Multidimensional Scaling Analysis using Neural Networks with Distance-Error Backpropagation

Lluís Garrido
Dept. ECM
Univ. de Barcelona
garrido@ecm.ub.es

Sergio Gómez
Dept. Eng. Informàtica
Univ. Rovira i Virgili
sgomez@etse.urv.es

Jaume Roca
Dept. ECM
Univ. de Barcelona
roca@ecm.ub.es

Abstract

We show that Neural Networks, trained with a suitable error function for backpropagation, can be used for Metric Multidimensional Scaling (*i.e.* dimensional reduction while trying to preserve the original distances between patterns) and are able to outdo other standard methods mainly because of the ability to model non-linear maps.

Keywords: Neural Networks, Multidimensional Scaling, Principal Component Analysis, Backpropagation.

1 Introduction

A standard problem in Multidimensional Scaling (MDS) (see [1], for example) consists in trying to map a collection of patterns represented as points in an n-dimensional space, $\boldsymbol{x}_a \in \mathbf{R}^n$ $(a = 1, \ldots, p)$, to a lower dimensional space,

$$\boldsymbol{x}_a \quad \mapsto \quad \boldsymbol{y}_a \in \mathbf{R}^m \quad (m < n), \tag{1}$$

in such a way that the distances between the projected points, $\boldsymbol{y}_a$, resemble as closely as possible the distances between the original ones. Typical values chosen for m are 2 or 3, since in this way the method can be used to visualize the most relevant features of the original n-dimensional configuration.

This can be set as a minimization problem once an energy function is given. Since we are concerned with the preservation of distances a natural choice is

$$E = \frac{1}{2} \sum_{a,b} \left(d_{ab}^{(n)} - d_{ab}^{(m)} \right)^2, \tag{2}$$

where $d_{ab}^{(n)}$ and $d_{ab}^{(m)}$ represent the Euclidean distance between patterns a and b in the original and projected spaces, respectively.

The optimal mapped configuration will be the one satisfying the set of $p \times m$ non-linear equations $\partial E / \partial \boldsymbol{y}_a = 0$. Solving them directly is generally out of reach and substitute approximate methods should be addressed. The standard one is just Principal Component Analysis (PCA). With this method one can determine, in the original space $\mathbf{R}^n$, the set of m principal directions (*i.e.* those directions along which the data have the highest variance). The projection onto them gives the mapped m-dimensional configuration, which makes actually the optimal solution among the restricted set of orthogonal projections. Other methods, such as Non-Linear PCA (NLPCA) [3, 4, 5], may be able to perform non-linear projections. However, NLPCA is really inappropriate for Metric MDS because it is only concerned with the (approximate) invertibility of the map but does not care at all of the distances between patterns in the projected space.

2 MDS with Neural Networks

Here we will give an alternative solution to this problem which involves the use of Neural Networks with a suitable error function for backpropagation. The main idea consists in building a net with n input units and a number of hidden layers, containing a *bottle-neck* layer with only m units and an output layer with n units. Backpropagation is invoked with an error function that contains, in addition to the quadratic error term between input and output, a new piece which is introduced to minimize the difference between the distances of pairs in the input and neck layers. Then, when enough iterations have been performed, the projected configuration is read

out from the neck layer.

In order to use the net in the most efficient way it is convenient to perform a translation and a global scaling of the initial data:

$$x_a \longrightarrow \xi_a^{\text{in}} = \lambda^{\text{in}}(x_a - \mathbf{a}), \tag{3}$$

so as to make $\xi_a^{\text{in}} \in [0,1]^n$. Then one can use ξ_a^{in} as the input to the net. The outcome of the neck layer, ξ_a^{nk}, lives in the region $[0,1]^m$ since we are using sigmoid activation functions. This implies that $0 \leq d_{ab}^{\text{nk}} \leq \sqrt{m}$ while $0 \leq d_{ab}^{\text{in}} \leq \sqrt{n}$ for any pair of input points $(\xi_a^{\text{in}}, \xi_b^{\text{in}})$, where d_{ab}^{nk} and d_{ab}^{in} stand for the distances between patterns a and b in the neck and initial layers, respectively.

The error function that we have considered in the backpropagation method is given by

$$E_{\text{BP}} = \alpha\, E_1 + (1-\alpha)\, E_2, \tag{4}$$

where

$$E_1 = \sum_a \left(\xi_a^{\text{out}} - \xi_a^{\text{in}}\right)^2 \tag{5}$$

and

$$E_2 = \sum_{a,b} \left(\frac{d_{ab}^{\text{in}}}{\sqrt{n}} - \frac{d_{ab}^{\text{nk}}}{\sqrt{m}}\right)^2, \tag{6}$$

and $\alpha \in [0,1]$ controls the relative contribution of each part. The term E_1 favors those maps for which the representation in the bottle-neck layer can be best accurately inverted to recover the original configuration. The second term, E_2, is the most important one since it forces this representation in the bottle-neck to inherit, as closely as possible, the metric structure of the original configuration. The different scalings for d_{ab}^{in} and d_{ab}^{nk} in this term are introduced in order to have both numbers in the same range. In this way we can guarantee that all possible configurations can still be covered with the use of sigmoids.

The various scalings performed in this process make the outcome of the neck layer not to be directly interpretable as the final m-dimensional configuration; we can undo all those scalings by setting

$$y_a = \lambda^{\text{out}} \xi_a^{\text{nk}}, \tag{7}$$

with $\lambda^{\text{out}} = \sqrt{n/m}\ \lambda^{\text{in}}$. However, a slightly better solution can be obtained by choosing instead

$$\lambda^{\text{out}} = \frac{\sum_{a,b} d_{ab}^{(n)} d_{ab}^{\text{nk}}}{\sum_{a,b} \left(d_{ab}^{\text{nk}}\right)^2}, \tag{8}$$

since this is the value of λ that minimizes the function $E(\lambda) = \frac{1}{2}\sum_{a,b}(d_{ab}^{(n)} - \lambda d_{ab}^{\text{nk}})^2$ for the given neck configuration, which is what we are ultimately trying to achieve with the whole procedure.

In the practical use of the neural network we have noticed that the best results are obtained by letting the parameter α fall to zero as the learning grows so that the error function E_{BP} reduces to E_2 after a certain number of iterations. Actually, a non-zero value of α is only useful in the early stages of the learning, in order to speed up convergence. In this situation, *i.e.* with $E_{\text{BP}} = E_2$, it is easy to prove analytically that the configuration minimizing E_{BP} differs from the one minimizing directly the original energy function E in eq. (2) only by a global scaling $\sqrt{n/m}$ of all coordinates. Thus, the (otherwise technically convenient) scalings that we have introduced above are completely harmless for the purpose of searching for the best mapped configuration.

It is well known that a network with just the input, output and neck layers, with linear activation functions and subject to self-supervised backpropagation is equivalent to PCA [2]. Our approach goes beyond PCA in two important instances. First, the presence of this new distance-error contribution, E_2, favors those configurations in the neck layer that approximate better the original distances; and second, the use of sigmoid activation functions and the addition of a number of hidden layers makes the neural net able to produce mappings which are more general (non-linear) than just the orthogonal projections of PCA. In fact, a comparative analysis of both approaches over several types of configurations shows that our method produces better results in the "tougher" situations, *i.e.* when some of the directions discarded by the PCA projection are still relatively important.

3 An illustrative example

As an application of both procedures we have considered a data set[1] consisting of different animal species, characterized by $n = 17$ attributes each (15 boolean + 2 numerical). The coordinates x_a and

[1] Original data extracted from the 'Zoo Database', created by Richard S. Forsyth (1990) (`ftp://ftp.ics.uci.edu:~/pub/machine-learning-databases/zoo`).

distances $d_{ab}^{(n)}$ have been obtained after scaling the numerical attributes to the range $[0,1]$ in order to assign an equal weight to all attributes[2] (implying that in this case we simply have $\xi_a^{\text{in}} = x_a$).

When using the neural net, the best scaling for the two-dimensional neck representation is given by $\lambda^{\text{out}} = 2.946$, which is in less than 1.1 % disagreement with the expected value of $\lambda^{\text{out}} = \sqrt{17/2}$.

The projected configurations obtained with each method are drawn in figure 1. Patterns are represented by their label. As shown in the plot, both approaches produce a fairly similar configuration. However, the computation of the overall relative error, *i.e.*

$$\varepsilon = \left(\frac{\sum_{a,b} \left(d_{ab}^{(n)} - d_{ab}^{(m)} \right)^2}{\sum_{a,b} \left(d_{ab}^{(n)} \right)^2} \right)^{\frac{1}{2}}, \qquad (9)$$

for each method shows that the neural network is giving out a slightly better result,

$$\varepsilon_{\text{PCA}} = 0.2728, \qquad \varepsilon_{\text{NN}} = 0.2346, \qquad (10)$$

which represents a 14.00 % improvement over PCA.

Acknowledgements

This work has been supported in part by a CICYT contract AEN95-0590 and by a URV project URV96-GNI-13.

References

[1] T.F. Cox and M.A.A. Cox, *Multidimensional Scaling*. Chapman & Hall, London 1994.

[2] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks* 2 (1989), 459.

[3] M.A. Kramer, Non-linear principal component analysis using autoassociative neural networks, *AICHE Journal* 37 (1991), 233.

[4] D. DeMers and G. Cottrell, Non-Linear Dimensionality Reduction, *NIPS* 5 (1994);

N. Kambhatla and T.K. Leen, Fast Non-Linear Dimension Reduction, *NIPS* 6 (1995).

[5] Ll. Garrido, V. Gaitán, M. Serra-Ricart and X. Calbet, Use of multilayer feedforward neural nets as a display method for multidimensional distributions, *Int. J. Neural Systems* 6 (1995), 273;

Ll. Garrido, S. Gómez, V. Gaitán and M. Serra-Ricart, A regularization term to avoid the saturation of the sigmoids in multilayer neural networks, *Int. J. Neural Systems* 7 (1996), 257.

[2] Although both methods are probably better suited when most of the attributes under consideration are numerical, rather than boolean, we believe that the large number of dimensions involved helps in making this issue less important.

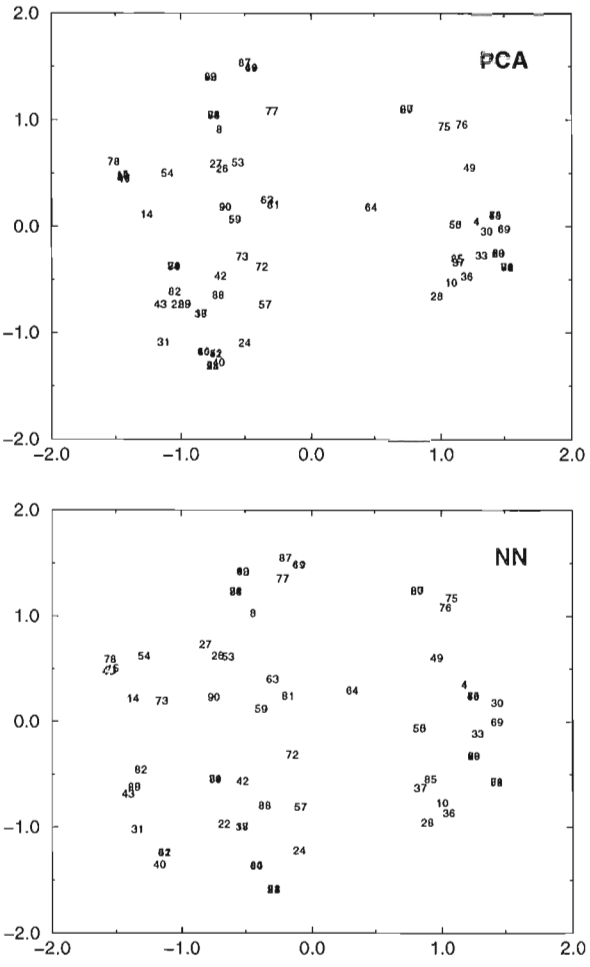

Figure 1: PCA vs. Neural Network projections