Normalized Gaussian Radial Basis Function Networks.

Guido Bugmann
Centre for Neural and Adaptive Systems, School of Computing,
University of Plymouth, Plymouth PL4 8AA, United Kingdom.
gbugmann@soc.plym.ac.uk, tel. (+44) 1752 23 25 66, fax (+44) 1752 23 25 40

March 1998

Abstract: *The performances of Normalised RBF (NRBF) nets and standard RBF nets are compared in simple classification and mapping problems. In Normalized RBF networks, the traditional roles of weights and activities in the hidden layer are switched. Hidden nodes perform a function similar to a Voronoi tessellation of the input space, and the output weights become the network's output over the partition defined by the hidden nodes. Consequently, NRBF nets loose the localized characteristics of standard RBF nets and exhibit excellent generalization properties, to the extent that hidden nodes need to be recruited only for training data at the boundaries of class domains. Reflecting this, a new learning rule is proposed that greatly reduces the number of hidden nodes needed in classification tasks. As for mapping applications, it is shown that NRBF nets may outperform standard RBFs nets and exhibit more uniform errors. In both applications, the width of basis functions is uncritical, which makes NRBF nets easy to use.*

Keywords: *Radial Basis Functions, Interpolation, Generalisation, Function Mapping, Dimensionality Curse, Classification, Voronoi tessellation.*

1. Introduction

Normalized Radial Basis Functions (NRBF) differ from standard Radial Basis Functions (RBF) by a seemingly minor modification of their equation (section 2). This results in novel computational properties which have attracted little attention in the neural network community. Moody and Darken (1989) were first to mention Normalised RBF nets without elaborating on their functional significance. However. Servin and Cuevas (1993) noted that normalization gave RBF nets the "same classification properties as nets using sigmoid functions". Cha and Kassam (1995) proposed that "a normalized Gaussian basis function features either localized behavior similar to that of a Gaussian or nonlocalized behavior like that of a sigmoid, depending on the location of its centre". Rao et al. (1997) interpreted NRBF nets as mixture of

expert models and Jang & Sun (1993) saw similarities with fuzzy inference systems. These multiple views reflect the fact that NRBF nodes in the hidden layer behave more like case indicators rather than basis functions proper, as is elaborated in section 2. This property leads to excellent performances in classification tasks, as shown in section 3. One of the key features of NRBF nets is their excellent generalization, a property that can be exploited to reduce the number of hidden nodes in classification tasks. This is achieved by using a new learning rule proposed in section 4 that is demonstrated in classification and mapping examples. NRBF nets have also given very good results in another class of application, trajectory learning in robotics (Althoefer and Bugmann, 1995, Bugmann et al, 1998).

2. Normalized Radial Basis Function networks

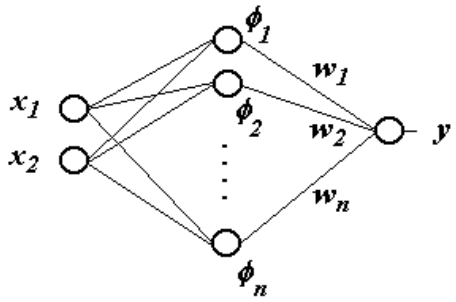


Figure 1: Network architecture for standard RBF nets and Normalized RBF nets (two inputs and one output shown).

Standard Radial Basis Functions (RBF) nets comprise a hidden layer of RBF nodes and an output layer with linear nodes (Broomhead and Lowe, 1988, Moody and Darken, 1989). The function of these nets is given by:

$$y_i(x) = \sum_{j=1}^{n} w_{ij}\, \phi(x - x_j) \qquad (1)$$

where y_i is the activity of the output node i, $\phi(\boldsymbol{x}\text{-}\boldsymbol{x}_j)$is the activity of the hidden node j, with a RBF function centred on the vector $\boldsymbol{x}_j$, $\boldsymbol{x}$ is the actual input vector and w_{ij} are the weights from the RBF nodes in the hidden layer to the linear output node. Such a net is a universal function approximator according to Powell (1987).

The RBF function $\phi(\boldsymbol{x}\text{-}\boldsymbol{x}_j)$ of a hidden node j used here is the Gaussian Radial Basis Function:

$$\phi(x - x_j) = \exp\left(-\frac{\sum_{k=1}^{K}(x_k - w_{jk})^2}{2\sigma^2}\right)$$

where σ is the width of the Gaussian and *K* is the dimension of the input space. The "weights" w_{jk} between node *k* in the input layer and node *j* in the hidden layer do not act multiplicatively as in other neuron models, but define the input vector $\boldsymbol{x}_j = (w_{j1},...,w_{jK})$ eliciting the maximum response of node *j* ($\boldsymbol{x}_j$ *is* the "centre of the receptive field").

In normalised RBF nets, the output activity is normalised by the total input activity in the hidden layer:

$$y_i(x) = \frac{\sum_j W_{ij}\,\phi(x - x_j)}{\sum_j \phi(x - x_j)} \qquad (2)$$

Moody and Darken (1989) proposed that normalisation be performed by the hidden nodes before the summation stage in the output node. In their approach, normalization is a non-local operation, requiring each hidden node to "know" about the outputs of the other hidden nodes. Hence a computationally costly convergence process is required. A similar view is taken in (Rao et al., 1997). In contrast, in our implementation the normalisation is done in the output layer. As it receives already information from all hidden units, the locality of the computational processes is preserved..

Equation 2 shows that, as a result of the normalization, the output activity becomes an activity-weighted average of the input weights in which the weights from the most active inputs contribute most to the value of the output activity. In other words, the roles of output weights and hidden nodes activities are switched. In standard RBF nets, the weights determine how much each hidden node contributes to the output. In NRBF nets, the activity of the hidden nodes determine which weights contribute most to the output. For instance, in the extreme case where only one of the hidden nodes is active, then the output of the net becomes equal to the weight corresponding to that hidden node, whatever its level of activity.

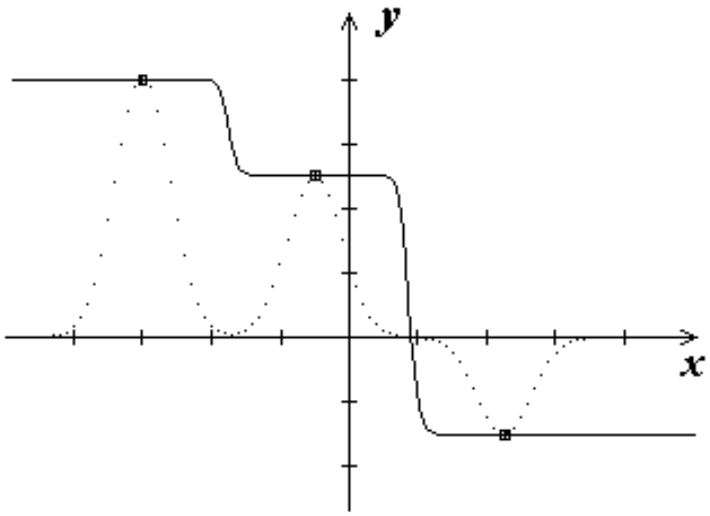


Figure 2. Illustration of the effects of normalization in RBF nets. *Full line:* output *y* of a NRBF net. *Dotted line:* Output of a standard RBF net. Both nets use three hidden nodes centred on the three training data indicated in the Figure.

Figure 2 shows that hidden nodes have a domain of influence in the input space in which they determine the output of the net. This domain is only limited by conflicts with other hidden nodes and has no limits outside of the domain covered by training data. This explains the "sigmoid-like" behaviour noted by Servin and Cuevas (1993) and Cha and Kassam (1995).

When the size of the Gaussians is small, the decay of the hidden nodes activity with distance is so fast that, for most point of the input space, there is usually only one hidden node that contributes significantly to equation 2. As a result, hidden nodes perform a parcelation of the input space similar to a Voronoi tessellation. This is also illustrated by a two-dimensional example in section 4. Due to normalization, RBF nodes become here case indicators rather than basis functions proper. In that sense, NRBF nets are similar to fuzzy inference systems, as discussed in (Jang & Sun, 1993, Andersen et al., 1998). In these systems normalization is a key element of the "centre of gravity defuzzification method (Brown and Harris, 1994, pp 388-404).

In equation 2, normalization is a way to select which weight becomes the output of the net. This is a special case of a more general approach where whole functions are selected. For instance in (Shao et al., 1993) the output y_i is a combination of linear functions $L_{ij}(\boldsymbol{x})$ weighted by the activity of their respective hidden nodes *j*.

Two types of networks are compared in this paper: a standard RBF net, as in equation (1) and a net with a normalizing output node as in equation (2). In both nets the hidden nodes have Gaussian receptive fields, with a width σ indicated in figure captions. The networks (Figure 1) have two inputs, one output and a number of hidden nodes determined by the recruitment procedures during training (section 3.1 and 4.1). Simulations are done on a PC with the neural network development package CORTEX-PRO[1].

3. Classification with Normalized and standard RBF nets.

3.1 Standard training procedure.

For the results shown in this section, training is done in a standard way (Bishop, 1995, p. 170), by recruiting hidden nodes in the first epoch, then, in subsequent epochs, adjusting the positions of the centres of the nodes and the weights to the output node to minimize the output error. Normalized RBF nets and standard RBF nets are trained with the same procedure:

i) recruiting a new hidden node centred on an input vector that was beyond a radius of 0.5σ from the centre of an existing node, or slowly shifting the centre $\boldsymbol{x}_j$ of an existing hidden node towards the new vector $\boldsymbol{x}$ using: $\boldsymbol{x}_j(t+1) = 0.8\boldsymbol{x}_j(t)+ 0.2\boldsymbol{x}$,

ii) modifying the output weight of the hidden nodes j within a radius 0.5σ of the current input vector $\boldsymbol{x}$ so that the output y_i becomes closer to the desired output y_{id} , using: $w_{ij}(t+1) = w_{ij}(t) + learnrate\ (y_{id} - y)$ (typically, $learnrate = 0.5$).

iii) showing the input vectors repeatedly to the net.

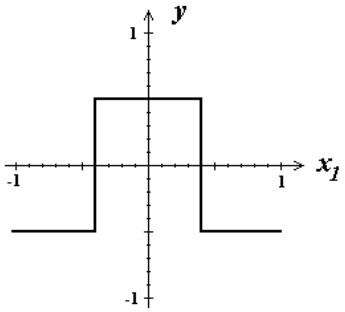
In this procedure, the recruitment of hidden node is purely input-driven, as it depends only on the distribution of training data in the input space. It is a simple procedure that converges rapidly .The number of epoch is indicated in figure captions.

The optimal width of the Gaussian basis functions differs for Normalized and standard RBF nets, and can be estimated by a simple calculation discussed in section 3.2.

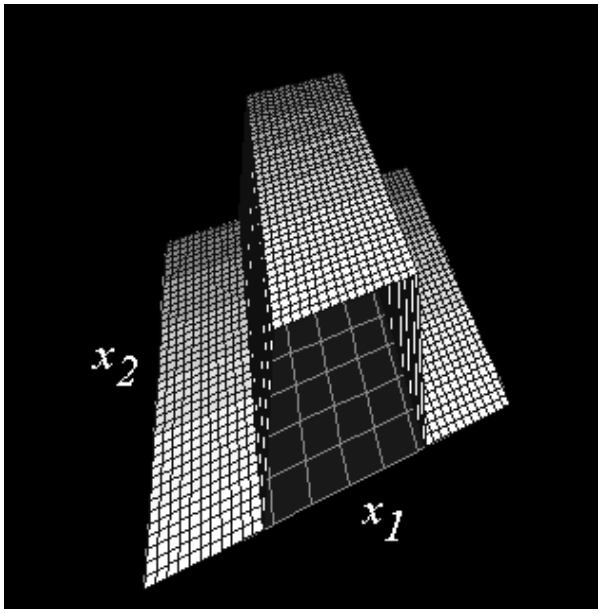
[1]Unistat Ltd, Unistat House, 4 Shirland Mews, London W9 3DY, UK. http://www.unistat.com

3.2 The "plateau-valleys" classification problem.

A simple example is used here to illustrate the difference between normalized RBF net and standard RBF nets. The insights gained will then motivate the design of a new learning procedure (section 4). In this example, a function $f(x_1,x_2)$ is used to divide the input space into two regions: a "plateau" where the $f(x_1,x_2) = 0.5$ and "valleys" where $f(x_1,x_2) = -0.5$ (Figure 3). The problem is to train a network that classifies the regions of the input space by using 70 data points picked at random from the function $f(x_1,x_2)$. The location of most of these points can be inferred from Figure 4.



A)



B)

Figure 3. "Plateau" function $y = f(x_1,x_2)$ used in this example. The data are sampled in the domain $-1<x_1<1$, $-1<x_2<1$. The output $y = -0.5$ everywhere except for $-0.4<x_1<0.4$, where $y = 0.5$. **A)** Cross-section of the function along the plane $\{x_1, y\}$. **B)** Perspective view of the function in the sampling domain. The base grid is placed at y = -0.5.

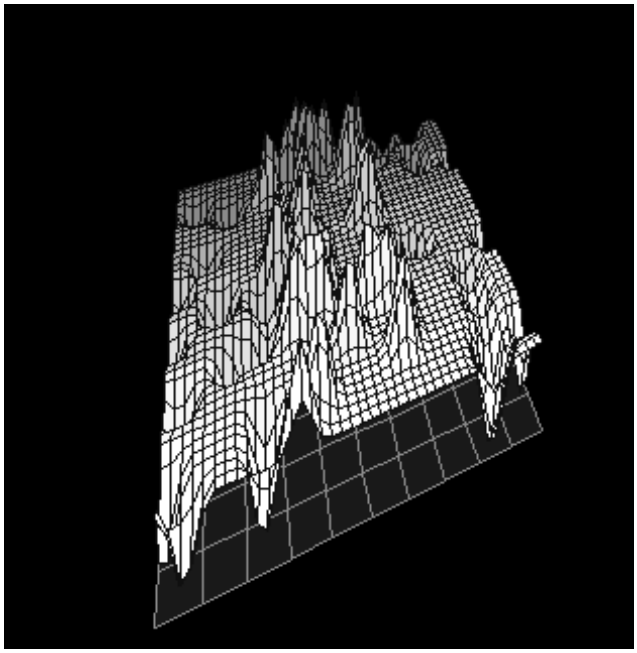


Figure 4: Function of a standard RBF net (equation 1) trained on 70 points sampled from the function shown in Figure 3. The net produces an output close to $y = 0$ over a significant proportion of the input space. The heights of the peaks are either $y = -0.5$ or $y = 0.5$. The base grid is located at $y = -0.5$. Parameters: $\sigma = 0.05$, Average RMS error < 0.003 after 15 epochs. 69 hidden nodes were recruited by the standard training procedure (section 3.1).

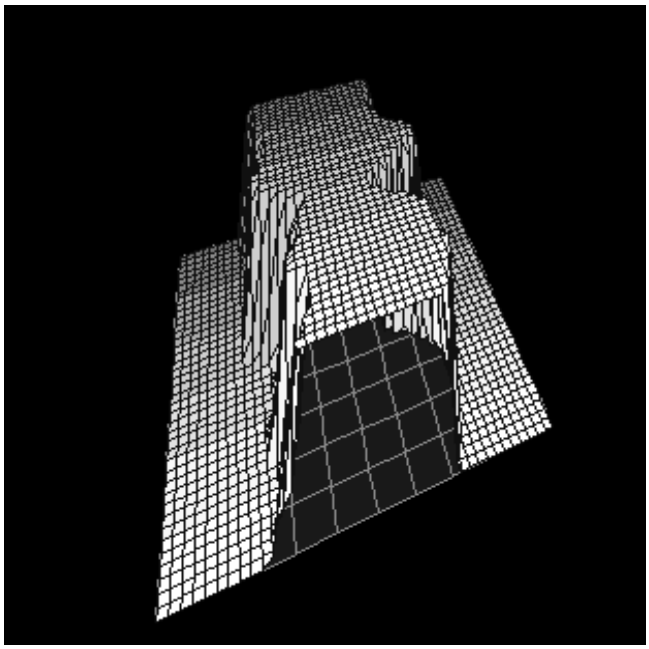


Figure 5: Function of the Normalized RBF net (equation 2) trained on the same points as in Figure 4 with the standard procedure (section 3.1). The outputs of the net are mostly either $y = -0.5$ or $y = 0.5$. Parameters: $\sigma = 0.05$, Average RMS error < 0.0002 after 2 epochs. 69 hidden nodes recruited by the standard procedure (section 3.1).

Figures 4 and 5 show the functions learned by the standard RBF net and the NRBF net using the same small width for the basis functions. A striking effect of normalization is the improved interpolation. Even in regions of the input space where no RBF hidden node produces a strong response, NRBF nodes with receptive fields in surrounding regions can generate a large output value. In contrast, in standard RBF nets, a significant output requires a hidden node with its centre close to the input vector.

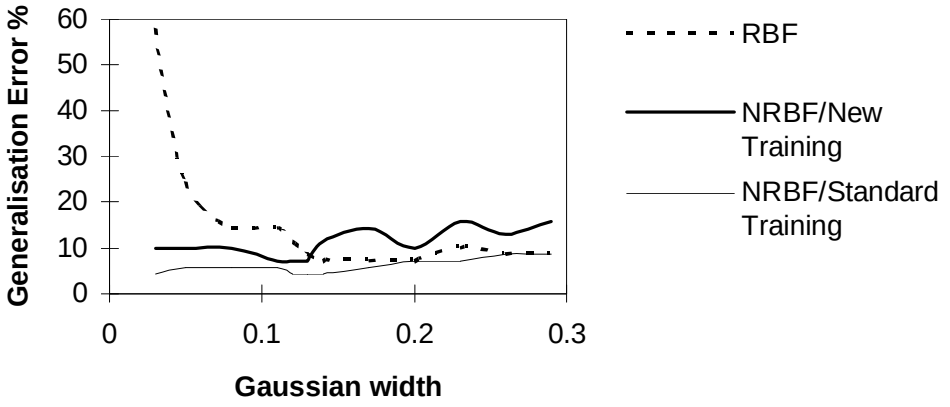


Figure 6: Generalisation error for Normalized and standard RBF nets in a classification task. The RBF net was trained with the standard procedure (section 3.1). The Normalized RBF net was trained with the standard and the new procedure (section 4.1). Both networks were able to classify correctly all training data for Gaussian widths up to 0.29. The curves show the proportion of incorrect classifications of a set of 70 test data randomly selected in the same way as the training data (section 3.1).

In order to estimate the optimal width σ of basis functions, the generalisation error was measured with a test set containing 70 random points selected in the same way as the training data (section 3.1). Figure 6 shows, as expected, that standard RBF nets need basis function sizes large enough for interpolating between data. Their performance then stays relatively stable for larger sizes. This is in part due to the progressively smaller number of nodes recruited as the size increases. In contrast, NRBF nets trained with the standard procedure show a good performance over a wide range of sizes, with better performances for small sizes. Interestingly, the best performance is achieved for a size $\sigma = 0.12$ which is exactly the *average* distance between a data point and its nearest neighbour in this training set. This size leads also to good generalization in mapping problems (section 4.3). This points to a simple rule of thumb for an initial selection of the sizes of basis functions for NRBF nets.

For standard RBF nets, the optimal size σ tends to be closer to one half of the *largest* distance between nearest neighbours (which is here 0.46), reflecting the need to ensure large enough output values over the largest empty space between training data.

3. 3 The two-spirals classification problem.

Figure 7 illustrates the extrapolation properties of NRBF networks with the two-spirals benchmark problem. The network was trained with all 194 data points shown in Figure 7. It can be seen that areas outside of the immediate neighbourhood of training data are classified according to their nearest neighbour in the training data, or more precisely their nearest hidden node centre. This results in a function very similar to the one obtained with Multi-Layer Perceptrons using sigmoid functions in the hidden layers (Lang & Witbrock, 1988). Indeed training is much faster here, with no classification errors after 4 epochs, and good generalisation may be inferred from Figure 7.

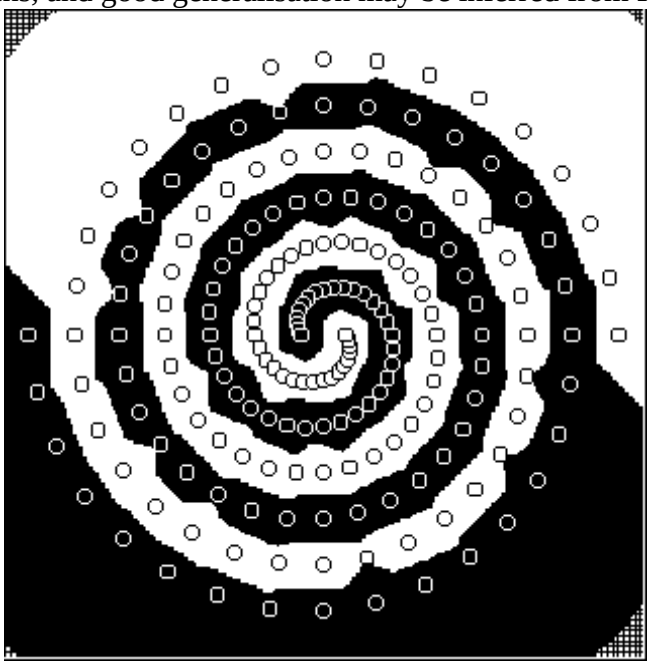


Figure 7: Two spirals classification. White and black areas indicate the two classes assigned to these areas by a Normalised RBF net. The hashed areas indicate unclassified areas due to activities of hidden nodes smaller than the precision of the computer. The 194 training data[2] are indicated by filled and empty symbols . Network parameters: σ = 0.28, Average RMS error = 0.057 after 4 epochs, 96 nodes recruited using the modified training procedure (section 4.1). Figure range: $-7 < x_1 < 7$, $-7 < x_2 < 7$.

[2] Data from the Carnegie Mellon AI repository: http://www-cgi.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/neural/bench/cmu/

NRBF nets have also been applied to the IRIS benchmark problem, matching the best published performances (Grant & Bugmann, in preparation).

4. Towards using less hidden nodes.

4.1 Modified training procedure.

The good interpolation and extrapolation properties shown in the previous section suggests that hidden nodes may need to be recruited only in crucial points, close to boundaries between two classes. To verify this hypothesis, the training procedure in section 3.1 was modified so that no new nodes are recruited if the network indicates the correct class by using existing nodes. The details of the modified procedure are:

i). Check if the output vector $\boldsymbol{y}$ of the net is correct, i.e. $|\boldsymbol{y}-\boldsymbol{y}_{desired}| < tolerance$, where a *tolerance* = 0.4 is adequate for classification problems (it may be reduced to much smaller values for mapping problems, where it represents the desired accuracy).

ii). If the output is correct. Do nothing, go to the next training data.

iii). If the output is incorrect and there is no node with its centre close to the current data point, recruit a new node centred on the data point and set its output weights to the desired output vector.

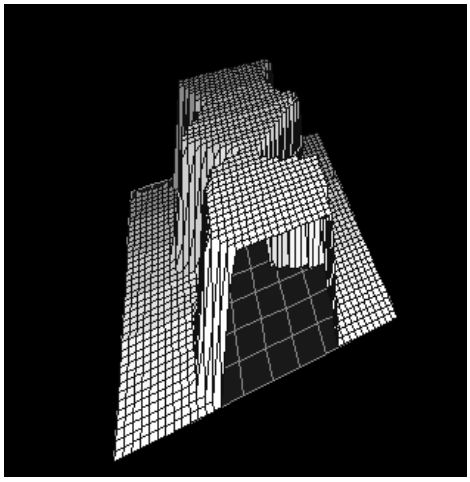
iv). If the output is incorrect and there is a node with centre close to input vector $\boldsymbol{x}$, move its centre $\boldsymbol{x}_j$ closer to the new vector $\boldsymbol{x}$ using: $\boldsymbol{x}_j(t+1) = 0.8\boldsymbol{x}_j(t)+ 0.2\boldsymbol{x}$, and modify its output weights using: $w_{ij}(t+1) = w_{ij}(t) + learnrate\cdot(y_{id} - y_i)$.

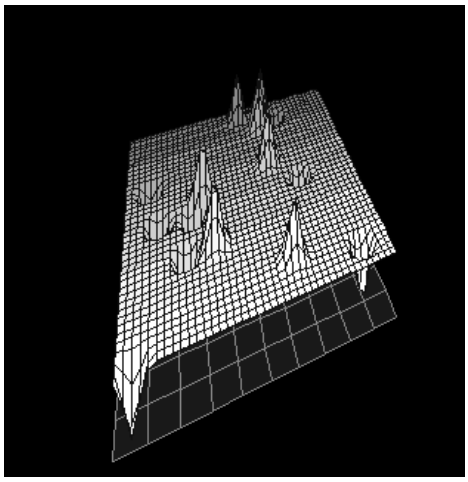
The main difference of this procedure with the standard one is that the recruitment of hidden nodes is also output-driven. It results in progressively less new nodes being recruited at each epoch, as the coverage of the class domains is refined. Generally, after 3 epochs, recruitment ceases and performance improvements are solely due to weights and centre modifications. It is not a optimal procedure[3], because the location of the centres of the recruited nodes depends on the order of presentation of the training data. This explains the variability of the curve in Figure 6.

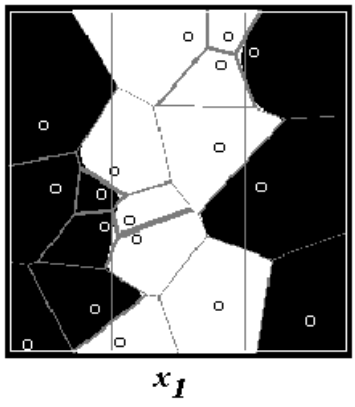
[3]This is of no consequence for the demonstration of the principle in this paper.

4.2 Classification with the modified learning rule.

Figure 8A shows the function of a Normalized RBF net trained with the new procedure on the same data as in the previous section. All training data are correctly classified using only 15 nodes, as compared to the 69 recruited by the standard procedure (Figure 4). Comparing Figure 5 and 8A reveals small differences such as a slightly wider plateau in front and a more narrow middle section in Figure 8A. As there are less nodes in the NRBF net for Figure 8A, each node is in charge of a wider area of the input domain. Figure 8C shows that some of the domains covered by each hidden nodes contain class boundaries. Thus coarser parcelation explains the larger generalisation errors of the modified procedure compared to the standard one (Figure 6).


A)


B)



C)

Figure 8: A) Function of the Normalized RBF net trained with the new procedure (section 4). *Parameters:* σ = 0.05, Average RMS error = 0.0041, 15 hidden nodes recruited. **B)** Non normalized output of the net, plotted to indicate the positions of the recruited nodes. **C)** Partitions of the input space generated by the recruited hidden nodes. The two vertical grey lines indicate the boundaries of the "plateau". Other grey lines indicate regions where neighbouring nodes have similar activities within a 20% margin.

Figure 8B shows the location of the recruited nodes. Comparison with Figure 4 indicates that recruited nodes are now mainly located at class boundaries, tending to form polarised pairs, with one member on each side of the boundary. This suggests that anti-symmetric basis functions reminiscent of the Gabor functions observed in visual cortex (Marcelja, 1980; Jones & Palmer, 1987), may constitute ideal oriented basis functions for classification problems with normalizing nets. The simplest candidate is probably:

$$\phi(x - x_j) = d \cdot \exp\left(-\frac{\sum_{k=1}^{K} (x_k - w_{jk})^2}{2\sigma^2}\right)$$

Where *d* is the distance from the class boundary, positive on one side and negative on the other.

4.3 Function mapping with the modified learning rule.

Normalization gives a wider range of action to each hidden node in NRBF nets. Over the whole of this extended range, a hidden node tends to cause the net to produce the same output value. This makes NRBF nets well suited to classification tasks. Thus it is somehow against expectations that NRBF nets perform also well in mapping tasks.

The example used here as target function is a half cylinder from which 70 training data have been sampled at random (Figure 9A). Normalized and standard RBF nets were trained with sizes of basis functions producing optimal generalization, determined from Figure 10. Generalization was tested using a set of 70 test data picked at random in the input domain $-1 < x_1 < 1$, $-1 < x_2 < 1$.

Comparing the parameters of Figure 9B and 9C shows that the NRBF net trained with the modified learning procedure uses slightly more nodes than the standard RBF net and generalizes slightly better. The NRBF net has a smaller training error and the outputs for all the training data are within the required 5% tolerance set in the procedure. The performances of the NRBF net can be improved by setting a smaller tolerance. For instance, with a tolerance of 1%, the av. RMS error drops to 0.007 in 100 epochs, with an Av. RMS generalization error of 0.051 but with 6% of training data still out of tolerance and 59 nodes being recruited. The standard RBF net cannot achieve such a small training error, due to the large size of the basis functions required for best generalization. It also cannot reach the 5% tolerance criteria for 9% of the

data points. Hence NRBF nets trained with the modified procedure produce better and more consistent performances.

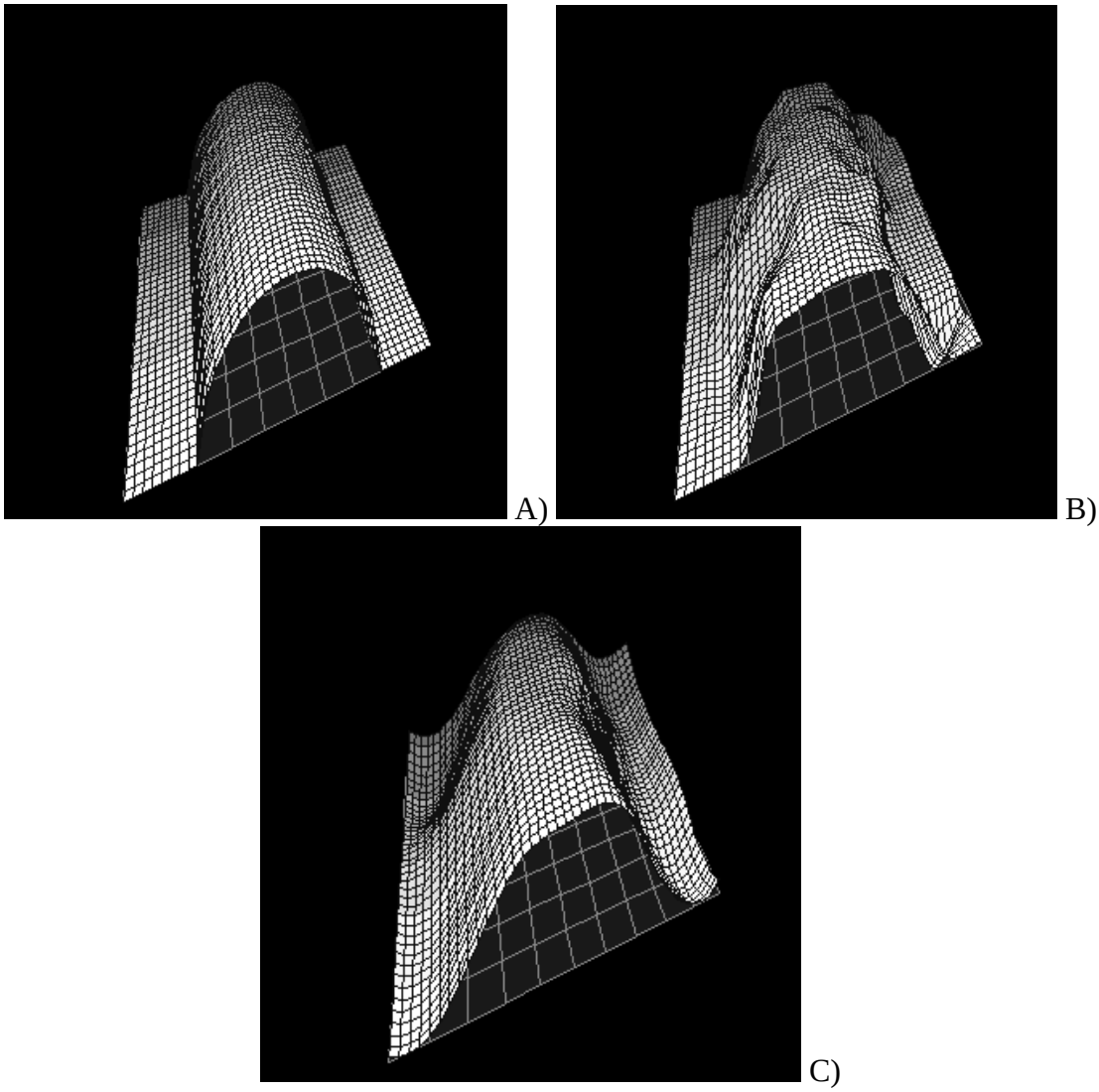


Figure 9. Function mapping with Normalized and standard RBF nets with parameters of best generalization determined in Figure 10. **A)** Function from which 70 training and test data points have been selected at random. The function is a half cylinder of radius 0.6 placed on a base at y = - 0.5. All other dimensions as in Figure 2. **B)** Mapping learnt by a NRBF net. Parameters: $\sigma = 0.12$, Av. RMS training error = 0.019 after 24 epochs (network output within a 5% tolerance for all training data), Av. RMS test error = 0.056. 46 nodes recruited using the modified procedure (section 4.1). **C)** Mapping learnt by a standard RBF net. Parameters: $\sigma = 0.37$, Av. RMS training error = 0.039 after 15 epochs (training stopped to avoid overfitting). The network output was not within the 5% tolerance for 9% of the training data. Av. RMS test error = 0.074. 43 nodes recruited by the standard training procedure (section 3.1).

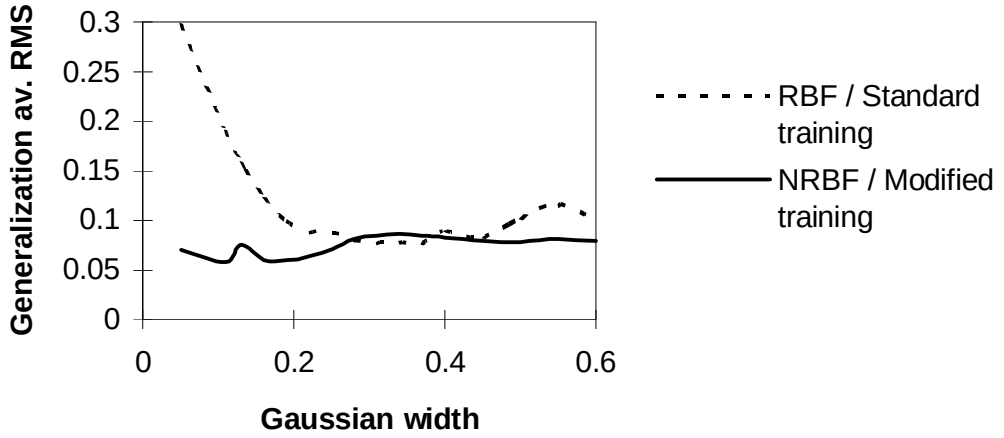


Figure 10: Generalisation error for Normalized and standard RBF nets in a mapping task. The RBF net was trained with the standard procedure (section 3.1). The Normalized RBF net was trained with the modified procedure (section 4.1). The generalization error was determined with 70 test data randomly selected in the same way as the training data (section 4.3).

5. Conclusion

These initial results show that Normalized RBF nets have very good generalisation properties that are beneficial in classification tasks. This is due to the property of normalised RBF nets to produce a significant output even for input vectors far from the centre of the receptive field of any of the hidden nodes.

Taking advantage of this, a modified learning procedure has been proposed in which hidden nodes are recruited only when neighbouring nodes do not point to the same output value. The modified learning procedure enables NRBF nets to operate with significantly less hidden nodes in classification tasks.

When applied to classification problems, the modified learning procedure results in nodes being recruited mainly along class boundaries. This obviates the need for a dense coverage of class domains, in contrast to standard RBF nets. Thus NRBF nets may contribute to soften the curse of dimensionality associated with networks of localized basis functions.

The modified learning rule also performes well in a function mapping example where the NRBF net outperforms the standard RBF network, in terms of training error and

generalization error. Interestingly, it leads to a more uniform performance over the training domain.

A very positive aspect of NRBF nets is that the size of basis functions is relatively uncritical for the performance in classification and mapping. That makes this type of network very easy to use.

Acknowledgements:
Stimulating comments and suggestions by anonymous referees are gratefully acknowledged and have helped to improve the final version of this paper.

References

Althoefer K. and Bugmann G.(1995) "Planning and Learning Goal-Directed Sequences of Robot-Arm movements", in Fogelman-Soulié F. and Gallinari P. (eds), Proc. of ICANN'95, Paris, Vol. 1, 449-454.

Andersen H.C., Lotfi A. & Westphal L. (1998) "Comments on 'Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems' Submitted to IEEE Trans. on Neural Networks, June, 1997 (http://www.elec.uq.edu.au/~andersen/).

Bishop C.M. (1995) "Neural networks for pattern recognition". Clarendon Press, Oxford, UK.

Bugmann G., Koay K.L., Barlow N., Phillips M. and Rodney D. (1998) "Stable Encoding of Robot Trajectories using Normalised Radial Basis Functions: Application to an Autonomous Wheelchair", To appear in Proc. 29th Intl. Symp. Robotics, 27-30 April, Birmingham, UK (Downloadable from: http://www.tech.plym.ac.uk/soc/staff/GuidBugm/Bugmann.html).

Broomhead D.S. and Lowe D. (1988) "Multivariable Functional Interpolation and Adaptive Networks", Complex Systems, 2, pp. 321-355.

Brown M. and Harris C. (1994) "Neurofuzzy Adaptive Modelling and Control", Prentice Hall, Hemel Hempstead, UK

Jang J.-S. R. & Sun C.-T. (1993) "Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems," IEEE Trans. on Neural Networks, vol. 4, pp. 156-159, Jan. 1993. (http://www.cs.nthu.edu.tw/~jang/publication.htm)

Cha I. & Kassam S.A. (1995) "Interference Cancelation Using Radial Basis Function Networks", Signal Processing, 47, pp. 247-268.

Jones J.P. & Palmer L.A. (1987) "The two-dimensional spatial structure of simple receptive fields in cat striate cortex". Journal of Neurophysiology, 58, pp. 1187-1211.

Lang K.J. & Witbrock M.J. (1988) "Learning to tell two spirals apart". In Touretzky D.S., Hinton G.E. & Sejnowski T.J. (eds). Proc. of the 1988 Connectionist Models Summer School, Morgan Kaufmann Publishers, San Mateo, CA.

Marcelja S. (1980) "Mathematical description of the response of simple cortical cells". J. Opt. Soc. Am., 70, pp. 1297-1300.

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

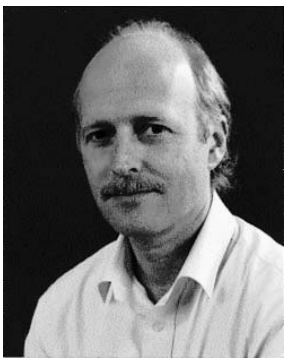
Powell M.J.D. (1987) "Radial Basis Functions for Multivariate Interpolation: A Review", in Mason J.C. and Cox M.G. (eds) "Algorithms for Approximation", Clarendon Press, Oxford, pp. 143-167.

Rao A.V., Miller D., Rose K. & Gersho A. (1997) "Mixture of experts regression modeling by deterministic annealing", IEEE Trans. on Signal Processing, 45, pp. 2811-2820.

Servin M. & Cuevas F.J. (1993) "A New Kind of Neural Network Based on Radial Basis Functions", Revista Mexicana de Fisica, 39:2, pp. 235-249.

Shao J., Kee Y.C. and Jones R. (1993) "Orthogonal Projection Method for Fast On-Line Learning Algorithm of Radial Basis Function Neural Networks", Proc. INNS World Congres on Neural Networks, Portland Oregon, USA, Vol.3, pp. 520-535.

Biosketch of the author



Dr Guido BUGMANN was born in 1953 and has two children. He studied Physics at the University of Geneva in Switzerland. In 1986 he completed a PhD on "Fabrication of photovoltaic solar cells with a-Si:H produced by anodic deposition in a DC plasma". He then worked at the Swiss Federal Institute of Technology in Lausanne on the development of a measurement system using an ultra-sound beam and neural networks to measure the size of air bubbles in bacterial cultures. In 1989 he joined the Fundamental Research Laboratories of NEC in Japan and modelled the function of biological neurons in the visual system. In 1992 he joined Prof. John G. Taylor at King's College London to develop applications of the pRAM neuron model and develop a theory of visual latencies. In 1993 he joined the group of Prof. Mike Denham at the University of Plymouth (UK) where he is developing artificial vision systems for robots and investigates path-planning and spatial memory.

Dr Bugmann has 3 patents and over 70 publications. He is member of the Swiss Physical Society, The Neuroscience Society and The British Machine Vision Association.