0truept

ON NORMALISING RADIAL BASIS FUNCTION NETWORKS

*Robert Shorten and Roderick Murray-Smith**

Daimler-Benz Systems Technology Research, Alt-Moabit 91b, 10559 Berlin, Germany
E-mail:shorten@DBresearch-berlin.de, murray@DBresearch-berlin.de.

ABSTRACT

Normalisation of the basis function activations in a radial basis function (RBF) network is a common way of achieving the partition of unity often desired for modelling applications. It results in the basis functions covering the whole of the input space to the same degree. However, normalisation of the basis functions can lead to other effects which are sometimes less desireable for modelling applications. This paper describes some side effects of normalisation which fundamentally alter properties of the basis functions, e.g. the shape is no longer uniform, maxima of basis functions can be shifted from their centres, and the basis functions are no longer guaranteed to decrease monotonically as distance from their centre increases – in many cases basis functions can re-appear far from the basis function centre. This paper examines how these phenomena occur, and analyses theoretically and experimentally the effect of normalisation on the least squares solution to the weights problem.

1. Introduction

Basis function networks have recently been the subject of increasing attention in the neural network, control and signal processing literature. Basis function networks, in particular Gaussian radial basis function (RBF) networks, have been successfully applied to a number of complex pattern recognition, modelling, control and signal processing tasks [1]. In many cases the use of *normalised basis functions* has resulted in an improvement in performance. Normalisation is sometimes desired because it results in every point in the input space being covered by the basis functions to the same degree, i.e. the basis functions sum to unity at every point. When this is the case a *partition of unity* across the input space is said to have been achieved. Partitioning of unity is an important property for basis function networks in many applications. It often results in a structure which can be less sensitive to poor centre selection and in cases where the network is being used within a local model structure, a partition of unity is highly desirable. While the approximation capabilities of normalised networks have been demonstrated [2], the side-effects of normalisation have not yet been considered in detail. In this paper the effect of normalisation, as used in [3][4][5][6] [7], on the behaviour of RBF networks is considered. Normalisation is most relevant for RBF nets, as other networks which partition the input space in an axis-orthogonal manner (e.g. B-Spline nets), can be designed to achieve a partition of unity without normalisation.

This paper is structured as follows. After an overview and discussion of normalised networks, the effects of normalisation on the physical properties of the basis functions and weight training is considered. It is demonstrated that, in addition to achieving a partition of unity, normalisation of the basis functions can lead to unexpected side effects. The most obvious of these is the change in shape of the basis functions and the possible loss in smoothness of representation. In addition, with non-compact basis functions, normalisation results in the whole of the input space being covered and not just the part of the space defined by the training data. This can lead to stability problems at the edge of the input space for non-compact[1] basis functions. It is also shown that in the case of irregular networks, i.e. those where centres are distributed unevenly, or differing sizes of basis functions are used, normalisation can give rise to two further phenomena, a shift in maximum and loss in monotonicity of the basis functions. The loss in monotonicity can lead to what we term 'reactivation', whereby the basis function can reappear far from its centre, thereby having more than one region of significant activity. It is then demonstrated that normalisation can affect the magnitude of the weights found, affecting the robustness of the final model. Finally, these effects will be illustrated by means of an example and their consequences for modelling discussed.

2. Modelling with Normalised Radial Basis Functions

The output of a normalised basis function network (BFN) is described by,

$$y = f(\mathbf{x}) = \sum_{i=1}^{M} w_i \bar{\phi}_i(\mathbf{x}), \tag{1}$$

where the basis functions $\bar{\phi}_k(\mathbf{x})$ are normalised

$$\bar{\phi}_k(\mathbf{x}) = \frac{\phi(d(\mathbf{x};\mathbf{c}_k,\sigma_k))}{\sum_{i=1}^{M}\phi(d(\mathbf{x};\mathbf{c}_i,\sigma_i))}, \tag{2}$$

where y is the network output[2], $\mathbf{x}$ is the vector of input variables, $\mathbf{c}_i$ is the centre of the ith basis function, σ_i is the width of the ith unit, M is the number of processing units,

[1]By compact it is meant that the basis functions are non-zero for some finite range. Outside of this range the basis function takes the value zero

[2]This paper considers single output systems only. However the analysis can easily be extended to multi-output systems.

*Published in Irish Neural Networks Conf. '94 University College Dublin, Sept. 1994

w_i is the weight associated with unit i , $d(\cdot)$ denotes some distance metric, and ϕ is the non-linear activation function before normalisation. The normalised form is $\bar{\phi}_i$, for unit i. In principle, this can be any non-linear function, but many authors use local[3] basis functions for a number of practical reasons. Local basis functions are advantageous because of the increased interpretability of the network, the ability to produce locally accurate confidence limits [8], and locality can also be utilised to improve computational efficiency.

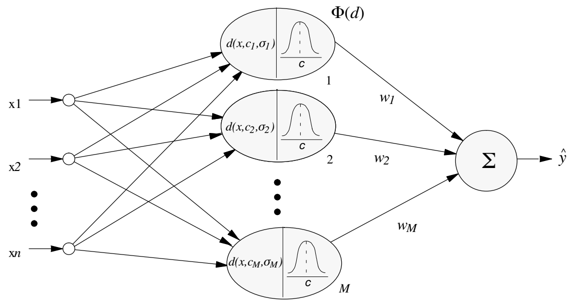


Figure 1: Unnormalised Radial Basis Function network

Figure 1 illustrates the network representation of equation (1). The output is a linear combination of the basis functions. A common choice for ϕ_i takes the form of a Gaussian although other activation functions have been proposed [9]. The Gaussian activation function used in this paper takes the form,

$$d(\mathbf{x}; \mathbf{c}_i, \sigma_i) = \left\| \frac{\mathbf{x} - \mathbf{c}_i}{\sigma_i} \right\|^2, \tag{3}$$

$$\phi_i(d(\mathbf{x}; \mathbf{c}_i, \sigma_i)) = \exp(-d(\mathbf{x}; \mathbf{c}_i, \sigma_i)), \tag{4}$$

Normalisation of the basis functions in such a network is often motivated by the desire to achieve a partition of unity across the input space. By partition of unity it is meant that at any point in the input space the sum of the normalised basis functions equals unity, i.e,

$$\sum_{i=1}^{M} \bar{\phi}_i(\mathbf{x}) = 1, \tag{5}$$

This has the effect of covering every point of the input space to the same degree, unlike the un-normalised case where points given different weightings. Normalised networks are attractive for a number of practical reasons. Because the space is covered to same degree at every point, they are often less sensitive to poor centre selection. In addition, it is desirable for many applications [10] that the cumulative sum of all basis functions at any point equals unity. Werntges [11] discusses the advantages of normalisation in RBF nets, promoting the advantages of a partition of unity produced by normalisation, but not considering the side-effects discussed in this paper.

The approximation capabilities of such networks have been considered in detail by Benaim [2] and it has been shown the NRBF's are capable of *universal approximation* [12] in a satisfactory sense.

[3] By local it is meant that the basis function is significantly active for some limited range of the input.

3. Side-effects of Normalisation for the Basis Functions

In order to achieve a partition of unity for many networks it is necessary to normalise. However, normalisation also leads to a number of important side effects which can have important consequences for the resulting network. In this section we describe these side effects and their consequences for the behaviour of the network.

3.1 Loss of Independence and Change of Shape of Basis Function

Unnormalised networks usually use homogeneous basis functions, sometimes with differing widths. In normalised nets this is not the case – the shape of the basis functions is usually quite different from the un-normalised basis function, and the shape is influenced not only by the basis function's width, but also by the proximity of the other functions in the network.

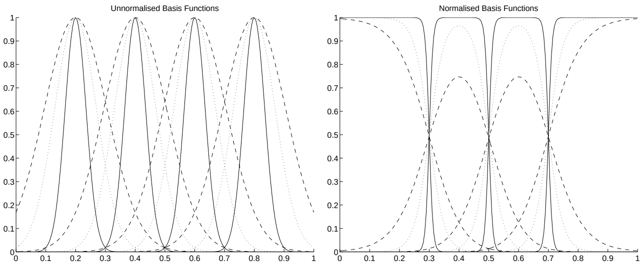


Figure 2: Change in shape due to normalisation

The effect on the shape of the basis functions can be seen from figure 2, with evenly spaced basis functions covering a one-dimensional space.It can be clearly seen that the basis functions have changed shape significantly. The smaller the width of the original basis function, the squarer the normalised basis function becomes. The maximum value of the basis functions also decreases, as width increases. As can be seen in figure 2, the smoothness of the network's representation can seriously affected by normalisation if the original basis functions are narrow in width. The effect in two dimensions is shown by the contour plot in figure 5.

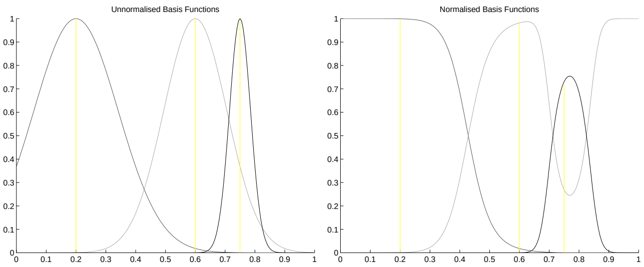
It also can be seen from equation 2 that each normalised basis function is a function of *all* the origional basis functions. Thus, changing any of the original basis functions affects all of the normalised basis functions. This can have important consequences for on-line applications where the network parameters are updated with each new data point.

3.2 Covering of the input-space

In the case where the basis function used is non-compact in nature, for example when Gaussians are used, then normalisation results the *whole* of the input space being covered and not just the region of the input space defined by the training data. It can be seen from figures 2 and 3 that in the normalised case the activation tends toward unity at the edges of the space. This can lead to unpredictable and often unstable behaviour in dynamic models if the input vector drifts outside the region of the input space that has been learnt during training.

3.3 Irregular Networks: reactivation and shift in maxima

A further difficulty with normalised basis functions involves two further phenomena. If centres are not uniformly spaced, or if basis functions of differing widths are used, the maximum of the basis function may no longer be at its centre. A further effect of varying basis widths is that the basis function can become multi-modal, meaning that it can now also increase as the distance function increases, instead of continuously decreasing – the unit 'reactivates'.



(a) Unnormalised Basis Functions

(b) Normalised Basis Functions

Figure 3: Shift in maxima and reactivation.

These effects are shown in figure 3. Note the reactivation of the centre basis function, the reduced maximum of the right hand basis function, and the shift in maximum for all three functions. The vertical lines show the positions of the basis functions centres to emphasise the centre-shift effect The reactivation occurs when neighbouring basis func-

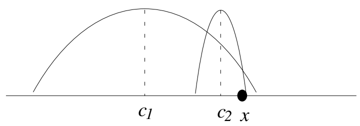


Figure 4: Simple example of reactivation

tions have differing widths. A one-dimensional example in Figure 4 using two basis functions illustrates how the phenomenon occurs. The point in the input space x where the basis function reactivates can be determined from the units' centres (c_1 and c_2, where c_1 is furthest from the input x) and their widths (σ_1 and σ_2). The reactivation point, assuming monotonically decreasing basis functions, is the point at which the distance metric d_1 is no longer smaller than d_2.

$$d_1 < d_2, \tag{6}$$

For a Euclidean distance metric,

$$\left(\frac{x - c_1}{\sigma_1}\right)^2 < \left(\frac{x - c_2}{\sigma_2}\right)^2, \tag{7}$$

$$\frac{\sigma_2}{\sigma_1} < \frac{|\ x - c_2\ |}{|\ x - c_1\ |}, \tag{8}$$

Equation (8) shows that reactivation only occurs when the ratio between σ_1 and σ_2 is less than the ratio of the unweighted distances from the centres. This implies that in networks with uniformly wide basis functions, reactivation cannot occur. The shift in the position of the activation function's maximum occurs when neighbouring basis functions are either unevenly spaced or have differing widths.

This behaviour can cause problems if the network is being used to estimate an underlying probability distribution as is the case when more general locally accurate models are being used to approximate the function in place of the simple weights of an RBF net, e.g. [6] [13]. Within this framework, reactivation can lead to models becoming significantly active in regions in which they were never intended to operate thus leading to less interpretable local models. The local learning methods for local model networks proposed in [14] require the partition of unity to be able to model the target function.

3.4 Effects of normalisation on multi-dimensional problems

The effects of normalisation can become more pronounced as the input dimension increases. Due to the increased number of neighbouring units in higher dimensions, the cumulative activation in a given region tends to increase with dimension, leading to normalised basis functions often having dramatically reduced maxima. Note also that the difference between the normalised radial and ellipsoidal basis functions is less extreme than the difference between the original functions – in many cases normalisation makes the use of more complex distance metrics less significant. Two dimensional contour plots are shown below.

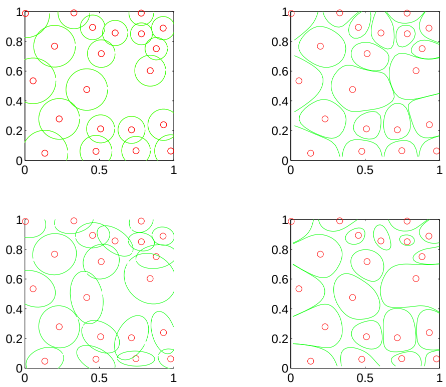


Figure 5: Contour plot of 2-dimensional network with radial and ellipsoidal units before and after normalisation.

4. Weight Estimation

The concepts of neural network stability and robustness are closely related to the size of the basis function weights. It is desirable to achieve a given network accuracy with weights of minimum magnitude since this improves the network's ability to generalise. For example, with noisy data, large weights can cause potentially large errors or even instability. It is therefore of interest to examine what happens to the *least squares* weight solution when a given network as defined by the network parameters $(M, \hat{\mathbf{w}}, \hat{\sigma}, \hat{\mathbf{c}})$, is normalised.

4.1 Theoretical limits on weights

We consider the system described by equation (1) where the exact form of the basis function is defined by the activation function used and whether it is normalised or not. We also assume that the output observations are all positive. In practice this is not a restriction since the output can be normalised to lie in the interval (0,1) during the preprocessing stage. At the output the inverse operation can be carried out. If all of the observations are grouped into matrix form with Y^T defined as,

$$\mathbf{Y}^T = [y_1, ..., y_N], \tag{9}$$

and

$$\psi_i{}^T = [\phi(d(\mathbf{x}_i; \mathbf{c}_1, \sigma_1))...\phi(d(\mathbf{x}_i; \mathbf{c}_M, \sigma_M))], \tag{10}$$

with

$$\mathbf{\Phi} = \begin{pmatrix} \psi_1^T \\ . \\ . \\ \psi_N^T \end{pmatrix} \tag{11}$$

then,

$$\mathbf{Y} = C\Phi\mathbf{w}, \tag{12}$$

where N is the number of observations, $\mathbf{\Phi}$ is the NxM design matrix of basis function activations from the training set, $\mathbf{w}$ is the Mx1 vector of weights and $\mathbf{C}$ is a NxN positive definite diagonal matrix (this assumes basis functions which positive for all of their support). In the unnormalised case $\mathbf{C}$ is simply the identity matrix, while in the normalised case $\mathbf{C}$'s entries are given by

$$c_{kk} = \left(\sum_{i=1}^{M} \phi(d(\mathbf{x}_k; \mathbf{c}_i, \sigma_i) \right)^{-1} \quad \forall \quad 1 \le k \le N. \tag{13}$$

Then,

$$\bar{\mathbf{Y}} = \mathbf{C}^{-1}\mathbf{Y} = \Phi\mathbf{w}, \tag{14}$$

since $\mathbf{C}$ is invertible. Therefore the solution to equation (14) can be written,

$$\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{C}^{-1}\mathbf{Y} \tag{15}$$

which can be written

$$\mathbf{w} = \mathbf{J}\mathbf{C}^{-1}\bar{\mathbf{Y}}, \tag{16}$$

where J is an MxN matrix and assuming that the inverse $(\Phi^T\Phi)^{-1}$ exists. Expanding equation (16) yields,

$$w_i = j_{i1}k_{11}y(t_1) + \cdots + j_{iN}k_{NN}y(t_N), \tag{17}$$

where w_i denoteds the ith normalised weight, j_{mn} is the mn entry of J and k_{ii} is the ith diagonal entry of $\mathbf{C}^{-1}$. After some manipulation the following inequality can be obtained from equation ,

$$k_{min}\bar{w}_i \le w_i \le k_{max}\bar{w}_i, \tag{18}$$

where k_{max} and k_{min} are the maximum and minimum entries of the $\mathbf{C}^{-1}$ matrix and $\bar{w}_i$ denotes the ith weight under the constraint that the $\mathbf{C}$ matrix is the identity matrix, i.e., the network is un-normalised.

Equation (18) indicates that the magnitude of optimal weights may be increased or decreased after normalisation of the basis functions. An increase in weights typically occurs when the widths are large (as k_{min} is then also large), whereas a decrease in weight magnitude tends to be associated with small widths. In multidimensional cases, the effect of large basis functions becomes even more dramatic, for the reasons described in section 3.3. It is therefore important not to normalise blindly, but to compensate for the normalisation by altering the design criteria for the structure (centre positions and width magnitudes) identification procedure.

5. Illustrative Example: Modelling a Pulse Function

In order to illustrate the effects noted in the previous sections a simple one-dimensional example of RBF modelling is presented. The function to be modelled is depicted in figure 6. A network consisting of three basis functions (centred at

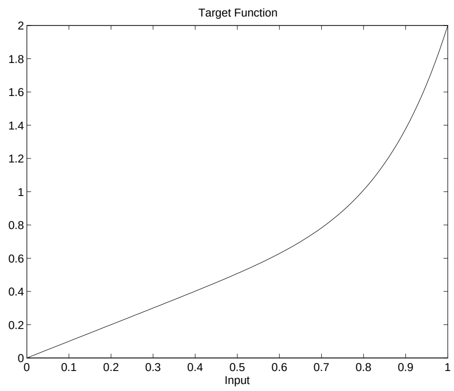


Figure 6: Target Function

$[0.3, 0.5, 0.9]$) was used for the modelling task. The widths of the various basis functions were varied to illustrate the effects noted above.

The modelling performance of the network with narrow basis functions is depicted in 7. The basis functions are deliberately chosen to be too narrow in order to exaggerate the effects of normalisation. The reactivation, covering of the input space and weight decrease effects can be clearly seen in figure 7. The approximation is very poor because the basis functions are too narrow.

The modelling performance of a network with wide basis functions is presented in figure 8. It can be clearly seen

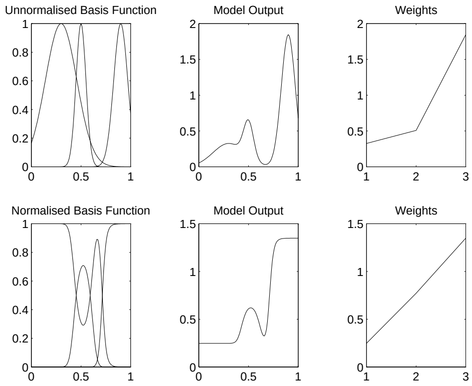


Figure 7: Narrow basis functions

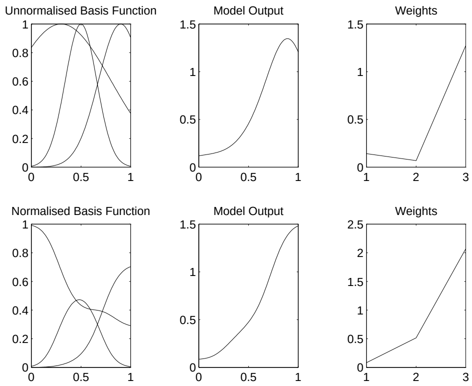


Figure 8: Wide basis functions

that the network approximation is better than with narrow basis functions. In this case reactivation is still occurs in the normalised case. Note the weight increase due to normalisation.

6. Conclusions

In this paper we have described phenomena which occur in basis function networks when a partition of unity is achieved by means of normalising the network. These effects can be summarised as follows:

1. Normalisation leads to a change in shape of basis function. This can lead to a loss in smoothness of representation if the widths of units are too narrow.
2. In the case where non-compact basis functions are used normalisation leads to a covering of the whole of the input space. This can result in stability problems for dynamical networks at the edges of the space defined by the training data.
3. For irregular networks the maxima of the units can shift away from the centres, and the units can reactivate in other parts of the input space. Reactivation, and the resulting non-localised behaviour of individual basis functions means that the very motivation behind much of the work carried on RBF nets, i.e. localised behaviour, is no longer guaranteed.
4. Normalisation also affects the magnitude of weights. This can subsequently effect the robustness and stability (for dynamical systems) of the network. The question of stability will be considered in more detail in a later publication.
5. Effects 1-4 become more pronounced as the input dimension increases.

While partitioning unity is highly desireable for many modelling applications, these phenomena, or side-effects, can lead to unpredictible network behaviour. It is therefore of crucial importance that researchers and users consider these effects when designing both networks and training algorithms.

Acknowledgements

This work was in part sponsored by the European Union Human Capital and Mobility programme (HCM), contract number ERBCHBICT930711. The authors also wish to acknowledge the many helpful comments made by J. Donne, A. D. Fagan, K. Hunt and D. Neumerkel.

7. References

[1] D. Neumerkel, R. Murray-Smith, and H. Gollee, "Modelling dynamic processes with clustered time-delay neurons," in *Intenational Joint Conference on Neural Networks, Nagoya, Japan*, 1993.

[2] M. Benaim, "On Functional Approximation with Normalised Gaussian Units," *Neural Computation*, vol. 6, no. 1, pp. 319–333, 1994.

[3] J. Moody and C. Darken, "Fast-learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294, 1989.

[4] C. Barnes, S. Brown, G. Flake, R. Jones, M. O'Rourke, and Y. C. Lee, "Applications of neural networks to process control and modelling," in *Artificial Neural Networks, Proceedings of 1991 Internat. Conf. Artif. Neur. Nets*, vol. 1, pp. p321–326, 1991.

[5] R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, P. S. Lewis, and S. Qian, "Function approximation and time series prediction with neural networks," Tech. Rep. 90-21, Los Alamos National Lab., New Mexico, 1989.

[6] T. Johansen and B. Foss, "Constructing NARMAX models using ARMAX models," *International Journal of Control*, vol. 58, no. 5, pp. 1125–1153, 1992.

[7] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications for modeling and control," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[8] J. A. Leonard and M. A. Kramer, "Radial Basis Function networks for classifying process faults," *IEEE Control Systems Magazine*, vol. 11, pp. 31–38, April 1991.

[9] K. Hlavácková and R. Neruda, "Radial Basis Function networks," *Neural Network World*, vol. 1, pp. 93–101, 1993.

[10] T. Johansen and B. Foss, "A NARMAX model representation for adaptive control based on local models," *Modelling, Identification and Control*, vol. 13, no. 1, pp. 25–39, 1992.

[11] H. W. Werntges, "Partitions of unity improve neural function approximation," in *Proc. IEEE Int. Conf. Neural Networks*, (San Francisco, CA), pp. 914–918, 1993. Vol. 2.

[12] W. Kaplan, *Advance Calculus*. Addison Wesley, 1991.

[13] T. Johansen and B. Foss, "Adaptive Control of MIMO Non-linear Systems using Local ARX models and Interpolation," *Modelling, Identification and Control*, vol. 13, no. 1, pp. 25–39, 1992.

[14] R. Murray-Smith, "Local Model Networks and Local Learning," in *Fuzzy Duisburg, '94*, pp. p404–409, 1994.

E-mail:murray@DBresearch-berlin.de.