. T. Lin, "Annealing Robust Radial Basis Function Networks for Function Approximation with Outliers," *Neurocomputing*, Vol. 56, 2004, pp. 123-139. doi:10.1016/S0925-2312(03)00436-3

[13] S.-C. Chan and Y.-X. Zou, "A Recursive Least M-Esimate Algorithm for Robust Filtering in Impulsive Noise: Fast Algorithm and Convergence Performance Analysis," *IEEE Transactions on Signal Processing*, Vol. 52, No. 4, 2004, pp. 975-991. doi:10.1109/TSP.2004.823496

[14] D. S. Pham and A. M. Zoubir, "A Sequential Algorithm for Robust Parameter Estimation," *IEEE Signal Processing Letters*, Vol. 12, No. 1, 2005, pp. 21-24. doi:10.1109/LSP.2004.839689

[15] J. Ni and Q. Soug, "Pruning Based Robust Backpropagation Training Algorithm for RBF Network Training Controller," *Intelligent and Robotic Systems*, Vol. 48, No. 3, 2007, pp. 375-396. doi:10.1007/s10846-006-9093-x

[16] G. Deng, "Sequential and Adaptive Learning Algorithms for M-Estimation," *EURASIP Journal on Advances in Signal Processing*, Vol. 2008, 2008, ID 459586.

[17] C.-C. Lee, Y.-C. Chiang, C.-Y. Shin and C.-L. Tsai, "Noisy Time Series Prediction Using M-Estimator Based Robust Radial Basis Function Network with Growing and Pruning Techniques," *Expert Systems with Applications*, Vol. 36, No. 3, 2008, pp. 4717-4724. doi:10.1016/j.eswa.2008.06.017

[18] E. Fogel and Y. E. Huang, "On the Value of Information in System Identification Bounded-Noise Case," *Automatica*, Vol. 18, No. 2, 1982, pp. 229-238. doi:10.1016/0005-1098(82)90110-8

[19] R. Lozano-Leal and R. Ortega, "Reformulation of the Parameter Identification Problem for Systems with Bounded Disturbances," *Automatica*, Vol. 23, No. 2, 1987, pp. 247-251. doi:10.1016/0005-1098(87)90100-2

[20] J. Chambers and A. Alvonitis, "A Robust Mixed-Norm Adaptive Filter Algorithm," *IEEE Signal Proceeding Letters*, Vol. 4, No. 22, 1997, pp. 46-48. doi:10.1109/97.554469

[21] Y. Zou, S. C. Chan and T. S. Ng, "A Recursive Least M-Estimate (RLM) Adaptive Filter for Robust Filtering in Impulse Noise," *IEEE Signal Proceeding Letters*, Vol. 7, No. 11, 2000, pp. 324-326. doi:10.1109/97.873571

[22] P. W. Holland and R. E. Welsh, "Robust Regression Using Iteratively Reweighted Least Squares," *Communications in Statistics-Theory Mathematics*, Vol. A6, 1977, pp. 813-827. doi:10.1080/03610927708827533

[23] S. Geman and D. McClure, "Statistical Methods for Tomographic Image Reconstruction," *Bulletin of the Inter-*

national Statistical Institut, Vol. L2, No. 4, 1987, pp. 4-5.

[24] K. S. Narendra and K. Parthasarathy, “Identification and Control of Dynamical Systems Using Neural Networks,” *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4-26. doi:10.1109/72.80202