Comparison of Unsupervised and Supervised Training of RBF Neural Networks. Case Study: Mapping of Contamination Data

V. Polishchuk, M. Kanevski
Institute of Nuclear Safety, B.Tulskaya 52, 113191 Moscow, Russia. e-mail: kotya@ibrae.ac.ru

Abstract

Algorithms of supervised and unsupervised learning of Radial Basis Function (RBF) neural network parameters were applied in environmental modelling domain. Both local (gaussian) and global (thin plate spline) RBF neural networks were studied. Geostatistical tools (variography) were used as qualitative and quantitative measure of the accuracy of the results obtained. Two real-world case studies (^{137}Cs soil contamination after Chernobyl fallout and heavy metal sediments in Lake Geneva) were considered.

Introduction

Radial Basis Function Neural Networks (RBFNN) are used widely in different machine learning tasks. Application of RBFNN can be found in the problems of time-series prediction, pattern and speech recognition, modelling natural phenomena etc. RBF networks possess the property of best universal approximation (Poggio and Girosi, 1989). The characteristic feature of RBFNN is strong motivation for unsupervised learning of the group of its parameters, specifically the RBF centres locations, via clustering algorithms. In the family of clustering algorithms employed for RBFNN training the Generalised Lloyd Algorithm (GLA, K-means, Lloyd, 1982) is most common; EM algorithm (Dempster et al., 1977) may provide further improvement of RBFNN performance; Kohonen's Self-Organising Nets (SOM, Kohonen, 1982) are also plausible. Previous work (Dowla and Leah, 1995; Wang et al., 1999; Demyanov et al, 1999) showed that RBFNN trained with clustering algorithms can be successfully applied to environmental modelling task. In this work we investigate weather supervised learning of RBFNN parameters with standard gradient descent optimisation methods may contribute to the decrease of RBFNN error rate on training and test data sets. The question of improvement of RBFNN performance by means of supervised learning has been addressed by a number of authors (Wettschereck and Dietterich, 1992; Nabney, 1999). From model-driven approaches for supervised learning the error accumulation method developed by Fritzke (Fritzke, 1994) is worth mentioning. Even such novel technique as Support Vector Machine (Cortes and Vapnik, 1995) is used for RBFNN centres positioning (Schölkopf et al., 1996). It was found that in almost all cases application of supervised learning for RBFNN training improves the RBFNN performance. The opposite results are rarely reported. The use of global RBF is also not common. This paper provides an example of case studies where supervised learning of RBFNN parameters failed to decrease the test error of the optimal net. In addition, the best net found was the one with global (tps) RBFs, for which the motivation for unsupervised learning of centres locations is weaker.

The environmental 2D data can be visualised simply while at the same time they are complex enough to offer a good case study for comparing different models and training algorithms. The most commonly occurred problems are multivariate and multiscale structures of data, high spatial variability, presence of noise and outliers. Spatial trends limit application of stationary geostatistical models like ordinary kriging. Artificial neural networks offer an alternative way of dealing with spatial information when no stationarity of the model is assumed.

In the present study geostatistics is widely used from the beginning of the analysis (exploratory data analysis, data splitting, spatial continuity description - variography) until the interpretation of the results.

Background

Standard 3-layer RBFNN expresses the regression function in the form:

$$f(x) = \sum_{j=1}^{M} w_j h(\| x - c_j \|^2 / 2\sigma_j^{\ 2}) + b \quad (1),$$

where M is the number of hidden nodes.

The parameters of RBFNN are c_j - positions of RBF centres, σ_j - scale factor (the variance if Gaussian RBF is used), w_j - second layer weights and b - bias.

RBFNN training may be divided into two stages. During the first stage relatively fast clustering methods for finding centres positions can be used with a given number of hidden units in the model. This stage makes use only of the monitoring network (distribution of points in the input space). There also exists a number of heuristics to set the variances if the model employs local variance for each RBF or to set the global scale factor if it is common for all hidden nodes.

When the number, type and centres' positions of basis functions are fixed, the problem of finding optimal weights becomes linear. Pseudoinverse activation matrix ($\|H_{ij}\| = h_j(x_i)$, x_i - training data set) can then be used for solving the equation for optimal weight vector:

$$w=(H^T H)^{-1} H^T t \qquad (2),$$

where t is the target data vector.

Scaled conjugate gradients (scg) algorithm, introduced by Muller, 1993, makes use of the fact that the product of the Hessian of the error function to be minimised by a vector can be evaluated efficiently without explicitly computing the Hessian itself.

In order to control the quality of the spatial model developed by RBFNN, geostatistical structural analysis (variography) was used. The main measure for the description of spatial continuity is the so-called variogram $g(h)=E\{\ (Z(x)\ -\ Z(x+h))^2\ \}$, where $E\{\ \}$ is an operator of mathematical expectation, h - vector of separation distance.

Case study

The case study considered deals with radioactive soil contamination with caesium 137 (^{137}CS) originated after the accident at Chernobyl Nuclear Power Plant. Training set consists of 300 and test set of 365 samples of contamination level as a function of two location co-ordinates. More detailed description of data can be found in Kanevski et al, 1999a, Demyanov et al., 1999.

RBFNN Training

A number of algorithms for RBFNN training has been developed and widely used. From those designed for unsupervised learning of RBFNN centres positions GLA and EM were chosen for they proved to be effective in the previous work (Demyanov et al., 1999). From gradient optimisation algorithms we opted for scaled conjugate gradients (Moller, 1993). Other algorithms (quasiNewton, BFG) gave rise to the same results. Besides, regardless of the preceding training procedure, second layer weights were finally recomputed using pseudoinverse activation matrix. It is known that pseudoinverse solution (2) for second layer weights yields theoretically the least squares solution to the approximation problem. But in practice pseudoinverting is a complicated task, especially when dealing with sparse matrices. Different algorithms developed for pseudoinverting may not produce satisfying result. Hence, adjusting second layer weights by means of pseudinversion might not lead to decrease in the training error. An independent investigation showed, however, that final training with pseudoinverting never increased the error, moreover, in most cases it diminished the training error considerably. Thus we have proved the efficiency of the pseudoinvertion algorithm used (pinv function of Matlab) and grounded our choice of final weights adjustment with pseudoinverse activation matrix.

4.1 Local (Gaussian) RBFNN

For EM algorithm is known to have many local minima, some simple heuristics for setting initial values of the parameters to be optimised with EM algorithm must be applied. GLA stands for such a heuristics. It seemed reasonable to apply GLA for initialising RBF centres locations not only during unsupervised learning of them, but also if further optimisation is carried out with standard optimisation algorithm. However, initial centres positioning with GLA not always helped to find optimal solution with scg. RBFNN trained with scg alone starting from random point almost always produced smaller training error. As far as the rate of convergence is concerned, it was found that scg converges almost equally fast regardless of weather the initial values for centres positions were found with GLA or were chosen randomly. Only a little negligible improvement in the rate of convergence was observed when GLA was the first stage of parameters optimisation.

Next we studied the influence of EM algorithm on the solution. Likewise GLA, EM algorithm did not contribute to reduction of RBFNN error. Being caught in local minima, scg applied after EM lead to decrease in RBFNN performance in comparison with "pure" scg.

4.2 Global (tps) RBFNN

On the contrary, training global RBFNN with scg failed to improve the RBFNN performance. At the same time the best RBFNN found (having minimum test error) was the one with thin plate spline - global - RBF. The net, having 85 hidden nodes, was trained with 10 iterations of GLA and 60 iterations of EM algorithm. The second layer weights was, naturally, determined using pseudoinverse activation matrix.

The full sequence of algorithms tried along with their classification is summarised in Table 1. Each of the algorithms may or may not be present in the training procedure. Conclusions for the usefulness of the algorithms are presented as well.

	Simple scheme for RBFNN training			Standard optimiser	RBF-peculiar
Learning	Unsupervised		Supervised	Supervised	
Algorithm	GLA	EM	Pseudoinv	scg	Pseudoinv
Local RBF	Unnecessary			Useful	
Global RBF	Useful			Unnecessary	

Table 1. Algorithms used for RBFNN training

Results

For RBFNN training modified Netlab code for MATLAB (available from http://www.ncrg.aston.ac.uk/ netlab) was used (Bishop and Nabney, 1997).

It can be seen (Figure 1) that RBF centres positions and variances found with EM algorithm settles on the data, revealing its clustering structure. RBFNN parameters found with scg are somewhat different.

RBFNN outputs vs. measurements (target data) are presented in Figure 2.

Omnidirectional variograms for the data sets, 85 tps RBFNN predictions and residuals are presented in Figure 3 for the 1st case study. Training and testing data are spatially correlated with correlation distance about 38 km. There exists spatial non-stationarity (variograms go above a priori variance level). Variograms of the residuals both for training and test data demonstrate pure nugget effects. From geostatistical point of view it means that all spatially correlated information has been extracted by RBF neural network. Geostatistical analysis of both original data and RBFNN outputs and residuals was carried out with GeoStat Office (Kanevski et al., 1999b) software (available from http://www.nsi.ibrae.ac.ru/~mkanev).

Training procedure, which includes clustering algorithm, may be used for training RBF net on the net's squares of the residuals (mapping of variance). It is possible then to draw «thick» contour isolines using the results of the function and variance predictions (Figure 4). «Thick» isolines consist of 3 isolines: conditional mean value predicted by RBFNN $Z_m(x,y)$ and two isolines corresponding to $Z_m(x,y)\ \pm 2\sigma(x,y)$. This kind of visualisation gives a fair impression of spatial uncertainty as well. It should be noted that in case of normally distributed data «thick» isolines correspond to regression mapping along with 95% confidence intervals.

Conclusion

Supervised learning of RBFNN parameters helped to minimise considerably the training error for local (Gaussian) RBFNN. Still, the error on test (unseen) data of local RBFNN trained with scg is big in comparison with the test error of global (thin plate spline) RBFNN trained with simple two-stage procedure. Supervised learning had no positive effect on global RBFNN performance. Thus, for our domain global RBFNN trained with clustering algorithm is the best - the fastest to train and has the least RMSE on both training and testing data sets.

It is known that in case of noisy data the model, having the least error on training set, lacks generalisation ability. Regardless of the model complexity, finding "superoptimal" parameters (i.e. vanishing the training error) leads to overfitting. One possible way of dealing with overfitting is searching for the models which are suboptimal on the training set, but are still good on the test. It seems that global RBFNN trained with clustering algorithms (which are rather motivated from the point of view of local RBF usage) gives an example of "suboptimal" net, combining both fitting to the training data and generalisation on the test set. An independent qualification (variography) of the results obtained confirmed this conclusion.

Acknowledgements

The work was supported in part by INTAS grants 96-1957 and 97-31726.

References

Bishop C. and I. Nabney. Netlab. http://www.ncrg.aston.ac.uk/Netlab/, 1997.

Cortes C. and V. Vapnik. Support Vector Networks. Machine Learning, 20: 273–297, 1995.

Dempster A.P., N.M. Laird and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, B 39 (1), 1-38, 1977.

Farid U. Dowla and Leah L. Rogers (1995). Solving Problems in Environmental Engineering and Geosciences with Artificial Neural Networks. The MIT Press, Cambridge, Massachusetts, London, England.

V. Demyanov, N. Gilardi, M. Kanevski, M. Maignan, V. Polishchuk. Decision-oriented Environmental Mapping with Radial Basis Function Neural Networks. Workshop W07 "Intelligent techniques for Spatio-Temporal Data Analysis in Evironmental Applications" Chania, Greece, 1999, pp. 33-42.

Fritzke B. Fast learning with incremental RBF Networks. Neural processing Letters, Vol. 1, No. 1, 2-5, 1994.

M. Kanevski, R. Arutyunyan, L. Bolshov, V. Demyamov, S. Chernov, E. Savelieva, V. Timonin, M. Maignan, M.F. Maignan. Mapping of Radioactively Contaminated Territories with Geostatistics and Artificial Neural Nerworks. I. Linkov and W.R. Schell (eds.), Contaminated Forests, 243-256. 1999a, Kluwer Academic Publishers. Printed in the Netherlands.

Kanevski M., V. Demyanov, S. Chernov, E. Savelieva, A. Serov, V. Timonin, M. Maignan. Geostat Office for Environmental and Pollution Spatial Data Analysis. Mathematische Geologie, N3, April 1999b, 1-12 pp.

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics 43, 59-69. Reprinted in Anderson and Rosenfeld (1988).

Lloyd, S. P. (1982). Least Squares Quantization in PCM. IEEE Transactions on Information Theory 28 (2), 129-137.

Moller, M. (1993). A scaled congugate gradient algorithm for fast supervised learning. Neural Networks 6 (4), 525-533

Nabney I. Efficient Training of RBF Networks for Classification. NCRG Technical Report NCRG/99/002, 1999.

Pogio, T. and F. Girosi (1989). Regularization Networks and Best Approximation Property. Proceedings of IEEE.

Schölkopf B., Sung K., Burges C., Girosi F., Niyogi P., Poggio T., Vapnik V. 1996. Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. AI Memo No. 1599; CBCL Paper No. 142, Massachusetts Institute of Technology, Cambridge. IEEE Transactions on Signal Processing, 45(11):2758-2765, November 1997.

L. Wang, P.M. Wong, M. Kanevski, T.D. Gedeon Combining Neural Networks with Kriging for Stochastic Reservoir Modeling. IN SITU, 23(2), 151-169 (1999). Copyright by Marcel Dekker, Inc.

Wettschereck D. and Dietterich T. Improving the Performance of Radial Basis Function Networks by Learning Center Locations. Advances in Neural Information Processing Systems 4. Edited by J.E. Moody, S.J. Hanson and R.P. Lippmann, Morgan Kaufmann Publishers, San Mateo, CA, 1992

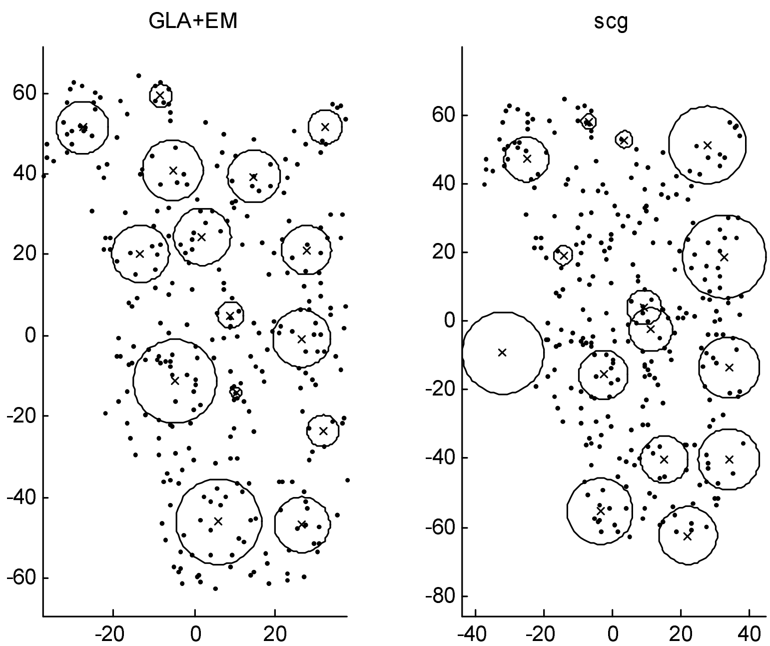

Figure 1. Centres and variances of 15 Gaussian RBFs found with GLA+EM (left) and scg (right) algorithms. Data plotted with dots

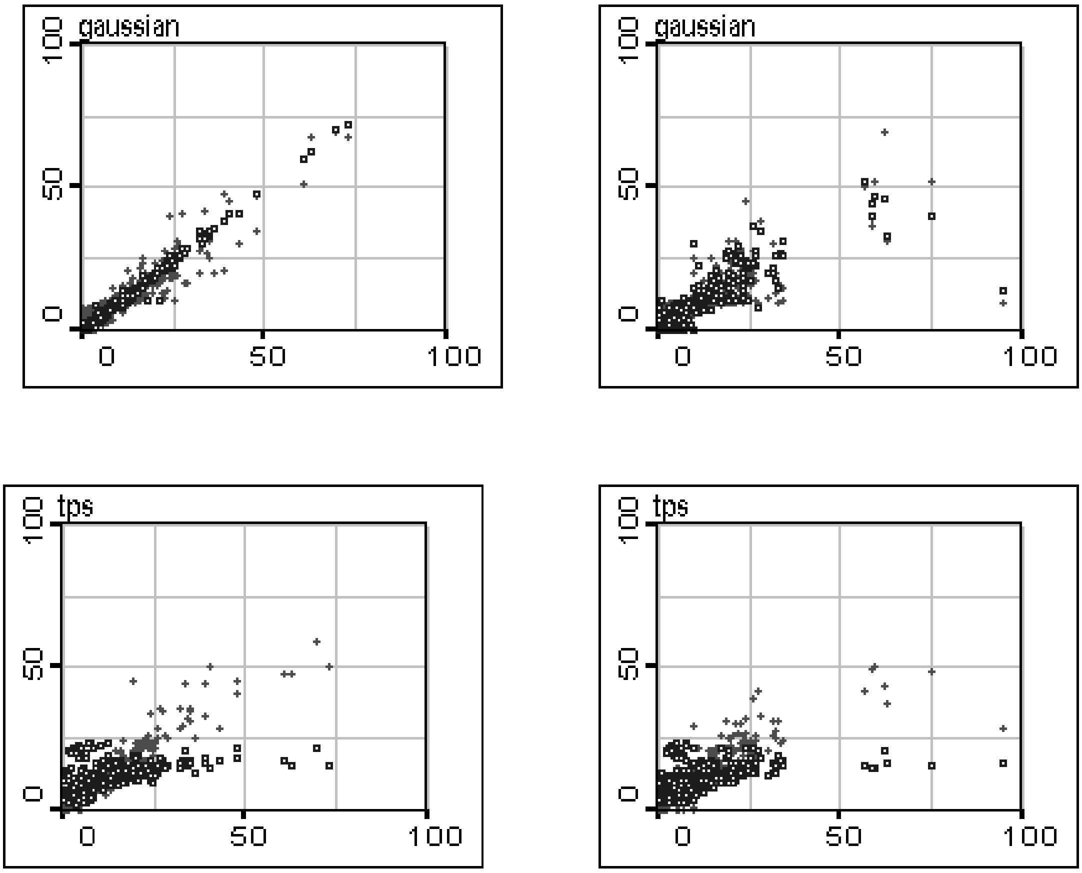

Figure 2. RBFNN outputs vs. measurements.. Left – training data, right – test. Dots – RBFNN trained with GLA+EM, crosses - scg. Above - 85 Gaussian RBFNN, below - 85 tps RBFNN

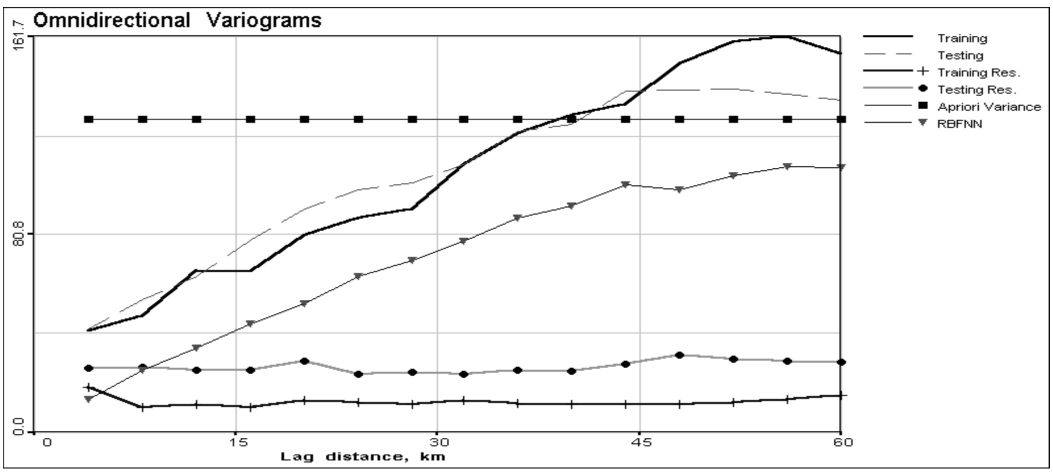

Figure 3. Omnidirectional variograms of training, testing, RBFNN output (training data), and residuals on training and testing data sets

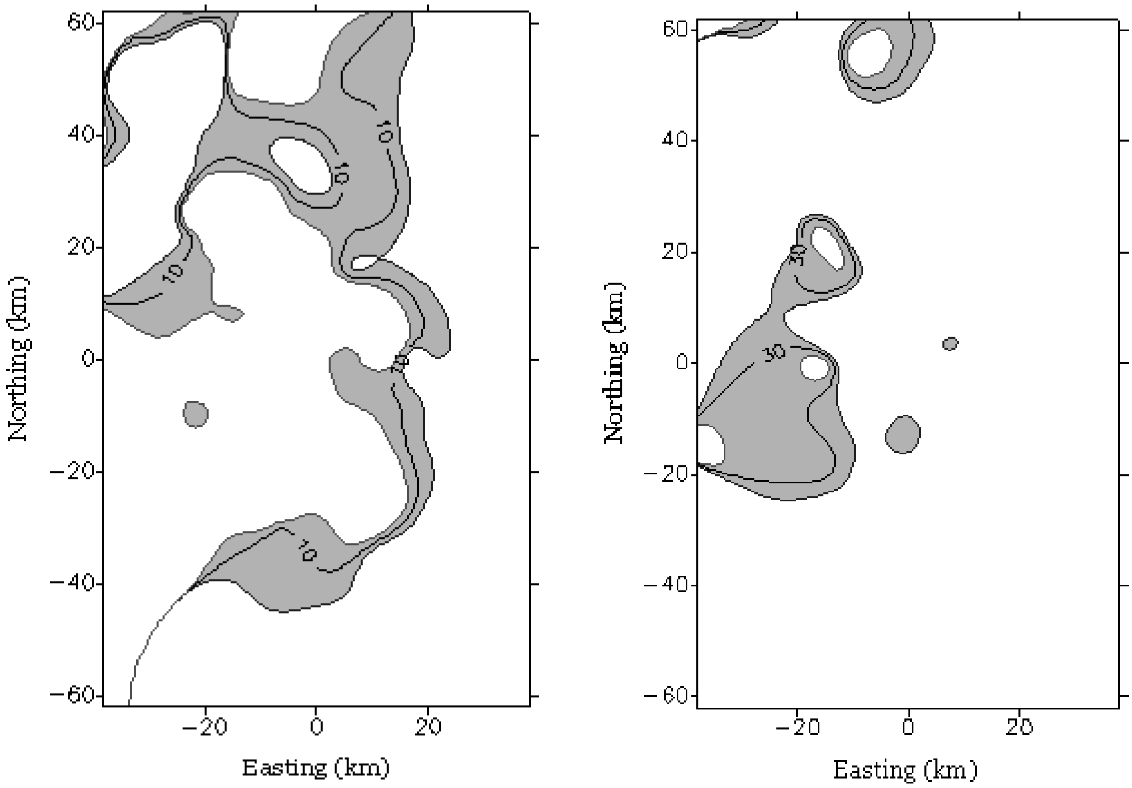

Figure 4. «Thick contour»: 10 Ci/sq.km (left) and 30 Ci/sq.km (right) isolines with uncertainty areas for CS137, Chernobyl fallout case study