A Comparison Study on Shape Parameter Selection in Pattern Recognition by Radial Basis Function Neural Networks

S Tavaen1,* and S Kaennakham1,2
1School of Mathematics, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand.
2Centre of Excellence in Mathematics, Bangkok 10400, Thailand.
*Corresponding Author: Sunisa Tavaen. Email: sunisat58@email.nu.ac.th

Abstract. This study investigates three choices of shape parameter selection when the so-called Radial Basis Function (RBF) is used. Under the problem of pattern recognition via RBF-Neural Network using RC-algorithm, three RBFs are focussed on; Gaussian (GA), Multiquadric (MQ), and Compactly-Supported (CS1). Two pattern recognition cases are tested and the best choice of shape parameter is validated using Model-Selection Criteria (MSC).

Keywords. Pattern Recognition, Radial Basis Function, Neural Networks, Gaussian, Multiquadric

1. Introduction
Over the past decades it has been noticeable that Radial Basis Function neural networks (RBFNs) have been slowly replacing the traditional Multilayer Perceptron (MLP) neural networks particularly in the applications of supervised learning concept. The main attractive feature is having the single hidden layer making the training process comparatively faster. RBFNs were introduced in 1987 by Powell M.J.D. [1] and have been developed by many researches and applied to many science and engineering problems. Some nice applications involve the process of interference cancellation [2], the diagnosis of damage of radial gate [3], on complex-valued Radial Basis Function Network in signal processing process [4], and the classification of incomplete feature vectors in pattern recognition [5] (see also the references therein).

Radial Basis Functions (RBF), $\varphi$, are commonly found as multivariate functions whose values are dependent only on the distance from the origin. This means that $\varphi(\mathbf{x}) = \varphi(r) \in \mathbb{R}$ with $\mathbf{x} \in \mathbb{R}^n$ and $r \in \mathbb{R}$; or, in other words, on the distance from a point of a given set $\{\mathbf{x}_j\}$, and $\varphi(\mathbf{x} - \mathbf{x}_j) = \varphi(r_j) \in \mathbb{R}$. Here, $r_j = \|\mathbf{x} - \mathbf{x}_j\|$ is the Euclidean distance expressed in $n$-dimensional space as;

$$ r_j = \|\mathbf{x} - \mathbf{x}_j\| = \sqrt{(x_{1j} - x_{1i})^2 + (x_{2j} - x_{2i})^2 + \ldots + (x_{nj} - x_{ni})^2} \quad (1) $$

and any function $\varphi$ satisfying $\varphi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$ is a radial function.

Despite of their great features one can benefit from using RBFNs, the key of success if still the choice of RBF to use. Many forms of popular RBFs contain the so-called ‘shape parameter’ and it is known to determine the quality of the whole model where the optimal choice is still problematic, one often needs to make somewhat an ‘ad-hoc’ decision. Figure 1 displays the clear influence on the function surface caused
by this shape for the case of inverse quadratic RBF type, \( \varphi(r) = \left(1 + \varepsilon r^2 \right)^{-1/2} \). This simply implies that as far as an RBFN goes, the pain will still be the choice of this shape, unless a non-parameterised RBF is considered. This issue is addressed and made the main objective of this work as the quest of assessing the quality of some popular RBFs when used with choices of parameters proposed in literature is set-out. The context of pattern recognition is where the applications of this study take place.

Figure 1 Different surface for different values of inverse quadratic RBF, \( \varphi(r) = \left(1 + \varepsilon r^2 \right)^{-1/2} \), shape parameter \( \varepsilon = 0.1, 0.5 \) and \( 1.0 \) respectively.

Section 2 provides the brief concept of mathematical concepts related to investigation before the details of radial basis functions and their choices of parameter are provided in Section 3. Numerical experiments with results are then provided in Section 4 before main findings can be listed in Section 5.

2. Mathematical Concept and Background

2.1 Pattern Recognition Problem Statement

Pattern recognition is the process of differentiating and dividing the data according to certain criteria or by general components, which are performed by special algorithms. Because pattern recognition helps to classification and prediction, it is one of the important components of machine learning technology [6] and is applied to image processing [7], industry [8], and medical [9] (see references therein).

The task of pattern recognition is to construct the model with unknown input-output mapping pattern. It is to construct the best model, if any, from the train data with some mapping functions and expect this model to best represent the rest of the data, called ‘training data’. Both sets of the data can be of the following form;

\[
D = \left\{ (x_i, y_i) \mid x_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = 1, 2, \ldots, n \right\}
\]

(2)

where \( x_i \) are inputs with the corresponding \( y_i \) are outputs. The main task is to find a mapping \( D \) from the \( d \) dimensional input space to 1-dimensional output space. Over the decade, there have been several models designed to tackle the problem and some are statistical model, structural model, template matching model, neural network based model, fuzzy based model, and hybrid model. Amongst these, very often that radial basis functions (RBF) are involved where the crucial factor is the shape parameter, mostly contained within the RBF used.

2.2 Radial Basis Function model structure
Model structure of radial basis function is depicted in Figure 2, when a data set \( \{(x_i, y_i)\}_{i=1}^{n} \) is given and it searches for the output estimate \( \hat{y} \) for input vector \( x \), represented by functional form:

\[
\hat{y} = f(x) = \sum_{j=1}^{m} w_j \phi_j(x) = \sum_{j=1}^{m} w_j \phi_j \left( \|x - \mu_j\| \right)
\]

(3)

where \( \|\cdot\| \) is the Euclidean distance norm, \( \phi(\cdot) \) is a basis function and \( m \ll n \), \( \mu_j \), \( w_j \) are the width of \( j \)-th basis functions, the number of basis functions, the centre of \( j \)-th basis functions and the weight associated with the \( j \)-th basis function, respectively.

![Figure 2 Radial basis function (RBF) neural network structure](image)

**2.3 Representational Capability (RC) Algorithm**

RC algorithm as proposed by Shin and Park (2000) [10] is a method to drag out information of interpolation matrix when RBF is in use. For given input data \( \{x_i, y_i\}_{i=1}^{n} \), the algorithm contains the following steps.

**Step 1:** Select a value for width (or the shape parameter) \( \varepsilon \) and effect of noise \( \delta \) which \( \delta \) is usually taken to be 0.1\% to 1.0\% and construct the interpolation matrix \( G \). For example, if Gaussian basis function is used, so that

\[
G = \begin{bmatrix}
    g_{11} & g_{12} & \cdots & g_{1n} \\
    g_{21} & g_{22} & \cdots & g_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    g_{n1} & g_{n2} & \cdots & g_{nn}
\end{bmatrix}
\]

(4)

where \( g_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{2\varepsilon^2} \right) \) for \( i, j = 1, 2, \ldots, n \).

**Step 2:** Determine the number of basis functions \( m \) by applied singular value decomposition of the interpolation matrix \( G \). This yields a diagonal matrix of singular values \( s_1 \geq s_2 \geq \cdots \geq s_n \geq 0 \). From these, \( m \) can be determined from the following:

\[
m = \max_{0 \leq i \leq n} \left\{ i \left| s_{i+1} \leq s_i \times \frac{\delta}{100} \right. \right\}
\]

(5)
Step 3: Determine the centres of basis function \( (\mu) \). Partition matrix \( V \) from singular value decomposition of the interpolation matrix \( G \) as:

\[
V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}
\]

\( m \times m \) \( n \times m \) \( m \times n-m \) \( \mu = \{\mu_j\}_{j=1}^m \) 

Next, generate matrix \( V' = \begin{bmatrix} v_{11}^T & v_{21}^T \end{bmatrix} \) and apply QR factorization with column pivoting of matrix \( V' \). And then compute \( X^T P \) and choose the first \( m \) elements in \( X^T P \) be the centres of basis function which are:

\[
\mu = \{\mu_j\}_{j=1}^m
\]

Step 4: Compute the weight parameters from the basis function \( (w) \). Consider,

\[
\phi = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_m \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m1} & \phi_{m2} & \cdots & \phi_{mm} \end{bmatrix}
\]

For \( i = 1, 2, \ldots, n \) and \( j = 1, 2, \ldots, m \).

Compute the \( m \) weights with

\[
w = \phi^* y
\]

where \( \phi^* \) denoted the pseudo inverse of \( \phi \).

3. RBFs and Their Parameter’s Choices

3.1 The Selected RBFs

For the purpose of comparison, three forms of radial basis functions are focused on and they are listed as follows.

Gaussian RBF (GA):

\[
\phi(r) = \exp\left(-r^2/2\varepsilon^2\right)
\]

Multiquadric RBF (MQ):

\[
\phi(r) = \sqrt{\varepsilon^2 + r^2}
\]

Compactly-Supported RBF (CS1):

\[
\phi(r) = \frac{112}{45} r^2 + \frac{16}{3} r^2 - \frac{14}{15} r^2 + \frac{1}{9}
\]

The first two are known to be popular in many areas of research and are determined by the choice of the so-called ‘shape parameter, \( \varepsilon \)’, [11], (details provided in the next section). The third one is included in this study with the main aim to monitor the performance with compared with the other two RBFs driven by \( \varepsilon \).

It is also discovered in our previous study [12] that amongst three forms of Compactly-Supported RBFs, CS1 is more simply implemented with no significant differences in final results quality. This is the main reason why it is being further investigated in this work.
3.2 The Parameter’s Choices

In this work, the main attention is paid on the effect of choices of the shape parameter proposed in three well-known numerical researches (under the applications of interpolation) and they are briefly detailed below.

**Choice I:** Hardy Shape (HS)

This was proposed by Hardy [13] and it is of the following form.

\[ e_{HS} = 0.815d \]  \hspace{1cm} (13)

where \( d = \frac{1}{N} \sum_{i=1}^{N} d_i \) and \( d_i \) is the mean distance from each data point \((x_i, y_i)\) to its nearest neighbour.

**Choice II:** Franke Shape (FS)

This was invented by Franke [14] and uses the following formula:

\[ e_{FS} = \frac{1.25D}{\sqrt{N}} \]  \hspace{1cm} (14)

Where \( D \) is the diameter of the smallest circle containing all data points.

**Choice III:** Carlson Shape (CS)

It was formed by Carlson [15] and it starts by computing the least squares bivariate quadratic polynomial fit to the data \( (x_i, y_i, z_i) \) and denoting this quadratic by \( q(x, y) \), i.e. compute

\[ V = \frac{\sum_{i=1}^{N} (z_i - q(x_i, y_i))^2}{N} \]

by setting \( \bar{x}_i = \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} \), \( \bar{y}_i = \frac{(y_i - y_{\min})}{(y_{\max} - y_{\min})} \) and \( \bar{z}_i = \frac{(z_i - z_{\min})}{(z_{\max} - z_{\min})} \).

The proposed form of shape parameter is then of the following.

\[ e_{CS} = \frac{1}{1+120V} \]  \hspace{1cm} (15)

These three interesting forms of \( e \) are numerically tested with 1D and 2D pattern applications and the results are discussed in the following section.

4. Numerical Experiments

For result validation, the mean square error norms (MSE) is employed and it is of the following form;

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \]  \hspace{1cm} (16)

With \( N \) being the number of data points involved in each case. Additionally, the following are experiment’s ingredients used in this study.

1. The whole dataset for each case is split into two parts namely ‘training dataset (TD)’ and ‘validation dataset (VD)’, and they are fixed throughout the investigation.
2. \( TD \cap VD = \{ \} = \text{empty set} \).
3. Define \( TVN \) represents the ratio of \( *TD : VD* \)
4. Defining ‘The Model Selection Criteria (MSC)’, the best model is ideally the one with the following properties:
   4.1. produces small validation error.
   4.2. requires small number of centres, i.e. small \( m \) in RC-algorithm.
   4.3. the number of centres, \( m \), should be as less affected by \( TD \) as possible.
(This is all to compromise the problem of over- and under-fitting scenarios.)

The observation of the performance of all the RBFs chosen and the choices of shape parameter is carried out using two benchmarking test cases where the details are following.

**Experiment 1: The Sine Function**

In the first case study, a sine function expressed below is investigated.

$$y = \sin(x) + \omega$$  \hspace{1cm} (17)

A set of 100 data on $[0, 2\pi]$ is generated and the noise $\omega$ is set to be a Gaussian distribution with mean zero and standard deviation 0.5. The data points are generated uniformly over the domain with 1,000 points.

The experiments are conducted covering a wide range of $TVN$ from 100:900 to 900:100 (only some are shown here, however). Table 1 provides information concerned in each case and it is clearly seen that the number of centres is significantly increasing with more training data being involved in the case of HS for both GA ($29 \leq m \leq 135$) and MQ ($22 \leq m \leq 49$). On the other hand, FS and CS lead to a much narrow range of $m$ over the same interval of $TD$ for both GA and MQ, indicating that FS and CS are comparatively less affected by $TD$. For all cases, it should be noted that the shapeless CS1 is not being at all affected by the increase in training dataset $TD$.

In terms of validation error trends, at every ratio of $TVN$, it can be seen from the table that HS produces the comparatively highest error for both GA and MQ. On the other hand, FS and CS are found to generate small error and stay very close to each other for all $TVN$ shown. It is also found that errors produced by CS1 is smaller than that obtained from HS but higher than that of FS and CS. For the case of GA only, pattern simulations obtained from all the shape choices are illustrated in Figure 3.

To sum up for this sine-function experiment, as long as the MSC is concerned, the shape parameter choice proposed by Franke [14] can be considered as the best.

**Table 1. Error comparison for some chosen different TVN cases.**

| $TVN$  | RBF | Type of | $\varepsilon$ | $\nu$ | $m$ | MSE         |
|-------|-----|---------|---------------|-------|-----|-------------|
|       |     |         |               |       |     | Training    | Validation |
| 100:900 | GA  | HS      | 0.0512 | 29     | 0.3721 | 0.9191 |
|        |     | FS      | 0.7775 | 4      | 0.4709 | 0.7535 |
|        |     | CS      | 0.2403 | 8      | 0.4362 | 0.8176 |
|        | MQ  | HS      | 0.0512 | 22     | 0.3926 | 0.8906 |
|        |     | FS      | 0.7775 | 4      | 0.4701 | 0.7526 |
|        |     | CS      | 0.2403 | 9      | 0.4383 | 0.8132 |
|        | CS1 |         | -       | -      | 13     | 0.4259 | 0.8352 |
| 250:750 | GA  | HS      | 0.0205 | 69     | 0.3449 | 0.7334 |
|        |     | FS      | 0.4947 | 5      | 0.4587 | 0.6316 |
|        |     | CS      | 0.3326 | 6      | 0.4577 | 0.6340 |
|        | MQ  | HS      | 0.0205 | 36     | 0.4093 | 0.6919 |
|        |     | FS      | 0.4947 | 5      | 0.4590 | 0.6328 |
|        |     | CS      | 0.3326 | 7      | 0.4579 | 0.6340 |
|        | CS1 |         | -       | -      | 13     | 0.4498 | 0.6428 |
| 500:500 | GA  | HS      | 0.0102 | 135    | 0.3415 | 0.6067 |
|        |     | FS      | 0.3505 | 6      | 0.4759 | 0.5014 |
|        |     | CS      | 0.3552 | 6      | 0.4758 | 0.5014 |
|        | MQ  | HS      | 0.0102 | 49     | 0.4271 | 0.5533 |
|        |     | FS      | 0.3505 | 7      | 0.4740 | 0.5020 |
|        |     | CS      | 0.3552 | 7      | 0.4740 | 0.5020 |
|        | CS1 |         | -       | -      | 13     | 0.4720 | 0.5044 |
Figure 3. Pattern reconstruction obtained from using Gaussian RBF with all three choices of shape parameter compared with that produced by the shapeless CS1-RBF where above) training phase, and below) validation phase, calculated using $TVN=(100:900)$.

**Experiment 2: The famous Franke’s Function**

This experiment is concerned with one of the most well-known testing functions invented by Franke in 1982 [16]. The complication of the function’s surface makes it a good test for validating any interpolation-based methods proposed over the past two decades. The expression of the function is as followed.

$$f(x, y) = 0.75\exp\left(\frac{-(9x - 2)^2 + (9y - 2)^2}{4}\right) + 0.75\exp\left(\frac{-(9x + 1)^2 - 9y + 1}{49}\right) + 0.5\exp\left(\frac{-(9x - 7)^2 + (9y - 3)^2}{4}\right) - 0.2\exp(-(9x - 4)^2 - (9y - 7)^2)$$  

(18)

Like done in the previous experiment, this function is disturbed with Gaussian distribution with mean zero and standard deviation 15.0 and for simplicity the following function is denoted.

$$f_i(x, y) = f(x, y) + \omega$$  

(19)

Several ratios of training to validation data points have been carried out. The data points are generated uniformly over the domain with $61\times61$ points. Figure 4 shows node-distribution in the case using $TVN=(121:3,600)$ and the corresponding exact surface (without noise).
Figure 4. (a) Node-distribution, projected on xy-plane, in the case with 121 training data points (depicted with blue dots) and 3600 validation data points (depicted with black dots), (b) Surface plot of \( f(x,y) \).

Similar to the previous case of sine function, a number of experiments are conducted and some discovery is provided in Table 2. From the Table, the mostly noticeable figure is the training error produced by using \( HS \) in both GA and MQ. This over-fitting phenomenon is directly the result of the fact that it uses the whole set of \( TD \) as its centres, i.e. it simply reflects itself for all \( TVN \). Even though it yields small validation errors, it cannot be crowned as the great model. On the other hand, \( FS \) and \( CS \) are seen to produce even smaller value of validation errors while the training errors still look reasonable. In both GA and MQ, it is found that \( CS \) leads to slightly better validation error than \( FS \) where the significant figure is revealed in terms of the number of centres required, i.e. \( m \). Not only \( CS \) is capable of producing good results with small value of \( m \), it is also seen to be mostly independent of the increase in training data. Even though CS1 is not being affected by \( TD \), the smallest \( m \) is still seen in the case of using CS (with GA in particular). For all these reasons, it is concluded in this experiment that \( CS \) shape parameter is the best choice for both GA-RBF and MQ-RBF. Solution distributions approximated using GA-\( CS \), MQ-\( CS \) and CS1 are illustrated in Figure 5.

| \( TVN \) | RBF | Type of \( \varepsilon \) | \( \varepsilon \) | \( m \) | \( MSE \) |
|---------|-----|----------------|--------|-----|-------|
|         |    | \( HS \) | 0.0815 | 121 | 1.5864e-30 | 0.0133 |
| 121:3600 | GA | \( FS \) | 0.1607 | 83  | 0.0023 | 0.0119 |
|         |    | \( CS \) | 0.4060 | 24  | 0.0063 | 0.0100 |
|         | MQ | \( HS \) | 0.0815 | 106 | 8.4055e-04 | 0.0123 |
|         |    | \( FS \) | 0.1607 | 53  | 0.0048 | 0.0098 |
|         |    | \( CS \) | 0.4060 | 24  | 0.0061 | 0.0097 |
|         | CS1 | - | - | 66 | 0.0037 | 0.0106 |
| 441:3280 | GA | \( HS \) | 0.0408 | 441 | 3.2124e-30 | 0.0137 |
|         |    | \( FS \) | 0.0842 | 257 | 0.0033 | 0.0114 |
|         |    | \( CS \) | 0.4264 | 19  | 0.0087 | 0.0097 |
| 961:2760 | MQ | \( HS \) | 0.0408 | 135 | 0.0052 | 0.0097 |
|         |    | \( FS \) | 0.0842 | 89  | 0.0059 | 0.0091 |
|         |    | \( CS \) | 0.4264 | 20  | 0.0079 | 0.0089 |
|         | CS1 | - | - | 66 | 0.0064 | 0.0086 |
|         | GA | \( HS \) | 0.0272 | 961 | 5.0043e-30 | 0.0137 |
|     | FS  |  CS  |  HS  | MQ  |
|-----|-----|-----|-----|-----|
| FS  | 0.0570 | 0.4103 | 0.0272 | 0.0570 |
| CS  | 0.0035 | 0.0093 | 0.0067 | 0.0070 |
| HS  | 0.0117 | 0.0091 | 0.0086 | 0.0085 |
| MQ  | 0.0117 | 0.0091 | 0.0086 | 0.0085 |
| CS1 | -   | -   | 66   | 0.0074 |
|     |     |     |     | 0.0081 |

Figure 5. Pattern reconstruction yielded by using (a) GA-CS, (b) MQ-CS, and (c) CS1 with the training phase being shown in the left column, and the right column is for the validation phase. (Note that: the ‘Exact’ means $f(x, y)$ and represented in red).
References

[1] Powell M J D 1987 Radial Basis Functions for Multivariable Interpolation: A Review *Algorithms for Approximation* 143–67

[2] Cha I and Kassam S A 1995 Interference Cancellation using Radial Basis Function Networks *Signal Processing* 47 247–268

[3] Jianwei Z, Yu Z, Yina Z and Longfei Z 2009 Damage Diagnosis of Radial Gate Based on RBF Neural networks *In Proceedings of 2009 International Forum on Computer Science-Technology and Applications* 3 399–402

[4] Chen S, McLaughlin S and Mulgrew B 1994 Complex-valued radial basic function network, Part I: Network architecture and learning algorithms *Signal Processing* 35 19–31

[5] Dybowski R 1998 Classification of incomplete feature vectors by radial basis function networks *Pattern Recognition Letters* 19 1257–64

[6] Holmström L and Koistinen 2010 Pattern recognition *WIREs Comp Stat* 2 404–13

[7] Bhowmik M K, Bhattacharjee D, Nasipuri M, Basu D K and Kundu M 2009 Classification of fused images using radial basis function neural network for human face recognition *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (Coimbatore, India: : IEEE)

[8] Escobar C A and Menendez R M 2017 Machine Learning and Pattern Recognition Techniques for Information Extraction to Improve Production Control and Design Decisions Advances in Data Mining Conference: *Industrial Conference on Data Mining*

[9] Zarbakhsh P and Addeh A 2018 Breast cancer tumor type recognition using graph feature selection technique and radial basis function neural network with optimal structure *J. Cancer Res. Ther.* 14 625–33

[10] Shin M and Park C 2000 A Radial Basis Function Approach to Pattern Recognition and Its Applications *ETRI Journal* 22

[11] Shin M 1998 Design and Evaluation of Radial Basis Function Model for Function Approximation Ph.D Dissertation, Syracuse University

[12] Tavaen S, Viriyapong R and Kaennakham S 2020 Performances of Non-Parameterized Radial Basis Functions in Pattern Recognition Applications *J. Phys.: Conf. Ser.* 1706 1–9

[13] Hardy R L 1971 Multiquadric equations of topography and other irregular surfaces *J. Geophys. Res.* 76 1905–15

[14] Franke R 1979 A critical comparison of some methods for interpolation of scattered data *Naval Postgraduate School, Technical Report NPS-53-79-003*

[15] Carlson R E and Foley T A 1991 The parameter R² in multiquadric interpolation *Computers Math. Appl.* 21 29–42

[16] Franke R 1982 Scattered Data Interpolation: Tests of Some Method *Mathematics of Computation* 38 181–200