JAST. Vol. 9, No. 2, pp 71-81

JAST
Journal of Aerospace Science and Technology

RESEARCH NOTE

Novel Radial Basis Function Neural Networks based on Probabilistic Evolutionary and Gaussian Mixture Model for Satellites Optimum Selection

M. Saraf[1] , M. R. Mosavi[2] and K. Mohammadi[3]

In this study, two novel learning algorithms have been applied on Radial Basis Function Neural Network (RBFNN) to approximate the functions with high non-linear order. The Probabilistic Evolutionary (PE) and Gaussian Mixture Model (GMM) techniques are proposed to significantly minimize the error functions. The main idea is concerning the various strategies to optimize the procedure of Gradient Descent (GD) in terms of the input feature vectors. The probability density of all feature vectors can help to optimize the learning rates of RBFNN by applying GMM. Another possibility is to utilize the Evolutionary Algorithms (EAs) to find the optimum solution. However, EAs often behave randomly which can't be mathematically controlled. So, a combined RBFNN based on novel PE algorithm has been proposed which has a soft behavior through the learning of non-linear function. The PE algorithm defines the occurrence probability of local minima in the space of extracted features as a Gaussian distribution correspondence to each chromosome. Then, it estimates the entire probabilities of local minima in an iterative procedure. These techniques have been utilized in the application of robust satellites subset selection. Geometric Dilution of Precision (GDOP) is the main factor to estimate the strength of goodness of each satellites subset. Then, the subset with the lowest value has been selected for improving the positioning performance, but it is so non-linear and imposes computational burden on navigation systems. These techniques have been implemented and the results on measured GPS data demonstrate that they significantly track the non-linearity of GPS GDOP comparison with the other conventional approaches.

***Keywords**: Neural Networks, Probabilistic Evolutionary, Gaussian Mixture, Satellites Selection.*

1 Introduction

Radial Basis Function Neural Network (RBFNN) is widely used in many fields such as data mining, artificial intelligence, pattern recognition and other non-linear approximation systems. The classical RBFNN defines the neurons with Gaussian distributions in order to map the input feature vectors to some sub-regions. It further uses a linear weighted vector to map that sub-regions to the RBFNN output. It has simple topological structure and universal approximation ability, but the important issue is the learning procedure in order to optimize the adjustable parameters including the center vectors, the variances and the linear output weights connecting the RBF hidden neurons to the output nodes [1]. The learning procedure of neural networks are utilized by several methods which have been developed such as Evolutionary Algorithms (EAs), Gradient Descent (GD) learning,etc. [2-4]. However, EAs suffer from the slow and premature convergence problems. GD algorithm is a common learning approach, but it is so sensitive to parameters initialization and learning rates. The procedure of learning is started with parameters initialization, then the resulted errors from the network and target values

1. M.Sc. Student, msaraf@elec.iust.ac.ir
2. (Corresponding Author), Professor, Department of Electrical Engineering, Iran University of Science and Technology, Narmak, Tehran 16846-13114, Iran, Email: m_mosavi@iust.ac.ir
3. Professor, mohammadi@iust.ac.ir

are computed and are propagated to the hidden layers [5-6]. Then according to these errors, the parameter updates are applied toward the negative of gradient in order to minimize the errors of learning. These procedures continue up to reaching the desired error values of network. The main issue for GA algorithm is the parameters initialization and the learning coefficients. Our first challenge in this study is why neurons must have the same strategies to learn all the feature vectors (dataset). It is better to design some intelligence procedures to distinguish the learning strategies of neurons for various feature vectors. We conducted some studies on the learning procedure in order to have different learning in neurons. With this assumption, the task is to evaluate the learning rate of each feature vector corresponding with its occurrence probability density. We propose Gaussian Mixture Model (GMM) to help the GD algorithm for better learning. GMM aims at providing a richer class of density models than the single Gaussian. Considering this proposed method, the estimation of the likelihood of the features according to arbitrary number of clusters is performed by GMM. Then some linear equations are assumed for evaluating the learning rate of such feature vector. Another main challenge of EANN is that common EAs cannot be mathematically controlled because of their huge randomness behavior. In this study, we propose a novel Probabilistic Evolutionary (PE) based RBFNN approximator which has a soft behavior through the learning of non-linear function. PE algorithm defines the occurrence probability of local minima in the feature space.

Geometric Dilution of Precision (GDOP) is a factor denoting how well selected satellites subset is for positioning system. Several methods based on GPS GDOP have been proposed to improve the GPS positioning accuracy [7-10]. The common approach is to use matrix inversion for all combinations and select the minimum one, but it requires numerous computational resources which cause computation burden [11-12]. In order to estimate non-linear GDOP factor, Simon and El-Sherief adopted a special set of features including traces of the measurement matrix as well as its second and third powers, and the determinant of the matrix [13]. They initially proposed the NN approach to approximate and classify the GPS GDOP factors. Extension work on BPNN based GPS GDOP approximation has been done by Jwo and Chin [14], but its learning takes so much time not to mention falling in local minima as yet another problem. There are some solutions for the local minima issue, such as applying robust EAs in order to find the better minima [15-16].

The rest of this paper is organized as follows. In section II, there are some discussions about preliminary backgrounds. Our proposed methods are discussed in section III. Section IV provides our experimental results. Conclusions are made in section V.

2 Preliminary Backgrounds

In this section, the preliminary backgrounds for our proposed methods are discussed.

2.1 Radial Basis Function Neural Network

The RBFNN is a forward neural network model with good performance and capability of global approximation. Furthermore, it is free from the local minima problems. It can be a multi-input multi-output system consisting of an input layer, a hidden layer and an output layer. During the data processing, the hidden layer performs non-linear transforms for the feature extraction and the output layer provides a linear combination of output weights [17]. Given a set of input vectors $\{x_1,\ldots,x_N\}$, along with the corresponding target values $\{t_1,\ldots,t_N\}$, the goal is to find a smooth function f(x) that fits every target value exactly, so that $f(x_n)=t_n$ for n=1,…,N. This is achieved by expressing f(x) as a linear combination of RBFs:

$$f(\mathrm{x}) = \sum_{i=1}^{L} w_i \exp\left(-\frac{\|\mathrm{x} - c_i\|^2}{2{\sigma_i}^2}\right) + b \tag{1}$$

Where L is the number of neurons, c_i and σ_i^2 are the center and variance of *i-th* neuron, and b is the bias of output layer neuron. Therefore, the fitting procedure continues with optimizing the adjustable parameters through learning procedure. This can be performed by GD algorithm which tries to minimize the learning error through the parameters updating in the direction of the negative gradient. Using sum of squared error function as a criterion of the matched design, we have:

$$E = \frac{1}{2}(t_i - f(\mathrm{x}_i))^2 \tag{2}$$

So according to the GD algorithm, the weights can be updated in direction of the negative of the gradient:

$$w(\tau + 1) = w(\tau) - \eta \nabla E(w(\tau)) \tag{3}$$

Considering $y_i = \exp\left(-\frac{\|\mathrm{x}-c_i\|^2}{2{\sigma_i}^2}\right)$, these equations can be proved:

$$\frac{\partial E}{\partial w_i} = -\big(t - f(\mathrm{x})\big)y_i \quad (4)$$

$$\frac{\partial E}{\partial c_{ij}} = -\big(t - f(\mathrm{x})\big)w_i \frac{y_i}{{\sigma_i}^2}\,(\mathrm{x} - c_{ij}) \quad (5)$$

$$\frac{\partial E}{\partial \sigma_i} = \big(t - f(\mathrm{x})\big)w_i \frac{2y_i}{{\sigma_i}^2}\ln(y_i) \quad (6)$$

In each iteration, the new data is introduced to the network and the parameters are updated according to above equations until it reaches the convergence condition. It is recommended that the learning coefficients decrease with iteration increases in order to guarantee the convergence.

2.2 Gaussian Mixture Model

GMM is a simple linear superposition of Gaussian distributions which provides a richer class of density models comparison with single Gaussian. Therefore, the density models can be given by:

$$p(\mathrm{x}) = \sum_{k=1}^{K} \pi_k N(\mathrm{x}|\mu_k, \Sigma_\mathrm{k}) \quad (7)$$

Where K is the number of clusters, π_k, μ_k and Σ_k are the mixing coefficients, mean vector and covariance matrix of the *k-th* cluster, respectively. So with these assumptions, the task is to optimize the adjustable parameters in order to have maximum likelihood solution. An elegant and powerful method for finding maximum likelihood solution for GMM is Expectation Maximization (EM) algorithm. Here the common approach for EM is introduced [18,19]:

Step 1: Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood:

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(\mathrm{x_n}|\mu_k, \Sigma_\mathrm{k})\} \quad (8)$$

Step 2: Evaluate the responsibilities using current parameter values (expectation step):

$$\gamma(z_{nk}) = \frac{\pi_k N(\mathrm{x_n}|\mu_k, \Sigma_\mathrm{k})}{\sum_{j=1}^{K} \pi_j N(\mathrm{x_n}|\mu_j, \Sigma_\mathrm{j})} \quad (9)$$

Step 3: The parameters are updated by using the current responsibilities (maximization step):

$$\mu_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})\,\mathrm{x_n} \quad (10)$$

$$\Sigma_\mathrm{k}^{\mathrm{new}} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})\,(\mathrm{x}_n - \mu_k^{new})(\mathrm{x}_n - \mu_k^{new})^T \quad (11)$$

$$\pi_k^{new} = \frac{N_k}{N} \quad (12)$$

Where:

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \quad (13)$$

Step 4: Evaluate the log likelihood and if the convergence criterion is not satisfied, then return to step 2.

Through this procedure, we can estimate the Gaussian mixture density of the dataset according to the selected number of clusters.

2.3 GPS GDOP

GDOP is a geometrically determined factor that describes the effect of geometry on the relationship between measurement error and position error. It is used to provide an indication of the quality of the solution. Some of the GPS receivers may not be able to process all visible satellites due to limited number of channels. Consequently, it is sometimes necessary to select the satellite subset that offers the optimal or acceptable solutions. The optimal satellite subset is sometimes obtained by minimizing the GDOP factor.

Leading to positioning error minimization, GPS GDOP helps us to find the subset which has the best geometric conditions. The least squares solution to the linearized GPS pseudo-range equation, $z=Hx+v$, is given in [20]:

$$\hat{\mathrm{x}} = (H^T H)^{-1} H^T \mathrm{z} \quad (14)$$

Where the dimension of the geometry matrix H is $n\times4$ with $n\geq4$. Consider the linearized pseudo-range equation, the estimated and true positions difference yield the quality of navigation solution.

$$\tilde{\mathrm{x}} = (H^T H)^{-1} H^T \mathrm{v} \quad (15)$$

Where v has zero mean and so does $\tilde{x}$. The covariance between the errors in the components of the estimated position is:

$$E\{\tilde{x}\tilde{x}^T\} = (H^T H)^{-1} H^T E\{vv^T\} H (H^T H)^{-1} \quad (16)$$

Where E{.} is the expected value operator. If all components of v are pair wise uncorrelated and have variance δ^2, then $E\{vv^T\}=\delta^2 I$ and consequently:

$$E\{\tilde{x}\tilde{x}^T\} = \delta^2 (H^T H)^{-1} \quad (17)$$

The GDOP factor is defined as:

$$GDOP = \sqrt{trace(H^T H)^{-1}} = \sqrt{\frac{trace[adj(H^T H)]}{det\,(H^T H)}} \quad (18)$$

By using cofactors and a determinant in equation (18), computation of GDOP has a closed-form solution in terms of the elements of H. Since the measurement matrix $M=H^T H$ is symmetric, it has four real valued eigenvalues $\lambda_1,\lambda_2,\lambda_3,\lambda_4$. Assuming that M is non-singular, then the GPS GDOP can be expressed as [21]:

$$GDOP = \sqrt{\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} + \lambda_4^{-1}} \quad (19)$$

Because eigenvalues of the inverse matrix are inverse of eigenvalues of the original matrix and the trace of a symmetric matrix equals the sum of all its eigenvalues. Considering optimal satellites subset selection, applying the matrix inversion method to all combinations of satellites and selecting the minimum one is a common approach, but its implementation has analytically high order and imposes a computational burden on the navigation computer. Suppose that a receiver has 20 channels to receive signals from 16 visible satellites, and then a total number of 16!/(4!×12!)=1820 GDOP values need to be computed in order to decide the best combinations of satellites. So GPS GDOP computing is a crucial procedure for real-time or mobile based GPS positioning. In order to approximate GPS GDOP by equation (19), Simon and El-Sherief adopted a special set of features [13]:

$$h_1(\lambda) = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = trace(M) \quad (20)$$

$$h_2(\lambda) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 + \lambda_4^2 = trace(M^2) \quad (21)$$

$$h_3(\lambda) = \lambda_1^3 + \lambda_2^3 + \lambda_3^3 + \lambda_4^3 = trace(M^3) \quad (22)$$

$$h_4(\lambda) = \lambda_1\lambda_2\lambda_3\lambda_4 = \det(M) \quad (23)$$

These equalities hold because M and its power are symmetric matrices. So we have:

$$f: R^4 \to R, \quad h \to GDOP \quad (24)$$

The task is to approximate the GDOP in terms of these features.

2.4 GA for RBFNN Learning

The main disadvantage of learning with GD algorithm is that the probability of falling the system error into local minima is high and it can't find the global minima. Various methods have been proposed for solving the problem. With assumption of NN's error function, one solution is that the network coefficients are updated utilizing EAs such as GA (optimization problem). For this purpose, at first the center vectors of Gaussian function are estimated utilizing K-means and the variances for all functions can be assumed as the same. Then the GA is run in order to decrease the error of the RBFNN. Each chromosome includes output weights coefficients and bias. With assumption of n neurons in hidden layer, we need $n+1$ genes for each chromosome. After evaluating the GA parameters with trial and error, the processes of selection, crossover and mutation are performed and run until our criteria condition is satisfied. The algorithm steps are as follows:

Step 1: Estimating the center vectors of Gaussian functions (RBFNN functions)

Step 2: Initializing the RBFNN weights (chromosomes)

Step 3: Computing the Root Mean Square Error (RMSE) of RBFNN for each chromosome

Step 4: Applying selection, crossover and mutation operators

Step 5: Computing the RMSE of RBFNN for each

chromosome. If the criteria condition is not satisfied, then go to step 4, otherwise the program is ended.

3 Proposed Methods

In this section, two novel learning algorithms based on GMM and PE algorithms applied on RBFNN are proposed in order to approximate the functions which have high non-linear order.

3.1 GMM-Guided GD for RBFNN Learning

Considering GD method, our main challenge is whether it is appropriate for the neurons to have the same learning strategy for all feature vectors? It is better to evaluate the learning rates with more intelligence strategies. With assumption of GD method, the task is to estimate the probability distribution of the dataset, which can select better strategy for learning of the non-linear function. In order to estimate the probability distributions, the GMM is utilized. Then the learning rate of the neurons for each feature vector is estimated as proportional to the its probability density. Therefore, the process of learning of RBFNN can be tracked through these steps:

Step 1: Estimation of the probability densities of each feature vector with GMM utilizing EM algorithm.

Step 2: With assumption of each feature vector density, the learning rate corresponding to that feature vector can be evaluated as follows:

$$\alpha=\alpha_0\,(1+p) \tag{25}$$

Step 3: The GD learning algorithm is run and the network's parameters are updated (the center vectors of the Gaussian functions are computed using K-means).

Step 4: Check the criteria condition and if it does not satisfy your purpose, then go to step 2.

3.2 Probabilistic Evolutionary for RBFNN Learning

In this section, the preliminary backgrounds of the proposed PE algorithm and its application for the learning procedure of RBFNN are discussed. Because of the randomness and variability of EAs, its optimum convergence is not robust, so the researchers are trying to improve its performance. By considering this assumption, we propose a PE algorithm which estimates the occurrence probability of local minima in the space of extracted features. Firstly, the chromosomes which are defined as the Gaussian probability centers (for occurrence probability of local minima) are initialized, and then the cost function is evaluated in terms of these chromosomes. Then, the normalized distance between the best chromosome and other chromosomes are computed as follows:

$$d(i)=\frac{|e_i-e_{best}|}{\sum_{i=1}^{N}|e_i-e_{best}|} \tag{26}$$

Furthermore, according to the evaluated distance, the value of each occurrence probability is updated with defined learning rate:

$$p(\text{best-chromosome})=p(\text{best-chromosome})+\eta\times d(\text{2nd-chromosome}) \tag{27}$$

$$p(\text{ith-chromosome})=p(\text{ith-chromosome})-\eta\times d(\text{ith-chromosome}) \tag{28}$$

For the best chromosome, the occurrence probability of local minima increases relative to its distance from the second chromosome and decreases for others. With increasing the normalized distance, the probability must decrease more. Then, the Gaussian centers are updated toward the best minima as the following:

$$m_i=m_i+\eta\times(m_{best}-m_i) \tag{29}$$

Then, for each probability density, according to that probability distribution, another point is selected and the cost function is evaluated. If this minimum is better than previous one, the Gaussian center will be relocated to this new point and the variance of that distribution is initialized. Otherwise, the probability value of that area decreases and the variance value increases in order to extend its searching area. At last, if the criteria condition is not satisfied, these procedures are continued from the step of distance normalization. The entire procedure of PE algorithm is depicted in Fig. 1. These processes help to search around the Gaussian centers. The searching behavior is controlled with some probability distributions. Like other EAs, some parameters such as populations and genes size, learning rates and the like must be evaluated with trial and error. So it can be operational to apply this algorithm to learn the weights coefficients of RBFNN.

4 Experimental Results

In this section, the embedded system and data analysis results are discussed.

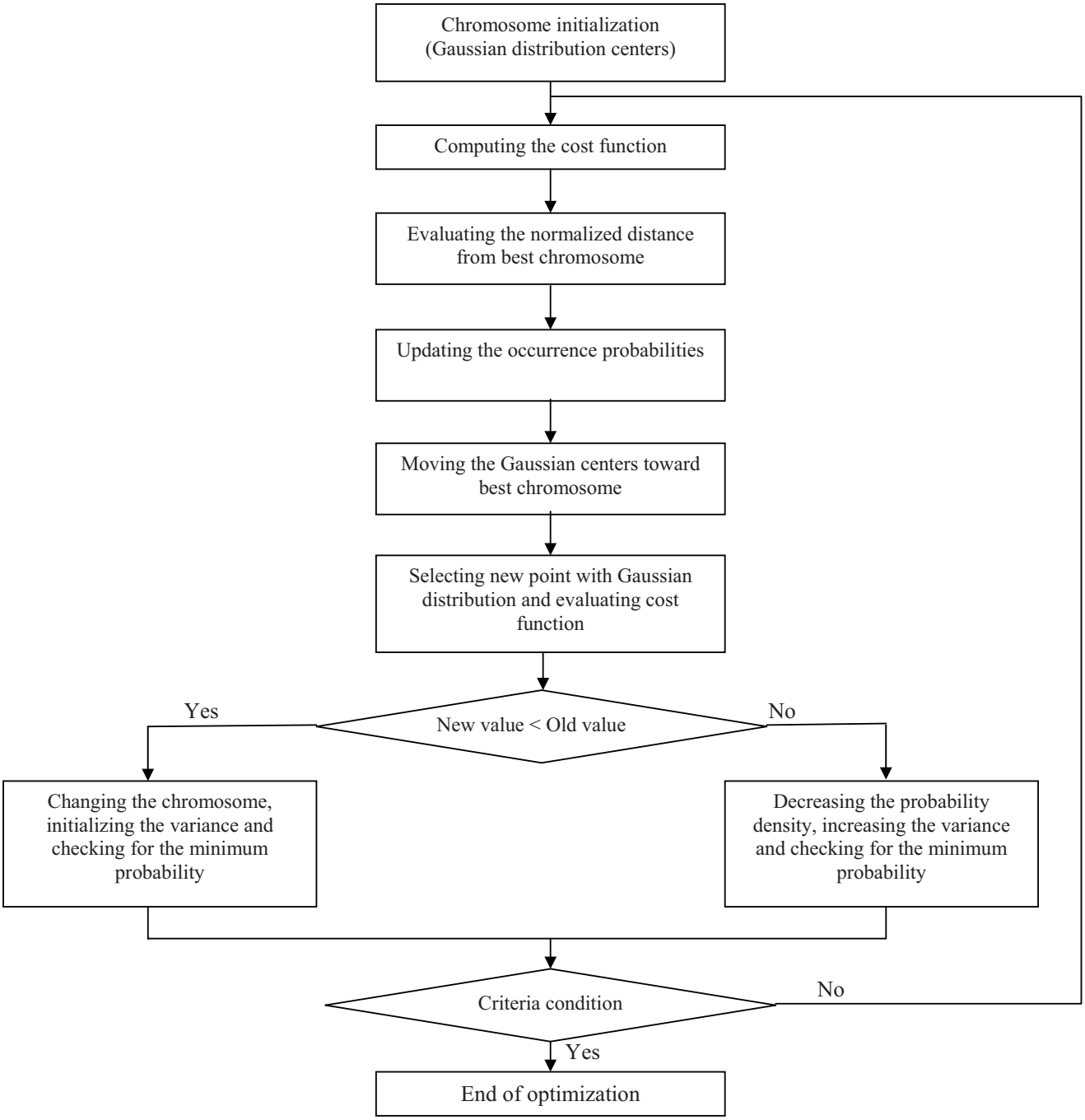

Figure 1. The block diagram of the PE algorithm.

4.1 Embedded Measurement System

In order to evaluate our proposed methods, we assumed the GPS GDOP approximation function a non-linear application. The embedded system has been designed for collecting the real GPS data such as elevation and azimuth of each tracking satellite and positioning coordinate. The FPGA performs the GPS protocol commands and PC USB interface. Two LEA-5 GPS modules from the U-blox company were used. The LEA-5 module series brings the high performance of the U-blox 5 positioning engine to the industry standard LEA form factor. These versatile and stand-alone receivers combine an extensive array of features with flexible connectivity options. Their ease of integration results in fast times-to-market for a wide range of automotive, consumer and industrial applications with strict size and cost requirements. It has 50 channels with over 1 million effective correlators and up to 4Hz position update rate. The PC interface has been implemented using FIFO USB and a handshake component has been developed in FPGA. After GPS starts the positioning, we can access the information through the UART port. These protocols, also the procedures of information saving on FPGA DRAMs, and the USB interfaces were implemented on FPGA. The VHDL program was developed and synthesized by ISE10.1 software. Then in PC, the program was developed using hybrid LABVIEW and MATLAB software in order to collect, save and perform the data analysis. At last, for each of the GPS pages with the number of tracking satellites of about 12, we selected four random satellites as a subset. Then by using conventional inverses matrix calculation, we evaluated the GPS GDOP values. The GPS measurements hardware system is shown in Fig. 2.

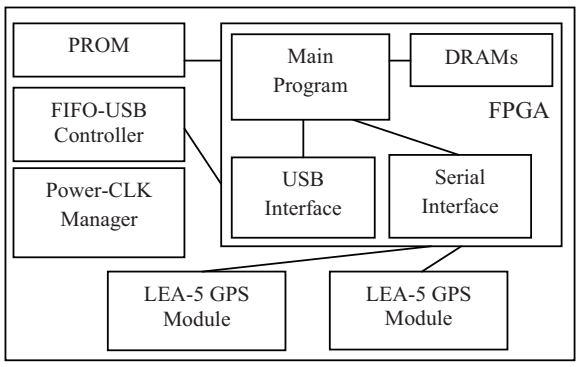

Figure 2. The configuration schematic of the hardware modules of embedded system.

4.2 GPS GDOP Approximation using GMM based RBFNN

We run the GMM with three clusters. At first, the GMM parameters were initialized with small random values. Then through the EM algorithm, it was tried to maximize the log likelihood. Ultimately, the outputs of the algorithms would be the mean and covariance matrices of each cluster and also the probability densities of feature vectors. Fig. 3 shows the data clusters and probability densities utilizing GMM in terms of the third and fourth feature which are more informative in comparison to others. GMM method estimates the combination of three Gaussian functions in the space of the 3rd and 4th features. By considering both features, it can be conceptually derived that clusters are approximately distinguished from each other. So this probability density function can help us to estimate the strength of learning over each input feature vector. In the space areas where the probability of occurrence is lower, the GD algorithms learned it with less efficiently (something like outliers).

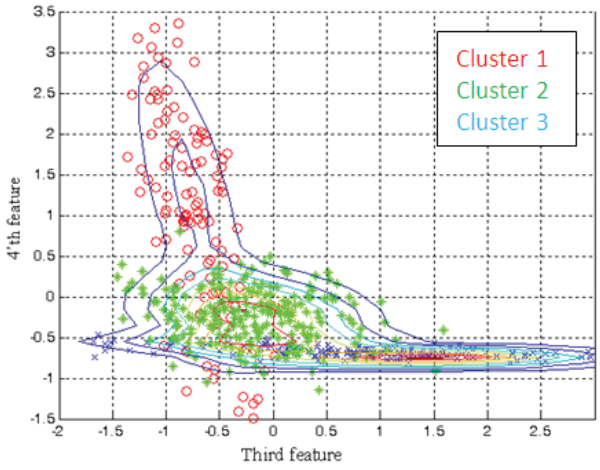

Figure 3. Data clusters and probability densities have been estimated utilizing GMM procedure.

Then, we initialized $\alpha_0 = 0.2$ and the learning rates were evaluated through $\alpha = \alpha_0(1+p)$ for each feature vector. The GD learning method was run for updating the weights coefficients of the RBFNN in order to decrease the error of the network. This helped the learning procedure attend more to the feature vector with higher probability distribution and the feature vectors with less probability were less attended.

Evaluation of the algorithm was continued by dividing that GPS GDOP data into learning and test parts. Each part contained 300 data. Then both RBFNNs with

the classical GD algorithm and our proposed approach were designed. In each procedure, the GD algorithm iterations were adjusted to 1000 and the number of neurons was 10. At first, the parameters of Gaussian centers, covariance matrices and weights were initialized with the low random values. The number of the GMM clusters, iterations and rules values were evaluated by trial and error. The number of neurons was selected so that RBFNN could learn the to-be-learned data and the generalization of algorithm could be satisfied. If we select the number of large neurons corresponding to learning data distributions, the neurons are fitted to learning data completely and the generalization is not satisfied. With trial and error, we found ten neurons in order to have the best learning.

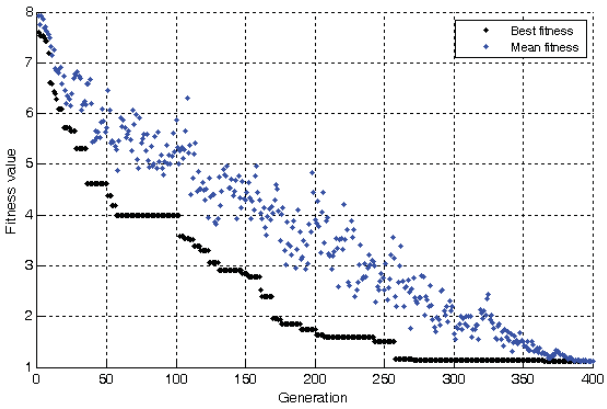

Figure 4. The searching convergence of the local minima of GDOP RMSE is shown utilizing GA based RBFNN.

4.3 GPS GDOP Approximation using GA based RBFNN

Firstly, the Gaussian centers were estimated utilizing K-means so that the variances of all distributions were the same. Then we ran the GA in order to decrease the error of the RBFNN. Each chromosome included the weights coefficients and bias of the network. So with assumption of ten neurons in hidden layer, we needed eleven genes for each chromosome. The population size was estimated ten with trial and error. The resulted error values for each chromosome were evaluated. Further 50% of the best chromosomes were selected for the next crossover operation for producing new offsprings. Even and odd crossover was performed and the crossover coefficient was set at 0.4. Eventually, in order to stay away from the local minima, the mutation operation was performed by adding Gaussian distribution with defined weights. The Gaussian distribution must have the variance of one with zero mean features. Fig. 4 shows the convergence of searching the local minima of the GPS GDOP approximation utilizing GA based RBFNN with assumption of 300 learning data and 300 test data. It shows the cost function of the network in terms of GA generations. From Fig. 4, the performance of the algorithm reaches 1.1. As it is shown, the mean of the RMSE of the network has the randomness behavior. The RMSEs of best chromosome and average chromosomes are shown by black and blue dots, respectively. As it is shown, it requires 400 generations of the algorithms to find its convergent minima. The chromosomes firstly search with more randomness to find better minima, then after some generations they converge to their own local minima.

4.4 GPS GDOP Approximation using PE based RBFNN

Since the mean cost function of common EAs is so random, it is not robust and the researchers are trying to improve it. For this purpose, we proposed the optimization method based on the occurrence probability of the local minima in feature space. At first, the chromosomes which were the Gaussian centers were initialized. Then, the cost function for each chromosome was evaluated. The normalized distances of all the chromosomes from the best chromosome were computed. Then, according to these distances, the probability values of each chromosome were increased or decreased with the coefficient value of 0.2. Then, the Gaussian centers were moved toward the best minimum. In the end, searching around each chromosome was performed with its Gaussian distribution and the chromosomes with probabilities less than 0.9 were omitted. These processes continued until the criteria condition would be satisfied. In order to learn the RBFNN, the PE algorithm tried to find the best weights for decreasing the error of the GPS GDOP approximation. Fig. 5 shows the GPS GDOP RMSE convergence in terms of the number of generations. As it is shown, during the evolution of algorithm, in order to find the best local minima, the searching pattern of error function is less random since it considers the probability density of each feature vector. As it is shown, the results demonstrate that the convergence time of the algorithm significantly reduced to 195 generations as well as it finding lower RMSE of 0.81, compared to the previous method. Fig. 6 shows the GPS GDOP approximation based on PE based RBFNN. The temporal patterns of real and estimated GPS GDOP are shown here. It represents the tractable capability of our proposed estimator for non-linear functions.

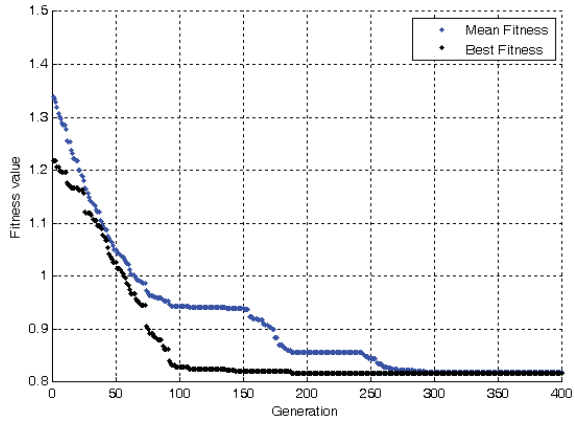

Figure 5. The GDOP RMSE convergence in terms of the number of generations.

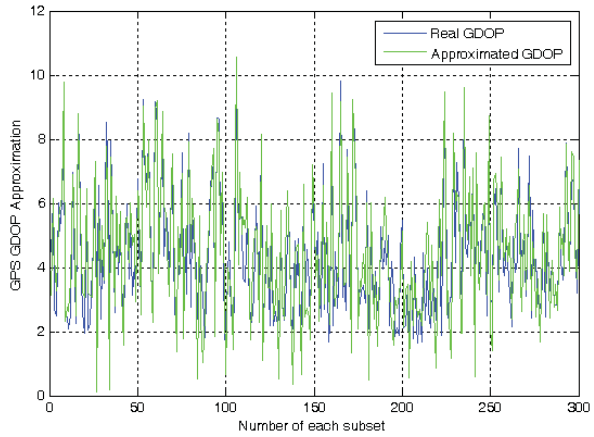

Fig. 6. The GPS GDOP approximation using PE based RBFNN.

The conventional RBFNN was also implemented with learning rate of 0.2. In order to compare the algorithms precisely, the statistical parameters such as RMSE, Max and Min of errors are given in Table 1. At it is shown, the PE based RBFNN have the best performance in comparison with other methods. Also the intelligence learning rate utilizing GMM on GD learning algorithm improved the RMSE up to 62%.

Table 1. Evaluation of the performance of four GPS GDOP approximation methods.

Methods / Parameters	RBFNN	GMM-RBFNN	GA-RBFNN	PE-RBFNN
Min	-1.1	-0.9	-1.6	-2.3
Max	5.3	3.1	2.76	2.1
RMSE	**1.45**	**0.89**	**1.1**	**0.81**

At last, after evaluating the GPS GDOP for all subsets of satellites by RBFNN, the one with lowest GPS GDOP value can be assigned as the best subset for positioning system. In comparison with the studies which were performed in this field [12,13,14,22], the accuracy of the proposed methods was significantly improved.

5 Conclusion

In this paper, in order to optimize the learning of RBFNN for GPS GDOP approximation, three methods were presented: 1) GMM-Guided gradient descent based, 2) GA based and 3) PE based learning for RBFNN. Two main issues were discussed: 1) how to select the learning rates for input feature vectors with different probability densities and 2) how to mathematically control the process of EA optimization. The strategies of selecting the learning rates were chosen with the estimation of the probability densities of feature vectors using GMM and then the learning rates were evaluated proportional to these densities. Furthermore, controlling the procedure of EA's randomness was proposed by defining the occurrence probability of local minima in order to search the feature space with some probabilities. The embedded GPS measurement system was implemented and the proposed algorithms were designed to approximate these measured GPS GDOP. The results demonstrate that the proposed methods have significantly better performance in comparison with conventional methods. The RMSE improvements in our proposed methods in comparison with conventional RBFNN led to 62%, 31% and 79%, respectively.

References

1. J. X. Peng, K. Li and D. S. Huang, "A Hybrid Forward Algorithm for RBF Neural Network Construction", IEEE Transaction on Neural Networks, Vol.17, No.6, pp.1439-1451, 2006.

2. J. Gonzalez, I. Rojas, J. Ortega, H. Pomares, F. J. Fernandez and A. F. Diaz, "Multiobjective Evolutionary Optimization of the Size, Shape, and Position Parameters of Radial Basis Function Networks for Function Approximation", IEEE Transaction on Neural Networks, Vol.14, No.6, pp.1478-1495, 2003.

3. F. H. F. Leung, H. K. Lam, S. H. Ling and P. K. S. Tam, "Tuning of the Structure and Parameters of a Neural Network Using an Improved Genetic Algorithm", IEEE Transaction on Neural Networks, Vol.14, No.1, pp.79-88, 2003.

4. N. B. Karayiannis, "Reformulated Radial Basis Neural Networks Trained by Gradient Descent", IEEE Transaction on Neural Networks, Vol.10, No.3, pp.657-671, 1999.

5. K. K. Shukla and Raghunath, "An Efficient Global Algorithm for Supervised Training of Neural Networks", Journal of Computers and Electrical Engineering, Vol.25, No.3, pp.195-218, 1999.

6. S. H. Nabavi-Kerizi, M. Abadi and E. Kabir, "A PSO-Based Weighting Method for Linear Combination of Neural Networks", Journal of Computers and Electrical Engineering, Vol.36, No.5, pp.886-894, 2010.

7. C. S. Chen and S. L. Su, "Resilient Back-Propagation Neural Network for Approximation 2D GDOP", Multiconference of Engineerings and Computer Scientists, pp.1-5, 2010.

8. J. Zhu, "Calculation of Geometric Dilution of Precision", IEEE Transactions on Aerospace and Electronic Systems, Vol.28, No.3, pp.893-895, 1992.

9. M. S. Phatak, "Recursive Method for Optimum GPS Satellite Selection", IEEE Transactions on Aerospace and Electronic Systems, Vol.37, No.2, pp.751-754, 2001.

10. M. Zhang and J. Zhang, "A Fast Satellite Selection Algorithm: Beyond Four Satellites", IEEE Journal of Selected Topics in Signal Processing, Vol.3, No.5, pp.740-747, 2009.

11. D. Y. Hsu, "Relations between Dilutions of Performance and Volume of the Tetrahedron Formed by Four Satellites", IEEE Symposium on Position, Location and Navigation, pp.669-676, 1994.

12. C. H. Wu, W. H. Su, and Y. W. Ho, "A Study on GPS GDOP Approximation Using Support-Vector Machines", IEEE Transactions on Instrumentation and Measurement, Vol.60, No.1, pp.137-145, 2011.

13. D. Simon and H. El-Sherief, "Navigation Satellite Selection using Neural Networks", Journal of Neurocomputing, Vol.7, pp.247-258, 1995.

14. D. J. Jwo and K. P. Chin, "Applying Back-Propagation Neural Networks to GDOP Approximation", Journal of Navigation, Vol.55, No.1, pp.97-108, 2002.

15. M. R. Mosavi, "Knowledge-based Methods for Optimum Approximation of Geometric Dilution of Precision", Journal of Computational Intelligence and Applications, Vol.9, No.2, pp.153-170, 2010.

16. M. R. Mosavi and M. Saraf, "Precise Modeling of GPS GDOP using Neural Network", International Review on Modelling and Simulations, Vol.3, No.4, pp.665-671, 2010.

17. O. Deperlioglu and U. Kose, "An Educational Tool for Artificial Neural Networks", Journal of Computers and Electrical Engineering, Vol.37, No.3, pp.392-402, 2011.

18. A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm (With Discussion)", Journal of the Royal Statistical Society, Series B, pp.1-38, 1977.

19. M. I. Jordan and R. A. Jacobs, "Hierarchical Mixtures of Experts and The EM Algorithm", Journal of Neural Computation, pp.181-214, 1994.

20. R. Yarlagadda, I. Ali, N. Al-Dhahir and J. Hershey, "GPS GDOP Metric", IEE Proc.-Radar, Sonar Navig., Vol.147, No.5, pp.259-264, 2000.

21. S. H. Doong, "A Closed-Form Formula for GPS GDOP Computation", Journal of GPS Solutions, Vol.13, No.3, pp.183-190, 2009.

22. D. J. Jwo and C. C. Lai, "Neural Network-based GPS GDOP Approximation and Classification", Journal of GPS Solutions, Vol.11, No.1, pp.51-60, 2007.

Authors

Morteza Saraf received his B.S. degree in Electronic Engineering from Department of Electrical and Computer Engineering, Tabriz University, Tabriz, Iran in 2009. He is currently a Master of Science student in Department of Electrical Engineering of IUST and a researcher in Cognitive Science School, IPM. His research interests include Artificial Intelligent Systems, Cognitive Neuroscience and Statistical Pattern Recognition.

Mohammad-Reza Mosavi received his B.S., M.S. and Ph.D. degrees from Department of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran in 1997, 1998 and 2004, respectively. He is currently a faculty member in Department of Electrical Engineering of IUST as professor. He is the author of about 160 scientific publications in journals and international conferences. His research interests include Artificial Intelligent Systems, Global Positioning Systems, Geographic Information Systems and Remote Sensing.

Karim Mohammadi received his Ph.D. degree from Department of Electrical Engineering, Oakland University, USA in 1981. He is currently a faculty member of Department of Electrical Engineering of IUST as professor.