Side Effects of Normalising Radial Basis Function Networks

Robert Shorten and Roderick Murray-Smith*

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 desirable 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 'reactivate', i.e. re-appear far from the basis function centre. This paper examines how these phenomena occur, discusses their relevance for non-linear function approximation and examines the effect of normalisation on the network condition number and weights.

1 Introductory remarks

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. In many cases the use of *normalised basis functions* has resulted in an improvement in performance.

This paper discusses the effect of normalisation, as used in (Moody and Darken 1989, Barnes *et al.* 1991, Jones *et al.* 1989, Johansen 1994), on the behaviour of RBF networks. 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. While the approximation capabilities of normalised networks have been analysed (Benaim 1994), the side-effects of normalisation have not yet been considered in detail.

Normalisation of basis functions arises naturally when basis function networks are viewed in a probabilistic framework, e.g. (Jordan and Jacobs 1994), and is also used in fuzzy logic systems (Takagi and Sugeno 1985). In other contexts 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

*Daimler-Benz Research, Alt-Moabit 96a, D-10559 Berlin, Germany. E-mail: `murray, shorten@DBresearch-berlin.de`

point. When this is the case a *partition of unity* across the input space is said to have been achieved. That the basis functions should form a partition of unity is an important property for basis function networks for a number of reasons. It often results in a structure which is less sensitive to poor basis function parameter selection (centres and widths).

In cases where the basis function network is used within a local model structure (Johansen 1994, Murray-Smith 1994) (i.e. $y = \sum_{i=1}^{M} f_i(\mathbf{x})\bar{\phi}_i(\mathbf{x})$), where the $f_i(\mathbf{x})$'s are local models, a partition of unity ensures that it is possible for the network to exactly represent the target system in the case of an exact structural match between the system and $f_i(\mathbf{x})$[1]. In the case of RBF nets, where $f_i(\mathbf{x})$ is equal to a constant w_i, this means that un-normalised networks are unable to exactly model a constant even in the absence of noise.

1.1 Paper structure

This paper is structured as follows: After an overview and discussion of normalised networks, the effects of normalisation on the form of the basis functions and the network parameters 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 important of these is the change in shape of the basis functions and the possible loss in smoothness of representation. In addition, for basis functions with non-compact support[2], normalisation results in the whole of the input space being covered and not just the part of the space populated by the training data. This can lead to stability problems at the edge of the input space. It is also shown that in the case of irregular networks[3], 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. The effect of normalisation on the least squares solution for the network weights is then discussed. Finally, these effects will be illustrated by means of an example and their consequences for modelling and other applications involving interpolation discussed.

2 Modelling with normalised radial basis functions

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

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

where y is the network output[4], $\mathbf{x}$ is the vector of input variables, M is the number of processing units, w_i is the weight associated with unit i, and ϕ is the non-linear activation function before normalisation.

[1] This assumes a network where the $f_i(\mathbf{x})$'s have the same model structure.

[2] By compact support 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.

[3] We use the term 'irregular' to describe networks where the basis function centres are distributed non-uniformly across the input space, or where differing sizes of basis functions are used.

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

We denote the normalised form of basis function k by $\bar{\phi}_k$, where

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

where $d(\cdot)$ denotes some distance metric, $\mathbf{c}_i$ is the centre of the ith basis function and σ_i is the width of the ith unit. In principle, the basis function can be any non-linear function, but many authors use local basis functions for a number of practical reasons ('local' implying that the basis function is only 'significantly active' for some limited range of the input). Local basis functions are advantageous because of the increased interpretability of the network, the ability to produce locally accurate confidence limits (Leonard *et al.* 1992), and because locality can also be utilised to improve computational efficiency (Murray-Smith and Johansen 1995).

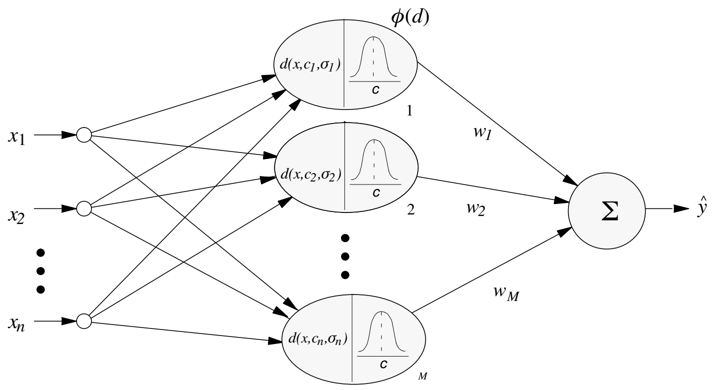


Figure 1: Un-normalised Radial Basis Function network

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 (Hlaváčková and Neruda 1993). 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 all normalised basis functions equals unity, i.e.

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

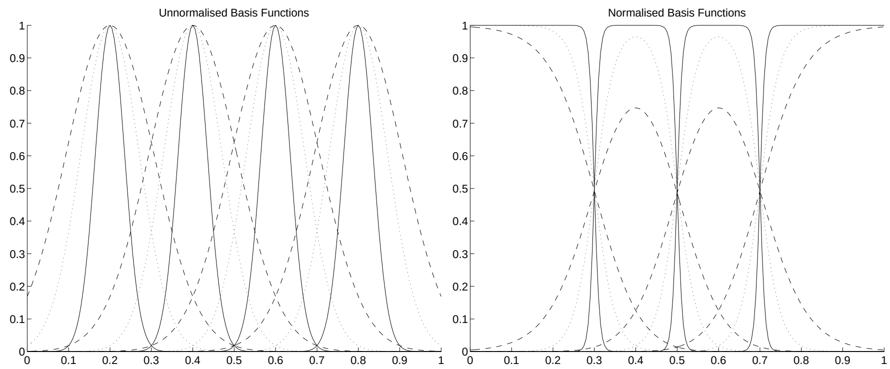
This has the effect of covering every point of the input space to the same degree, unlike the un-normalised case where the total weighting over the input space is non-uniform. The approximation capabilities of such networks have been considered in detail by Benaim (Benaim 1994) and it has been shown that normalised RBF's are capable of *universal approximation* (Kaplan 1991) in a satisfactory sense. Werntges (Werntges 1993) discusses the advantages of normalisation in RBF nets, describes the advantages of a partition of unity produced by normalisation, but does not consider the side-effects discussed in this paper.

3 Side-effects of normalisation

In order to achieve a partition of unity for many network types it is necessary to normalise. However, when the basis functions in the network are non-compact, 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

Un-normalised radial basis function networks usually use homogeneous basis functions, sometimes with differing parameters. 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. Note the decrease in basis function maxima in the normalised case shown in figure 2(b). As the width of the basis function decreases the normalised network becomes less smooth, and tends towards a crisp nearest-neighbour classifier. The effect in two dimensions is shown by the contour plot in figures 6 and 7.



(a) Un-normalised basis functions (b) Normalised basis functions

Figure 2: Change in shape due to normalisation. As the original functions become wider the evenly spaced normalised basis functions become less square and their maxima are reduced. Edge basis functions tend to unity as they move to the limits of their support.

It can also be seen from equation (2) that each normalised basis function is a function of *all* the original basis functions; normalisation introduces a coupling between the original basis functions. Therefore, changing the parameters of one basis function affects all other normalised basis functions in the network. This can have important consequences for on-line applications where the network parameters are updated with each new data point, i.e., a change in the parameters of a single basis function affects all other basis functions in the network.

3.2 Covering of the input-space

In the case where the basis functions used are non-compact in nature, for example when Gaussians are used, then normalisation results in the *whole* of the input space being covered and not just the region of the input space populated by the training data. It can be seen from figures 2(b) and 3 that in the normalised case the activation tends toward unity at the edges of the space.

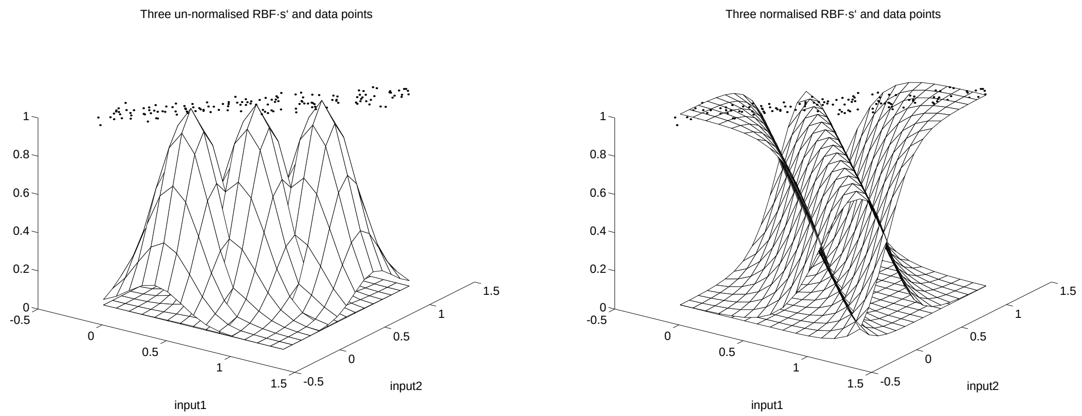


Figure 3: Covering of input space due to normalisation

3.3 Irregular networks: reactivation and shift in maxima

A further difficulty with normalised basis functions involves two phenomena which occur due to normalisation. 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'. These effects are shown in figure 4.

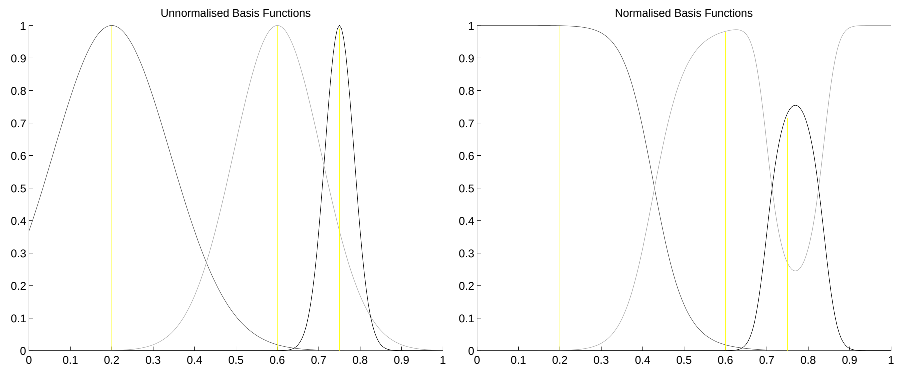
Reactivation occurs when neighbouring basis functions have differing widths. A one-dimensional example shown in figure 5 using two basis functions illustrates how this phenomenon occurs. The reactivation point x, assuming monotonically decreasing basis functions, is the point at which the distance metric d_1 (equation 3) for the first basis function is no longer smaller than d_2 for the second basis function. For a Euclidean distance metric this implies,

$$\left(\frac{x - c_1}{\sigma_1}\right)^2 < \left(\frac{x - c_2}{\sigma_2}\right)^2, \tag{6}$$

leading to a condition for reactivation given by,

$$\frac{\sigma_2}{\sigma_1} < \frac{\mid x - c_2 \mid}{\mid x - c_1 \mid}. \tag{7}$$

Equation (7) 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. In networks with



(a) Un-normalised Basis Functions (b) Normalised Basis Functions

Figure 4: Shift in maxima and reactivation. 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 centres of basis functions to emphasise the centre-shift effect

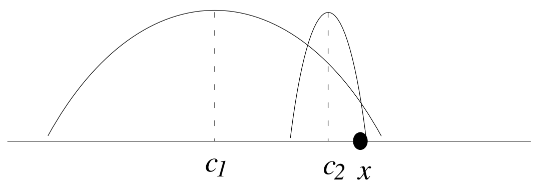


Figure 5: Reactivation example. The point in the input space x where the basis function reactivates can be determined from the units' centres (c_1 and c_2, where c_1 is furthest from the input x) and their widths (σ_1 and σ_2).

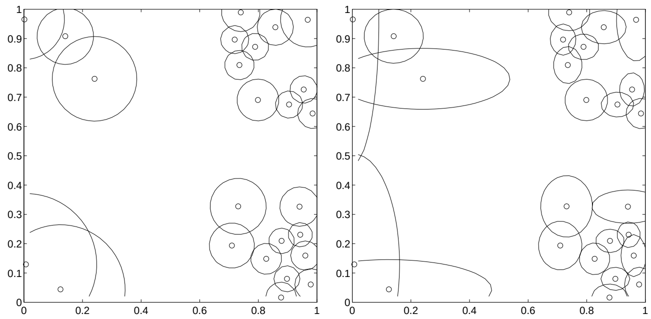
uniformly wide basis functions reactivation cannot occur. The shift in the position of the activation function's maximum occurs when neighbouring basis functions are either unevenly spaced or have differing widths.

This behaviour can cause problems if the network is being used to estimate an underlying probability distribution as is the case when local linear models are weighted by the basis functions used to approximate the function, as in (Johansen 1994, Murray-Smith 1994)[5]. Within this framework, reactivation can lead to local models becoming significantly active in regions in which they were never intended to operate.

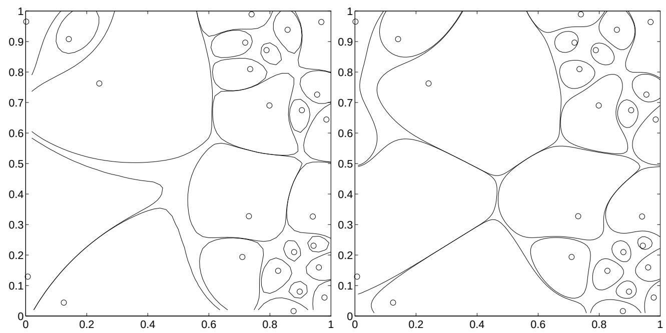
[5]The local learning methods for local model networks discussed in (Murray-Smith and Johansen 1995) require a partition of unity to be able to model the target function.

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 in figure 6.

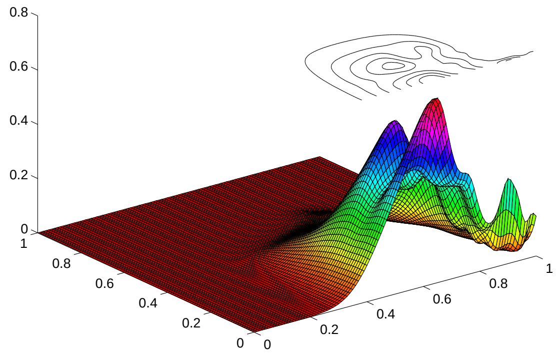


(a) Un-normalised basis functions

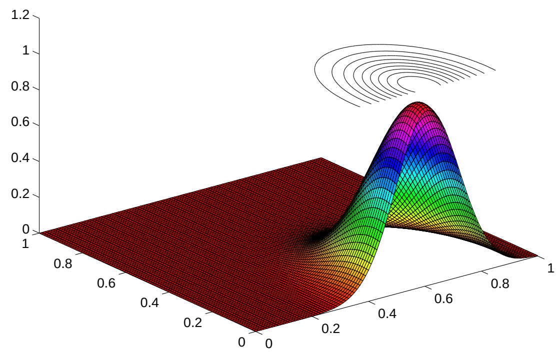


(b) Normalised basis functions

Figure 6: Normalised Radial and Ellipsoidal basis functions. Contours (drawn at a level of 0.5) of the basis functions are shown for a two-dimensional problem. The matrix of basis function centres is the same in each case. Networks employing radial and ellipsoidal distance metrics are used in (a), and the effect of normalising the two networks is depicted in (b).



(a) Normalised Basis Function



(b) Un-normalised Basis Function

Figure 7: Effect of Normalisation on Basis Function Shape. The normalised basis function has a far more complex surface than before, with many local maxima and minima.

4 Network weights and condition number

The concepts of neural network robustness (with regards to variance of the optimisation process and generalisation ability for new data) are closely related to the magnitude of the networks weights $\mathbf{w}$ and the condition number κ of the design matrix ($\mathbf{\Phi}$, defined below). The condition number is a useful indicator of how robust the solution for the network parameters actually is, i.e. how the solution for $\mathbf{w}$ is affected by slightly perturbing the training data. Similarly, large weights can lead to non-robust models. It is therefore of interest to examine the effect of normalisation on the *least squares* weight solution for the network parameters, and the effect of normalisation on the condition number of the network's design matrix.

The decrease in basis function magnitude brought about by normalisation in networks with large levels of overlap (see figure 2) typically results in a compensating increase in weight magnitude. This has the effect of increasing the sensitivity of the resulting model to noisy data or model mismatch. The condition number is strongly dependent on the degree of 'similarity'[6] of overlapping basis functions. Intuitively this implies that, for regular networks with narrow basis functions, normalisation results in an increase in condition number, and for networks with wide basis functions, a decrease in condition number. Therefore, for networks with wide basis functions, normalisation can result in a significant weight increase while at the same time causing an improvement in the condition of the design matrix. Figure 8 shows the effect of basis function width on the condition number of regular networks before and after normalisation.

[6]The degree of similarity is quantified by the linear correlation of basis function responses in their region of overlap.

4.1 Normalisation and the least squares solution

This section discusses the effect of normalisation on the least squares solution for the network weights and on the condition number of the design matrix. We consider approximating the function $y = f(\mathbf{x})$ by a basis function expansion of the form of equation (1), where the basis functions are normalised as in equation (2).

If all of the observations are grouped into vector form with $\mathbf{y}^{\mathbf{T}}$ defined as,

$$\mathbf{y}^T = [y_1, ..., y_N], \tag{8}$$

and

$$\psi_i{}^T = [\phi(d(\mathbf{x}_i; \mathbf{c}_1, \sigma_1))..\phi(d(\mathbf{x}_i; \mathbf{c}_M, \sigma_M))], \tag{9}$$

with

$$\mathbf{\Phi} \;\; = \;\; \begin{pmatrix} \psi_1^T \\ . \\ . \\ \psi_N^T \end{pmatrix}. \tag{10}$$

then,

$$\hat{\mathbf{y}} = \mathbf{C}\mathbf{\Phi}\mathbf{w}, \tag{11}$$

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 are positive over their entire support). In the un-normalised case $\mathbf{C}$ is simply the identity matrix, while in the normalised case $\mathbf{C}$'s entries are given by

$$c_{kk} = \left(\sum_{i=1}^{M} \phi(d(\mathbf{x}_k; \mathbf{c}_i, \sigma_i) \right)^{-1} \qquad \forall \;\; 1 \leq k \leq N, \; C_{ik} = 0, \; \forall i \; \neq k. \tag{12}$$

The solution to the least squares minimisation problem is then given by finding a $\mathbf{w}$ which minimises

$$(\hat{\mathbf{y}} - \mathbf{y})^T (\hat{\mathbf{y}} - \mathbf{y}). \tag{13}$$

Minimisation of (13) for equation (11) yields an expression for $\mathbf{w}$ given by,

$$\mathbf{w} = (\mathbf{C}\mathbf{\Phi})^{+}\mathbf{y}, \tag{14}$$

where $(\cdot)^{+}$ denotes the Moore-Penrose pseudoinverse and is equal to (Noble and Daniel 1988, Golub and Van Loan 1989),

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

when the rank of $\mathbf{\Phi}$ is M. In the following analysis we wish to examine the effect of normalisation on the network weights and on the condition number of the design matrix $\mathbf{C}\mathbf{\Phi}$. Before proceeding some definitions and basic results are necessary:

Definition 4.1 *(Noble and Daniel 1988) The* 2 *norm of a real matrix* $\mathbf{J}$ *is defined,*

$$\begin{aligned} \| \mathbf{J} \|_2 &= \sigma_{max}(\mathbf{J}), \\ &= \sqrt{\lambda_{max}(\mathbf{J}^T\mathbf{J})}, \end{aligned} \tag{16}$$

where $\sigma(\mathbf{J})$ *denotes a singular value of* $\mathbf{J}$ *and* $\lambda(\mathbf{J})$ *denotes an eigenvalue of* $\mathbf{J}$.

Definition 4.2 *(Golub and Van Loan 1989) The condition number $\kappa(\mathbf{J})$ of a rectangular matrix $\mathbf{J}$ is given by,*

$$\kappa(\mathbf{J}) = \frac{\sigma_{max}(\mathbf{J})}{\sigma_{min}(\mathbf{J})}, \tag{17}$$

where $\sigma_{max}(\mathbf{J})$ denotes the maximum singular value of $\mathbf{J}$ and where $\sigma_{min}(\mathbf{J})$ denotes the minimum singular value of $\mathbf{J}$.

Lemma 4.1 *(Söderström and Stoica 1989)*

$$\| \mathbf{J}^{+} \|_2 = \frac{1}{\sigma_{min}(\mathbf{J})}. \tag{18}$$

Proof
The above result follows directly from the definition of the pseudo-inverse.

Theorem 4.1 *Let $\mathbf{Z} = \mathbf{C\Phi}$, where $\mathbf{C}$ is an $N \times N$ diagonal matrix whose diagonal entries are real and positive, and $\mathbf{\Phi}$ is a NxM real matrix. Let $\sigma(\mathbf{J})$ denote the singular values of some matrix $\mathbf{J}$. Then,*

$$\sigma^2_{min}(\mathbf{Z}) \geq c^2_{min}\sigma^2_{min}(\mathbf{\Phi}), \tag{19}$$

$$\sigma^2_{max}(\mathbf{Z}) \leq c^2_{max}\sigma^2_{max}(\mathbf{\Phi}), \tag{20}$$

where c_{min} is the minimum entry of $\mathbf{C}$.

Proof
The singular values of a real matrix $\mathbf{J}$ satisfy,

$$\sigma^2(\mathbf{J}) \quad = \quad \lambda(\mathbf{J}^T\mathbf{J}).$$

From the Raleigh quotient (Noble and Daniel 1988),

$$\sigma^2_{min}(\mathbf{Z}) \quad = \quad \frac{\mathbf{x}^T_{min}\mathbf{\Phi}^T C^2 \mathbf{\Phi}\mathbf{x}_{min}}{\mathbf{x}^T_{min}\mathbf{x}_{min}},$$

when $\mathbf{x}_{min} \neq 0$ is the eigenvector of $\mathbf{Z}^T\mathbf{Z}$ corresponding to the minimum eigenvalue of $\mathbf{Z}^T\mathbf{Z}$. Then,

$$\sigma^2_{min}(\mathbf{Z}) \quad \geq \quad c^2_{min}\frac{\mathbf{x}^T_{min}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{x}_{min}}{\mathbf{x}^T_{min}\mathbf{x}_{min}}.$$

But from the Raleigh inequality,

$$\sigma^2_{min}(\Phi) \quad \leq \quad \frac{\mathbf{x}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \; \forall \; \mathbf{x}.$$

Therefore,

$$\sigma^2_{min}(\mathbf{Z}) \quad \geq \quad c^2_{min}\sigma^2_{min}(\mathbf{\Phi}).$$

The inequality

$$\sigma^2_{max}(\mathbf{Z}) \quad \leq \quad c^2_{max}\sigma^2_{max}(\mathbf{\Phi}),$$

follows using the same arguments as above for the maximum singular values.

Theorem 4.2 *Let* $\mathbf{w} = \mathbf{H}\mathbf{x}$*, where* $\mathbf{w}$ *is a* Mx1 *vector,* $\mathbf{H}$ *is a* MxN *matrix of rank* $M \leq N$*, and* $\mathbf{x}$ *is a* Nx1 *vector. Then,*

$$\| \mathbf{w} \|_2 \geq \frac{\| \mathbf{x} \|_2}{\| \mathbf{H}^+ \|_2}. \tag{21}$$

Proof
A bound on the network weights can be written,

$$\begin{aligned}
\mathbf{w}^T\mathbf{w} &= \mathbf{y}^T\mathbf{H}^T\mathbf{H}\mathbf{y} \\
&= \mathbf{y}^T\mathbf{P}^T\mathbf{D}\mathbf{P}\mathbf{y}, \ \textit{where} \ \ \mathbf{P}^T\mathbf{P} = \mathbf{I} \\
&\geq \lambda_{min}(\mathbf{H}^T\mathbf{H})\mathbf{y}^T\mathbf{y} \\
&= \frac{\mathbf{y}^T\mathbf{y}}{\sigma^2_{max}(\mathbf{C}\mathbf{\Phi})}
\end{aligned}$$

The proof of the theorem follows directly from the definition 4.1.

4.2 Bounds on network weights

Now consider equation (14). By using the triangular inequality an upper bound for the network weights can be written,

$$\begin{aligned}
\| \mathbf{w} \|_2 &\leq \| (\mathbf{C}\mathbf{\Phi})^+ \|_2 \| \mathbf{y} \|_2 \\
&= \frac{1}{\sigma_{min}(\mathbf{C}\mathbf{\Phi})} \| \mathbf{y} \|_2, \text{ from lemma 4.1,} \\
&\leq \frac{1}{c_{min}\sigma_{min}(\mathbf{\Phi})}, \text{ from theorem 4.1,} \\
&= \frac{1}{c_{min}} \| \mathbf{\Phi}^+ \|_2 \, .
\end{aligned} \tag{22}$$

Similarly, using theorem 4.2 and equation (20) from theorem 4.1 the lower bound for the network weights can be written,

$$\begin{aligned}
\| \mathbf{w} \|_2 &\geq \frac{\| \mathbf{y} \|_2}{\| (\mathbf{C}\mathbf{\Phi}) \|_2}, \\
&= \frac{\| \mathbf{y} \|_2}{\sigma_{max}(\mathbf{C}\mathbf{\Phi})} \text{ from definition 4.1,} \\
&\geq \frac{\| \mathbf{y} \|_2}{c_{max} \| (\mathbf{\Phi}) \|_2} \text{ from theorem 4.1.}
\end{aligned} \tag{23}$$

Equations[7] (22) and (23) demonstrate that the bounds on the network weights are dependent on the entries of $\mathbf{C}$ (i.e. the inverse sum of the basis functions). It is noted that c_{min} and c_{max} can be very small when there is a large amount of overlap between the network basis functions. In this case the bound on the network weights will increase significantly as a result of normalisation. An increase in the weights typically occurs when the widths of the un-normalised basis functions are 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 3.4.

[7] These equations are trivial to derive when both $\mathbf{\Phi}$ and $\mathbf{C}$ are square matrices.

4.3 Normalisation and network condition number

Finally, it is noted that equations (20) and (19) can be used to find an upper bound for the condition number of the design matrix $\mathbf{C\Phi}$,

$$\begin{aligned}\kappa(\mathbf{C\Phi}) &= \frac{\sigma_{max}(\mathbf{C\Phi})}{\sigma_{min}(\mathbf{C\Phi})}, \\ &\leq \frac{c_{max}\sigma_{max}(\mathbf{\Phi})}{c_{min}\sigma_{min}(\mathbf{\Phi})} \text{ from theorem 4.1 .} \end{aligned} \tag{24}$$

Equation (24) suggests that the ratio $\frac{c_{max}}{c_{min}}$ is very important in determining the condition number of $\mathbf{C\Phi}$. Note that for large amounts of overlap between the unnormalised basis functions, both c_{max} and c_{min} can be very large, whereas the ratio $\frac{c_{max}}{c_{min}}$ need not be significantly greater than unity. This implies that although the network weights increase significantly, this increase does not imply a significant increase in network condition number. The converse is true for networks with basis functions which do not overlap significantly; a significant increase in condition number can correspond to a decrease in network weights.

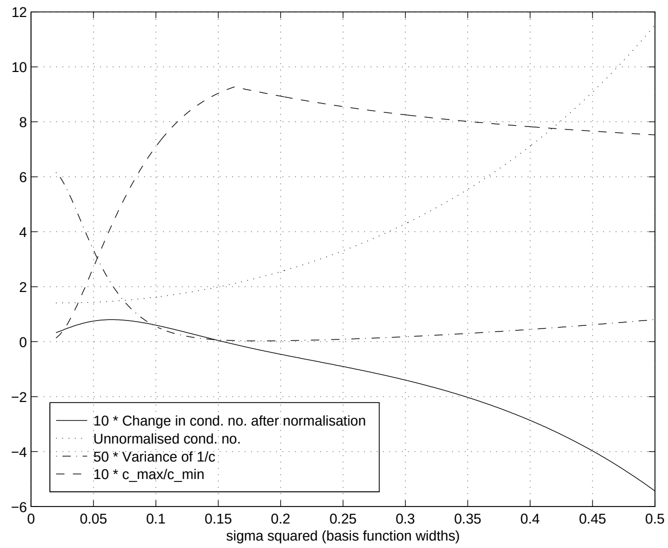


Figure 8: The difference in un-normalised and normalised condition number as a function of overlap between neighbouring basis functions. The ratio $\frac{c_{max}}{c_{min}}$ and the variance of the cumulative network activation (indicating how close the basis functions are to a uniform partition of the input space) are also depicted. Note that for narrow units normalisation results in an increase in condition number, whereas for wider basis functions, it results in a lower condition number. The 'cross-over' point, i.e. the point from which normalisation actually improves network condition is also the point at which $\frac{c_{max}}{c_{min}}$ is closest to unity. $\frac{c_{max}}{c_{min}} = 1$ implies that the un-normalised basis functions form a uniform partition in the area covered by the input data.

In figure 8 we illustrate the effect of overlap on the condition of the design matrix for 11 basis functions spread uniformly on a single input dimension from 0 to 1. The condition numbers of the design matrices are calculated for basis function widths ranging from

$\sigma^2 = 0.02...0.5$, and 10000 data points spread uniformly from 0 to 1. The change in condition in this example is unlikely to have a major effect in learning, but the example illustrates the processes involved and provides tools for their analysis. The effect of normalisation on matrix condition becomes more important in higher dimensional spaces because of the increased level of overlap, and in irregular networks, where reactivation often increases the coupling between basis functions.

5 Illustrative example

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 9. Uniform white noise of amplitude 0.2 was superimposed on the target as depicted in figure 9.

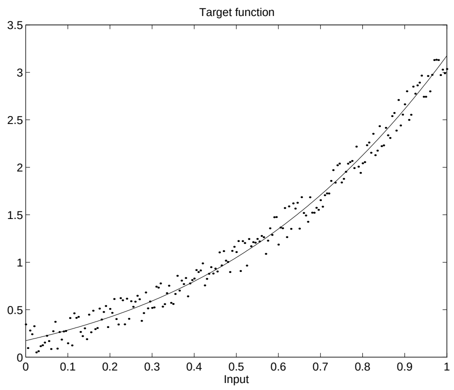


Figure 9: Target function and training data

5.1 Effect of varying the basis function overlap

A network consisting of three basis functions (centred at $[0.3, 0.5, 0.9]$) 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 figure 10. The basis functions are deliberately chosen to have a low level of overlap in order to exaggerate the effects of normalisation. The reactivation and covering of the input space effects can be clearly seen in figure 10. The approximation is very poor because the basis functions are too narrow.

The effect of wide basis functions is presented in figure 11. It can be clearly seen that the network approximation is better than with narrow basis functions, due to the increased level of overlap. Reactivation still occurs in the normalised case.

The diagonal elements of the $\mathbf{C}$ matrix for the above example are depicted in figure 12. It can be clearly observed that the ratio $\frac{c_{max}}{c_{min}}$ is greater for the case of the narrow basis functions. This explains why normalisation increased the condition number in the case

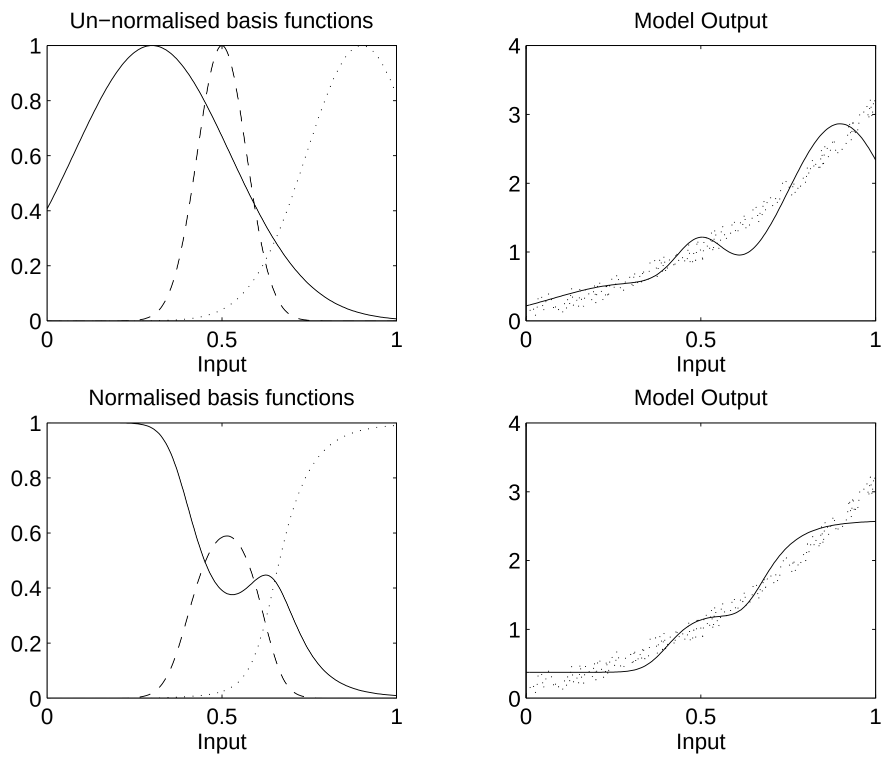


Figure 10: Narrow basis functions (relative to figure 11): The magnitude of the network weights $\mathbf{w}$ decreases due to normalisation (not shown). The condition number of the design matrix in the un-normalised case is $\kappa(\mathbf{\Phi}) = 2.33$ and in the normalised case $\kappa(\mathbf{C\Phi}) = 3.09$. Note the reactivation effects.

of basis functions with a lower level of overlap, despite decreasing the magnitude of the network weights.

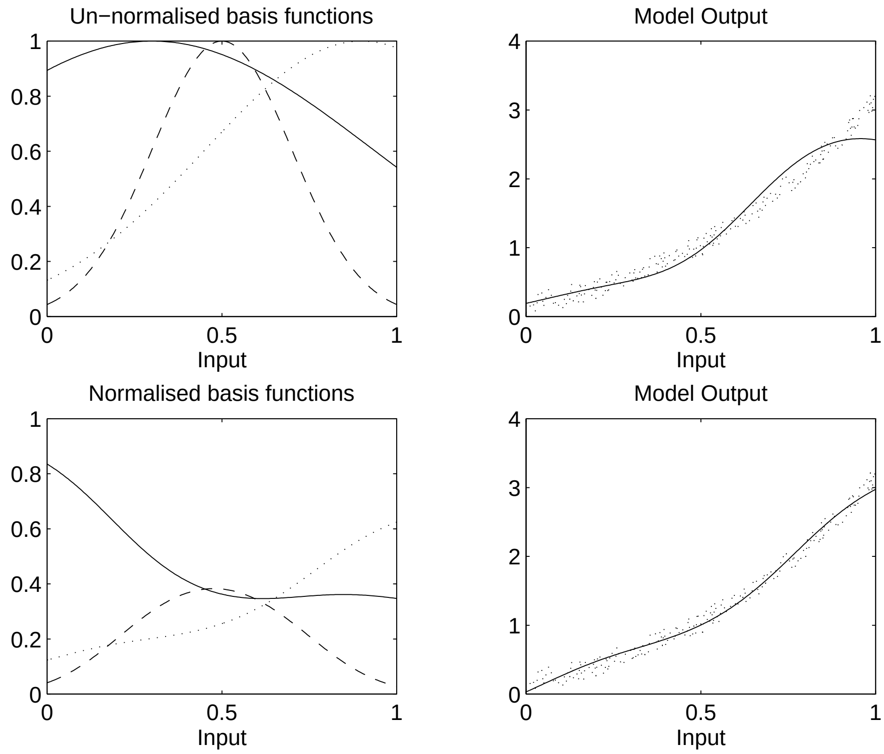


Figure 11: Wide basis functions: Note the better approximation. The shape of the original basis functions change due to normalisation. The network weights also increase significantly (not shown). The condition number of the design matrix in the un-normalised case is $\kappa(\mathbf{\Phi}) = 4.71$ and in the normalised case $\kappa(\mathbf{C\Phi}) = 4.26$

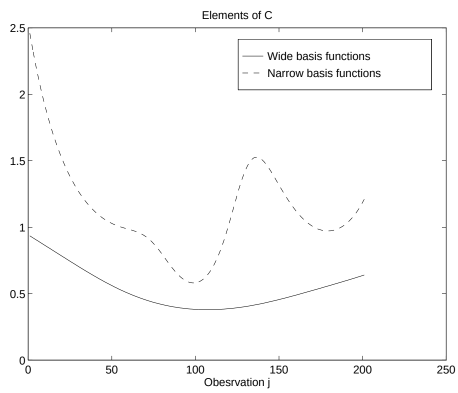


Figure 12: Diagonal elements of $\mathbf{C}$ matrix for wide and narrow basis functions from the examples shown in figure 9, with ordered inputs. Note the large variation of c for the narrower basis functions.

6 Conclusions

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

1. Normalisation leads to a change in shape of basis function.
2. If non-compact basis functions are used, normalisation leads to the whole of the input space being covered. This can result in stability problems at the edges of the space populated by the training data, when networks are used to represent dynamic systems. Ideally, a network should also return an input-dependent confidence estimate which rejects data points outside the area covered by the training data, but in practice the reliable estimation of the probability distribution function is often difficult to achieve.
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 out on RBF nets, i.e. localised behaviour, is no longer guaranteed.
4. The analysis in section 4.1 shows how normalisation affects both the condition number of the design matrix and the magnitude of the parameters $\mathbf{w}$ for a given network. It was demonstrated that large overlap between basis functions leads to a increase in network weights but not necessarily to an increase in network condition number.
5. Effects 1-4 become more pronounced as the input dimension increases.

A partition of unity is highly desirable for many modelling applications. In many network types this partition of unity can only be achieved by normalising the network basis functions. However, normalising non-compact basis functions leads to unavoidable side-effects, which can lead to unpredictable network behaviour. The effects noted in this paper are relevant for any application where an interpolation function is used. In particular, reactivation is an important consideration for intelligent control paradigms (Shorten to appear). Researchers and users of basis function nets, local model nets, and fuzzy systems should consider these effects when designing, interpreting and validating 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 D. Neumerkel, J. Franz, R. Bjørgan and J. Kalkkuhl. The authors also wish to thank the anonymous reviewer who pointed out errors in an earlier version of the manuscript and for making several useful suggestions regarding the content and structure of the paper.

References

Barnes, C., Stan Brown, Gary Flake, Roger Jones, Michael O'Rourke and Y. C. Lee (1991). 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.

Benaim, M. (1994). 'On Functional Approximation with Normalised Gaussian Units'. *Neural Computation* **6**(2), 319–333.

Golub, G. and C. Van Loan (1989). *Matrix Computations.* John Hopkins University Press.

Hlaváčková, K. and R. Neruda (1993). 'Radial Basis Function Networks'. *Neural Network World* **3**(1), 93–101.

Johansen, T. A. (1994). Operating Regime Based Process Modelling and Identification. Ph.D. Thesis. Norges Tekniske Høgskole. Trondheim, Norway.

Jones, R. D., Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, P. S. Lewis and S. Qian (1989). Function approximation and time series prediction with neural networks. Technical Report 90-21. Los Alamos National Lab., New Mexico.

Jordan, M. and R.A. Jacobs (1994). 'Hierarchical mixtures of experts and the EM algorithm'. *Neural Computation* **6**, 181–214.

Kaplan, W. (1991). *Advanced Calculus.* Addison Wesley.

Leonard, J. A., M. A. Kramer and L. H. Ungar (1992). A neural network architecture that computes its own reliability. MIT Industrial Liason Report 3-7-92. MIT, Dept. of Chemical Engineering.

Moody, J. and C. Darken (1989). 'Fast-learning in networks of locally-tuned processing units'. *Neural Computation* **1**, 281–294.

Murray-Smith, R. (1994). A Local Model Network Approach to Nonlinear Modelling. Ph.D. Thesis. Department of Computer Science, University of Strathclyde. Glasgow, Scotland. E-mail:murray@DBresearch-berlin.de.

Murray-Smith, R. and Tor Arne Johansen (1995). Local learning in local model networks. In '4th IEE Intern. Conf. on Artificial Neural Networks'. pp. p40–46.

Noble, B. and J Daniel (1988). *Applied Linear Algebra.* Prentice Hall.

Shorten, R. (to appear). A Study of Hybrid Dynamical Systems with Application to Automotive Control. PhD thesis. Department of Electrical Engineering, University College Dublin, Republic of Ireland, submitted February 1996.

Söderström, T. and P. Stoica (1989). *System Identification.* Prentice Hall.

Takagi, T. and M. Sugeno (1985). 'Fuzzy identification of systems and its applications to modeling and control'. *IEEE Trans. on Systems, Man and Cybernetics* **SMC-15**(1), 116–132.

Werntges, H. (1993). Partitions of unity improve network performance. In 'Proc IEEE Int Conf. Neural Networks, pp. 914-918'.