Mon. Not. R. Astron. Soc. **283**, 207–221 (1996)

Neural computation as a tool for galaxy classification: methods and examples

O. Lahav,[1] A. Naim,[1] L. Sodré, Jr[2] and M. C. Storrie-Lombardi[1]

[1] *Institute of Astronomy, Madingley Road, Cambridge CB3 0HA*
[2] *Instituto Astronômico e Geofísico da Universidade de São Paulo, CEP CP 9638, 01065-970, São Paulo, Brazil*

Accepted 1996 June 28. Received 1996 April 17; in original form 1995 July 3

ABSTRACT

We apply and compare various artificial neural network (ANN) and other algorithms for the automated morphological classification of galaxies. The ANNs are presented here mathematically, as non-linear extensions of conventional statistical methods in astronomy. The methods are illustrated using a selection of subsets from the ESO-LV catalogue, for which both machine parameters and human classifications are available. The main methods we explore are: (i) principal component analysis (PCA), which provides information on how independent and informative the input parameters are; (ii) encoder neural networks, which allow us to find both linear (PCA-like) and non-linear combinations of the input, illustrating an example of an unsupervised ANN; and (iii) supervised ANNs (using the back-propagation or quasi-Newton algorithm) based on a training set for which the human classification is known. Here the output for previously unclassified galaxies can be interpreted as either a continuous (analogue) output (for example T-type) or a Bayesian a posteriori probability for each class. Although the ESO-LV parameters are suboptimal, the success of the ANN in reproducing the human classification is 2 T-type units, similar to the degree of agreement between two human experts who classify the same galaxy images on plate material. We also examine the aspects of ANN configurations, reproducibility, scaling of input parameters and redshift information.

Key words: methods: data analysis – galaxies: general.

1 INTRODUCTION

The exponential growth of data in extragalactic astronomy calls for new approaches to analysis and interpretation. Observations with large ground-based telescopes, automatic measuring machines and satellites have produced large data bases of imaging and spectroscopy of galaxies. The advance in producing 'Gigabytes of data' has not, however, been matched by artificial intelligence techniques of classification and interpretation. In spite of several attempts (e.g. Murtagh & Heck 1987; Thonnat 1989; Lauberts & Valentijn 1989; Okamura, Kodaira & Watanabe 1984; Spiekermann 1992; Storrie-Lombardi et al. 1992; Doi, Fukugita & Okamura 1993; Abraham et al. 1994; Lahav et al. 1995; Naim et al. 1995b), the morphological classification of galaxies still remains a human-intensive process, dependent on the eyes of a handful of dedicated individuals.

The motivation for classifying galaxies is two-fold.

(i) RC3-like catalogues for millions of galaxies are needed for statistical studies (for example correlation functions or density–morphology relations) and as a target list of selected types for observational projects, such as Tully–Fisher measurements.

(ii) Classification is important for quantifying the astrophysics of galaxies, in analogy with the HR diagram for stars. It allows us to incorporate multiwavelength and dynamical properties of galaxies, with the hope that a new 'physical Hubble sequence' will emerge.

Automated procedures are the only practical way of classifying the enormous number of data produced by machine scans like those obtained in the Cambridge Automated Plate Measuring (APM) facility and the Sloan Digital Sky Survey (SDSS). The artificial neural network (ANN) method is a novel technique for classifying objects, which has not been used much in astronomy. In a pilot study (Storrie-Lombardi et al. 1992; hereafter SLSS) we investigated the ANN technique for classifying galaxies. Using a back-propagation algorithm, we showed that we could reproduce the ESO-LV classification (into five classes) at a success rate of 64 per cent 'perfect match'. More recently, we have shown (Naim et al. 1995b; Lahav et al. 1995) that ANNs can replicate the human classification of APM-selected galaxies to the same degree of agreement as that between two human experts, to 1.8 T-type units. While previous papers (Storrie-Lombardi et al. 1992; Naim et al. 1995; Lahav et al. 1995) focused on the applications of ANNs to galaxy classification, this paper provides the theoretical framework and mathematical details of the methods.

Other recent applications of ANNs in astronomy include adaptive optics (e.g. Angel et al. 1990), star/galaxy separation (e.g. Odewahn et al. 1991; Mahonen & Hakala 1995), meteor monitoring (Fraser, Khan & Levy 1992), and classification of stellar spectra (von Hippel et al. 1994) and galaxy spectra (Folkes, Lahav & Maddox 1996). For a review of astronomical applications, see also Serra-Ricart et al. (1993), Miller (1993) and Storrie-Lombardi & Lahav (1994, 1995). Non-astronomical applications similar to our problem are speech recognition and identification of hand-written characters. ANNs have several practical advantages over traditional techniques. ANN algorithms make no prior assumptions about the statistical distribution of test objects, and invoke no neuristics to define class membership. The algorithms are general-purpose and can be applied to a variety of problems.

Surprisingly, in spite of the wide application of CCD imaging and the theoretical interest in the Hubble sequence, there is no large uniform data set of galaxy images available. The largest available uniform samples include less than 1000 galaxies (e.g. Kent 1985; Simien & de Vaucouleurs 1986; Kodaira, Watanabe & Okamura 1986). The recent APM-selected sample of Naim et al. (1995a) includes 830 galaxy images. Here we use the ESO-LV data sets, although they are far from being optimal for our problem. They are based on plate material, the galaxies were not classified uniformly by one expert (but by A. Lauberts, E. A. Valentijn and H. G. Corwin over a decade), and the machine parameters do not optimally reflect structural parameters like spiral arms which are so apparent to the human eye. This is, however, a large data set (more than 5000 galaxies), which includes both machine parameters and human classification. The results presented here should be regarded as a *lower limit* to what can be done with the ANN approach to classification in the future, for example with uniform large samples of CCD images which are currently measured (e.g. Madore et al., in preparation; White et al., in preparation).

In this paper we shall also briefly address the astrophysical implications of our ANN results. One open question is whether galaxies were formed in a self-similar way, or in a way that mainly depends on their total mass or potential well. For example, Simien & de Vaucouleurs (1986) showed a tight correlation between the disk-to-bulge ratio (a distance-independent property) and the Hubble type, while Meisles & Ostriker (1984) argued that the absolute luminosity of the spheroidal component (a distance-dependent property) is the major parameter determining the Hubble sequence. We shall examine this question using ESO-LV diameters.

The structure of the paper is as follows. As the ANN methods are general and are currently scattered in the ANN literature (for example in journals of engineering and biology), we present them mathematically in Appendices A (principal component analysis and its non-linear extensions), B (the back-propagation and quasi-Newton minimization algorithms used in our applications) and C (Bayesian classification, Wiener filtering and weight decay). The main text of the paper gives examples of applications of these methods to the ESO-LV galaxies. Following a general Introduction (Section 1), Section 2 presents the ESO-LV parameters, Section 3 illustrates the use of principal component analysis for both data compression and unsupervised inference, while Section 4 presents a variety of applications of supervised non-linear ANNs. Future work is discussed in Section 5.

2 THE DATA SETS

Here we illustrate the method using ESO-LV galaxies (Lauberts & Valentijn 1989; hereafter LV89) at high Galactic latitudes ($|b| > 30°$). We shall consider several samples. There are three aspects to consider when defining the samples for training by the ANN: the sample selection (for example by apparent diameter), the galaxy machine parameters used, and the binning into galaxy classes.

The first sample, composed of 13 galaxy parameters, hereafter called P13, is the same one we used in SLSS, i.e. galaxies with visual diameters larger than 1 arcmin (the claimed completeness of the ESO-LV catalogue). Only galaxies with morphological classification performed by visual examination of the galaxy image were considered in our analysis. We use the 13 catalogue parameters shown in Table 1 of SLSS to describe each galaxy. Briefly, they are: (1) the average blue minus red colour; (2) the exponent in the generalized de Vaucouleurs law in the blue; (3) the log of the ratio of diameters that include 80 and 50 per cent of the blue light; (4) an indicator of the degree of asymmetry of the galaxy image; (5) the central blue surface brightness; (6) the log of the ratio of the minor to major diameters; (7) the error in the ellipse fit to blue isophotes; (8) the gradient of the blue surface-brightness profile at half-light radius; (9) the log of the ratio of the blue 26 mag arcsec^{-2} diameter to the half-light diameter; (10) the exponent in the generalized de Vaucouleurs law in the red; (11) the average blue surface brightness within a 10-arcsec diameter circular aperture; (12) the blue surface brightness at half-light radius; and (13) the red surface brightness at half-light radius. These 13 parameters were chosen because they are almost distance-independent, and they are very similar to those used by LV89 to perform the automated classification presented in the ESO-LV catalogue. This allows us to compare meaningfully the success rate of the classifications provided by our ANN with the classifications in ESO-LV. After selecting only galaxies with all 13 parameters available, our final data set had 5217 galaxies. We then randomly divided these galaxies into two independent sets of 1700 and 3517 objects for training and testing, respectively. We also normalized our input data between 0 and 1 by using the minimum and maximum values of each parameter (and have also tried normalizing by the variance). We grouped the ESO-LV catalogue subclasses in three ways: (i) by keeping the original range of classes $-5.0 \le T \le 10.0$, where T is the coded type; (ii) by binning into five major classes (as in SLSS), i.e. E ($-5.0 \le T < -2.5$, 466 galaxies), S0 ($-2.5 \le T < 0.5$, 851 galaxies), Sa + Sb ($0.5 \le T < 4.5$, 2403 galaxies), Sc + Sd ($4.5 \le T < 8.5$, 1132 galaxies), and Irr ($8.5 \le T \le 10.0$, 365 galaxies); and (iii) by splitting into two classes: early-type, (E + S0, $T < 0.5$, 1317 galaxies) and late-type ($T \ge 0.5$, 3900 galaxies).

The second sample, hereafter called D7, is also extracted from ESO-LV. It includes galaxies larger than 2 arcmin (as defined by the old ESO sample) which also have redshift information, and information on seven diameters. D_e, D_{70}, D_{80} and D_{90} are the major diameters of the ellipses at 50, 70, 80 and 90 per cent of the total B light, while D_{25}, D_{26} and D_{27} are the major diameters of the ellipses at B surface brightnesses of 25, 26 and 27 mag arcsec^{-2}. We then converted these into metric diameters using their redshifts. This sample includes 791 galaxies, which were mainly classified by one expert, H. Corwin.

3 HOW INFORMATIVE ARE THE INPUT PARAMETERS?

A key question when providing an ANN with an input is how many input parameters to present, and how to compress them in an efficient and informative way. There is of course a trade-off between keeping the number of parameters small and the amount of information presented. The principal component analysis (PCA) described below is useful for two purposes: (i) unveiling correlations between parameters and reducing the dimension of the input parameter space, hence acting as an 'unsupervised' method, where the data points are allowed to organize themselves without pre-defined classes, and (ii) compressing the data into a small number of new parameters, which can then be fed into 'supervised' ANNs (discussed in Section 4).

3.1 Standard PCA

PCA is a widely used method which allows us to judge how many independent parameters are needed, by looking at directions along which the variance is maximal. The formulation of PCA is given in Appendix A1. It is worth emphasizing that PCA is only meaningful for linear parameters (or 'the nearest to linear', for example by taking the log of the original variables), and may suffer from scaling problems. Nevertheless, it is a useful tool for reducing the dimensionality of the input parameter space. In the context of this paper it can be viewed as a data compression technique for the input to the ANN, as well as an 'unsupervised method' for exploring the parameter space.

We begin by applying the method to the log of the seven metric diameters given in the D7 sample (of 791 galaxies), each scaled to have zero-mean. We do not normalize here by the variance of each variable, as they all have the same metric, and we wish to represent their relative values. Not too surprisingly, the correlation matrix indicates strong correlation between the log-diameters. Table 1 gives the eigenvalues (ordered by values) and the corresponding eigenvectors for the log-diameters. The eigenvectors, which give more insight than the original covariance matrix, tell us how to build linear combinations from the raw parameters. 95 per cent of the variance is in the first principal component (which is found to be approximately the average of the seven log-diameters). As we show in Section 4.4, however, when using the ANN it is *not* sufficient to use just the first principal component to represent the 7D data space for classification.

We then applied PCA to the 13 distance-independent parameters of the P13 sample of 5217 galaxies, with the

Table 1. The eigenvalues and eigenvectors from the PCA (zero-mean) of seven log-diameters using 791 ESO-LV galaxies.

Eigenvalue 1 :	0.54	95.1 %					
Eigenvector 1 :	0.37	0.39	0.40	0.41	0.35	0.36	0.37
Eigenvalue 2:	0.01	2.5 %					
Eigenvector 2:	0.20	0.28	0.33	0.37	-0.62	-0.43	-0.25
Eigenvalue 3:	0.01	2.1 %					
Eigenvector 3:	-0.73	-0.20	0.12	0.49	-0.17	0.07	0.37
Eigenvalue 4:	0.00	0.2 %					
Eigenvector 4:	0.30	-0.06	-0.18	-0.28	-0.47	-0.04	0.76
Eigenvalue 5:	0.00	0.1 %					
Eigenvector 5:	-0.41	0.55	0.39	-0.57	-0.16	0.19	0.00
Eigenvalue 6:	0.00	0.0 %					
Eigenvector 6:	0.13	-0.18	-0.08	0.08	-0.46	0.80	-0.29
Eigenvalue 7:	0.00	0.0 %					
Eigenvector 7:	0.13	-0.63	0.73	-0.23	0.04	-0.05	0.01

parameters normalized to have zero-mean and unit-variance, as here the parameters are made of 'apples and oranges'. Indeed, one should be cautious about applying PCA to a set of parameters that are of different characters and may well be non-linear. The results do, however, give some insight into the amount of useful information in this parameter space. Tables 2(a) and 2(b) give the 13 eigenvalues and the eigenvectors corresponding to the largest three eigenvectors. We find that the first seven linear combinations give 90 per cent of the variance.

The projection of the 13 parameters onto the first and second principal components is shown in Figs 1(a) and (b). Although the distribution of all galaxies looks like a fuzzy cloud, the different morphological types occupy distinct regions in this new parameter space. We can even see a slight separation between E and S0 galaxies, although it is difficult to tell just from the plot if a galaxy is an E or S0.

Table 2(a). The 13 sorted eigenvalues (and their fractional contribution) from the PCA (zero-mean and unit-variance) of 13 parameters using 5217 ESO = LV galaxies.

1	5.43	41.7 %
2	2.34	18.0 %
3	1.21	9.3 %
4	1.14	8.8 %
5	0.91	7.0 %
6	0.69	5.3 %
7	0.45	3.4 %
8	0.33	2.5 %
9	0.19	1.5 %
10	0.16	1.2 %
11	0.08	0.6 %
12	0.07	0.5 %
13	0.01	0.0 %

This plot illustrates how PCA could compress a 13D parameter space into a 2D space. Although the physical meaning of the new space is not easy to interpret, it allows us to segregate different classes of objects.

3.2 Encoder and neural PCA

Generally, a multilayer ANN consists of nodes (analogous to human neurons) arranged in a series of layers. The nodes in a given layer are connected to the nodes in the next layer. The strength of the connection between node i in one layer and node j in the next layer is called the weight, w_{ij}. The weights are the free parameters of the ANN and they are determined by least-squares of the difference between the input and the desired output, the so-called cost function:

$$E=\frac{1}{2}\left\langle\sum_{k} (o_k-d_k)^2\right\rangle, \qquad (1)$$

where the sum is over the components of the vector $(k=1, M)$ and the average is over the galaxies. Layers between the input and the output layers are called 'hidden layers' and allow non-linear mapping. The least-squares minimization can be done by a variety of algorithms, for example back-propagation and quasi-Newton, which are described in detail in Appendix B. We use both algorithms in this paper. They differ in methods and in the speed of convergence, but the results are affected very little by the choice of algorithm. In non-linear minimization there is no guarantee of reaching the global minimum. For this reason it is recommended that the ANN be run several times with different initial random weights. The values of the weights are of secondary importance, since the sought minimum of equation (1) can be arrived at by more than one combination of weights. Moreover, the weights of a non-linear ANN are usually not easy to interpret. In this paper we prefer to judge the performance of the network by showing the network's output rather than by considering the individual weights. To avoid a situation in which the network is trapped in a local minimum with unreasonably large weights, it is sometimes worth regularizing the minimization. This procedure (called weight decay) is described in Appendix C3.

We begin by demonstrating an encoder network in which

Table 2(b). The three largest eigenvalues and corresponding eigenvectors from the PCA (zero-mean and unit-variance) of 13 parameters using 527 ESO-LV galaxies.

```
Eigenvalue  1 : 5.43    41.7 %
Eigenvector 1 :
            0.26  -0.26   0.33  -0.21  -0.40   0.09   0.17
            0.16   0.38  -0.26  -0.36  -0.24  -0.30

Eigenvalue  2 : 2.34    18.0 %
Eigenvector 2 :
           -0.03  -0.34   0.21  -0.28  -0.07   0.35   0.13
            0.02  -0.21  -0.33   0.13   0.50   0.44

Eigenvalue  3 : 1.21     9.3 %
Eigenvector 3 :
           -0.19   0.42   0.12  -0.42   0.13   0.56   0.07
            0.27   0.07   0.41  -0.12  -0.07   0.00
```

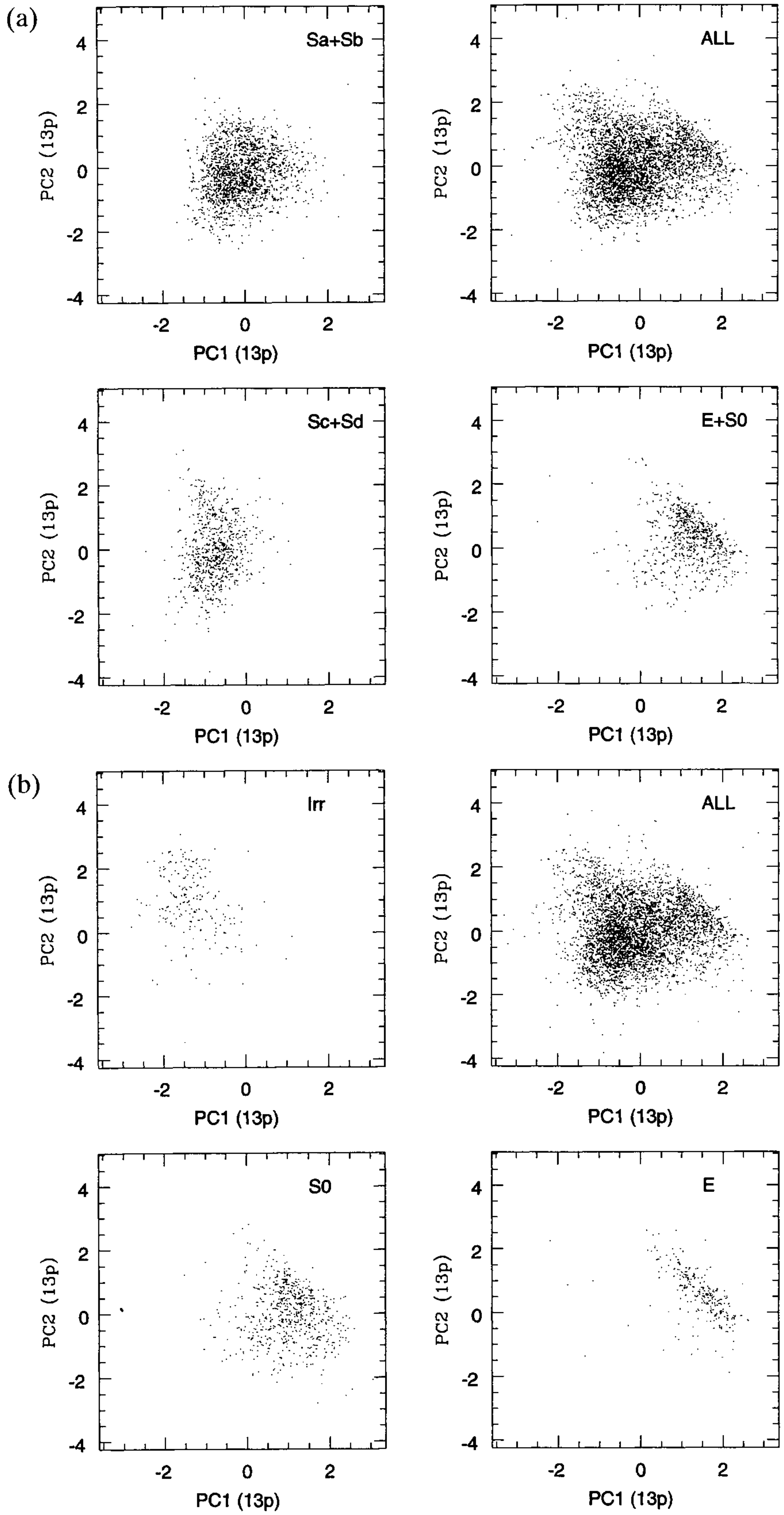

Figure 1. (a) The distribution of 5217 ESO-LV galaxies of all morphological types (top right) in the two dimensions defined by the first and second principal components as derived from PCA using 13 galaxy parameters (P13 sample). The other three panels show subsets of this fuzzy cloud according to their classification labels Sa + Sb, Sc + Sd and E + S0 as given in ESO-LV. The different morphological types occupy distinct regions in this new parameter space. (b) The top-right panel is as in (a), using the P13 sample, and the other three panels are for the clases E, S0 and Irr. Note that E and S0 galaxies are segregated.

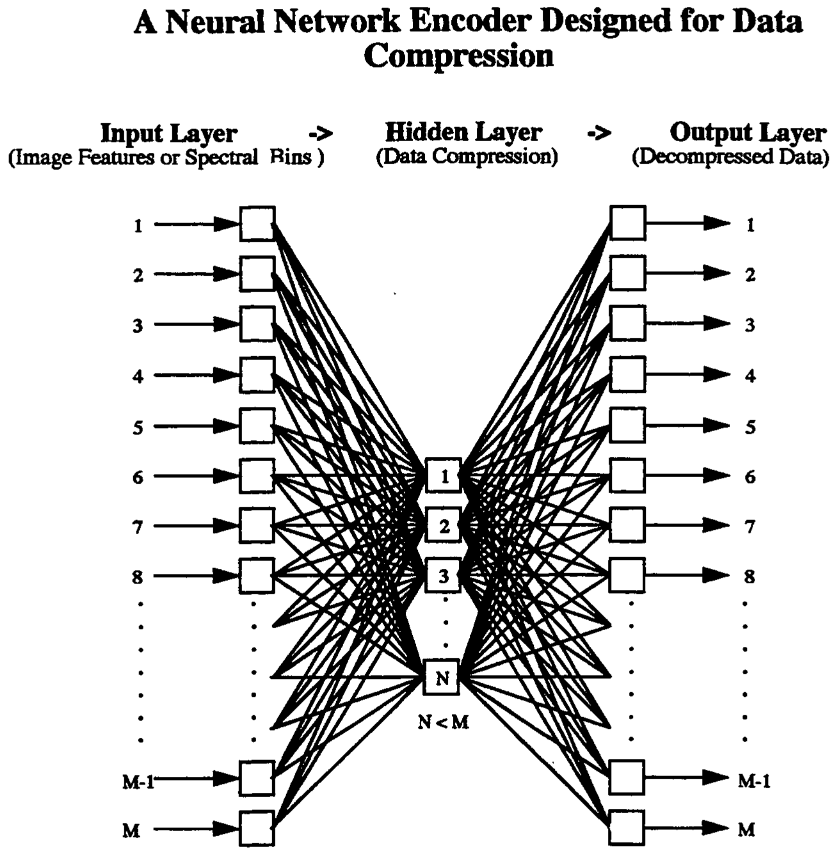
A Neural Network Encoder Designed for Data Compression
Input Layer (Image Features or Spectral Bins)
->
Hidden Layer (Data Compression)
->
Output Layer (Decompressed Data)
1
2
3
4
5
6
7
8
M-1
M
1
2
3
N
N<M

Figure 2. A schematic diagram of an encoder network with M input parameters, N (denoted by M' in the text) nodes in the hidden layer, and M output nodes. N will range between 1 and M, depending on the desired data compression factor. During training, set Input = Output to teach the encoder to reproduce a given input vector at the output layer. This network performs PCA-like dimension reduction when the transfer function is linear, and can be extended to perform non-linear mapping.

the desired output is the input itself, as explained in detail in Appendix A2. Fig. 2 shows an M:M':M network configuration, where M' is the number of 'neck' units (in the 'hidden layer'), or number of linear combinations in the PCA language (M' is denoted by N in Fig. 2). While a linear network in this configuration simply reproduces a standard PCA, a non-linear transfer function, such as $f(z) = \tanh(z)$ or $1/[1 + \exp(-z)]$ (sigmoid) can allow 'non-linear PCA'. Appendix B gives further details on the transfer functions and their role.

We now apply a non-linear encoder network, with a sigmoid threshold function. In Fig. 3 we plot the cost function (calculated over the 5217 ESO-LV galaxies) versus the number of hidden units M'. Clearly, when the input 13 parameters are uncorrelated we will need 13 hidden units to recover fully the 13 parameters at the output layer. If, on the other hand, the 13 parameters are identical, then one hidden unit will be sufficient. The figure shows that the cost function drops exponentially as a function of the number of hidden units. this behaviour may serve as a guide for selecting the number of hidden units for the classification network (see below). Serra-Ricart et al. (1993) have developed this unsupervised approach further, illustrating for our P13 data set that a non-linear encoder can identify classes in this data set much better than a standard PCA. Other algorithms for neural PCA, such as Oja's rule, are discussed in Appendix A3.

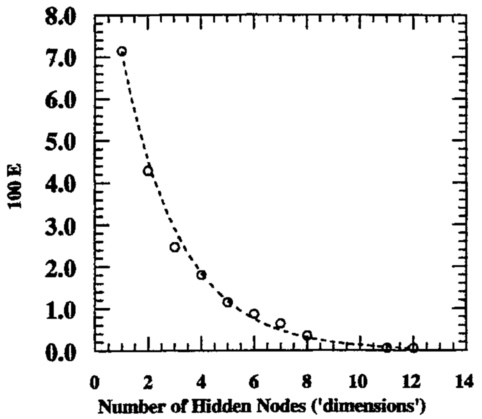

Figure 3. The cost function versus the number of hidden units in an encoder network with a sigmoid transfer function. The network was trained on the 5217 ESO-LV galaxies, each with 13 parameters. The cost function seems to drop roughly exponentially with the number of hidden nodes.

4 SUPERVISED CLASSIFICATION OF GALAXIES WITH ANNs

We now apply 'standard' supervised ANNs for classifying the ESO-LV galaxies. In the 'training' process, the input vectors, containing the galaxy parameters, are presented to the network. The weights ('free parameters') w_{ij} are computed by least-squares minimization with the back-propagation or the quasi-Newton algorithm (explained in detail in Appendix B). The ANN is then ready to handle new unclassified data for which only the machine parameters are available. We shall present three different network configurations: (i) a single output ('analogue') network; (ii) a two-class classifier; and (iii) a five-class classifier. We wish to emphasize that *supervised* ANNs do not produce an 'objective' unique classification. Supervised networks replicate the choices of their trainer – a network trained according to the classification made by Hubble or de Vaucouleurs will classify new data in a manner similar to the original expert.

4.1 Single continuous output

Although galaxy morphology is probably a continuous sequence (Hubble 1936), human experts provide us with a 'true answer', usually given in quantified units, to a first decimal point, for example $T=5.3$. It is to our benefit that the single-output configuration of the network can approximate a 1D continuous sequence.

It is common in astronomy to fit a model with several free parameters to the observations. This regression is usually done by means of χ^2 minimization. A simple example of a model is a polynomial with the free parameters as the coefficients. Consider now the specific problem of morphological classification of galaxies. If the type is T (for example using the numerical system $[-5, 10]$), and we have a set of parameters $\boldsymbol{x}$ (for example isophotal diameters and colours), then we would like to find the free parameters $\boldsymbol{w}$ ('weights') such that the cost function,

$$E=\frac{1}{2}\sum_i [T_i-f(\boldsymbol{w},\boldsymbol{x}_i)]^2, \qquad (2)$$

is minimized. The function $f(\boldsymbol{w},\boldsymbol{x})$ is the 'network'. Usually, f is written in terms of the variable

$$z=\sum_k w_k x_k, \qquad (3)$$

where the sum here is over the input parameters to each node. A 'linear network' has $f(z)=z$, while a non-linear threshold function could be a sigmoid $f(z)=1/[1+\exp(-z)]$ or $f(z)=\tanh(z)$. Another element of non-linearity is provided by the 'hidden layers'. The hidden layers allow curved boundaries around clouds of data points in the parameter space. A typical configuration with one hidden layer and a single output is shown in Fig. 4. While in most computational problems we only have 10–1000 nodes, in the brain there are $\sim 10^{10}$ neurons, each with $\sim 10^4$ connections. We do not, of course, regard our simple ANN algorithm as a model for the human brain, but rather as a non-linear statistical method.

The determination of many free parameters, the weights w_i in our case, might be unstable. It is therefore convenient to regularize the weights, for example by preventing them from growing too much. In the ANN literature this is called 'weight decay'. This approach is analogous to the method of maximum entropy (Gull 1989), and can be justified by Bayesian arguments, with the regularizing function acting as the prior in the weight space. One possibility is to add a quadratic prior to the cost function and to minimize

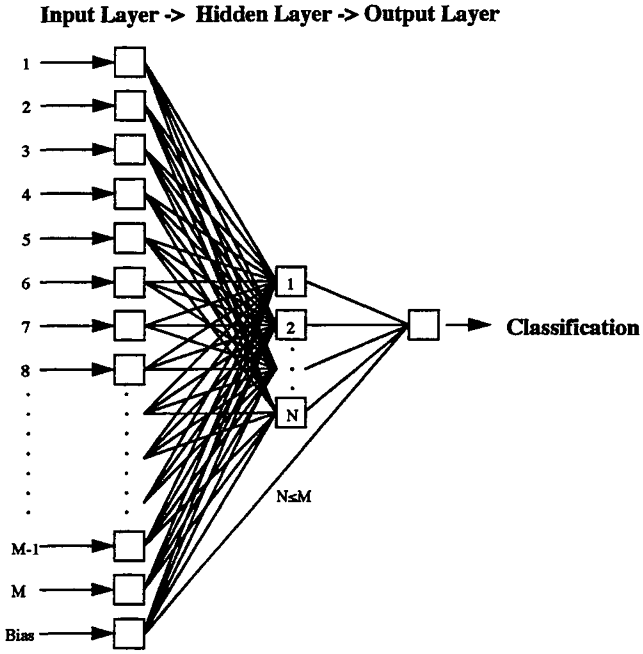

Figure 4. An ANN configuration with M input parameters, N hidden nodes and a single 'analogue' output. Such a network can perform a non-linear regression, and is used in our problem to predict the T-type, based on input galaxy parameters. All nodes in a given layer are connected to all nodes in the next layer. The 'bias' node allows additive constants in the network equation.

$$E_{\text{tot}}=\alpha E_{\text{w}}+\beta E_{\text{D}}, \qquad (4)$$

where E_{D} is our usual cost function, based on the data (for example equation 2), and

$$E_{\text{w}}=\frac{1}{2}\sum_{i=1}^{Q} w_i^2 \qquad (5)$$

is the chosen regularizing function, where Q is the total number of weights. The coefficients α and β can be viewed as Lagrange multipliers. While sometimes they are specified ad hoc, it is possible to evaluate them 'objectively' by Bayesian arguments in the weight-space. We discuss this procedure in detail in Appendix C3.

To illustrate the above ideas, we built a network with configuration 13:13:1, resulting in 46 free weights (including the 'bias' node, which represents an additive constant). In the training process the network was presented with 13 parameters (from sample P13) for each galaxy, using a subset of 1700 galaxies. Both the input parameters and the 'true' answer T-type (in the range $-5 \le T \le 10$) were scaled to the range [0, 1], so all the weights were treated on an equal footing in the regularization process. The transfer function used was a sigmoid. By the procedure outlined in Appendix C3 we found the weight decay regularization coefficient to be $\alpha/\beta=0.001$. We then applied least-squares minimization using a quasi-Newton method (as implemented in a code kindly provided to us by B. D. Ripley).

As in other optimization problems, it is crucial to decide when to stop the minimization. One approach is to stop when the cost function drops below a certain value, or changes little between successive iterations. Particularly when the sample size is small (relative to the number of weights), however, this may result in an 'over-fitting' ('memorizing') of the data (including the noise). Usually, the cost function with respect to the training set shows monotonic decline, and it is difficult to define a minimum for stopping. Instead, we calculate, at each iteration, the cost function for the testing set (with the weights derived of course from the training set). In this way we monitor the ability of the ANN to 'generalize' its choice of weights to data it was not trained on. Usually the cost function with respect to the testing set decreases to a minimum and then increases, so it is easy to decide where to stop according to this minimum.

Once the training phase was completed, we presented the network with a testing set (of 1700 galaxies in this case), but for which a human classification was known. On a Sun Sparc workstation, the training of the network on 1700 galaxies takes about 1 min (CPU), while testing on a sample of similar size takes only 1 s (CPU).

Fig. 5 shows the network type $T_{\rm net}$ versus the ESO-LV human classification $T_{\rm eso}$. The Spearman rank-order correlation coefficient is $r_s = 0.83$. As another way of quantifying the network performance we calculate the variance between the network and the ESO-LV types over the number of galaxies $N_{\rm gal}$,

$$\sigma^2 = \frac{1}{N_{\rm gal}} \sum (T_{\rm net} - T_{\rm eso})^2, \qquad (6)$$

and we find $\sigma \approx 2.0$ T-type units. We note in Fig. 5 that there are a few extremely deviant points for some cases (in particular at $T_{\rm eso} = -5$ and $T_{\rm net} = +7$ to 8). There is also a slight curvature at the plot for the extreme types, i.e. too few galaxies are classified as $T_{\rm net} = -5$ or $T_{\rm net} = 10$. This is probably due to the small number of galaxies with these extreme types and the difficulty the network has in to fitting 'abnormal' galaxies. By a similar statistic we can compare the runs of similar network configurations that start the minimization with different random weights. Fig. 6 shows the results of the two runs. The scatter between the two runs is much smaller than that in Fig. 5. Here, the Spearman coefficient is $r_s = 0.98$, and the typical 'reproducibility' scatter is $\sigma \approx 0.6$ T-type units. In Fig. 6 we note a 'break' in the transition from early-type ($T < 0$) to late-type ($T > 0$). It may be that the non-linearity of the network was not sufficient to fit both classes by the same weights (i.e. in each minimization the network finds a different compromise of weights to satisfy both early- and late-type galaxies), or that the quality of parameters for early- and late-types is different. There is possibly another 'break' at $T = 5$.

In the study of the blue APM images (Naim et al. 1995b; Lahav et al. 1995) we have shown that ANNs can replicate the classification by a human expert to the same degree of agreement as that between two human experts, to within 1.8 T-type units (based on a comparative study where the same images were classified by six experts independently). The ESO-LV data give a slightly weaker result, 2 T-type units, probably due to the parameters' being less informative, although they include blue minus red colour as a parameter, which is lacking in the APM sample.

4.2 Two-class (E, S) classifier

In a network with multiple outputs, the output vector can be interpreted in a probabilistic way. The jth component of this vector can be viewed as the probability for class j given the

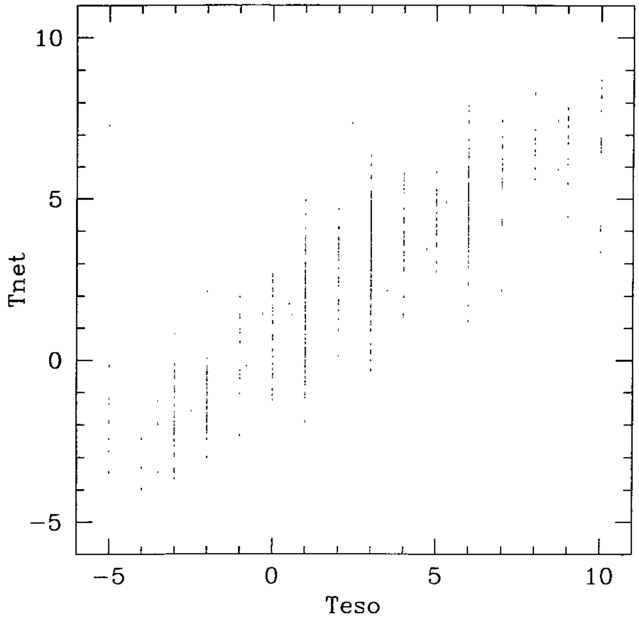

Figure 5. The type $T_{\rm net}$ predicted by the ANN for 1700 ESO-LV galaxies (based on a different set of 1700 galaxies) against the ESO-LV human classification $T_{\rm eso}$. The Spearman correlation coefficient in this diagram is 0.83, and the average rms dispersion is 2.0 T-type units.

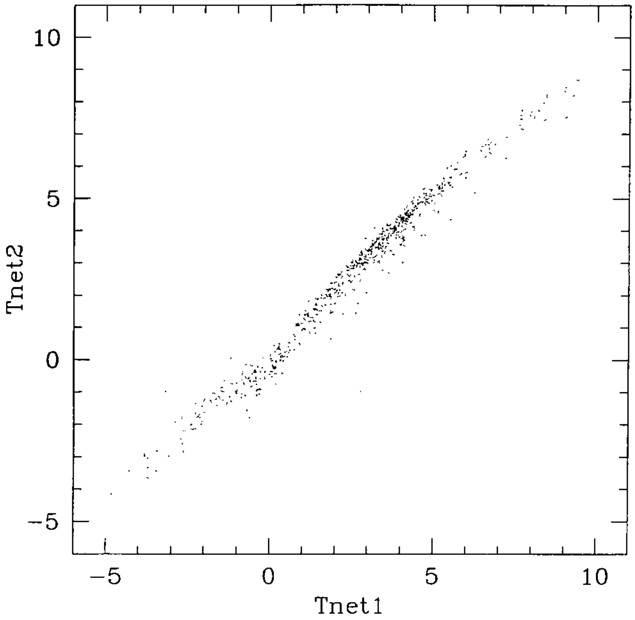

Figure 6. ANN reproducibility: A comparison between the predicted T-type of the network for 1700 ESO-LV galaxies from two runs, starting the minimization with different random weights. The Spearman correlation coefficient is 0.98 and the rms dispersion is 0.6 T-type units. Note the transition between early and late types, discussed in the main text.

input parameters, $P(C_j|x)$. In fact, it can be proved theoretically (Appendix C1) that the output of an ideal ANN is indeed a Bayesian a posteriori probability. Moreover, as our experiments confirm, the sum of the output vector components is $\sum_k o_k \approx 1$, as expected for a probabilistic classifier. It is worth noting that, unlike discrete classification of handwritten characters, galaxies form a continuous sequence. Hence the combination of probabilities assigned to different 'eigenclasses' may reflect an intermediate class. The 'most likely class' can be defined as the class associated with the largest output component. Here we do not include weight decay, as when it is included the interpretation of the output is no longer strictly Bayesian (see Appendix C1).

To classify into early-type ($-5 \leq T \leq 0.5$) and late-type ($0.5 < T \leq 10$) galaxies we have used a back-propagation algorithm (Appendix B), and a network configuration of 13:10:2 with a tanh threshold function, a learning coefficient of $\eta = 0.01$, and a momentum coefficient of $\alpha = 0.9$ (see equation B6). The network was trained on 1700 ESO-LV galaxies, and was tested on the remaining 3517 galaxies. Of the 898 galaxies classified as early-type by ESO-LV, 681 were classified as such by the network, while 217 were classified as late-type. Of the 2619 galaxies classified as late-type by ESO-LV, 2471 were classified as such by the network, while 148 were classified as early-type. This means a success rate of 90 per cent.

Breaking down the early class into ellipticals and lenticulars (S0s) demonstrates that the vast majority of the variance is in the classification of S0s. Of the 311 galaxies classified as ellipticals ($-5 \leq T < -2.5$) by ESO-LV, the ANN agreed on 94 per cent of them and disagreed for only 6 per cent. On the other hand, of the 587 galaxies classified as S0 ($-2.5 \leq T < 0.5$) by ESO-LV, the ANN agreed only for 66 per cent and disagreed for 34 per cent. This is yet another indication that the S0s form a 'transition class' along the Hubble sequence.

4.3 Five-class classifier

This is essentially the network we presented in SLSS. The input layer consists of the 13 parameters and the output layer consists of the five classes described in Section 2. The configuration used was (13:13:5) with a sigmoid as the non-linear transfer function. The learning and momentum coefficients were kept constant at $\eta = 0.5$ and $\alpha = 0.2$, for all layers.

During training (using 1700 ESO-LV galaxies of the P13 sample), the ANN compared the output of these five nodes to the visual classification decisions of LV89. We then tested the ANN against the remaining 3517 galaxies of the P13 sample. Morphological classification was performed by assigning the galaxy to the class corresponding to the maximal output component. Further experiments that we carried out with a variety of network configurations showed that the number of hidden units and layers, the epoch, the size of training set, the number of iterations, and the learning and momentum coefficients had little effect.

Our main result, shown in Table 2 of SLSS still stands. The percentage of galaxies correctly classified was 64 per cent (and 96 per cent to within the nearest class; if either the first or the second highest outputs are considered in the comparison with the visual classification, the success rate is 90 per cent). On the other hand, a simple Bayesian classifier we constructed (assuming a Gaussian multivariate function, see Appendix C1, equations C1 and C2) only gave 56 per cent. This is the same success rates as reported by LV89 for their linear classifier. This clearly shows that non-linear ANNs can be superior to linear classifiers, and that the classifier itself is of great importance, not only the parameters. We note that an improvement of 8 per cent in classification success rate is very significant for the new large surveys with $\sim 10^6$ galaxies (Sloan and 2dF).

4.4 PCA data compression as input to ANNs

In this section we address the question how many principal components are needed to recover the same classification as achieved with ANN using the full input data. To illustrate this point we use the D7 sample of 791 ESO-LV galaxies, where the input parameters are the logs of seven *metric* diameters. The network architecture was 7:3:1, with both input and output scaled to [0, 1]; a quasi-Newton algorithm was used, with a weight-decay coefficients of $\alpha/\beta = 0.001$. Training was done on 600 galaxies, and testing on the remaining 191 galaxies. The resulting rms scatter (equation 6) evaluated over several runs is $\sigma = 2.2$. Using as input only the first principal component, which was derived in Section 3.1 and accounts for 95 per cent of the variance, we find a much larger scatter, $\sigma = 3.6$. Only when the first three PCs are used does one recover the scatter achieved by using all seven diameters. This shows that the fractional variance on its own is not sufficient to tell us how many PCs are needed for classification.

The failure of the first principal component to recover on its own the classification might be due to non-linearity in the data, the effect of noise on the deduction of principal components, or the fact that classification requires more information than that given just by the maximal variance (i.e. the second moment of the distribution function). We note that the fractional variance of the eigenvalues is often used as the sole criterion in compressing the data prior to applying a classification procedure. For example, Okamura et al. (1984) deduced their 'concentration parameter' by using only the first principal component (cf. our first PC in Table 1 for the seven diameters). However, this criterion may underestimate the importance of the minor principal components. It may well be that classification can be improved by using more principal components. Furthermore, our experience shows that, in some cases, minor principal components are more important for classification than major principal components.

4.5 Scaled parameters versus absolute parameters

So far, in this paper as well as in our previous studies (SLSS; Naim et al. 1995b; Lahav et al. 1995), we have not used the distance (as estimated from the redshift) to the galaxies. Our input parameters were always scaled, such that they were distance-independent. In a sense, we have assumed that what matters in classification is the relative properties of the galaxies: for example, that two ellipticals with high and low absolute luminosities will be classified as the same type if one is a scaled-down (or scaled-up) version of the other.

The astrophysical question whether galaxies were formed in a self-similar way, or in a way that mainly depends on their total mass or potential well is still open. For example, Simien & de Vaucouleurs (1986) showed a tight correlation between the disk-to-bulge ratio (a distance-independent property) and the Hubble type, while Meisels & Ostriker (1984) argued that the absolute luminosity of the spheroidal component (a distance-dependent property) is the major parameter determining the Hubble sequence.

To test this question we used the D7 data as described in Section 4.4 and fed the ANN with the logs of the ratios of six diameters to the half-light diameter. The resulting scatter was larger, $\sigma = 2.4$, than the scatter of 2.2 when all seven metric diameters were presented. Our tentative conclusion is that absolute parameters are not much more informative than the scaled properties. However, the quality of the data and the parameters used (diameters) are not sufficient to prove the theoretical prejudice some may have that only scaled (self-similar) properties control the fate of a galaxy along the Hubble sequence.

5 DISCUSSION

In this paper we have attempted to de-mystify ANNs by showing how they generalize other statistical methods commonly used in astronomy and other fields. The methods were illustrated using the ESO-LV galaxy data, showing that ANNs can successfully replicate human classification. These results for ESO-LV are in accord with our results for the APM sample of 830 galaxy images (Lahav et al. 1995; Naim et al. 1995b): an ANN can replicate the classification by a human expert to within 2 T-type units, similar to the scatter between two human experts.

ANNs are sometime considered as being esoteric methods. Questions commonly asked by 'neuro-sceptics' are: (i) can we understand what the ANNs are doing, or are they just 'black boxes'? (ii) if one has already selected 'good parameters', does it matter what classifier is to be used? We have shown that the ANNs approach should be viewed as a general statistical framework. Some special cases of ANNs are statistics we are all familiar with. However, the ANNs can do better, by allowing non-linearity. There is, of course, freedom in choosing what kind of 'non-linearity' to apply, but sensible choices show that significant improvement can be achieved over the linear approaches. For cosmologists, there is an analogy here with N-body simulations of gravitational systems. Linear theory is reasonably well understood, but is not sufficient to describe complicated dynamics. One then needs to use numerical simulations, producing results that are not always understood by intuition or by analytic methods. One can, however, verify what is happening by considering simple cases (for example the spherical in-fall model) to gain confidence in what the simulations give. Our approach to the ANNs is similar.

This paper does not, of course, cover all possible approaches to classification. For example, as described in Appendix C2, one can use just a linear network (in which the weights effectively act like a Wiener filter), but modify the input parameters to be non-linear (in a somewhat ad hoc way). In some cases such networks can do as well as the non-linear ANNs. Another important issue, not discussed here, is how to handle noisy data (see, for example, Folkes et al. 1996).

An even more challenging task is to devise 'unsupervised' algorithms, where there is no external 'teacher', and the data speak for themselves. Such methods could be either 'cooperative' (for example PCA, non-linear encoder, or the Kohonen 1989 self-organizing map) or 'competitive' (for example cluster analysis). For preliminary applications of unsupervised methods to galaxy classification see Naim (1995). These unsupervised algorithms may well explore new features in the data set which were previously ignored by the human experts.

In astrophysics, the goals are to incorporate dynamical properties of galaxies (for example circular velocities) and multiwavelength data (from radio to the UV). The hope is that these methods will help to define a new physical space of galaxies, in analogy with the HR diagram for stars.

ACKNOWLEDGMENTS

We thank W. Fitzgerald, S. Folkes, M. Fukugita, J. Hertz, M. Irwin, C. Jones, J. Lasenby, D. Lynden-Bell, D. MacKay, B. Ripley and T. von Hippel for stimulating discussions. OL acknowledges the hospitality of the Hebrew University and the Weizmann Institute, where parts of this paper were written, AN is grateful for an Overseas Research Studentship and an Isaac Newton Studentship, LSJ acknowledges FAPESPO and CNP for financial support, and MCSL thanks the Sheepshanks Fund and the Harrison-Watson Foundation of Clare College for financial support.

REFERENCES

Abraham R., Valdes F., Yee H. K. C., van den Bergh S., 1994, ApJ, 432, 75
Angel J. R. P., Wizinowich P., Lloyd-Hart M., Sandler D., 1990, Nat, 348, 221
Doi M., Fukugita M., Okamura S., 1993, MNRAS, 264, 832
Folkes R. S., Lahav O., Maddox S. J., 1996, MNRAS, in press
Francis P., Hewett P. C., Foltz C. B., Chafee F. H., 1992, ApJ, 398, 480
Fraser D. D., Khan Z., Levy D. C., 1992, in Aleksander I., Taylor J., eds, Artificial Neural Networks. Elsevier, Amsterdam
Geva S., Sitte J., 1992, IEEE, 3, 621
Gish H., 1990, in Proc. IEEE Conf., Acoustics Speech and Signal Processing. p. 1361
Gull S. F., 1989, in Skilling, J., ed., Maximum Entropy and Bayesian Methods. Kluwer, Dordrecht, p. 53
Hebb D. O., 1949, The Organization of Behavior. Wiley, New York
Hertz J., Krogh A., Palmer R. G., 1991, Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City
Hubble E., 1936, The Realm of Nebulae. Yale Univ. Press, New Haven
Kent S., 1985, ApJS, 59, 115
Kodaira K., Watanabe M., Okamura S., 1986, ApJS, 62, 703
Kohonen T., 1989, Self-Organization and Associative Memory 3rd edn. Springer-Verlag, Berlin
Lahav O., 1995, in Paturel G., Petit C., eds, The World of Galaxies II, Lyon 1994, Astrophysical Letters & Communications, 31, 73, Gordon & Breach
Lahav O., Gull S. F., 1989, MNRAS, 240, 753
Lahav O. et al., 1995, Science, 267, 859

Lasenby J., Fitzgerald W. J., 1993, CUED/F-INENG/TR.142

Lauberts A., VAletijn E. A., 1989, The Surface Photometry Catalogue of the ESO – Uppsala Galaxies, ESO (LV89)

MacKay D. J. C., 1992, PhD thesis, Caltech

Mahonen P. H., Hakala P. J., 1995, ApJ, 452, L77

Meisles A., Ostriker J. P., 1984, AJ, 89, 1451

Miller A. S., 1993, Vistas in Astron., 36, 141

Murtagh F., Heck A., 1987, Multivariate Data Analysis. Reidel, Dordrecht

Naim A., 1995, PhD thesis, Univ. Cambridge

Naim A. et al., 1995a, MNRAS, 274, 1107

Naim A., Lahav O., Sodré L., Jr, Storrie-Lombardi M. C., 1995b, MNRAS, 275, 567

Odewahn S. C., Stockwell E. B., Pennington R. L., Humphreys R. M., Zumach W. A., 1991, AJ, 103, 318

Oja E., 1982, J. Math. Biol., 15, 267

Oja E., 1992, Neural Networks, 5, 927

Okamura S., Kodaira K., Watanabe M., 1984, ApJ, 280, 7

Pao Y. H., 1989, Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, New York

Parker D. B., 1985, Report TR-47. MIT, Center for Computational Research in Economics and Management Science, Cambridge, MA

Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 1992, Numerical Recipes, 2nd edn. Cambridge Univ. Press, Cambridge

Rayner J. W., Lynch M. R., 1989, IEEE, D7.10

Richard M. D., Lippmann R. P., 1991, Neural Computation, 3, 461

Ripley B. D., 1993, in Mardia K. V., ed., Statistics and Images. Carfax, Abingdon

Rumelhart D. E., Hinton G. E., Williams R. J., 1986, Nat, 323, 533

Rybicki G. B., Press W. H., 1992, ApJ, 398, 169

Sanger T. D., 1989, Neural Networks, 2, 459

Serra-Ricart M., Calbet X., Garrido L., Gaitan V., 1993, AJ, 106, 1685

Simien, de Vaucouleurs G., 1986, ApJ, 302, 564

Spiekermann G., 1992, AJ, 103, 2102

Storrie-Lombardi M. C., Lahav O., 1994, guest eds, Vistas in Astron., spec. iss. on ANNs in Astronomy, 38 (3)

Storrie-Lombardi M. C., Lahav O., 1995, in Arbib M. A., ed., Handbook of Brain Theory and Neural Networks. MIT Press, Boston

Storrie-Lombardi M. C., Lahav O., Sodré L., Storrie-Lombardi L. J., 1992, MNRAS, 259, 8P (SLSS)

Thonnat M., 1989, in Corwin H. G., Jr, Bottinelli L., eds., The World of Galaxies. Springer-Verlag, New York, p. 53

von Hippel T., Storrie-Lombardi L. J., Storrie-Lombardi M. C., Irwin M., 1994, MNRAS, 269, 97

Werbos P. J., 1974, PhD thesis, Harvard Univ., Cambridge, MA

Wiener N., 1949, Extrapolation and Smoothing of Stationary Time Series. Wiley, New York

Zaroubi S., Hoffman Y., Fisher K. B., Lahav O., 1995, ApJ, 449, 446

APPENDIX A: PCA AND ANN

A1 Standard PCA

A pattern can be thought of as being characterized by a point in an M-dimensional parameter space. One may desire a more compact data description, where each pattern is described by M' quantities, with $M' \ll M$. This can be accomplished by principal component analysis (PCA), a well-known statistical tool commonly used in astronomy (e.g. Murtagh & Heck 1987 and references therein). The PCA method is also known in the literature as the Karlhunen–Loéve or Hotelling transform, and is closely related to singular value decomposition. By identifying the *linear* combination of input parameters with maximum variance, PCA finds the M' variables ('principal components') that can be most effectively used to characterize the inputs.

The first principal component is taken to be along the direction in the M-dimensional input parameter space with the maximum variance. More generally, the kth component is taken along the maximum-variance direction in the subspace perpendicular to the subspace spanned by the first $(k-1)$ principal components. It is convenient to apply PCA to data that are already standardized, for example transformed to zero-mean and unit-variance. However, while this scaling is appropriate for data composed of 'apples and oranges' as in the present paper for the 13 ESO-LV parameters, in other problems such as the seven ESO-LV diameters and spectral analysis of quasars and galaxies (se Francis et al. 1992; Lahav 1995) it is more sensible not to divide by the variance of each channel (over an ensemble of objects), in order to keep the relative strengths of the lines.

The formulation of standard PCA is as follows. Consider a set of N objects $(i=1, N)$, each with M parameters $(j=1, M)$. If r_{ij} are the original measurements, we construct normalized properties as follows:

$$x_{ij}=\frac{r_{ij}-\bar{r}_j}{s_j}, \qquad \text{(A1)}$$

where $\bar{r}_j=(\sum_{i=1}^{N} r_{ij})/N$ is the mean, and $s_j^2=[\sum_{i=1}^{N}(r_{ij}-\bar{r}_j)^2]/N$ is the variance. We then construct a correlation matrix

$$C_{jk}=\frac{1}{N}\sum_{i=1}^{N} x_{ij}x_{ik}. \qquad \text{(A2)}$$

It can be shown that the axis along which the variance is maximal is the eigenvector $\boldsymbol{u}_1$ of the matrix equation

$$\mathbf{C}\boldsymbol{u}_1=\lambda_1\boldsymbol{u}_1, \qquad \text{(A3)}$$

where λ_1 is the largest eigenvalue, which is in fact the variance along the new axis. The other principal axes and eigenvectors obey similar equations. It is convenient to sort them in decreasing order, and to quantify the fractional variance by $\lambda_\alpha/\sum_\alpha \lambda_\alpha$. It is also convenient to renormalize each component by $\sqrt{\lambda_\alpha}$, to give unit-variance along the new axis. We note that the weakness of PCA is that it assumes linearity and also depends on the way the variables are scaled. In contrast, ANNs generally allow non-linearity.

A2 Principal components from an encoder

PCA is, in fact, an example of 'unsupervised learning', in which an algorithm or a 'linear network' discovers for itself features and patterns (see e.g. Hertz, Krogh & Palmer 1991 for a review). A simple network configuration M:M':M (see Fig. 2) with linear transfer functions allows us to find M' linear combinations of the original M parameters. The idea is to force the output layer to reproduce the input layer, by least-squares minimization (for example using the backpropagation algorithm, see Appendix B). If the number of 'neck units' M' equals M then the output will exactly reproduce the input. If $M' < M$, however, the network will find, after minimization, the optimal linear combination. By changing the threshold function from linear to non-linear (for example a sigmoid) one can allow 'non-linear PCA'. Some authors (e.g. Geva & Sitte 1992; Serra-Ricat et al. 1993) advocate a configuration of $M{:}2M+1{:}M'{:}2M+1{:}M$ to obtain optimal reconstruction of non-linear shapes, for example a circle.

A3 Neural PCA: Oja's neural network

One interesting aspect of ANN theory is that a very simple artificial neuron can be trained to extract the first principal component of the input parameters (Oja 1982). Consider an artificial neuron, which receives a set of n scalar-valued input $w_1, \ldots, w_n$ and produces an output Y:

$$Y = \sum_{i=1}^{n} w_i x_i. \tag{A4}$$

During the training of this neuron, the weights w_i can be changed after the presentation of a pattern according to the Hebbian rule (Hebb 1949)

$$\Delta w_i = \eta Y x_i, \tag{A5}$$

where η controls the rate of learning. Oja (1982) generalized this rule by incorporating a normalization

$$w_i(t+1) = \frac{w_i(t) + \eta Y x_i}{\left\{\sum_{j=1}^{n} [w_j(t) + \eta Y x_j]^2\right\}^{1/2}}, \tag{A6}$$

where η is the 'learning coefficient'. Expanding this expression as a power series in η and retaining only the first-order term yields the learning equation known as Oja's rule:

$$w_i(t+1) = w_i(t) + \eta Y(t)\,[x_i - Y(t) w_i(t)]. \tag{A7}$$

Oja's rule, after training, chooses the direction of the weights vector $\boldsymbol{w}$ to lie in the maximal eigenvector direction of the correlation matrix $\langle \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \rangle$ (assuming zero-mean data, and using here matrix multiplication notation, $\boldsymbol{x}^{\mathrm{T}}$ being the transposed vector). Moreover, this turns out to be also the direction that maximizes the variance of the output $\langle Y^2 \rangle = \boldsymbol{w}^{\mathrm{T}} \langle \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \rangle \boldsymbol{w}$ (see e.g. Hertz et al. 1991). Oja (1982) also showed that, after training, the normalization $\sum_{i=1}^{n} w_i^2$ tends to be bounded and close to one. Other rules to extract the first and higher principal components have been proposed by, for example, Sanger (1989) and Oja (1992). While these learning rules give some insight into the link between PCA and ANN, in practice it is easier to extract the principal components by the standard method (Appendix A1) or by a linear encoder (Appendix A2).

APPENDIX B: MINIMIZATION ALGORITHMS

There are many algorithms that can be used for minimization in multiparameter space. They differ in the methods used and in the seed of convergence. We show below two algorithms that we used in various applicarions; both gave very similar results.

B1 The back-propagation method

The back-propagation algorithm has been re-invented several times (e.g. Werbos 1974; Parker 1985; Rumelhart, Hinton & Williams 1986) and is one of the most popular ANN algorithms. A typical configuration is shown in Fig. 4. For a given network architecture, the first step is the 'training' of the ANN. In this step the weights w_{ij} (the 'free parameters') are determined by minimizing 'least-squares'. The novel aspect of back-propagation is the way this minimization is done, using the chain rule (gradient descent).

Each node (except the input nodes) receives the output of all nodes in the previous layer and produces its own output, which then feeds the nodes in the next layer. A node at layer s calculates a linear combination over the input $x_i^{(s-1)}$ from the previous layer $s-1$ according to

$$I_j^{(s)} = \sum_{i=0}^{n} w_{ij}^{(s)} x_i^{(s-1)}, \tag{B1}$$

where the w_{ij} are the weights associated with that node. Commonly, one takes $x_0 = 1$, with w_{0j} playing the role of a 'bias' or DC level. The node then fires a signal

$$x_j^{(s)} = f(z), \tag{B2}$$

where z here stands for $I_j^{(s)}$, and f is a non-linear transfer function usually of the sigmoid form

$$f(z) = 1/[1 + \exp(-z)] \tag{B3}$$

in the interval [0, 1], or

$$f(z) = \tanh(z) \tag{B4}$$

in the interval $[-1, 1]$.

For each object (pattern) in the training set, the network compares its output vector in the 'classification space' $\boldsymbol{o}$ to the desired vector $\boldsymbol{d}$ determined by the 'true answer' (for example as given by a human expert). For example, the elements of the vector $\boldsymbol{d}$ can be defined as zeros except for one element set to 1 corresponding to an actual class, for example we define $\boldsymbol{d} = (1, 0, 0, 0, 0)$ for elliptical galaxies.

The comparison is done in terms of a cost function, usually of the form

$$E = \frac{1}{2} \sum_k (o_k - d_k)^2, \tag{B5}$$

where the sum is over the components of the vectors. This cost function, averaged over all the training galaxies presented to the ANN is minimized with respect to the free parameters, the weights w_{ij}. The weights are updated by gradient descent *backwards* (hence the name back-propagation) from the output layer to one or more hidden layers, by a small change in each time-step,

$$\Delta w_{ij}(t+1) = -\eta \frac{\partial E}{\partial w_{ij}} + \alpha \Delta w_{ij}(t), \tag{B6}$$

where the 'learning coefficient' η and the 'momentum' α are 'knobs' that control the rate of learning and the inertia from the previous iteration respectively (see e.g. Hertz et al. 1991).

The elegance of the back-propagation algorithm is in the way that the derivative is evaluated. Let us consider the case of a sigmoid output

$$o_j^{(s)} = x_j^{(s)} = f(I_j^{(s)}), \tag{B7}$$

where

$$f(z) = \frac{1}{1+e^{-z}}. \tag{B8}$$

In this case note that $f' = f(1-f)$ and the derivative can be written as

$$\frac{\partial E}{\partial w_{ij}} = o_i^{(s-1)} o_j^{(s)} (1-o_j^{(s)}) \beta_j, \tag{B9}$$

where

$$\beta_j = (o_j - d_j) \tag{B10}$$

for nodes in the *output* layer, and

$$\beta_j = \sum_k w_{jk} o_k (1-o_k) \beta_k \tag{B11}$$

for nodes in *hidden* layers (the sum is over k nodes in the layer *above* node j).

The 'hidden layers' allow curved boundaries around clouds of data points in the parameter space in a non-parametric way. The interpretation of the output depends on the network configuration. For example, a single output node provides a continuous output (for example predicting the T-type as in Section 4.1 or the luminosity of a galaxy), while several output nodes can be used to assign probabilities to different classes (for example five morphological types of galaxies), as explained in Appendix C.

B2 The quasi-Newton method

There are methods, other than back-propagation, for minimizing the non-linear function equation (B5). A more efficient method is quasi-Newton. In short, the cost function $E(\boldsymbol{w})$ in terms of the weight vector $\boldsymbol{w}$ is expanded about a current value $\boldsymbol{w}_o$:

$$E(\boldsymbol{w}) = E(\boldsymbol{w}_0) + (\boldsymbol{w}-\boldsymbol{w}_0)\nabla E(\boldsymbol{w}_0) + \tfrac{1}{2}(\boldsymbol{w}-\boldsymbol{w}_0)\cdot \mathbf{H} \cdot (\boldsymbol{w}-\boldsymbol{w}_0) + \dots, \tag{B12}$$

where $\mathbf{H}$ is the Hessian with elements $H_{ij} = \partial^2 E / \partial w_i \, \partial w_j$ evaluated at $\boldsymbol{w}_0$. The minimum occurs at approximately

$$\nabla E(\boldsymbol{w}) \approx \nabla E(\boldsymbol{w}_0) + \mathbf{H}\cdot(\boldsymbol{w}-\boldsymbol{w}_0) = 0. \tag{B13}$$

Hence an estimation for the optimal weight vector is

$$\boldsymbol{w} = \boldsymbol{w}_0 - \mathbf{H}^{-1}\nabla E(\boldsymbol{w}_0). \tag{B14}$$

In the standard Newton's method, a previous estimate of $\boldsymbol{w}$ is used as the new $\boldsymbol{w}_0$. Calculating the Hessian exactly is expensive computationally, and in the quasi-Newton method an iterative approximation isused for the inverse of the Hessian (e.g. Press et al. 1992; Hertz et al. 1991.

APPENDIX C: RELATIONS BETWEEN ANNs AND OTHER CLASSIFIERS

C1 Bayesian classification and probabilities

A classifier can be formulated from first principles according to Bayes theorem:

$$P(T_j|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|T_j)P(T_j)}{\sum_k P(\boldsymbol{x}|T_k)P(T_k)}, \tag{C1}$$

i.e. the a posteriori probability for a class T_j given the parameter vector $\boldsymbol{x}$ is proportional to the probability for data given a class (as can be deduced from a training set) times the *prior* probability for a class (as can be evaluated from the frequency of classes in the training set). However, application of equation (C1) requires parametrization of the probabilities involved. It is common, although not always adequate, to use multivariate Gaussian function

$$P(\boldsymbol{x}|T_j) = (2\pi)^{-M/2} |\mathbf{C}_j|^{-1/2} \exp\left(-\tfrac{1}{2}\boldsymbol{x}\mathbf{C}_j^{-1}\boldsymbol{x}^{\mathrm{T}}\right), \tag{C2}$$

where $\boldsymbol{x}$ is of dimension M and has zero-mean, $\boldsymbol{x}^{\mathrm{T}}$ is its transposed vector, and $\mathbf{C}_j = \langle \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \rangle_j$ is the covariance matrix *per class j*. This matrix is similar to the one used in the PCA (Appendix A1) for *all the classes*. As in PCA, the matrix $\mathbf{C}_j$ can be diagonalized, hence simplifying equation (C2).

It can be shown that certain ANN configurations behave like Bayesian classifiers, i.e. the output nodes produce Bayesian a posteriori probabilities (see e.g. Gish 1990; Richard & Lippmann 1991), although it does not implement Bayes theorem directly. To illustrate this important property of the networks, we follow Gish (1990) for a simple heuristic example. Let the network's single output be written as $f(x, w)$ where x stands for the input parameters and w stands for the weights (more generally these quantities are vectors). We consider a two-class problem for which the desired output of the network is 1 if x is in class T_1 and 0 if it is in class T_2. The cost function over all objects N is then (cf. equation B5)

$$E = \frac{1}{N}\left\{ \sum_{x\in T_1} [f(x,w)-1]^2 + \sum_{x\in T_2} [f(x,w)-0]^2 \right\}. \tag{C3}$$

For large N, and if the number of samples from each of the classes is proportional to the a priori probability of class membership $P(T_j)$, this can be replaced by an integral:

$$E = \int [f(x,w) - P(T_1|x)]^2 P(x)\,\mathrm{d}x + P(T_1) - \int P^2(T_1|x)P(x)\,\mathrm{d}x. \tag{C4}$$

The minimum of this function with respect to w clearly occurs for

$$f(x,w) = P(T_1|x), \tag{C5}$$

so the output of the network can be interpreted as the a posteriori probability. This can be generalized for multiple output. It is reassuring (and should be used as a diagnostic) that the probabilities in an 'ideal' network add up to approximately unity. Moreover, if both the training and testing sets are drawn from the same parent distribution, then the frequency distribution $P(T_j)$ for the objects as classified by the ANN is similar to that of the training set. The link between minimum variance and probability also illustrates why a classification scheme where one calculates the Euclidean distance of the ANN output from the vector representing each of the possible classes, and then assigns the object to the class producing the minimum distance, is equivalent to assigning a class according to the highest probability (cf. Richard & Lippmann 1991). For a sigmoid output (equation B3) it can be shown (Gish 1990) that the argument of the sigmoid, $z(x, w) = \ln [f(x, w)/(f(x, w) - 1)]$, with $f(x, w) = P(T_1|x)$ (equation C4) and $P(T_2|x) = 1 - P(T_1|x)$ gives

$$z(x, w) = \ln \frac{P(T_1|x)}{P(T_2|x)}, \tag{C6}$$

i.e. the argument of the sigmoid models the log-likelihood ratio of the two classes. With the transfer function tranh $(z) = 2/[1 + \exp(-2z)] - 1$ the interpretation is similar. We note that the above analysis (equation C4) does not give any information about the network architecture, and it only holds for 'idealized' network and data. For more rigorous and general Bayesian approaches for modelling ANNs see MacKay (1992).

C2 Linear networks and Wiener filtering

The weights, the free parameters of the ANN, can have a simple interpretation when the network is linear $[f(z) = z]$ without hidden layers, commonly called the 'perceptron'. For simplicity of notation we consider a network with a single continuous output, for example yielding the type T. In this case we can write the cost function as

$$E = \frac{1}{2} \frac{1}{N} \sum_{\mu=1}^{N} \left[T^{\mu} - \sum_{k=0}^{M} w_k x_k^{\mu} \right]^2, \tag{C7}$$

where $\mu = 1, \ldots N$ labels the objects, and $k = 0, \ldots M$ the parameters. The index $k = 0$ stands for the 'bias' term (with $x_0 = -1$), and it plays the role of an additive constant w_0 in the network equation. The minimum of E with respect to the weights occurs at

$$\frac{\partial E}{\partial w_j} = \frac{1}{N} \sum_{\mu} \left[T^{\mu} - \sum_{k} w_k x_k^{\mu} \right] x_j^{\mu} = 0, \tag{C8}$$

giving

$$\sum_k \langle x_k x_j \rangle w_k = \langle T x_j \rangle, \tag{C9}$$

where $\langle \ldots \rangle$ are averages over the N objects.

The solution of this set of linear equations (for $j = 1, \ldots M$) for the optimal weights vector $\boldsymbol{w}$ can be written as

$$\boldsymbol{w}_{\rm opt} = \mathbf{A}^{-1} \boldsymbol{b}, \tag{C10}$$

where $A_{jk} = \langle x_k x_j \rangle$ and $b_j = \langle T x_j \rangle$. More generally, if there are multiple output units (say a vector $\boldsymbol{s}$) so the weights form a matrix $\mathbf{W}$ (not necessarily square), the minimum variance $\langle (\boldsymbol{s} - \mathbf{W}\boldsymbol{x})(\boldsymbol{s} - \mathbf{W}\boldsymbol{x})^{\rm T} \rangle$ with respect to the weights occurs for

$$\mathbf{W}_{\rm opt} = \langle \boldsymbol{s}\boldsymbol{x}^{\rm T} \rangle \langle \boldsymbol{x}\boldsymbol{x}^{\rm T} \rangle^{-1}. \tag{C11}$$

This is in fact the standard Wiener (1949) filter known in digital filtering and image processing, commonly applied for signal-plus-noise problems with $\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n}$ (e.g. Rybicki & Press 1992 for a review, and Zaroubi et al. 1995 and references therein for recent cosmological applications). We note that the same result can be derived by conditional probabilities with Gaussian probability distribution functions, as well as by regularization with a quadratic prior.

For an alternative, somewhat more complicated, expression see Hertz et al. (1991, p. 102), where the weights are given in terms of a covariance matrix of the *objects* (useful for the case of many features and fewobjects).

One can go one step further to generalize the above to non-linear *input*. This can be done by, for example, expanding the elements of the input vector as products of their powers. For example, if the input parameters are x_1 and x_2 the expanded input vector is

$$[1, x_1, x_2, x_1^2, x_1x_2, x_2^2, \ldots].$$

This is sometime called the Volterra connectionist model (VCM; see e.g. Rayner & Lynch 1989; Pao 1989; Lasenby & Fitzgerald 1993). Other alternatives for non-linear input are 'radial basis functions' and spherical harmonics. In fact, this can be viewed as an ad hoc hidden layer which forces the input to a new non-linear form. The advantage of a VCM network is that the global minimum is unique. This provides a reproducible solution and allows fast training. On the other hand, as the connections between input and 'hidden' layer are 'hard-wired', the freedom of the network for difficult data sets is limited.

C3 Regularization and weight decay

As in other inversion problems, the determination of many free parameters, the weights w_i in our case, might be unstable. It is therefore useful to regularize the weights, for example by preventing them from growing too much. In the ANN literature this is called 'weight decay'. This approach is analogous to the maximum entropy method, and can be justified by Bayesian arguments, with the regularizing function acting as the prior in the weight-space. Note that this is a different application of Bayes theorem from the one discussed in Section C1, applied in the class-space.

One possibility is to add a quadratic prior and to minimize

$$E_{\rm tot} = \alpha E_{\rm w} + \beta E_{\rm D}, \tag{C12}$$

where $E_{\rm D}$ is our usual cost function, based on the data (for example equations B5 and C6), and

$$E_{\rm w} = \frac{1}{2} \sum_{i=1}^{Q} w_i^2 \tag{C13}$$

is the chosen regularizing function, where Q is the total number of weights. The coefficients α and β can be viewed as 'Lagrange multipliers'. While they are sometimes specified ad hoc, it is possible to evaluate them 'objectively' using Bayesian arguments in the weight-space. This has been done in the context of ANNs by MacKay (1992, see also Ripley 1993, following earlier analysis in relation to maximum entropy by Gull (1989; see also Lahav & Gull 1989). The Bayesian analysis gives the conditions on α and β as

$$\chi_{\mathrm{w}}^2 = 2\alpha\hat{E}_w = \gamma \tag{C14}$$

and

$$\chi_{\mathrm{D}}^2 = 2\beta\hat{E}_{\mathrm{D}} = N - \gamma, \tag{C15}$$

where N is the number of data points (objects) and

$$\gamma = \sum_{q=1}^{Q} \frac{\lambda_q}{\lambda_q + \alpha}, \tag{C16}$$

where the λ_qs are the eigenvalues of the Hessian (in the weight-space) $\beta\nabla\nabla E_{\mathrm{D}}$, evaluated with the weights at which E_{tot} is minimum. The parameter γ has an interesting interpretation, the number of 'well-determined' weights. If $\lambda_q \geq \alpha$ then $\gamma \approx Q$ (the total number of weights). In this case $\chi_{\mathrm{D}}^2 \approx N - Q$, which is similar to the usual condition that χ^2 equals the number of degrees of freedom. Moreover, if $Q \ll n$ then

$$E_{\mathrm{tot}} \approx \frac{1}{\sigma_{\mathrm{w}}^2} E_w + \frac{1}{\sigma_{\mathrm{D}}^2} E_{\mathrm{D}}, \tag{C17}$$

where $\sigma_{\mathrm{w}}^2 = 2\hat{E}_{\mathrm{w}}/Q = \sum w_i^2/Q$ and $\sigma_{\mathrm{D}}^2 = 2\hat{E}_{\mathrm{D}}/N$, as expected for Gaussian probability distribution functions. We note that this analysis makes sense if the input and output are properly scaled, for example on the range [0, 1] with sigmoid transfer functions, so all the weights are treated in the regularization process on an 'equal footing'. It can be generalized for several regularizing functions, for example one per layer.

We note that the addition of the regularization term E_{w} changes the location of the minimum, now satisfying

$$\nabla E_{\mathrm{D}} = -\frac{\alpha}{\beta} \nabla E_{\mathrm{w}} = -\frac{\alpha}{\beta} \boldsymbol{w}, \tag{C18}$$

as from equation (C13) $\nabla E_{\mathrm{w}} = \boldsymbol{w}$. The effect of the regularization term here is similar to the restoring force of a harmonic oscillator: the larger $\boldsymbol{w}$ is, the more it will be suppressed. The addition of the regularization term to equation (C4) gives a minimum for the extended cost function which does not satisfy equation (C5), i.e. it violates the probabilistic interpretation in the class-space. However, one could construct a network with regularization that will produce probabilities self-consistently (e.g. MacKay 1992). The weight-decay term also modifies the Wiener solutions in Section C2.