Incremental Radial Basis Function Computation for Neural Networks

Vaclav Skala Department of Computer Science and Engineering Faculty of Applied Sciences, University of West Bohemia CZ 306 14 Plzen, Czech Republic skala@kiv.zcu.cz

Abstract: - This paper present a novel approach for incremental computation of Radial Basis Functions (RBF) for Fuzzy Systems and Neural Networks with computational complexity of $O(N^2)$ is presented. This technique enables efficient insertion of new data and removal of selected or invalid data. RBF are used across many fields, including geometrical, image processing and pattern recognition, medical applications, signal processing, speech recognition, etc. The main prohibitive factor is the computational cost of the RBF computation for larger data sets or if data set is changed and RBFs have to be recomputed.

The presented technique is applicable in general to fuzzy systems as well offering a significant speed up due to lower computational complexity of the presented approach. The *Incremental RBF Computation* enables also fast RBF recomputation on "sliding window" data due to fast insert/remove operations. This is a very significant factor especially in guided Neural Networks case. Generally, interpolation based on RBF is very often used for scattered scalar data interpolation in *n*-dimensional space. As there is no explicit order in data sets, computational complexity of RBF for *N* values is of $O(N^3)$ or $O(k N^2)$, *k* is a number of iterations if an iterative method is used, which is prohibitive for many real applications. The inverse matrix can also be computed by the Strassen algorithm based on matrix block notation with $O(N^{2.807})$ complexity. Even worst situation occurs when interpolation has to be made over non-constant data sets, as the whole set of equations for determining RBFs has to be recomputed when data set is changed. This situation is typical for applications in which some values are becoming invalid and new values are acquired.

Key-Words - RBF, interpolation, incremental computation, neural networks, fuzzy systems, algorithm, matrix inversion.

1 Introduction

Radial basis functions interpolation was originally introduced by [Hardy 1971] by introduction of multiquadric method, which he called Radial Basis Function (RBF) method, which is based on interpolation formula

$$f(x) = \sum_{i=1}^{N} \lambda_i \, \phi(r_i)$$

where: $\phi(r_i) = \phi(||\mathbf{x} - \mathbf{x}_i||)$ and \mathbf{x} is generally *n*dimensional vector and λ_i are weights. Since then many different RFBF interpolation schemes have been developed with some specific properties, e.g. [Duchon 1977] uses $\phi(r) = r^2 lg r$, which is called Thin-Plate Spline (TPS), a function $\phi(r) = e^{-(\epsilon r)^2}$ was proposed by [Shagen 1979] and [Wetland 2005] introduced Compactly Supported RBF (CSRBF) as

$$\phi(r) = \begin{cases} (1-r)^q \ P(r), \ 0 \le r \le 1\\ 0, \ r > 1 \end{cases}$$

where: P(r) is a polynomial function and q is a parameter. An analysis of algorithms of RBF for neural networks can be found in [Stastny et all 2007].

Theoretical problems with stability and solvability were solved by [Micchelli 1986] and [Wright 2003] and he has extended the RBF by adding a polynomial function $P_k(x)$ of degree k to the RBF that resulted to:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + P_k(\mathbf{x})$$
$$= \sum_{i=1}^{N} \lambda_i \phi_i(\mathbf{x}) + P_k(\mathbf{x})$$

and additional conditions were introduced:

$$\sum_{i=1}^N \lambda_i = 0 \qquad \qquad \sum_{i=1}^N \lambda_i \, \boldsymbol{x} = \boldsymbol{0}$$

Usually a linear polynomial is used, i.e. the polynomial $P_k(\mathbf{x})$ is taken as

$$P_k(\boldsymbol{x}) = a_0 + \boldsymbol{a}^T \, \boldsymbol{x}$$

As the values $f(\mathbf{x}_i)$ at points \mathbf{x}_i are known, the equations above form a system of linear equations that has to be solved in order to determine coefficients λ_i and a_0, \mathbf{a} , i.e.

$$f(\mathbf{x}_j) = \sum_{i=1}^N \lambda_i \phi(\|\mathbf{x}_j - \mathbf{x}_i\|) + P_k(\mathbf{x}_j)$$
$$= \sum_{i=1}^N \lambda_i \phi_{i,j} + P_k(\mathbf{x}_j)$$
$$j = 1, \dots, n$$

It can be seen that for *n*-dimensional case and *N* points given a system of (N + n + 1) has to be solved, where *N* is a number of points in the dataset and *n* is dimensionality of data. For n=2 vectors \mathbf{x}_i and \mathbf{a} are given as $\mathbf{x}_i = [x_i, y_i]^T$ and $\mathbf{a} = [a_x, a_y]^T$.

Using the matrix notation we can write for 2-dimensions:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,N} & x_N & y_N & 1 \\ x_1 & \cdots & x_N & 0 & 0 & 0 \\ y_1 & \cdots & y_N & 0 & 0 & 0 \\ 1 & \cdots & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ a_y \\ a_y \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \\ a_y \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} B & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \quad Ax = b$$
$$a^T x_i + a_0 = a_x x_i + a_y y_i + a_0$$

It can be seen that for 2-dimensional case and N points given a system of (N + 3) linear equations has to be solved. If "global" functions, e.g. TPS $(\phi(r) = r^2 lg r)$, are used the matrix **B** is "full", if CSRBF functions are used, the matrix **B** can be sparse.

Applications of RBF in Fuzzy Systems and Neural Networks are widely known, e.g. [Hunt et al 1996], [Gonzalez et al 2003], [Maglogiannis et al. 2008] and [Berthold et al 1998]. Unfortunately they use standard RBF computation.



Fig.1: Surface reconstruction (438000 points) [Carr et al. 2001]



Original image Reconstructed image [Bertalmio et al. 2000] [Uhlir and Skala 2006] **Fig.2:** Reconstruction of inpainting



Fig.3a: Original image with 60% of damaged pixels



Fig.3b: Reconstructed image

There are also other interesting problems that can be solved using RBF interpolation quite effectively, e.g. surface reconstruction from scattered data [Carr et al. 2001], [Ohtake et al. 2005], reconstruction of damaged images [Uhlir and Skala 2006], [Zapletal et al. 2009], inpainting removal [Bertalmio et al. 2000], [Wang and Kwok 2009] etc.

Let us consider, for simplicity, geometrical analogy. It should be noted that in geometrically based applications we have to handle many and many points, so extreme efficiency is needed.

All those applications of RBFs based interpolation has one significant disadvantage – the cost of

computation. This is especially severe in applications where data are not static.

Generally we are looking for RBF interpolation of z=f(x,y) in two dimensional case, where *x*, *y* are "points", i.e. independent variables while *z* is the value associated with this point.

There are actually two cases:

- 1. **Position of points is fixed**, but the value associated with a point is changed. In this case iterative methods are usually faster than explicit computation of an inverse matrix.
- 2. **Position of points is changed**. It means that the whole system of linear equations has to be form and recomputed which leads generally to $O(N^3)$ computational complexity and unacceptable time consuming computation.

In some applications a "*sliding window*" on data is required, especially in time related applications, when old data should not be used in the interpolation and new data should be included. This is typical situation in signal processing applications.

Considering facts above there is a questions how to compute RBF incrementally with a lower computational complexity?. This question will be answered in the following section.

2 Incremental RBF computation

The main question to be answered is:

Is it possible to use already computed RFB interpolation if a new point is included to the data set?

If the answer is positive it should lead to significant decrease of computational complexity. In the following we will present how a new point can be inserted (new data given), a selected point can be removed (non-relevant data are to be removed) and also how to select the best candidate for a removal according to an error caused by this point removal.

The above mentioned operation are needed especially in neural networks and they lead to change of the dimensionality of the matrix A, resp. M.

Sherman-Morrison formula

Sherman-Morrison formula states, that inverse matrix of *A* perturbed by $\boldsymbol{u} \otimes \boldsymbol{v} = \boldsymbol{u} \, \boldsymbol{v}^T$ (note that it is column x row operation – the result is a matrix) can be computed as

$$(\boldsymbol{A} + \boldsymbol{u} \otimes \boldsymbol{v})^{-1} = \boldsymbol{A}^{-1} \frac{(\boldsymbol{A}^{-1}\boldsymbol{u}) \otimes (\boldsymbol{v}^{T}\boldsymbol{A}^{-1})}{1 + \lambda}$$

where: $\lambda = \boldsymbol{v}^{T}\boldsymbol{A}^{-1}\boldsymbol{u}$.

Please, see section 2.1 for actual matrix structure, now. We can see that the matrix $\boldsymbol{u} \otimes \boldsymbol{v}$ is a very special as it will contain only last column and last row of non-zero values generally.

Then, in our case, the incremental computation is made actually in 3 steps:

- Generation of a new matrix **B** as an extension of the matrix **A** by one row and one column so that $B_{i,j} = \begin{cases} A_{i,j} & 1, \dots n \\ 1 & i,j = (n+1,n+1) \\ 0 & \text{othewise} \end{cases}$ **B**_1 = **B** + last row of **u** $\otimes v$
- $B_2 = B_1 + \text{last column of } u \otimes v$

Now we can apply the original Sherman-Morison formula for the case that a new point – new data – is inserted to the data set. The problem is with the removal of improper point- improper data from the given set. The Sherman-Morrison formula is aimed for general the perturbation case.

In the case of the incremental RBF interpolation we do have a very specific case, where the matrix A is symmetric with a very special structure, as it was presented above. This should lead to more efficient computational method.

Let us consider some operations with block matrices (we will assume that all operations are correct and matrices are non-singular in general etc.).

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

Let us consider a matrix M of $(n+1) \times (n+1)$ and a matrix A of $n \times n$ in the following block form:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^T & \boldsymbol{c} \end{bmatrix}$$

Then the inverse of the matrix M applying the rule above can be written as:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \left(\boldsymbol{A} - \frac{1}{c}\boldsymbol{b}\boldsymbol{b}^{T}\right)^{-1} & -\frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b} \\ -\frac{1}{k}\boldsymbol{b}^{T}\boldsymbol{A}^{-1} & \frac{1}{k} \end{bmatrix}$$

$$= \begin{bmatrix} A^{-1} + \frac{1}{k}A^{-1}bb^{T}A^{-1} & -\frac{1}{k}A^{-1}b \\ -\frac{1}{k}b^{T}A^{-1} & \frac{1}{k} \end{bmatrix}$$

where: $k = c - \boldsymbol{b}^T \boldsymbol{A}^{-1} \boldsymbol{b}$

We can easily simplify this equation if the matrix *A* is symmetrical as:

$$\boldsymbol{\xi} = \boldsymbol{A}^{-1}\boldsymbol{b} \qquad \boldsymbol{k} = \boldsymbol{c} - \boldsymbol{\xi}^{T}\boldsymbol{b}$$
$$\boldsymbol{M}^{-1} = \frac{1}{k} \begin{bmatrix} \boldsymbol{k}\boldsymbol{A}^{-1} + \boldsymbol{\xi} \otimes \boldsymbol{\xi}^{T} & -\boldsymbol{\xi} \\ -\boldsymbol{\xi}^{T} & 1 \end{bmatrix}$$

where: $\boldsymbol{\xi} \otimes \boldsymbol{\xi}^{T}$ means the tensor multiplication. It can be seen that all computations needed are of $O(N^2)$ computational complexity.

It means that we can compute an inverse matrix incrementally with $O(N^2)$ complexity instead of $O(N^3)$ complexity required originally in this specific case. It can be seen that the structure of the matrix M is "similar to the matrix of the RBF specification.

Now, there is a question how the incremental computation of an inverse matrix can be used for RBF interpolation?

We know that the matrix **A** in the equation Ax = b is symmetrical and non-singular if appropriate rules for RBFs are kept.

2.1 Point Insertion

Let us imagine a simple situation. We have already computed the interpolation for N points and we need to include a new point into the given data set. A brute force approach of full RBF computation on the new data set can be used with $O(N^3)$ complexity computation.

Let us consider RBF interpolation for N+1 points and the following system of equations is obtained:

$$\begin{bmatrix} \phi_{1,1} & \dots & \phi_{1,N} & \phi_{1,N+1} & x_1 & y_1 & 1 \\ \vdots & \dots & \vdots & \vdots & \vdots & 1 \\ \phi_{N,1} & \vdots & \phi_{N,N} & \phi_{N,N+1} & x_N & y_N & 1 \\ \phi_{N+1,1} & \phi_{N+1,N} & \phi_{N+1,N+1} & x_{N+1} & y_{N+1} & 1 \\ x_1 & \dots & x_N & x_{N+1} & 0 & 0 & 0 \\ y_1 & \dots & y_N & y_N & 0 & 0 & 0 \\ 1 & \dots & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ \lambda_{N+1} \\ a_x \\ a_y \\ a_0 \end{bmatrix}$$
$$= \begin{bmatrix} f_1 \\ \vdots \\ f_N \\ f_{N+1} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where: $\phi_{i,i} = \phi_{j,i}$

Reordering the equations above we get:

$$\begin{bmatrix} 0 & 0 & 0 & x_{1} & \dots & x_{N} & x_{N+1} \\ 0 & 0 & 0 & y_{1} & \dots & y_{N} & y_{N+1} \\ 0 & 0 & 0 & 1 & \dots & 1 & 1 \\ x_{1} & y_{1} & 1 & \phi_{1,1} & \dots & \phi_{1,N} & \phi_{1,N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N} & y_{N} & 1 & \phi_{N,1} & \dots & \phi_{N,N} & \phi_{N,N+1} \\ x_{N+1} & y_{N+1} & 1 & \phi_{N+1,1} & \dots & \phi_{N+1,N} & \phi_{N+1,N+1} \end{bmatrix} \begin{bmatrix} a_{x} & a_{y} \\ a_{y} \\ a_{0} \\ \lambda_{1} \\ \vdots \\ \lambda_{N} \\ \lambda_{N+1} \end{bmatrix}$$

We can see that last row and last column is "inserted". As RBF functions are symmetrical the recently derived formula for iterative computation of the inverse function can be used. So the RBF interpolation is given by the matrix M as

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^T & \boldsymbol{c} \end{bmatrix}$$

where the matrix A is the RBF matrix $(N+3) \times (N+3)$ and the vector b (N+3) and scalar value c are defined as:

$$\boldsymbol{b} = [x_{N+1} \ y_{N+1} \ 1 \ \phi_{1,N+1} \ \dots \ \phi_{N,N+1}]^T$$

 $\boldsymbol{c} = \phi_{N+1,N+1}$

It means that we know how to compute the matrix M^{-1} if the matrix A^{-1} is known.

That is exactly what we wanted!

Recently we have proved that iterative computation of inverse function is of $O(N^2)$ complexity, that offers a significant performance improvement for points insertion. It should be noted that some operations can be implemented more effectively, especially $\xi \otimes \xi^T = A^{-1}bb^T A^{-1}$ as the matrix A^{-1} is symmetrical etc.

2.2 Point Removal

In some cases it is necessary to remove a point from the given data set. It is actually an inverse operation to the insertion operation described above. Let us consider a matrix M of the size $(N+1) \times (N+1)$ as

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^T & \boldsymbol{c} \end{bmatrix}$$

Now, the inverse matrix M^{-1} is known and we want to compute matrix A^{-1} , which is of the size $N \times N$.

Recently we derived opposite rule:

$$M = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix}$$
$$\xi = A^{-1}b \qquad k = c - \xi^T b$$
$$M^{-1} = \begin{bmatrix} A^{-1} + \frac{1}{k} \xi \otimes \xi^T & -\frac{1}{k} \xi \\ -\frac{1}{k} \xi^T & \frac{1}{k} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

It can be seen that

$$\boldsymbol{Q}_{11} = \boldsymbol{A}^{-1} + \frac{1}{k}\boldsymbol{\xi}\otimes\boldsymbol{\xi}^{T}$$

and therefore

$$\boldsymbol{A}^{-1} = \boldsymbol{Q}_{11} - \frac{1}{k} \boldsymbol{\xi} \otimes \boldsymbol{\xi}^{T}$$

Now we have both operations, i.e. insertion and removal, with effective computation of $O(N^2)$ computational complexity instead of $O(N^3)$. It should be noted that vectors related to the point assigned for a removal must be in the last row and last column of the matrix M^{-1} .

2.3 Point selection

As the number of points within the given data set could be high, the point removal might be driven by a requirement of removing a point which causes a *minimal error of the interpolation*. This is a tricky requirement as there is probably no general answer. The requirement should include additional information which interval of x is to be considered.

Generally we have a function

$$f(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \, \phi_i(\mathbf{x}) + P_k(\mathbf{x})$$

and we want to remove a point x_j which causes a minimal error ε_j of interpolation, i.e.

$$f_j(\boldsymbol{x}) = \sum_{i=1, i \neq j}^N \lambda_i \phi_i(\boldsymbol{x}) + P_k(\boldsymbol{x})$$

and we want to minimize

$$\varepsilon_j = \int_{\Omega} |f(\mathbf{x}) - f_j(\mathbf{x})| d\mathbf{x}$$

where Ω is the interval on which the interpolation is to be made. It means that if the point x_j is removed the error ε_j is determined as:

$$\varepsilon_j = \lambda_j \int_{\Omega} \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) d\boldsymbol{x}$$

As we know the interval Ω on which the interpolation is to be used, we can compute or estimate the error ε_i for each point x_i in the given

data set and select the best one. For many functions ϕ the error ε_j can be computed or estimated analytically as the evaluation of ε_j is simple for many functions, e.g.

Vaclav Skala

$$\int r^m \ln dr = r^{m+1} \frac{\ln r}{m+1} - \frac{1}{(m+1)^2}$$

It means that for TPS function $r^2 \ln r$ the error ε_k is easy to evaluate. In the case of CSRBF the estimation is even simpler as they have a limited influence, so generally λ_i determines the error ε_i .

It should be noted, that a selection of a point with the lowest influence to the interpolation precision in the given interval Ω is of O(N) complexity only.

We have shown a novel approach to RBF computation which is convenient for larger data sets. It is especially convenient for t-varying data and for applications, where a "sliding window" is used. Basic operations – point insertion and point removal – have been introduced. These operations have $O(N^2)$ computational complexity only, which makes a significant difference from the original approach used for RBFs computation. Fig. 4 presents differences of computational time if one point is added or removed to the given set with *N* points. Fig.5 presents actual speed up defined as a ratio of the standard and incremental computation.



Fig.4: Comparison of computational time [ms] for standard and incremental method Dimension means number of points



Fig.5: Speed up for incremental computation (1 point added /removed)

3 **Implementation aspects**

Radial basis functions are very powerful method as it enables interpolation of scattered data generally in k-dimensional space.

As experiments proved the incremental computation of RBF is significantly faster than the :standard: one.

Of course, it requires more computational power than methods specialized on order data sets. Unorganized data sets are very typical for some applications like neural networks, data mining, computer graphics etc. As computer graphics applications process huge amount of data, typically 10^{6} - 10^{7} if points, we used computer graphics to evaluate behavior of the RBF interpolation on large data sets. A nice example of RBF use is a reconstruction of images, as it is two dimensional, damaged pixels are un-ordered, the complexity of computation grows with the square of the image resolution. As the image reconstruction is an iterative process the RBF interpolation is called many, many times. The extension of the RBF interpolation to k-dimensional space is simple and the matrix of RBF grows $O(N^2)$, i.e. if the we change the image size from 1000x1000 to 2000x2000 we have to count with 4-times more elements of the matrix for RBF interpolation.

As the computational requirements grow fast, there is a possibility to use Graphical processing Unit (GPU) available on all PCs and notebooks today. First experiments we made proved that speed up on single GPU can be achieved by a factor 10-200.

Also, there are "personal supercomputers" based on GPU like TESLA/FERMI from NVIDIA. In this case significant speed up can be achieved as well.

4 Conclusion

The proposed Incremental RBF Computation method has advantages over the standard techniques based RBF interpolation used in Fuzzy systems and Neural Networks due to incremental insertion and/or removal of points with decreased computational complexity from $O(N^3)$ to $O(N^2)$. It enables to apply this approach in applications when interpolation of data in a "sliding window" and / or t-varying interpolation data are required; in applications when some data are becoming invalid, new data are acquired and need to be included into the processed data set, i.e. in typical cases of RBF use in Fuzzy Systems or Neural Networks. Due to lower computational complexity it is possible to handle larger data sets in which scalar values are associated with t-varying points, i.e. it is possible to handle non-static data.

It is expected that the presented approach can lead to development of new algorithms in Neural Networks and Fuzzy Systems. As the proposed *Incremental RBF Computation* uses vector – matrix operations exclusively, the presented approach is suitable for matrix-vector architectures including GPU/Larabee architectures as well.

5 Acknowledgments

The author thanks to colleagues at the University of West Bohemia for their critical comments, suggestions, to anonymous reviewers for their critical view and comments that improved the manuscript significantly. Special thanks belong to former PhD and MSc. students, especially to Vit Ondracka, for their help with implementations and recent experiments made.

The research was supported by the Ministry of Education of the Czech Republic, project LC-CPG No. LC06008.

6 **References**

- [1] Baxter,B.J.Ch: The Interpolation Theory of Radial Basis Functions, PhD thesis, Trinity College, Cambridge University, U.K., 1992
- [2] Bertalmio, M., Sapiro, G., Ballester, C. and V. Caselles, V.: Image Inpainting, Proceedings of SIGGRAPH'00, Computer Graphics, New Orleans, 23-28 July 2000, pp.417-424.
- [3] Berthold, M.R., Diamond,J.: Constructive training of probabilistic neural networks, Neurocomputing 19, Vol.19., pp.167-183, 1998

- [4] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., Evans, T. R.: "Reconstruction and representation of 3D objects with radial basis functions", Computer Graphics (SIGGRAPH 2001 proceedings), pp. 67-76, August 2001.
- [5] Duchon, J.: "Splines minimizing rotationinvariant semi-norms in Sobolev space", Constructive Theory of Functions of Several Variables, Springer Lecture Notes in Math, 21 (1977), pp. 85-100.
- [6] Hardy, L.R.: Multiquadric equations of topography and other irregular surfaces, J. Geophy. Res., 76 (1971), pp. 1905-1915.
- Hunt,K.J., Hass,R., Murray-Smith,R.:
 Extending the Functional Equivalence of Radial Basis Functions and Fussy Interference Systems, IEEE Trans. on Neural Networks, Vol.7., No.3, pp.766-781, 1996
- [8] Gonzales, J., Rojas, I., Ortega, J., Pomares, H., Fernandez, J., Diaz, A.: Multiobjective Evolutionary Optimization of the Size Shape and Position para, eters of Radial Baisi Function network for Function Approximation, IEEE Trans. on Neural Networks, Vo.l. 14, No.6., pp. 1478-1495, 2003
- [9] Maglogiannis,I., Sarimveis, H., Kiranoudis, C.T., Chatziioannou, A.A., Oikonomou, N., Aidinis,V.: Radial Basis Function Neural networks Classification of the Recognition of Idiopathic Pumonary Fibrosis in Microscopic Images, IEEE Trans. on Information Technology in Biomedicine, Vol.12, No.1, pp.42-54, 2008
- [10] Micchelli, C.A.: "Interpolation of scattered data: distance matrice and conditionally positive definite functions", Constr. Approx., 2 (1986), pp. 11-22.
- [11] Ohtake, Y., Belyaev, A., Seidel, H.-P.: 3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs, Graphical Models, Vol.67, No.3., pp.150-165, 2005.
- [12] Or,M.J.L.: Local Smoothing of Radial Basis Function Network, Int.Symposium on Artificial Neural Network, Hsinchu, Taiwan, 1995.

- [13] Pan,R.J., Skala,V.: Implicit surface modeling suitable for inside/outside tests with radial basis functions, 10th International Conference on Computer Aided Design and Computer Graphics (CAD/Graphics 2007)
- [14] Schagen, I.P.: Interpolation in Two Dimension – A New Technique, J. Inst. Maths Applics, 23 (1979), pp. 53-59.
- [15] Stastny, J., Skorpil, V.: Analysis of Algorithms for Radial Basis Function Neural Network, Springer Boston, ISBN 978-0-387-74158-1, pp. 54-62, 2007
- [16] Uhlir,K., Skala,V.: Radial basis function use for the restoration of damaged images. *In* Computer vision and graphics. Dordrecht: Springer , 2006. s. 839-844. ISBN 1-4020-4178-0
- [17] Wang,Ch.C.L., Kwok,T.-H.: Interactive Image Inpainting using DCT Based Expemplar Matching, ISVC 2009, LNCS 5876, pp.709-718, 2009
- [18] Wendland, H.: Computational aspects of radial basis function approximation, in K. Jetter et al. (eds.) *Topics in Multivariate Approximation and Interpolation*, Elsevier B.V., pp. 231-256, 2005.
- [19] Wright, G.B., "Radial Basis Function Interpolation: Numerical and Analytical Developments", University of Colorado, Thesis, 2003.
- [20] Zapletal, J., Vanecek, P., Skala, V.: RBF-based Image Restoration Utilizing Auxiliary Points, CGI 2009 proceedings, ACM ISBN 978-60558-687-8, pp.39-44, 2009.

WEB references

- FastRBF: http://www.farfieldtechnology.com/
- Brenner,S., Sung,L.-Y.: The Sherman-Morison Formula, <u>http://www.math.sc.edu/calclab/526-F05/lab07-PartI.pdf</u>, 2005
- Mathworld: <u>http://mathworld.wolfram.com/Sherman-MorrisonFormula.html</u>
- Wiki: <u>http://en.wikipedia.org/wiki/Sherman-</u> <u>Morrison_formula</u>

Additional references (related but not directly) which could be of a reader interest if more deeply interested in RBF

- Face recognition using point symmetry distance-based RBF network
- Applied Soft Computing, Volume 7, Issue 1, January 2007, Pages 58-70
- Jamuna Kanta Sing, Dipak Kumar Basu, Mita Nasipuri, Mahantapas Kundu

Corrective action planning using RBF neural network

- Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 1055-1063
- Daya Ram, Laxmi Srivastava, Manjaree Pandit, Jaydev Sharma

Suitability of different neural networks in daily flow forecasting

Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 968-978

- Pankaj Singh, M.C. Deo
- Application of neural networks in forecasting engine systems reliability
- Applied Soft Computing, Volume 2, Issue 4, February 2003, Pages 255-268
- K. Xu, M. Xie, L. C. Tang, S. L. Ho
- A predictive and probabilistic load-balancing algorithm for cluster-based web servers
- Applied Soft Computing, In Press, Corrected Proof, Available online 25 January 2010
- Saeed Sharifian, Seyed A. Motamedi, Mohammad K. Akbari
- Technical data mining with evolutionary radial basis function classifiers
- Applied Soft Computing, Volume 9, Issue 2, March 2009, Pages 765-774
- Markus Bauer, Oliver Buchtala, Timo Horeis, Ralf Kern, Bernhard Sick, Robert Wagner
- Radial Basis Function Network based on Time Variant Multi-Objective Particle Swarm Optimization for Medical Diseases Diagnosis
- Applied Soft Computing, In Press, Accepted Manuscript, Available online 5 May 2010
- Sultan Noman Qasem, Siti Mariyam Shamsuddin
- Clustered-Hybrid Multilayer Perceptron network for pattern recognition application
- Applied Soft Computing, In Press, Corrected Proof, Available online 5 May 2010
- Nor Ashidi Mat Isa, Wan Mohd Fahmi Wan Mamat
- Filter design using radial basis function neural network and genetic algorithm for improved operational health monitoring
- Applied Soft Computing, Volume 6, Issue 2, January 2006, Pages 154-169
- Niranjan Roy, Ranjan Ganguli
- Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations

- Applied Soft Computing, Volume 9, Issue 1, January 2009, Pages 20-29
- Yazdan Shirvany, Mohsen Hayati, Rostam Moradian
- Estimation of current-induced scour depth around pile groups using neural network and adaptive neuro-fuzzy inference system
- Applied Soft Computing, Volume 9, Issue 2, March 2009, Pages 746-755
- Mohammad Zounemat-Kermani, Ali-Asghar Beheshti, Behzad Ataie-Ashtiani, Saeed-Reza Sabbagh-Yazdi

Soft computing system for bank performance prediction

- Applied Soft Computing, Volume 8, Issue 1, January 2008, Pages 305-315
- V. Ravi, H. Kurniawan, Peter Nwee Kok Thai, P. Ravi Kumar
- Prediction of a Lorenz chaotic attractor using two-layer perceptron neural network
- Applied Soft Computing, Volume 5, Issue 4, July 2005, Pages 333-355

Sanjay Vasant Dudul

- A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems
- Applied Soft Computing, In Press, Corrected Proof, Available online 24 December 2009
- Y.F. Li, S.H. Ng, M. Xie, T.N. Goh
- The use of computational intelligence in intrusion detection systems: A review
- Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 1-35

Shelly Xiaonan Wu, Wolfgang Banzhaf

- Classification of communications signals using an advanced technique
- Applied Soft Computing, In Press, Corrected Proof, Available online 16 December 2009
- A. Ebrahimzadeh, S.E. Mousavi
- Extrapolation detection and novelty-based node insertion for sequential growing multi-experts network
- Applied Soft Computing, Volume 3, Issue 2, September 2003, Pages 159-175
- Loo Chu Kiong, Mandava Rajeswari, M. V. C. Rao
- An approach to mitigate the risk of voltage collapse accounting uncertainties using improved particle swarm optimization
- Applied Soft Computing, Volume 9, Issue 4, September 2009, Pages 1197-1207
- L.D. Arya, L.S. Titare, D.P. Kothari
- Hybrid fuzzy set-based polynomial neural networks and their development with the aid of genetic optimization and information granulation
- Applied Soft Computing, Volume 9, Issue 3, June 2009, Pages 1068-1089
- Sung-Kwun Oh, Witold Pedrycz, Seok-Beom Roh
- Interval type-2 fuzzy logic and modular neural networks for face recognition applications
- Applied Soft Computing, Volume 9, Issue 4, September 2009, Pages 1377-1387
- Olivia Mendoza, Patricia Melín, Oscar Castillo

Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 208-214

- Xiao-ning Song, Yu-jie Zheng, Xiao-jun Wu, Xi-bei Yang, Jingyu Yang
- Nonlinear dynamic system identification and control via constructivism inspired neural network
- Applied Soft Computing, Volume 3, Issue 3, November 2003, Pages 237-257

Loo Chu Kiong, Mandava Rajeswari, M. V. C. Rao

- Fuzzy ARTMAP dynamic decay adjustment: An improved fuzzy ARTMAP model with a conflict resolving facility
- Applied Soft Computing, Volume 8, Issue 1, January 2008, Pages 543-554

Shing Chiang Tan, M.V.C. Rao, Chee Peng Lim

- Modified fuzzy c-means and Bayesian equalizer for nonlinear blind channel
- Applied Soft Computing, Volume 9, Issue 3, June 2009, Pages 1090-1096

Soowhan Han, Imgeun Lee, Witold Pedrycz

- Type-2 fuzzy logic-based classifier fusion for support vector machines
- Applied Soft Computing, Volume 8, Issue 3, June 2008, Pages 1222-1231
- Xiujuan Chen, Yong Li, Robert Harrison, Yan-Qing Zhang
- Patent classification system using a new hybrid genetic algorithm support vector machine
- Applied Soft Computing, In Press, Corrected Proof, Available online 24 December 2009

Chih-Hung Wu, Yun Ken, Tao Huang

- Numerical solution of PDEs via integrated radial basis function networks with adaptive training algorithm
- Applied Soft Computing, In Press, Corrected Proof, Available online 25 January 2010

Hong Chen, Li Kong, Wen-Jun Leng

- Direct adaptive neural control for affine nonlinear systems
- Applied Soft Computing, Volume 9, Issue 2, March 2009, Pages 756-764

Indrani Kar, Laxmidhar Behera

Robust incremental growing multi-experts network

Applied Soft Computing, Volume 6, Issue 2, January 2006, Pages 139-153

- Chu Kiong Loo, Mandava Rajeswari, M.V.C. Rao
- Non-linear channel equalization using adaptive MPNN
- Applied Soft Computing, Volume 9, Issue 3, June 2009, Pages 1016-1022
- Sasmita Kumari Padhy, Siba Prasada Panigrahi, Prasanta Kumar Patra, Santanu Kumar Nayak
- An RBFN–Wiener hybrid filter using higher order signal statistics
- Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 915-922

Noriaki Suetake, Eiji Uchino

A transmission line fault locator based on Elman recurrent networks

- Applied Soft Computing, Volume 9, Issue 1, January 2009, Pages 341-347
- Sami Ekici, Selcuk Yildirim, Mustafa Poyraz
- A distributed PSO–SVM hybrid system with feature selection and parameter optimization
- Applied Soft Computing, Volume 8, Issue 4, September 2008, Pages 1381-1391
- Cheng-Lung Huang, Jian-Fan Dun
- Parameter determination of support vector machine and feature selection using simulated annealing approach
- Applied Soft Computing, Volume 8, Issue 4, September 2008, Pages 1505-1512
- Shih-Wei Lin, Zne-Jung Lee, Shih-Chieh Chen, Tsung-Yuan Tseng
- Fast Constructive-Covering Algorithm for neural networks and its implement in classification
- Applied Soft Computing, Volume 8, Issue 1, January 2008, Pages 166-173

Di Wang

- An incremental adaptive neural network model for online noisy data regression and its application to compartment fire studies
- Applied Soft Computing, In Press, Corrected Proof, Available online 25 January 2010
- Eric Wai Ming Lee
- Modified Recursive Least Squares algorithm to train the Hybrid Multilayered Perceptron (HMLP) network
- Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 236-244
- Mohammad Subhi Al-Batah, Nor Ashidi Mat Isa, Kamal Zuhairi Zamli, Khairun Azizi Azizli
- A computer vision approach for weeds identification through Support Vector Machines
- Applied Soft Computing, In Press, Corrected Proof, Available online 25 January 2010
- Alberto Tellaeche, Gonzalo Pajares, Xavier P. Burgos-Artizzu, Angela Ribeiro

CPBUM neural networks for modeling with outliers and noise

Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 957-967

Chen-Chia Chuang, Jin-Tsong Jeng

- Enhancing the classification accuracy by scatter-search-based ensemble approach
- Applied Soft Computing, In Press, Corrected Proof, Available online 4 February 2010
- Shih-Chieh Chen, Shih-Wei Lin, Shuo-Yan Chou
- Facing classification problems with Particle Swarm Optimization
- Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 652-658
- I. De Falco, A. Della Cioppa, E. Tarantino
- Threshold accepting trained principal component neural network and feature subset selection: Application to bankruptcy prediction in banks
- Applied Soft Computing, Volume 8, Issue 4, September 2008, Pages 1539-1548
- V. Ravi, C. Pramodh

- No-reference image quality assessment using modified extreme learning machine classifier
- Applied Soft Computing, Volume 9, Issue 2, March 2009, Pages 541-552
- S. Suresh, R. Venkatesh Babu, H.J. Kim
- Weather analysis using ensemble of connectionist learning paradigms
- Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 995-1004

Imran Maqsood, Ajith Abraham

- Tracking performance control of a cable communicated underwater vehicle using adaptive neural network controllers
- Applied Soft Computing, Volume 10, Issue 3, June 2010, Pages 908-918
- A. Bagheri, T. Karimi, N. Amanifard
- A novel clustering approach: Artificial Bee Colony (ABC) algorithm
- Applied Soft Computing, In Press, Corrected Proof, Available online 22 December 2009

Dervis Karaboga, Celal Ozturk

- A generic optimising feature extraction method using multiobjective genetic programming
- Applied Soft Computing, In Press, Corrected Proof, Available online 26 February 2010

Yang Zhang, Peter I. Rockett

- Employing Multiple-Kernel Supp ort Vector Machines for Counterfeit Banknote Recognition
- Applied Soft Computing, In Press, Accepted Manuscript, Available online 7 May 2010

Chi-Yuan Yeh, Wen-Pin Su, Shie-Jue Lee

- Feature extraction and gating techniques for ultrasonic shaft signal classification
- Applied Soft Computing, Volume 7, Issue 1, January 2007, Pages 156-165
- Kyungmi Lee, Vladimir Estivill-Castro

Power signal classification using dynamic wavelet network

- Applied Soft Computing, Volume 9, Issue 1, January 2009, Pages 118-125
- B. Biswal, P.K. Dash, B.K. Panigrahi, J.B.V. Reddy
- Dynamic classification for video stream using support vector machine
- Applied Soft Computing, Volume 8, Issue 4, September 2008, Pages 1314-1325
- Mariette Awad, Yuichi Motai

Uncertainty measures for fuzzy relations and their applications Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 1135-1143

Daren Yu, Qinghua Hu, Congxin Wu

- A new Kernelized hybrid c-mean clustering model with optimized parameters
- Applied Soft Computing, Volume 10, Issue 2, March 2010, Pages 381-389

Meena Tushir, Smriti Srivastava

- Hybrid ANN reducing training time requirements and decision delay for equalization in presence of co-channel interference
- Applied Soft Computing, Volume 8, Issue 4, September 2008, Pages 1536-1538
- Siba Prasada Panigrahi, Santanu Kumar Nayak, Sasmita Kumari Padhy
- Time sequence data mining using time-frequency analysis and soft computing techniques
- Applied Soft Computing, Volume 8, Issue 1, January 2008, Pages 202-215

P.K. Dash, H.S. Behera, I.W.C. Lee

- Multi-class iteratively refined negative selection classifier
- Applied Soft Computing, Volume 8, Issue 2, March 2008, Pages 972-984
- Urszula Markowska-Kaczmar, Bartosz Kordas
- A Type-2 Neuro-fuzzy System Based on Clustering and Gradient Techniques Applied to System Identification and Channel Equalization
- Applied Soft Computing, In Press, Accepted Manuscript, Available online 2 May 2010
- Rahib H. Abiyev, Okyay Kaynak, Tayseer Alshanableh, Fakhreddin Mamedov

Ex-ray: Data mining and mental health

- Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 923-928
- Joachim Diederich, Aqeel Al-Ajmi, Peter Yellowlees
- Hybrid neural network models for hydrologic time series forecasting
- Applied Soft Computing, Volume 7, Issue 2, March 2007, Pages 585-592

Ashu Jain, Avadhnam Madhav Kumar

- Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach
- Applied Soft Computing, Volume 9, Issue 2, March 2009, Pages 599-607
- Hyunchul Ahn, Kyoung-jae Kim
- Word segmentation of handwritten text using supervised classification techniques
- Applied Soft Computing, Volume 7, Issue 1, January 2007, Pages 71-88
- Yi Sun, Timothy S. Butler, Alex Shafarenko, Rod Adams, Martin Loomes, Neil Davey
- On-line system identification of complex systems using Chebyshev neural networks
- Applied Soft Computing, Volume 7, Issue 1, January 2007, Pages 364-372
- S. Purwar, I.N. Kar, A.N. Jha
- A model updating strategy for predicting time series with seasonal patterns
- Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 276-283
- Jose A. Guajardo, Richard Weber, Jaime Miranda
- A neural network-based sliding-mode control for rotating stall and surge in axial compressors
- Applied Soft Computing, In Press, Corrected Proof, Available online 11 February 2010

J. Javadi Moghaddam, M.H. Farahani, N. Amanifard

- Large population size IGA with individuals' fitness not assigned by user
- Applied Soft Computing, In Press, Corrected Proof, Available online 25 January 2010

Dunwei Gong, Jie Yuan

- Unsupervised adaptive neural-fuzzy inference system for solving differential equations
- Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 267-275
- Hadi Sadoghi Yazdi, Reza Pourreza
- A Clustering-based Differential Evolution for Global Optimization
- Applied Soft Computing, In Press, Accepted Manuscript, Available online 13 May 2010
- Zhihua Cai, Wenyin Gong, Charles X. Ling, Harry Zhang
- Memetic Pareto Evolutionary Artificial Neural Networks to determine growth/no-growth in predictive microbiology
- Applied Soft Computing, In Press, Corrected Proof, Available online 16 December 2009
- J.C. Fernández, C. Hervás, F.J. Martínez-Estudillo, P.A. Gutiérrez

Designing a hierarchical fuzzy logic controller using the differential evolution approach

Applied Soft Computing, Volume 7, Issue 2, March 2007, Pages 481-491

France Cheong, Richard Lai

- Soft computing methods applied to the control of a flexible robot manipulator
- Applied Soft Computing, Volume 9, Issue 1, January 2009, Pages 149-158

B. Subudhi, A.S. Morris

- Development of dynamic cognitive networks as complex systems approximators: validation in financial time series
- Applied Soft Computing, Volume 5, Issue 2, January 2005, Pages 157-179
- D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris, C.D. Zopounidis

Hybrid multi-agent framework for detection of stealthy probes

- Applied Soft Computing, Volume 7, Issue 3, June 2007, Pages 631-641
- Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham
- Non-Mercer hybrid kernel for linear programming support vector regression in nonlinear systems identification
- Applied Soft Computing, Volume 9, Issue 1, January 2009, Pages 94-99
- Zhao Lu, Jing Sun
- Eliciting transparent fuzzy model using differential evolution Applied Soft Computing, Volume 8, Issue 1, January 2008, Pages 466-476

M. Eftekhari, S.D. Katebi, M. Karimi, A.H. Jahanmiri

Incremental modeling with rough and fine tuning method Applied Soft Computing, In Press, Corrected Proof, Available online 16 December 2009

Sung-Suk Kim, Keun-Chang Kwak

- B-spline neural network design using improved differential evolution for identification of an experimental nonlinear process
- Applied Soft Computing, Volume 8, Issue 4, September 2008, Pages 1513-1522

Leandro dos Santos Coelho, Fabio A. Guerra

- Modeling nonlinear elastic behavior of reinforced soil using artificial neural networks
- Applied Soft Computing, Volume 9, Issue 3, June 2009, Pages 954-961

Shouling He, Jiang Li

- Optimal matching by the transiently chaotic neural network
- Applied Soft Computing, Volume 9, Issue 3, June 2009, Pages 863-873

Abdolreza Mirzaei, Reza Safabakhsh

- Direct adaptive neural flight control system for an unstable unmanned aircraft
- Applied Soft Computing, Volume 8, Issue 2, March 2008, Pages 937-948
- S. Suresh, N. Kannan

Competitive fuzzy edge detection

- Applied Soft Computing, Volume 3, Issue 2, September 2003, Pages 123-137
- Lily Rui Liang, Carl G. Looney
- Neural networks modeling of shear strength of SFRC corbels without stirrups
- Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 135-148

Shailendra Kumar, S.V. Barai

- Automatic extraction and identification of chart patterns towards financial forecast
- Applied Soft Computing, Volume 7, Issue 4, August 2007, Pages 1197-1208
- James N.K. Liu, Raymond W.M. Kwong
- A novel ensemble of classifiers for microarray data classification Applied Soft Computing, Volume 8, Issue 4, September 2008,
- Pages 1664-1669 Yuehui Chen, Yaou Zhao

Steady-state image processing

- Applied Soft Computing, Volume 1, Issue 1, June 2001, Pages 53-62
- Mario Köppen, Javier Ruiz-del-Solar
- BCH coding and intelligent watermark embedding: Employing both frequency and strength selection
- Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 332-343

Imran Usman, Asifullah Khan

- Multiple Instance Learning with Multiple Objective Genetic Programming for Web Mining
- Applied Soft Computing, In Press, Corrected Proof, Available online 24 October 2009

Amelia Zafra, Eva L. Gibaja, Sebastián Ventura

A soft computing-based approach for integrated training and rule extraction from artificial neural networks: DIFACONNminer Applied Soft Computing, Volume 10, Issue 1, January 2010, Pages 304-317

Lale Özbakır, Adil Baykasoğlu, Sinem Kulluk

A hybrid learning algorithm with a similarity-based pruning strategy for self-adaptive neuro-fuzzy systems

Applied Soft Computing, Volume 9, Issue 4, September 2009, Pages 1354-1366

Gang Leng, Xiao-Jun Zeng, John A. Keane