3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)

Proceedings of Engineering & Technology (PET)
pp. 134-140

Construction of an RBF Classifier Based on Logarithmic Spiral

M. W. GUERFALA, A. SIFAOUI and A. ABDELKRIM
L.R. Automatique, ENIT
Ecole Nationale d'Ingénieurs de Tunis, ENIT
BP 37, Le Belvédère 1002 Tunis, Tunisie
mohamed.wajih.guerfala@gmail.com
amel.sifaoui@enit.rnu.tn
afef.a.abdelkrim@ieee.org

***Abstract*— Clustering is the organization of a set of data in homogeneous classes. It aims to simplify the representation of the initial data. The automatic classification recovers all the methods allowing the automatic construction of such groups. This paper describes the design of neural classifiers with radial basis function (RBF) using a new algorithm for characterizing the hidden layer structure. This algorithm, called k-means Mahalanobis distance, groups the training data class by class in order to calculate the optimal number of clusters of the hidden layer, using the validity index of Davis Bouldin. To initialize the initial clusters of k-means algorithm, we have used the method of logarithmic spiral golden angle. Two examples of data sets are considered to show the efficiency of the proposed approach and the obtained results are compared with basic literature classifier.**

Keywords—Radial Basis Function neural network; classification; k -means; validity index; Mahalanobis distance; Logarithmic spiral; golden angle; golden ratio.

I. INTRODUCTION

Introduced into the neural network literature by Broomhead and Lowe [1], the radial basis function neural networks have been widely used for function approximation, pattern classification and recognition due to their structural simplicity and faster learning ability [2], [3]. However, their design still remains a difficult task due to the absence of systematic method giving an optimal architecture [4].

RBF neural networks consists of three layers: an input, a hidden and an output one. The input layer corresponds to the input vector feature space and the output layer to the pattern classes [5]. So the whole architecture is fixed only by determining the hidden layer and the weights between the middle and the output layers [6].

Its training procedure is usually split into two successive steps. First, the centers of hidden layer (HL) neurons are selected by clustering algorithms such as k-means [7], [8], support vector machine (SVM) [9] or hierarchical clustering [10]. Second, the weights connecting the hidden layer with the output layer are determined by supervisor algorithms such as Neural Networks. One of the used techniques to find the optimal number of this HL is the logarithmic spiral which have seen a significant amount of research on nature-inspired optimization techniques such as neuro-computing in the past 25 years, evolutionary and genetic algorithms, particle swarm optimization. Most recently, a new multipoint meta-heuristics search method has emerged for 2-dimensional continuous optimization problems based on the analogy of spiral phenomena in nature, called 2- dimensional spiral optimization first proposed by Tamura and Yasuda in 2010 [11].

Focused spiral phenomena are approximated to logarithmic spirals, which frequently appear in nature, such as whirling currents, nautilus shells and arms of spiral galaxies. A remarkable observation about logarithmic spirals is that their discrete processes generating spirals can realize effective behavior in meta-heuristics. Two-dimensional spiral optimization uses the feature of logarithmic spirals [11].

In this paper, a new learning algorithm is proposed for construction of the radial basis function networks solving classification problems. It determines the proper number of hidden neurons automatically and calculates the centers values of radial basis functions. After the selection of the hidden neurons, the widths of nodes are determined by the P-nearest neighbors heuristic, and the weights between the hidden layer and the output layer are calculated by the pseudo-inverse matrix.

The aim of this approach consists in transforming the problem of determing the number of hidden layer neurons to a clustering problem. In order to determine the clusters number in the data of each class, the k-means algorithm is combined with the validity index of Davis Bouldin. In k-means algorithm, the used distance corresponds to the Mahalanobis distance. We also give a solution to overcome the problem of initialization of initial centers needed to start this algorithm using our method " The logarithmic spiral golden angle". Two different real databases are used in order to evaluate the proposed classifier performances.

II. THE LOGARITHMIC SPIRAL GOLDEN ANGLE

The logarithmic spiral golden angle is a specific case of the logarithmic spiral. It represents a plane curve centered in a

ISSN: 2356-5608

starting point and parameterized by the radius r, the angle θ and the Golden Ratio φ.

A. The logarithmic spiral

A Logarithmic Spiral is a plane curve for which the angle between the radius vector and the tangent to the curve is a constant [12]. Such spirals can be approximated mathematically defined by the following equation on the 2-dimensional polar coordinates system (r,θ) as [11] :

$$r = a\, e^{b\theta} \qquad (1)$$

where a and b are positive real with $a > 0$ and b is not zero. Equation (1) can be transformed into Cartesian coordinate as follows:

$$\begin{cases} x(\theta) = r(\theta)\ \cos(\theta) = ae^{b\theta}\ \cos(\theta) \\ y(\theta) = r(\theta)\ \sin(\theta) = ae^{b\theta}\ \sin(\theta) \end{cases} \qquad (2)$$

In our work, we have set the factor b of the logarithmic spiral to zero ($b = 0$), it goes back to simplify the polar radius as follows: $r = ae^{b\theta} = ae^{0\times\theta} \Rightarrow r = a$

We obtain the following equation of the logarithmic spiral golden angle:

$$\begin{cases} x(\theta) = r\ \cos(\theta) = a\ \cos(\theta) \\ y(\theta) = r\ \sin(\theta) = a\ \sin(\theta) \end{cases} \qquad (3)$$

B. The Golden ratio

The irrational number, golden ratio is also known as golden section by the ancient Greeks, golden proportion, divine proportion or golden number [13].

The golden ratio φ, has many properties in which people are eager to know. It is a number that is equal to the reciprocal of its own with the addition of 1: $\varphi = \frac{1}{\varphi} + 1$.

Likewise, the ratio of any two consecutive Fibonacci numbers converges to give approximates of 1.618, or its inverse, 0.618. This shows the relationship between Fibonacci numbers and golden ratio [14].

If we can divide a line in such a way that the ratio of the whole length to the length of the longer segment happen to be equal to the ratio of the length of the longer segment to the length of the shorter segment, then we can say the ratio is golden ratio [13].

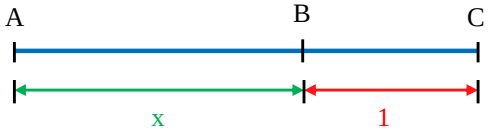

Figure 1. Divising of a whole length AC into two segments AB and BC.

This gives mean ratio if $\frac{AB}{BC} = \frac{AC}{AB}$. If we set the value of AB to be x, and use 1 to represent the length of BC, then $\frac{x}{1} = \frac{1+x}{x}$. Then the irrational number is the only positive solution of the equation $x^2 - x - 1 = 0$, so $x = \frac{1+\sqrt{5}}{2}$

where the Greek letter phi (φ) represents the golden ratio. Its value is: $\varphi = \frac{1+\sqrt{5}}{2} \approx 1{,}6180339887$.

C. The Golden angle

In geometry, the golden angle is created by dividing the circumference of a circle c in two sections, a longer arc of length a and a smaller arc of length b such that: $c = a + b$ and $\varphi = \frac{a+b}{a} = \frac{c}{a} = \frac{a}{b}$

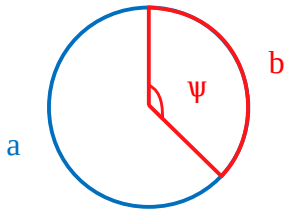

Figure 2. Golden angle measurement

The angle formed by the arc b of circle c is called the golden angle ψ. It derives from the golden ratio φ.

$$\psi = 2\pi - \left(\frac{2\pi}{\varphi}\right) = 2\pi\frac{(\varphi - 1)}{\varphi} \approx 137{,}5°$$

III. PROPOSED INITIALIZATION OF THE K-MEANS ALGORITHM WITH LOGARITHMIC SPIRAL GOLDEN ANGLE

The k-means algorithm aims to minimize the distance between the object and the center of its groups. In this section, we present the k-means algorithm based on the Mahalanobis distance.

A. K-means algorithm Mahalanobis distance specifications

There are different types of distances such as: Minkowski distance, the average, the family of metrics, Euclidean Weighted and the Euclidean distance which is the most used one, e.g. applied in the RBF Networks. [15]

Moreover, the Mahalanobis distance is a distance measure and its utility is a way to determine the similarity between two multidimensional random variables, and it differs from Euclidean distance, because the Mahalanobis distance takes into account the correlation between random variables, [15]. The Mahalanobis distance is defined by:

$$d(x,y) = \sqrt{(x-y)\times Cov(X)^{-1}\times(x-y)^{T}} \qquad (4)$$

where $Cov(X)$ is the covariance matrix. If the elements x and y are independent, the covariance matrix is the identity and the Mahalanobis distance is equal to the Euclidean distance. The algorithm based on the Mahalanobis distance k-means is described by the following steps:

Algorithm: Function Kmeans_distance_Mahalanobis (KDM)

Begin

Input: - The database $X = \{x_1, x_2, ..., x_N\} \in R^d$.

- The position of each center $C = \{c_1, c_2, ..., c_k\} \in R^d$.

Output: - The new position of each center $C^* = \{c_1^*, ..., c_k^*\} \in R^d$

Step 1:

- Determine the size N of the data base of X .
- Determine the number of k centers to use in the observation space C .
- Initialize the vector of new positions centers C^* to zero.

Step 2: - Determine the covariance matrix $Cov(X)$ with the following equation:

$$Cov(X) = \frac{1}{N-1} \sum_{i=1}^{N} \left(X_{ij} - \bar{X}_j\right)\left(X_{ij} - \bar{X}_j\right)^T$$

with $X_{ij} \in X$; $i = 1,...,n$ and $j = 1,..., p$.

Where $\overline{X}_j = \sum_{i=1}^{n} X_{ij}$ with $\overline{X}_j$: arithmetic averages.

While : The new centers as not undergo significant displacement **Do :**

Step 3: - Assign each observation (dot) group nearest center $c_j : x_l \in c_j$ according to the Mahalanobis distance formula:

$$d(i, j) = \sqrt{\left(x_i - c_j\right) \times Cov\left(X\right)^{-1} \times \left(x_i - c_j\right)^T}$$

with $l = 1,...,\ N$ and $j = 1,...,\ k$.

Step 4: - Recalculate the position of each new center :

$$c^*_j = \frac{1}{N_j} \sum_{x_l \in c_j}^{N} x_l$$

with N_j = the set of points belonging to the center c_j and $j = 1,...,k$.

End While

End

The execution of the k-means algorithm requires the determination of the initial centers values. These values are usually selected in a random manner. Each boot (initialization) is a different solution (local optimum) which can in some cases be far from the optimal solution (global optimum) [16] . A simple solution to this problem is to run the algorithm several times with different initialization and retain the best combination found. The use of this solution is limited because of its cost and we can find a better score in a single execution [17].

To solve this problem, we propose a solution to initialize the k-means algorithm using the logarithmic spiral golden angle. parameterized by the radius r , the angle θ and the Golden ratio φ and the Golden angle Φ .This solution is divided en several steps:

The first step is to calculate the maximum distance between two individual points (a,b) belonging to the database, then to define the middle ground G between these two individuals and determine the radius $R = \overline{Gb}$ *"Fig. 3"*.

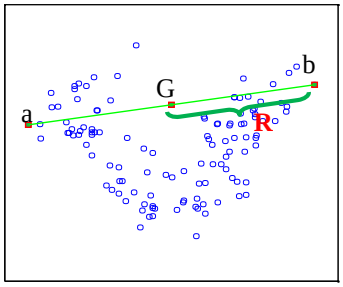

Figure 3. Tracing of the two most distant individuals $(a,\ b)$ and their medium G .

The second step is to calculate the golden number φ by applying the following formula: $\varphi = \frac{1+\sqrt{5}}{2} \approx 1{,}6180339887$

The third is to initialize the values of the logarithmic spiral golden angle on the polar coordinates system (r, θ): the radius $r = a = a_0 = 0$ and the angle $\theta = \theta_0 = 0$ .The angle θ increases by the factor $d\theta = \Phi = 2\pi\left(\frac{\varphi - 1)}{\varphi}\right.$ and the radius r increases by the factor $da = \frac{R}{k_{\max}} = \frac{\overline{Gb}}{k_{\max}}$.

To determine $k_{\max}$ we adopted the suggestion of Bezdek [18] as follows : $k_{\max} = \sqrt{N}$, (N is the size of the database) .

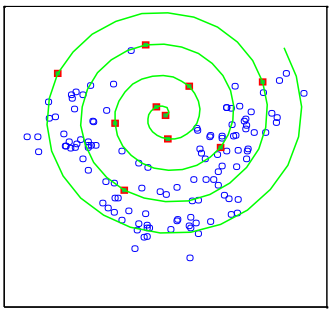

Figure 4. Tracing of initialization of the k centers maximum on the outline of the logarithmic spiral golden angle

The forth step is to determine the positions of the centers of the logarithmic spiral golden angle with center G and radius $r = a$ *"Fig. 4"*. The calculation of the center position C_k is performed by applying the following formula:

$$\begin{cases} C_{kx} = G_x + a \times \cos(\theta) \\ C_{ky} = G_y + a \times \sin(\theta) \end{cases} \quad (5)$$

With $\theta = \theta + d\theta$; $a = a + da$ and $k = 1,...,k_{\max}$.

Thereafter, we will save the positions of these $k_{\max}$ centers in the variable $C_k = \{c_1,...,c_{k\max}\}$.

The basic principle of the strategy adopted is summarized in the following algorithm:

Algorithm: Init_Centres_Kmeans_ Logarithmic_Spiral

Begin

Input: - The database $X = \{x_1, x_2, ..., x_N\} \in R^d$.

- The maximum number of centroids $k_{\max}$.

Output: - The position of each center $C = \{c_1, .., c_k, ..., c_{k\max}\} \in R^d$

Step 1: - Calculate the maximum distance D between two points belonging to the base X .

- Calculate the center G of D and the radius $R = \overline{Gb}$.

Step 2: - Calculate the golden ratio ¨ by applying the following formula: $¨ = \frac{1+\sqrt{5}}{2}$

Step 3: - Initialize the values of the logarithmic spiral golden angle on the polar coordinates system $(r, ¯)$: the radius $r = a = a_0 = 0$ and the angle $¯ = ¯_0 = 0$.

- Fixing the increment of the angle ¯ by the factor $d¯ = · = 2® \frac{(¨ - 1)}{¨}$

- Fixing the increment of the radius r by the factor $da = \frac{1}{R} = \frac{1}{\overline{Gb}}$

Step 4: - Determine the positions of the centers belonging to the logarithmic spiral golden angle with center G and radius $r = a$ according to the following formula:

$$\begin{cases} C_{kx} = G_x + a \times \cos(¯) \\ C_{ky} = G_y + a \times \sin(¯) \end{cases}$$

With $¯ = ¯ + d¯$; $a = a + da$ and $k = 1, ..., k_{\max}$.

Step 5: - Save the positions of the centers found in $C = \{c_1, .., c_k, ..., c_{k\max}\} \in R^d$.

End

B. Evaluation Measures

Using a non-supervised clustering algorithm, such as k-means algorithm, requires the determination of the number k of groups leading to the execution of the algorithm repeatedly for different values of this parameter. For optimal number of groups, a criterion should be used to evaluate the result of the algorithm. This criterion is known as the validity index [19]–[22] [15, 16, 17, and 18] name based on the notions of compactness and separation.

In literature, there are a lot of validity indexes, most of them are based on the notions of compactness within different groups and separability between these different groups. In this article, we will use the Davies-Bouldin index as an index of validity of neural classifiers.

C. Index Davies-Bouldin

This index takes into account both the compactness and the separability of groups [23] and its value is much lower than the groups are compact and well separated. It promotes hyperspherical groups and is therefore particularly well suited for a use with the k-means algorithm. The I_{DB} index is defined by the following expression:

$$I_{DB} = \frac{1}{K} \sum_{i=1}^{k} \max_{i \neq j} \frac{\{d_c(c_i) + d_c(c_j)\}}{D_{cc}(c_i, c_j)} \quad (6)$$

Where $d_c(c_i)$ is the average distance between an object and its group c_i following the center and $D_{cc}(c_i, c_j)$ is the distance between the centers of groups c_i and c_j with:

$$d_c(c_i) = \frac{1}{N_l} \sum_{l=1}^{N_l} \|x_l - c_j\| \quad (7)$$

$$D_{cc}(c_i, c_j) = \|c_i - c_j\| \quad (8)$$

IV. NEW ALGORITHMS FOR THE CONSTRUCTION OF THE HIDDEN LAYER OF THE RBF CLASSIFIER

We proposed a new algorithm to characterize the hidden layer classifier i.e. to determine the number of centers of different Gaussian and the value of each center.

In what follows, we present the principle of the proposed algorithm and explain how the validity index of Davies-Bouldin is combined with the k-means algorithm with Mahalanobis distance to determine automatically the number k of groups.

However, it is necessary to fix a maximum number of centroids $k_{\max}$. The $k_{\max}$ value can be defined by the user if he knows the structure of his database. Given that it's not always the case, the Bezdek [18] suggestion is adapted, so we choose $k_{\max} = \sqrt{N}$ (N is the size of database).

Applying this algorithm to all classes and summing the number of obtained groups, the number of neurons in the hidden layer is determined. A neuron is then assigned to each group. For this RBF classifier we partition the data base $X = \{x_1, x_2, ..., x_N\} \in R^d$ in individual blocks according to the number of output classes $\Omega_j = 1, 2, ..., m$. We get $X_d = (X_{\Omega 1} \quad X_{\Omega 2} \quad . \quad . \quad X_{\Omega m})^T$.

Then, we apply the principal component analysis (PCA) with the data base X_d to reduce it to a new base in two dimensions $X^*_d = \{x_1, x_2\} \in R^2$.

Principal component analysis (PCA) is a widely used statistical technique for unsupervised dimension reduction. K-means clustering is a commonly used data clustering for performing unsupervised learning tasks [24]. The PCA is based on the calculation of averages, variances and correlation coefficients. The main basis of dimension reduction is that PCA picks up the dimensions with the largest variances. In our case, we choose the two largest variances.

The next step is to determine the number of centers and the center position of each class $C_{\Omega 1} = \{c_1, ..., c_k\} \in R^d$ through the classifier based on the k-means algorithm with Mahalanobis distance.

We group the centers of each class $\Omega_j = 1,...,m$ found in the $C = \begin{pmatrix} C_{\Omega 1} \\ . \\ . \\ C_{\Omega m} \end{pmatrix}$ matrix and we apply the k-means algorithm Mahalanobis distance for the new positions of the centers $C^* = \{c_1^*, c_2^*, ..., c_K^*\} \in R^d$.

To complete the construction of the hidden layer classifier, there is a second parameter to consider in the neurons, which is the width factor $\pm_j$ for each centroid c_j ($j = 1,...,k$). This factor is calculated using the following formula:

$$\pm_j = \frac{1}{N_a \sqrt{8}} \sum_{i=1}^{Na} \left\| x_i - c_j \right\| \quad (9)$$

A. RBF classifier based on the k-means algorithm with Mahalanobis distance KMD-LS-IDB

The proposed algorithm based on the k-means algorithm with Mahalanobis distance (KMD-LS-IDB) determining the number and the centers values is described below.

Begin
Input:
- The block database $X_{\Omega j} = \{X_{\Omega 1}, ..., X_{\Omega m}\} \in R^d$ of one class of data base X_d , Taking the case of the block. (The same approach for other classes).

Output: - The position of each center $C_{\Omega 1} = \{c_1, ..., c_k\} \in R^d$.

Step 1:
- Determine the size n and the number of characters (attributes) p of the data base $X_{\Omega 1}$.

Step 2: - Initialize the minimum number of centroids $K_{\min} = 2$ and then look for the maximum number of centroids by $K_{\max} = \sqrt{n}$.

- Initialize the variables $a = k = K_{\min}$ and $b = 1$.

Step 3: - Apply Algorithm *Init_Centres_Kmeans_ Logarithmic_Spiral* which initializes the centers for kmeans algorithm of the data base $X_{\Omega 1}$.

Step 4: - **Repeat** the following steps **until** $k = K_{\max}$

Step 4.1: -**If** $k \prec K_{\max} + 1$ **Then**:

- Take the following positions centers $C = \{c_1, ..., c_a\} \in R^d$.

- Deduct the number of centers k .

End If

Step 4.2: - Apply **k-means** algorithm with **Mahalanobis** distance to determine the new positions of the $C^* = \{c_1^*, ..., c_a^*\} \in R^d$ centers.

Step 4.3: - Calculate the compactness and separability of groups with the Davies- Bouldin Index by variable I_{DB} .

Step 4.4: - Save I_{DB} variable in the table called Tab_IDB .

Step 4.5: - Increment variables $a = a + 1$, $b = b + 1$.

Step 5: - Determine the I_{DB} lowest index of the table Tab_IDB in variable .

- Take the following positions centers $C_{\Omega 1} = \{c_1, ..., c_{+1}\} \in R^d$ as the optimal number required classifying Ω_1 class centers.

End

B. Calculation of synaptic weight

After determining the parameters of the proposed classifier hidden layer, the learning is finished by the calculation of the synaptic weight w_{ij} , connecting the hidden layer neurons to those of the output layer. The linearity property of the outputs $y_j(x_l)$ of the network is used**.** The expression of each of the m outputs is written in the form:

$$y_j(x_l) = h_j(x_l) \times w_{ij} \quad (10)$$

The global output of the network is written as follows:

$$Y = H \times W \quad (11)$$

The objective is to determine the matrix W that minimizes an error function, chosen as the square of the sum of classification errors. The weight of the output layer can be calculated by the following matrix equation:

$$\underbrace{\begin{bmatrix} .. & .. & ..... & .. \\ 11 & 12 & ..... & 1M \\ .. & .. & ..... & .. \\ 21 & 22 & ..... & 2M \\ ... & ... & ..... & ... \\ .. & .. & ..... & .. \\ N1 & N2 & ..... & NM \end{bmatrix}}_{H} \times \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ ... \\ w_M \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_M \end{bmatrix}}_{Y} \quad (12)$$

With H : The matrix Gaussian widths ($..._{ij}$).

Y : The matrix of the output layer.

W : The weight matrix of centroids.

The above equation is of the form:

$$H \times W = Y \Rightarrow W = H^{-1} \times Y \quad (13)$$

Given that the H matrix is rarely square, the pseudo- inverse of the matrix H is applied according to equation (14):

$$W = \left[H^T \times H \right]^{-1} \times H^T \times Y \quad (14)$$

V. EVALUATION OF RBF CLASSIFIERS

The purpose of this section is to evaluate the performance of proposed RBF neural classifier. The performance of the RBF neural networks classifier is tested with two different databases: Iris and Wine among the different data sets available from the machine learning community by the University of California at Irvine (UCI) [25].

The first test is carried out with Iris database which is one of the most popular data set to examine the performance of novel methods in pattern recognition and machine learning. It is composed of three classes (i.e., iris Setosa, iris Versicolor and iris Virginica) each having 50 patterns with four features.

The second test is done with the Wine database which includes the results of a chemical analysis of wines produced in different regions of Italy from different grape varieties. The concentration of 13 components are indicated for each of the 178 wines (patterns) which are analyzed and divided into three classes (59 in Class 1, 71 in Class 2 and 48 in Class 3).

To evaluate the proposed classifier performances, the holdout method is used. It consists on dividing the initial data into two independent sets: one for training and the other for testing the classifier performances.

The results given by the RBF classifier built with our algorithm are compared with those obtained with other neural classifiers: the Learning Vector Quantization (LVQ) classifier proposed by Kohonen, the RBF neural network classifier for which the hidden layer is obtained using adaptive Pattern Classifier (APCIII) [26], the Multi-Layer Perceptrons classifier (MLP) and with a reference one, the K nearest Neighbor (KNN). The present comparative results of different classifiers over Iris and Wine are illustrated in *Table I* and *Table II*.

Considering Wine database, the best recognition rate is obtained by the KMD-LS-IDB proposed classifier. For Iris database the best recognition rate is given for the KNN classifier, however the difference with the proposed classifier is not important.

Then, the proposed algorithm gives good results in term of recognition rate but the most powerful of them is the classifier KMD-LS-IDB.

TABLE I. RESULTS OF THE RECOGNITION RATE OVER IRIS DATABASE

Classification algorithms	Database : Iris
KMD-LS-IDB	93.46 %
LVQ	94,00 %
APCIII	93,33%
MLP	96,66 %
KNN	96,70 %

TABLE II. RESULTS OF THE RECOGNITION RATE OVER WINE DATABASE

Classification algorithms	Database : Wine
KMD-LS-IDB	98,88 %
LVQ	66,14 %
APCIII	67,04 %
MLP	73,80 %
KNN	70,45 %

VI. CONCLUSION

In this paper, a new algorithm is proposed, to design RBF neural network classifier and to select the centers of the hidden layer neurons in particular.

The basic idea of this approach is to select the training data from the database class by class and to decide of the optimal number of neurons in each class by using the validity index of Davies-Bouldin which is integrated in the K-means algorithm with the Mahalanobis distance.

We also proposed a solution to overcome the problem of initialization of centers necessary for the start of the K-means algorithm using the method of the logarithmic spiral golden angle. The obtained classifier results are satisfactory in comparison with other considered classifiers in the literature for two real databases (Iris and Wine).

REFERENCES

[1] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," Complex Syst., vol. 2, pp. 321–355, 1988.

[2] G. P. Zhang, "Neural networks for classification: a survey," IEEE Trans. Syst. Man Cybern. Part C (Applications Rev., vol. 30, no. 4, pp. 451–462, 2000.

[3] M. Verleysen and K. Hlavackova, "An optimized RBF network for approximation of functions," Neural Networks, no. April, pp. 175–180, 1994.

[4] T. Y. Kwok and D. Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," IEEE Trans. Neural Networks, vol. 8, no. 3, pp. 630–645, 1997.

[5] J. K. Sing, D. K. Basu, M. Nasipuri, and M. Kundu, "Improved k-means algorithm in the design of RBF neural networks," Conf. Converg. Technol. Asia-Pacific Reg. TENCON, pp. 841–845, 2003.

[6] S. Song, Z. Yu, and X. Chen, "A novel radial basis function neural network for approximation," Int. J. Inf. …, vol. 11, no. 9, pp. 46–53, 2005.

[7] J. Moody and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," Neural Comput., vol. 1, pp. 281–294, 1989.

[8] R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Trans. Neural Networks, vol. 16, no. 3, pp. 645–678, 2005.

[9] M. Vogt, "Combination of Radial Basis Function Neural Networks with Optimized Learning Vector Quantization," Proc. ICNN'93, Int. Conf. Neural Networks, vol. III, pp. 1841–1846, 1993.

[10] M. Kubat, "Decision trees can initialize radial-basis function networks," IEEE Trans. Neural Networks, vol. 9, pp. 813–821, 1998.

[11] N. Siddique and H. Adeli, "Spiral Dynamics Algorithm," vol. 23, no. 6, pp. 1–24, 2014.

[12] R. H. Bacon, "Logarithmic Spiral : An Ideal Trajectory for the Interplanetary Vehicle with Engines of Low Sustained Thrust," vol. 164, no. 1959, pp. 164–166, 2012.

[13] G. Markowsky, "Misconceptions about the Golden Ratio," Coll. Math. J., vol. 23, no. 1, pp. 2–19, 1992.

[14] L. D. G. Sigalotti and A. Mejias, "The golden ratio in special relativity," Chaos, Solitons & Fractals, vol. 30, no. 3, pp. 521–524, 2006.

[15] R. J. Praga-Alejo, L. M. Torres-Treviño, D. S. González-González, J. Acevedo-Dávila, and F. Cepeda-Rodríguez, "Analysis and evaluation in a welding process applying a Redesigned Radial Basis Function," Expert Syst. Appl., vol. 39, no. 10, pp. 9669–9675, 2012.

[16] A. Sifaoui, A. Abdelkrim, and M. Benrejeb, "On the Use of Neural Network as a Universal Approximator," Int. J. Sci. Tech. Autom. Control Comput., vol. 2, no. July, pp. 386–399, 2008.

[17] A. Sifaoui, A. Abdelkrim, and M. Benrejeb, "On New RBF Neural Network Construction Algorithm for Classification," SIC, vol. 18, no. 2, pp. 103–110, 2009.

[18] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker, "Convergence Theory for Fuzzy C-Means: Counterexamples and Repairs," Syst. Man Cybern. IEEE Trans., vol. 17, no. 5, pp. 873–877, 1987.

[19] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Clustering algorithms and validity measures," Proc. Thirteen. Int. Conf. Sci. Stat. Database Manag. SSDBM 2001, pp. 3–22, 2001.

[20] M. Halkidi and M. Vazirgiannis, "Clustering Validity Assessment : Finding the optimal partitioning of a data set," in In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, 2001, pp. 187–194.

[21] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," J. Intell. Inf. Syst., vol. 17, no. 2–3, pp. 107–145, 2001.

[22] M. Sassi, A. G. Touzi, and H. Ounelli, "Using Gaussians Functions to Determine Representative Clustering Prototypes," pp. 1–5, 2006.

[23] D. L. Davies and D. W. Bouldin, "A cluster separation measure.," IEEE Trans. Pattern Anal. Mach. Intell., vol. 1, no. 2, pp. 224–227, 1979.

[24] C. Ding and X. He, "K-means Clustering via Principal Component Analysis," Int. Conf. Mach. Learn., 2004.

[25] C. L. Blake and C. J. Merz, "UCI Repository of machine learning databases," University of California. p. http://archive.ics.uci.edu/ml/, 1998.

[26] Y. Hwang and S. Bang, "An Efficient Method to Construct a Radial Basis Function Neural Network Classifier," Neural Networks, 1997.