SHADOW TARGETS: A NOVEL ALGORITHM FOR TOPOGRAPHIC PROJECTIONS BY RADIAL BASIS FUNCTIONS

Michael E. Tipping
David Lowe

Neural Computing Research Group, Aston University, Birmingham, UK

ABSTRACT

The archetypal artificial neural network topographic paradigm, Kohonen's self-organising map, has proven highly effective in many applications but nevertheless has significant disadvantages which can limit its utility. Alternative feed-forward neural network approaches, including a model called 'NEUROSCALE', have recently been developed based on explicit distance preservation criteria. Excellent generalisation properties have been observed for such models, and recent analysis indicates that such behaviour is relatively insensitive to model complexity. As such, it is important that the training of such networks is performed efficiently, as computation of error and gradients scales in the order of the square of the number of patterns to be mapped. We therefore detail a novel training algorithm for NEUROSCALE which outperforms present approaches and we illustrate the algorithm in practice.

INTRODUCTION

Topographic mappings are an important class of information processing techniques which offer considerable potential for the visualisation and exploratory analysis of data, as well as for generic feature extraction. In a topographic map, N data vectors $\{\mathbf{x}_i\}$ in $\mathbb{R}^p$ are transformed to a corresponding set of feature vectors $\{\mathbf{y}_i\}$ in $\mathbb{R}^q$, such that the topology, or *geometric structure*, of the feature vectors is representative of that of the data vectors. Generally, $q < p$ as dimension reduction is desired, with $q = 2$ in visualisation applications. The archetypal neural network topographic paradigm, Kohonen's self-organising map (SOM), is seen to offer less informative projections in many visualisation and exploratory analysis applications [6, 7, 11].

An alternative topographic paradigm, the *Sammon mapping* [1] [10], can offer better results. The mapping is generated by minimisation of an error, or 'STRESS', measure which explicitly embodies the topographic constraint:

$$E = \sum_{i}^{N} \sum_{j>i}^{N} (d^*_{ij} - d_{ij})^2, \text{ with} \tag{1}$$
$$d^*_{ij} = \| \mathbf{x}_i - \mathbf{x}_j \|,$$
$$d_{ij} = \| \mathbf{y}_i - \mathbf{y}_j \| .$$

The inter-point distances in the feature space are thus desired to match as closely as possible, the inter-point distances in the data space. Standard nonlinear optimisation techniques may be employed such that by differentiation of E with respect to each of the N feature vectors $\mathbf{y}_i$, those vectors may be simultaneously directly adjusted so as to minimise the STRESS measure (Sammon used a variant of gradient-descent in his original paper). A key disadvantage of the mapping is thus that it only acts as a look-up table; previously unseen data cannot be located in the feature space without an expensive re-optimisation procedure of the complete dataset. Because of this, there has been significant recent interest in development of distance-preserving mappings which incorporate a parameterised transformation from the data space to the feature space [5, 1, 8, 12, 6]. By defining $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i; \mathbf{w})$, where $\mathbf{w}$ is a parameter vector, E may be differentiated with respect to the parameters, and the feature vectors *indirectly* adjusted so as to minimise the STRESS. In the 'NEUROSCALE' model [6], the transformation $\mathbf{f}(\cdot)$ is effected by a radial basis function neural network, and a schematic of the model is illustrated in figure 1. To train the network, *pairs* of points are presented, and while there is no corresponding pair of explicit target vectors, there is a measure of *relative* error and this training approach has been intuitively termed *relative supervision* [5].

The major remaining limitation of the Sammon mapping, and which should be the only reason to use less suitable methods such as the SOM, is that its computational requirements grow in the order of the square of the number of data points to be mapped. Any neural network implementation will also suffer from this undesirable $O(N^2)$ scaling behaviour. Note that unlike in feed-forward applications, the parameters of the RBF occur nonlinearly and hence the usual pseudo-inverse approaches are no longer directly applicable. It is therefore important

[1] The Sammon mapping is also a very close relative of the statistic field of *multidimensional scaling*.

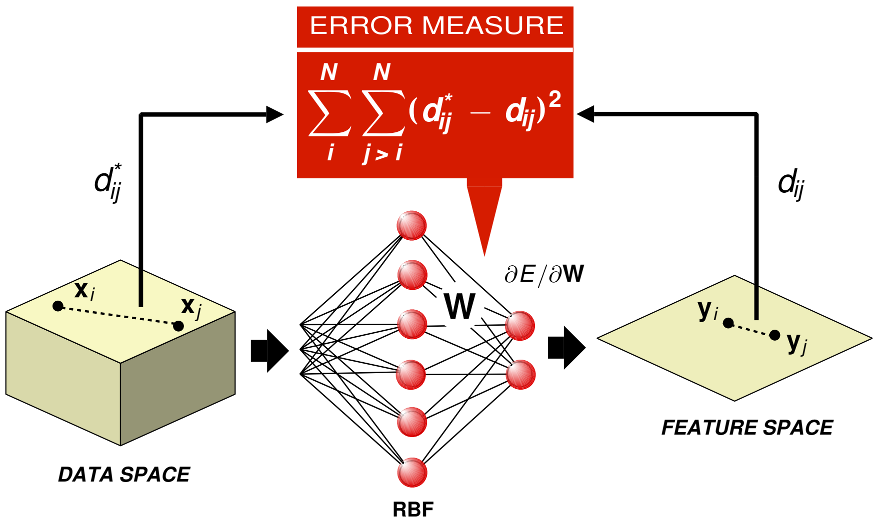

Figure 1: The NEUROSCALE architecture.

that the method chosen for optimising the network weights is as efficient as possible. Until recently the most appropriate methods of solution have employed quasi-Newton approaches.

This paper presents a new algorithm for optimisation of the weight parameters in NEUROSCALE and which, within the constraint of $O(N^2)$ scaling, outperforms present techniques both in speed of training and quality of final projection by exploiting the properties of the RBF. The trick we employ is reminiscent of the same problem which occurs in complex maximum likelihood optimisations which led to the introduction of the expectation-maximisation (EM) algorithm. The EM algorithm is an iterative approach to solving maximum likelihood functions. Rather than solve the complicated likelihood function directly, the EM algorithm instead splits up the likelihood problem into a sequence of solving a set of pseudo-complete problems where the parameter updates of these pseudo-steps can be obtained in closed form. Convergence results guarantee that the parameters obtained for these pseudo-problems converge to the parameters of the original problem. The EM algorithm thus does not need recourse to complicated gradient based nonlinear optimisation methods. The algorithm introduced in this paper is the equivalent in concept to the EM algorithm, but developed for relative supervision rather than maximising likelihood.

First, however, we will motivate the application of the NEUROSCALE model by reference to its generalisation properties.

GENERALISATION PROPERTIES

An important result concerning NEUROSCALE is that generalisation performance is relatively insensitive to the complexity of the RBF model, including basis function and smoothing parameter selection [7]. As an example, we consider examination of some satellite image data as used in the STATLOG study [9]. Each example in the data set represents a 3x3 region of a satellite land image, and for each of the nine pixels, there are four measurements of intensity taken at different wavelengths (approximately red, green and two in the near infrared). This gives a total of 36 variables for each data point, which has an additional label indicating the type of land represented by the central pixel. In figures 2 and 3 below, we visualise the structure of this data for four particular land types: red soil, grey soil, damp grey soil and very damp grey soil.

A NEUROSCALE model is trained (with $q = 2$) on 100 examples of each class and the resulting projection is shown in figure 2. In figure 3, 30 new data points of each soil type are projected by the previously trained network. Training STRESS is 0.001451 squared-distance-error per point-pair, and the generalisation STRESS is 0.001565 — only 7.9% higher than that for the training set. This quality of generalisation implies that if the latter data points were unlabelled, we might attempt to infer them from the test projection with relative confidence. Note that this method of data analysis does *not* use the class information to determine the clustering in projection space. Only relative separability in the input measurement space is employed in the discussions in this paper.

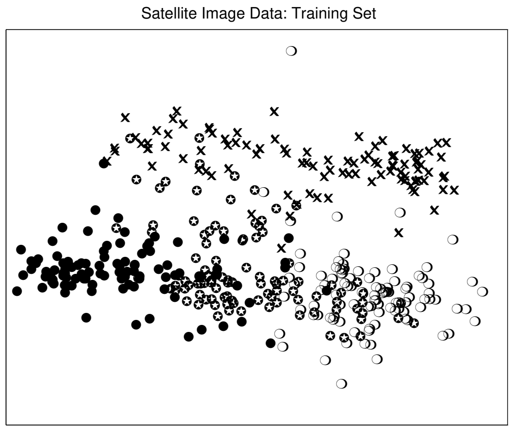

Figure 2: Training projection of the satellite image data set. The classes are ✕: red soil, ❍: grey soil, ✪ damp grey soil and ●: very damp grey soil.

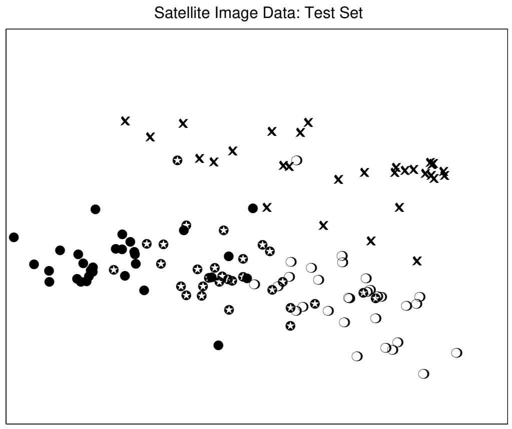

Figure 3: Generalisation projection of the satellite image data set.

OPTIMISING TOPOGRAPHIC TRANSFORMATIONS

There are a number of standard optimisation techniques for nonlinear problems which may be applied to minimisation of STRESS in topographic mapping contexts. Many of these are gradient-based techniques, although there have been alternatives proposed. The method of *iterative majorisation* was employed by [12], although it had been found to be "generally slow, and sometimes intolerably slow" in a previous study [2]. Statistical mechanics approaches have been exploited by [4] and [3]. The former applied the simulated annealing technique, although the authors found the approach to be largely impractical. The latter developed a scheme based on a mean-field approximation, but are yet to report on the efficacy of their algorithm.

One aspect of the above techniques is that they do not require calculation of the gradients $\partial E/\partial \mathbf{y}_i$. In cases where the gradient calculation is expensive (compared to calculation of error), such approaches are sensible. However, examination of the gradient equation (4), later, reveals that all the terms that must be computed are those also required for calculation of the STRESS measure. This observation holds also for a parameterised model linear in its weights, such as an RBF, but for an MLP, a back-propagation step would be required for the gradient calculation, which thus becomes considerably more expensive. Therefore for the Sammon mapping and linear parameterised models, computation of the gradient should be relatively inexpensive and so it should not be surprising if gradient-based techniques were more effective. A simulated annealing approach can also be appropriate when the cost surface in question is characterised by many local minima. Although this is generally the case for the Sammon mapping, it has been observed to be much less problematic for smooth transformational models such as NEUROSCALE [11].

Figure 4 below shows a comparison of training times for NEUROSCALE for three optimisation methods, from random initialisations. The plot shows the performance of the *scaled conjugate-gradients*[2] (SCG), on-line gradient descent with momentum (as used by [8]) and an algorithm named *shadow-targets*, which is described shortly, on a 400-point data set. Note that the on-line method required some considerable tuning of parameters for the purposes of the comparison.

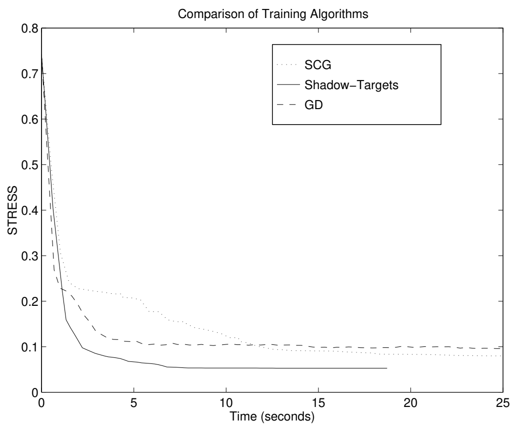

Figure 4: Relative performance of optimisation algorithms.

The shadow-targets algorithm offers the best performance, while the on-line gradient descent is initially better than SCG. In the on-line

[2] An improved variant of the conjugate-gradients algorithm

approach, pairs of patterns are selected from the data set at random, and the weights updated after each such pair, rather than in a 'batch' fashion as in SCG. This approach seems sensible for larger datasets, where the computational demands of the $N(N-1)/2$ loops are prohibitive. In the example above it can be seen to be very effective, but as would be expected in the stochastic case, performance deteriorates as the error approaches the minimum. This on-line technique was employed by [8], although it was applied with an MLP model. It is important to note that in such a case, an on-line approach becomes much less efficient, and the plot for an MLP is not so favourable as figure 4 where the model is an RBF. This is because the weight update for *every* pattern-pair would require an expensive back-propagation step. It should be clear that to adjust the weights in this manner for $N(N-1)/2$ pairs of data points would require $N(N-1)$ back-propagations. By contrast, for a batch method, $N(N-1)/2$ pairs of data points would require only N such computations and that approach thus becomes much more efficient.

THE SHADOW-TARGETS ALGORITHM

We now detail a new algorithm for the training of topographic models that are linear in their weights, such as radial basis function neural networks. An advantage gained by using such networks in supervised problems with a sum-of-squares error function — that they could be trained via a single pass pseudo-inverse approach — is lost in the topographic case where the STRESS measure introduces quartic terms. The following algorithm effectively decomposes the training problem into a two-step procedure, one step of which is linear and can be computed efficiently. This is reminiscent of the EM philosophy to likelihood maximisation.

The Algorithm

In the NEUROSCALE model, the output of the network for a set of N data points may be given as:

$$\mathbf{Y} = \mathbf{J}\mathbf{W}, \tag{2}$$

where $\mathbf{Y}$ is a matrix of row feature vectors $\mathbf{y}_i$, $\mathbf{W}$ is the weight matrix and $\mathbf{J}$ is the matrix:

$$\mathbf{J} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \dots & \phi_M(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \dots & \phi_M(\mathbf{x}_2) \\ \dots & \dots & \dots & \dots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \dots & \phi_M(\mathbf{x}_N) \end{bmatrix},$$

where $\phi(\cdot)$ are the M basis functions.

Then, referring to equation (1), the equations for calculating the weight derivatives are:

$$\frac{\partial E}{\partial w_{kr}} = \sum_i^N \frac{\partial E}{\partial \mathbf{y}_i} \cdot \frac{\partial \mathbf{y}_i}{\partial w_{kr}}, \quad \text{where} \tag{3}$$

$$\frac{\partial E}{\partial \mathbf{y}_i} = -2 \sum_{j \neq i} \left(\frac{d^*_{ij} - d_{ij}}{d_{ij}} \right) (\mathbf{y}_i - \mathbf{y}_j). \tag{4}$$

Equation (3) may be expressed in matrix form (for each separate output dimension r) as:

$$\boldsymbol{\nabla} E = \mathbf{J}^{\mathrm{T}} \mathbf{e}_r, \tag{5}$$

where

$$\boldsymbol{\nabla} E = \left(\frac{\partial E}{\partial w_{1r}}, \frac{\partial E}{\partial w_{2r}}, \dots, \frac{\partial E}{\partial w_{Mr}}\right)^{\mathrm{T}},$$

and

$$\mathbf{e}_r = \left(\frac{\partial E}{\partial \mathbf{y}_{1r}}, \frac{\partial E}{\partial \mathbf{y}_{2r}}, \dots, \frac{\partial E}{\partial \mathbf{y}_{Nr}} \right)^{\mathrm{T}}.$$

The set of linear equations implicit in equation (5) above, for the gradient of the STRESS measure with respect to the weights, is closely related to the set of *normal equations* of a linear least-squares problem. In such a *supervised* problem, the error E is given by $E = \frac{1}{2}\sum_i \| \mathbf{y}_i - \mathbf{t}_i \|^2$, where the vectors $\mathbf{t}_i$ are the explicit *targets*, such that

$$\frac{\partial E}{\partial \mathbf{y}_i} = (\mathbf{y}_i - \mathbf{t}_i). \tag{6}$$

In the relative supervision algorithm, there is an expression for $\partial E/\partial \mathbf{y}_i$ given by equation (4). This equation may be combined with equation (6) above to give a set of vectors that can be considered to represent *estimated targets* $\hat{\mathbf{t}}_i$:

$$\hat{\mathbf{t}}_i = \mathbf{y}_i - \frac{\partial E}{\partial \mathbf{y}_i}, \tag{7}$$

$$= \mathbf{y}_i + 2 \sum_{j \neq i} \left(\frac{d^*_{ij} - d_{ij}}{d_{ij}} \right) (\mathbf{y}_i - \mathbf{y}_j). \tag{8}$$

The vectors $\hat{\mathbf{t}}_i$ represent those *exact* targets for the network that would lead to an identical expression for the weight derivatives in the RBF, in the least-squares supervised case, as those obtained from the relative supervision approach. Of course, the problem cannot simply be solved in the one step as these targets $\hat{\mathbf{t}}_i$ are not fixed, but are dependent upon the current outputs of the network, $\mathbf{y}_i$, and thus the weights also.

For a fixed set of estimated targets $\hat{\mathbf{t}}_i$, the normal equations can be solved directly by

$$\mathbf{W} = \mathbf{J}^{+} \hat{\mathbf{T}}, \tag{9}$$

where $\hat{\mathbf{T}} = (\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2, \ldots, \hat{\mathbf{t}}_N)^{\mathrm{T}}$ and $\mathbf{J}^+$ denotes the *pseudo-inverse* of $\mathbf{J}$.

An approach, therefore, to minimising E would be to repetitively estimate a set of targets and then for each successive set, solve the above least-squares problem directly. However, in the early stages of training when STRESS is high, the targets given by equation (7) are poor and may lead to an increase in STRESS. A more effective approach is to introduce a parameter η, which is initially small, and estimate the targets as

$$\hat{\mathbf{t}}_i = \mathbf{y}_i - \eta \frac{\partial E}{\partial \mathbf{y}_i}, \tag{10}$$

and increase η as STRESS decreases during the training procedure.

The training algorithm as implemented in NEUROSCALE then becomes:

1. Initialise weights $\mathbf{W}$ to small random values.
2. Initialise η to some small value.
3. Calculate $\mathbf{J}^+$, where $\mathbf{J}_{ik} = \phi_k(\mathbf{x}_i)$.
4. Calculate estimated targets $\hat{\mathbf{t}}_i$ from equation (10).
5. Solve for weights using $\mathbf{W} = \mathbf{J}^+\hat{\mathbf{T}}$.
6. Calculate STRESS.
7. - If STRESS has increased, $\eta = \eta \times k_{down}$ where $0 < k_{down} < 1$ is a constant.
   - If STRESS has decreased, $\eta = \eta \times k_{up}$ where $k_{up} > 1$ is also a constant.
8. If not converged, return to Step 4.

Note that, for fixed basis functions, the potentially computationally expensive pseudo-inverse calculation need only be performed once and the $(M \times N)$ matrix $\mathbf{J}^+$ stored. Appropriate values for k_{down} and k_{up} are, for example, 0.1 and 1.2 respectively. These values appeared acceptably robust for various data sets.

Convergence Behaviour

It can be shown that this new scheme will both minimise STRESS and that its stationary points are identical to those of other gradient-based nonlinear optimisation methods.

At time step t let

$$\mathbf{J}\mathbf{W}^t = \mathbf{Y}^t, \quad \text{and} \tag{11}$$

$$\begin{aligned} \hat{\mathbf{T}}^t &= \mathbf{Y}^t - \eta \left.\frac{\partial E}{\partial \mathbf{Y}}\right|_{\mathbf{Y}=\mathbf{Y}^t}, \\ &= \mathbf{Y}^t - \eta \boldsymbol{\Delta}^t, \end{aligned} \tag{12}$$

$$\text{letting} \quad \boldsymbol{\Delta}^t = \left.\frac{\partial E}{\partial \mathbf{Y}}\right|_{\mathbf{Y}=\mathbf{Y}^t}$$

Then,

$$\begin{aligned} \mathbf{Y}^{t+1} &= \mathbf{J}\mathbf{W}^{t+1}, \qquad (13) \\ &= \mathbf{J}(\mathbf{J}^+\hat{\mathbf{T}}^t), \\ &= \mathbf{J}\mathbf{J}^+(\mathbf{Y}^t - \eta\boldsymbol{\Delta}^t), \\ &= \mathbf{Y}^t - \eta\mathbf{J}\mathbf{J}^+\boldsymbol{\Delta}^t, \qquad (14) \end{aligned}$$

since $\mathbf{J}\mathbf{J}^+\mathbf{Y}^t = \mathbf{Y}^t$ from equation (11). (Note that in general, $\mathbf{J}\mathbf{J}^+ \neq \mathbf{I}$.)

Now, $\boldsymbol{\Delta}^t$ is the direction of steepest descent, in $\mathbf{Y}$-space, on the STRESS surface. For the proposed scheme to minimise STRESS, it is required that the change in weights, and thus in $\mathbf{Y}$, implied by equation (14) represents a *descent direction* upon the cost surface. For this to be the case, $\mathbf{J}\mathbf{J}^+$ should be positive definite. Since by considering its singular value decomposition as $\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$, $(\mathbf{J}\mathbf{J}^+)$ is equal to $\mathbf{U}\mathbf{U}^{\mathrm{T}}$ and is thus non-negative definite. The shadow-targets algorithm will thus descend the surface of the STRESS measure, excepting when when $\boldsymbol{\Delta}^t = \mathbf{0}$ or the subspace of $\boldsymbol{\Delta}^t$ lies in the null-space of $(\mathbf{J}\mathbf{J}^+)$.

Following through the above analysis for gradient descent minimisation would give

$$\mathbf{Y}^{t+1} = \mathbf{Y}^t - \eta\mathbf{J}\mathbf{J}^{\mathrm{T}}\boldsymbol{\Delta}^t. \tag{15}$$

which has stationary points again when $\boldsymbol{\Delta}^t = \mathbf{0}$, and also when the subspace of $\boldsymbol{\Delta}^t$ lies in the null-space of $(\mathbf{J}\mathbf{J}^{\mathrm{T}})$, which is identical to the null-space of $\mathbf{J}\mathbf{J}^+$.

Interpretation

Firstly, consider equation (10). The estimated target $\hat{\mathbf{t}}_i$ is identical to the new point obtained after one step of a gradient-descent optimisation of the standard Sammon Mapping. In this sense, the proposed algorithm is effectively *shadowing*, step by step, the standard Sammon Mapping generation procedure. For this reason, the algorithm is referred to as the *shadow-targets* algorithm.

However, in contrast to the Sammon mapping, by reference to (14), the combination of reduced basis functions and smoothing parameter(s) in the RBF network serves to transform this standard optimisation step $\boldsymbol{\Delta}$ to an alternative step, $(\mathbf{J}\mathbf{J}^+)\boldsymbol{\Delta}$. The matrix $\mathbf{J}\mathbf{J}^+$ is a projection matrix that projects the columns of the error matrix $\boldsymbol{\Delta}$ onto the subspace spanned by the columns of $\mathbf{J}$. If there is a full complement of basis functions of sufficiently small σ (such that $\mathbf{J}$ is non-singular), then $\mathbf{J}\mathbf{J}^+ = \mathbf{I}$, and the algorithm effectively defaults to a repetitive fitting of an evolving Sammon map. The disadvantages of such an approach, in terms of robustness to local minima and the generalising properties of the final transformation, were made explicit in [11]. Typically, however, either there are fewer basis functions than patterns or, as discussed in [7], the width parameter of the basis functions is chosen to be sufficiently large such that $\mathbf{JJ}^+$ is singular.

CONCLUSIONS

Distance-preserving criteria, as incorporated in the Sammon mapping, are capable of extracting relevant feature space representations of data in many applications, and often more effectively than alternative topographic approaches. Previous work has shown how neural network implementations of such mappings can exhibit excellent generalisation properties which are considerably robust. The case for a more widespread application of such models is thus well founded.

The major disadvantage of the distance-preserving approach, that of $O(N^2)$ scaling, thus requires that the training algorithms should be maximally efficient. Current approaches, of employing conventional nonlinear optimisation algorithms, are relatively slow and have also been shown to introduce additional local minima into the training procedure [11]. The shadow-targets algorithm described in this paper both avoids the introduction of these minima and is an order of magnitude faster in converging. In the light of the established robust generalisation properties, the demonstrated efficiency of the shadow-targets algorithm indicates that the NEUROSCALE model offers considerable potential for topographic feature extraction.

Acknowledgements: This work was partially supported under EPSRC contract J75425.

REFERENCES

[1] T. F. Cox and G. Ferry. Discriminant analysis using non-metric multidimensional scaling. *Pattern Recognition*, 26(1):145–153, 1993.

[2] J. de Leeuw. Covergence of the majorisation method for multidimensional scaling. *Journal of Classification*, 5:163–180, 1988.

[3] T. Hofmann and J. Buhmann. Multidimensional scaling and data clustering. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 459–466. Cambridge, Mass: MIT Press, 1995.

[4] R. W. Klein and R. C. Dubes. Experiments in projection and clustering by simulated annealing. *Pattern Recognition*, 22(2):213–220, 1989.

[5] D. Lowe. Novel 'topographic' nonlinear feature extraction using radial basis functions for concentration coding in the 'artificial nose'. In *3rd IEE International Conference on Artificial Neural Networks*. London: IEE, 1993.

[6] D. Lowe and M. E. Tipping. Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications*, 4:83–95, 1996.

[7] D. Lowe and M. E. Tipping. Neuroscale: Novel topographic feature extraction with radial basis function networks. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, page to appear. Cambridge, Mass: MIT Press, 1997.

[8] J. Mao and A. K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6(2):296–317, 1995.

[9] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, 1994.

[10] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

[11] M. E. Tipping. *Topographic Mappings and Feed-Forward Neural Networks*. PhD thesis, Aston University, Aston Street, Birmingham B4 7ET, UK, 1996. Available from `http://www.ncrg.aston.ac.uk/`.

[12] A. R. Webb. Multidimensional scaling by iterative majorisation using radial basis functions. *Pattern Recognition*, 28(5):753–759, 1995.