Advances in Engineering Software 40 (2009) 913–919

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Visual RBF network design based on Star Coordinates

Otto M. Machado Filho *, Nelson F.F. Ebecken

COPPE/Federal University of Rio de Janeiro, Centro de Tecnologia, P.O. Box 68506, Bloco B – Sala 100, 21945-970 Rio de Janeiro, RJ, Brazil

ARTICLE INFO

Article history:
Received 8 December 2008
Received in revised form 8 December 2008

Keywords:
Neural network design
Radial Basis Function Network
Multi-dimensional visualization
Star Coordinates

ABSTRACT

The performance of neural networks is greatly affected by their design, yet the question of finding optimal designs remains open and inspires a considerable amount of research. Most of the researches have been focused on developing automatic algorithms for neural network configuration. This paper addresses the problem of Radial Basis Function Network (RBFN) design definition with a visual technique, called Star Coordinates. The purpose of this approach is to enable the RBFN design revision and refining process, capitalizing on the power of visualization and interactive operations.

1. Introduction

Radial Basis Function Networks (RBFNs) [1–3] are effective computational tools that have attracted a lot of interest in the areas of systems modeling and pattern recognition. The main RBFN advantages are the simplicity of its structure and the speed of the learning algorithms it employs.

RBFN are commonly trained following a hybrid procedure that operates in two stages, where: the centers locations (hidden nodes) are computed first and the connection weights are calculated in the second stage.

The most popular training technique based on two stage approach [4], uses *k*-means clustering [5–7] to determine the centers locations, whilst the weights of the output layer are trained by a single-shot process using pseudo-inverse matrices [8–10].

Besides the *k*-means based approach, a variety of algorithms have been suggested to determine the configuration of RBF Network automatically and include: orthogonal least squares algorithm [11], constructive methods [12–14] (the network is incrementally built), pruning methods [15–17] (a large number of centers is reduced as the algorithm proceeds) and optimization methods based on genetic algorithms [18–21].

The users are almost excluded from the RBFN configuration process when applying these automatic algorithms, which reduce their work but restrict the process manipulation. Usually, the user sets the parameters before the algorithm execution and repeats the entire process if the results are inadequate.

In this paper, we propose an approach, based on a visual technique called Star Coordinates [22–24], which allows the user to be involved into the RBF Network configuration process via interactive visualization. The Star Coordinates enables a visual RBFN design rendering from any algorithmic results and then lets the user interactively participate in the design evaluation and refining.

The proposed technique may be useful for academic and business areas, although it is expected a better interest within students, educators and researches, once they have the appropriate knowledge to enable a wider and more efficient utilization of the available resources.

We organize the paper as following. The basic intuition and mathematics behind Star Coordinates is introduced in Section 2, followed by the proposed methodology in Section 3; descriptions of the visual framework are presented in Section 4. In Section 5, an empirical example is demonstrated and, finally, Section 6 concludes the paper.

2. Star Coordinates

The Star Coordinates is a coordinate transformation that enables plotting multi-dimensional data in 2D space. Each dimension is represented by an axis and a point represents each multi-dimensional data object.

Basically, it is a curvilinear coordinate system, which can be related to the Cartesian Coordinates by defining a sequence of two dimensional vectors representing the axes. The axes are arranged on a circle with equal angles between each other and origin at the circle center.

The minimum value of each dimension is mapped to the axis origin and the maximum to the axis end. It is established that the origin of the Star Coordinates system coincides with the origin of a Cartesian Coordinates system to simplify the calculation.

* Corresponding author.
E-mail addresses: otto.machado@accenture.com, edt2009@bhvt.org.uk (O.M. Machado Filho).

0965-9978/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2008.12.020

Considering a multi-dimensional dataset represented by a matrix D_{nxm}, where n = total number of data objects, m = total number of dimensions.

Each row of the matrix D_{nxm} corresponds to an object D_j, as shown next:

$$D_j = (d_{j1}, d_{j2}, \ldots, d_{ji}, \ldots, d_{jm}), \tag{1}$$

where d_{ji} = dimension "i" of object "j"

There will be a sequence of "m" two dimensional vectors representing the Star Coordinates axes, as follows:

$$\vec{C} = (\vec{C}_1, \vec{C}_2, \ldots, \vec{C}_i, \ldots, \vec{C}_m), \tag{2}$$

where $\vec{C}_i$ = vector that represents the axis of the dimension "i".

The unit vector of each axis is calculated to enable a value scale through the axis length:

$$|\vec{u}_i| = \frac{|\vec{C}_i|}{\max_i - \min_i}, \tag{3}$$

where

$$\min_i = \min\{d_{ji}, 1 \leqslant j \leqslant n\} \tag{4}$$
$$\max_i = \max\{d_{ji}, 1 \leqslant j \leqslant n\}. \tag{5}$$

The unit vectors are represented by their components in the Cartesian system:

$$\vec{u}_i = (u_{xi,} u_{yi}). \tag{6}$$

The mapping of an object D_j from a dataset D_{nxm} to a point (P_j) in the two dimensional Cartesian Coordinates is determined by the sum of all unit vectors on each dimension multiplied by the value of the object for that dimension, as shown next:

$$P_j(x, y) = \left(\sum_{i=1}^{n} u_{xi} \cdot (d_{ji} - \min_i), \sum_{i=1}^{n} u_{yi} \cdot (d_{ji} - \min_i) \right). \tag{7}$$

In order to illustrate the method, Fig. 1 shows the calculation of a point location (P_j) for an object of a dataset with three dimensions ($D_j = (d_{j1}, d_{j2}, d_{j3})$), where the unit vectors are:

$$\vec{u}_1 = (u_{x1,} u_{y1}) = (\cos\alpha, \sin\alpha), \tag{8}$$
$$\vec{u}_2 = (u_{x2,} u_{y2}) = (\cos\beta, \sin\beta), \tag{9}$$
$$\vec{u}_3 = (u_{x3,} u_{y3}) = (\cos\gamma, \sin\gamma). \tag{10}$$

In this example, the final position of P_j on the Star Coordinates system, representing D_j, is given by:

$$P_j(x) = (\cos\alpha \cdot (d_{j1} - \min_1) + \cos\beta \cdot (d_{j2} - \min_2) + \cos\gamma \cdot (d_{j3} - \min_3)), \tag{11}$$

$$P_j(y) = (\sin\alpha \cdot (d_{j1} - \min_1) + \sin\beta \cdot (d_{j2} - \min_2) + \sin\gamma \cdot (d_{j3} - \min_3)). \tag{12}$$

Data objects with similar attribute values always map to nearby positions on the display, thus, only with interactive dynamic transformations such as axis scaling and axis rotation it is possible make sense of data distribution.

Scaling transformation allows users to change the length of an axis, thus increasing or decreasing the contribution of a particular data attribute on the resultant visualization. When scaled the data points are remapped according to the new scaling factor based on the length of the axis.

Rotation transformation modifies the direction of the unit vector of an axis, thus making a particular data attribute more or less correlated with other attributes. As a result, these transformations provide a restructuring of the data points in the visualization based on the criteria chosen.

3. The new visual approach for RBF network design

With automatic RBFN configuration algorithms the user can only define the parameters before the algorithm execution and then wait for the results evaluation. Several sets of parameters are tried before finding the appropriate ones, which it is usually very time-consuming.

We propose a methodology that enables an interactive visual RBFN design rendering to get human more involved in the process. The steps related to the basic methodology employed in visual RBFN design evaluation and refining are given as follows (Fig. 2).

3.1. Importing RBFN training dataset into the Star Coordinates system

The dataset used to train the RBF Network is imported into the visual rendering system and represented by simply points. This visual resource provides a good overall understanding of the training dataset.

3.2. Importing RBFN centers into the Star Coordinates system

The RBFN centers, from any RBFN design algorithm, are imported into the visual rendering system and represented by triangles. The RBFN centers distribution act as the guidance to re-define the center position or width.

3.3. Applying interactive operations to refine the RBFN design

It is assumed that once the users have a better overall understanding of the training dataset and the RBFN centers, they will know where to look for RBFN design improvements. The approach taken is to provide interactive operations on the visualization that will enable users to analyze and to improve the automatic algorithms results. These operations include: to edit center position, to edit center width, to create center and to delete center.

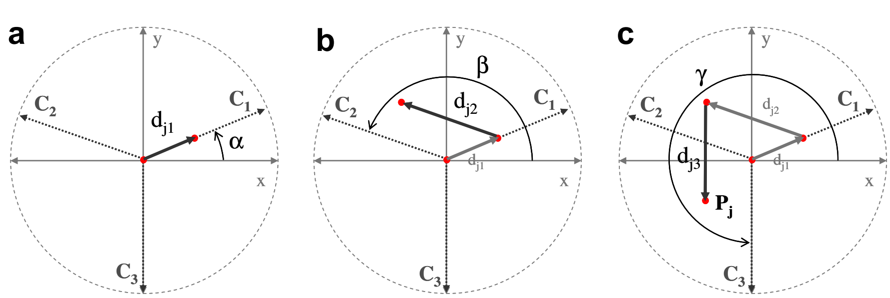

Fig. 1. Data point location for a three dimensional dataset. (a) Component of dimension 1, (b) Component of dimension 2, (c) Component of dimension 3.