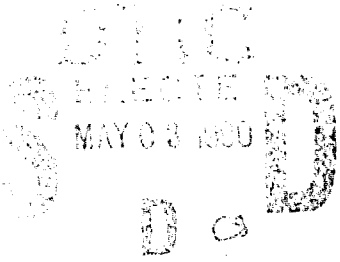
RSRE
MEMORANDUM No. 4343

ROYAL SIGNALS & RADAR ESTABLISHMENT



INCORPORATING PRIOR PROBABILITIES AND MISCLASSIFICATION COSTS INTO NETWORK TRAINING: AN EXAMPLE FROM MEDICAL PROGNOSIS

Authors: D Lowe & A R Webb

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
R S R E MALVERN,
WORCS.

R S R E MEMORANDUM No. 4343

UNLIMITED

CONDITIONS OF RELEASE

0064778 BR-113093

..... DRIC U

..... DRIC Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

Royal Signals and Radar Establishment
Memorandum 4343

Incorporating Prior Probabilities and Misclassification Costs into Network Training: an Example from Medical Prognosis.

David Lowe and Andrew R. Webb

1st November 1989.

Abstract

Feed-forward layered networks trained on a pattern classification task in which the number of training patterns in each class is nonuniform, bias strongly in favour of those classes with largest membership. This is an unfortunate property of networks when the relative *importance* of classes with smaller membership is much greater than that of classes with many training patterns. In addition, there are many pattern classification tasks where different penalties are associated with misclassifying a pattern belonging to one class as another class. It is not generally known how to compensate for such effects in network training. This paper discusses an analytical regularisation scheme whereby prior expectations of class importance *occurring in the generalisation data* and misclassification costs may be incorporated into the training phase, thus compensating for the uneven and unfair class distributions occurring in the training set. The effects of the proposed scheme on the feature extraction criterion employed in the hidden layer of the network is discussed. An illustration of the results is presented by considering a real medical prognosis problem concerning data collected from head-injured coma patients. Relationships between least mean square error minimisation and Bayesian minimum risk estimation is mentioned and the importance and relevance of input/output coding schemes for network performance is considered.

THIS PAGE IS LEFT BLANK INTENTIONALLY

Contents

List of Figures

List of Tables

1 Introduction

Connectionist models based on adaptive layered networks have been used with some success when operating as static pattern classifiers in problems as diverse as sonar [11] and radar [1] classification, speech recognition [16] and medical diagnosis [3]. The ability of feed-forward layered networks to perform static pattern discrimination stems from their potential to create a *specific* nonlinear transformation into a space spanned by the outputs of the hidden units in which class separation is easier [19, 15] (these comments will be discussed in more detail later and mathematically summarised in the appendices). This transformation is constrained to maximise a *feature extraction criterion* which may be viewed as a nonlinear multi-dimensional generalisation of Fisher's Linear Discriminant function [9]. Since this criterion involves the *weighted* between class covariance matrix (where the weighting is determined by the *square* of the number of patterns in each class), adaptive networks trained on a 1-from-c classifier problem (for a c class problem) bias strongly in favour of those classes which have the largest membership in the training data. Thus, in order to minimise the sum square error over the entire training set, the optimum solution for the network parameters is such that the network misclassifies patterns in classes with smallest representation in favour of those with larger representation in the training set, irrespective of the frequency of occurrence or relative class importance in actual 'operation'.

This is an undesirable feature of networks (and many other standard classifiers) in problems where information on one particular class may be more difficult or expensive to obtain than other classes, and where the relative importance of the classes follows another distribution to their frequency of occurrence. For instance, in speech recognition the bulk of the continuous acoustic signal consists of silence whereas the dominant information content is contained in the subword units ('phonemes') which themselves have differing importance to their frequency of occurrence. Another example which illustrates asymmetric misclassification costs is in the problematic realm of medical prognosis: *given* a feature pattern as determined from a set of observations on a patient, what are the likely future health prospects of that patient. Clearly it is more important to diagnose a serious ailment correctly than to diagnose a minor complaint correctly. However, and more complicated, it is a more serious error to predict incorrectly that a given patient will die if that patient would in fact recover, than to predict incorrectly that a patient would survive when he actually dies, particularly if resources had to be limited to those in most need who would gain maximum benefit. Thus, the problem is compounded by asymmetric misclassification costs.

These are illustrative of real pattern classification problems where the distribution of patterns amongst the different classes in the training set is nonuniform and also could follow a different distribution to the *expected* occurrence or the relative importance of the classes in operation. In addition, there may be further prior knowledge which could be used to associate a penalty of misclassification of each class with any other. In spite of the obvious practical relevance of such aspects in real-world pattern processing problems, it is an area of network research where very little detailed analysis has been performed. One of the aims of this paper is to create an awareness of the existence of such problems in the naïve application of adaptive networks to real data, and how they arise. A second aim is to provide the theoretical justification behind our proposed solutions to the problems raised.

It is possible, of course, to develop heuristic methods which attempt to compensate for some of the mentioned effects in training adaptive networks. For instance, the classes of

the training data may be sampled according to a distribution which reflects the importance or expected frequency of occurrence of patterns in that class and the network subsequently can be trained on the sampled data. Alternatively, if training proceeds iteratively and sequentially, the number of iterations in the learning cycle of a network may be varied for each class or pattern which would have a similar compensatory effect. Equivalently, the sum square error minimised by the network during training may be weighted by the frequency of occurrence of the patterns in each class. It has not been obvious what the effects of such methods have had on the feature extraction mechanisms employed by adaptive networks.

In this paper, an analytic regularisation scheme [15] is reviewed and discussed. This allows for effects such as uneven class membership and importance, to be compensated for during training so as to produce a network with the desired characteristics *in operation*. In particular, a network may be 'tuned' during the learning phase by an appropriate choice of error weighting *and* target coding by exploiting prior knowledge specific to the problem under study. The effects of these factors on the space spanned by the outputs of the hidden units will be considered from the point of view of a generalised feature extraction criterion. The essential results are stated in the following section with relevant concise mathematical details contained in the appendices. The second half of the paper applies the results to a real medical prognosis problem taken from an analysis of 1000 patients suffering severe head injury and contrasts the results obtained by a feed-forward network with those achieved by various standard pattern classifiers. For this particular problem, one would like to identify quite early on in the treatment, those patients likely to require long term intensive care and therapy. It happens that this class of patient has the smallest frequency of occurrence and so a simple application of adaptive network techniques would find a solution which totally misclassifies this class of patient (this will be illustrated). Indeed this is a common problem with most traditional statistical diagnostic techniques. It will be illustrated how it is possible to improve the likelihood of performing a correct prognosis on that class of patients requiring the greatest long term care by exploiting the results of the next section.

2 Networks, Feature Extraction and Discriminant Analysis

We consider generic feed-forward networks with an arbitrary number of hidden layers (although only one hidden layer is necessary for approximating a given function mapping arbitrarily closely[14]) and each hidden node may have a different nonlinearity. Also, the combination rule transforming patterns from the output of one layer to the input of the next layer may be the usual scalar product rule as used in traditional multi-layer perceptrons, although other combination rules can be used [5] without altering our arguments. However, the transfer functions of the output nodes are restricted to be *linear* to allow us to exploit the properties of linear least mean square optimisation methods in the final layer of weights. This is not a severe restriction. For instance associative mappings of unscaled data require output transfer functions which do not restrict the range of possible outputs. In this paper, network training is viewed as a problem in optimisation by minimising the total residual between the desired target values and the actual network outputs over the entire training set. There are other criteria for training adaptive networks which do not attempt to obtain the best (even locally) minimum error solution. However the theoretical basis of least mean square error minimisation allows a rigorous understanding and analysis on optimum network performance.

The output of the k-th output node of the network may be expressed as

$$O_k = \lambda_{0k} + \sum_{j=1}^{n_0} \lambda_{jk} \phi_j(y_j), \quad k = 1, 2, \ldots, c \tag{1}$$

where λ_{jk} is the connecting weight from the j-th hidden node (of which there are n_0) to the k-th output node (of which there are c), λ_{0k} is a bias term attached to the k-th output node and $\phi_j(y_j)$ is the output from the j-th hidden node in the final hidden layer which is a nonlinear function of the scalar input y_j. The input y_j is a parameterised function of the previous layer patterns. For instance in a multilayer perceptron

$$y_j = \mu_{0j} + \sum_{i=1}^{n} x_i \mu_{ij} \quad j = 1, \ldots, n_o \tag{2}$$

where μ_{0j} is a bias term, μ_{ij} is the weight connecting the i-th node of the previous layer to the j-th node of the current layer, and x_i is the i-th component of the pattern vector output at the previous layer.

Denoting the actual network output vector of the p-th pattern as $\mathbf{o}^p$ and the desired prototype target pattern as $\mathbf{t}^p$, generally one wishes to minimise the error

$$E = \frac{1}{D} \sum_{p=1}^{P} d_p \left\| \mathbf{t}^p - \mathbf{o}^p \right\|^2 \tag{3}$$

where d_p is the scalar weighting associated with the p-th pattern and is usually assumed to be unity. Since the network output represents a (differentiable) flexible though parameterised model, *training* consists of adapting the parameters of the network by any suitable optimisation strategy [21] to minimise this residual error. *Generalisation* ability depends on the network being complex enough (as determined by the number of hidden units in this case) to model the structure in the data adequately without being too complex which would

allow the network to fit the superimposed noise on the data. Previous work [5] has made explicit this relationship between *training* and curve fitting, and between *generalisation* and interpolation to the fitted surface.

Since the output transfer functions are linear the transformation performed by the final layer of weights may be inverted by pseudo-inverse techniques to determine the optimum distribution of patterns at the output of the n_0 hidden nodes which minimises the error. One finds that the choice of weight parameters in the first set of layers of the network distorts the training patterns by a nonlinear transformation into the space spanned by the outputs of the final hidden layer. This distortion is performed so as to maximise a *specific* feature extraction criterion,

$$J = Tr\{\boldsymbol{S}_B \boldsymbol{S}_T^+\} \tag{4}$$

where $Tr\{\boldsymbol{A}\}$ is the trace and $\boldsymbol{A}^+$ is the Moore-Penrose pseudo-inverse of matrix $\boldsymbol{A}$. The mathematical form of the matrices $\boldsymbol{S}_B$, $\boldsymbol{S}_T$ are given in Appendix A. Their precise interpretation depends on the specific output target coding scheme and on the error weighting factors, d_p. However they may be considered as the between class and total covariance matrices of the nonlinearly transformed patterns at the outputs of the hidden layer. Thus, the optimum network solution is obtained by forcing the weights in the primary layers of a network to produce a transformation of patterns into the space spanned by the outputs of the final layer of hidden units which maximises the separability of the classes (through $\boldsymbol{S}_B$) whilst maintaining overall normalisation (through $\boldsymbol{S}_T$). This transformation of patterns is equivalent to an optimum feature extraction criterion (maximising (4)) matched to the (linear) discrimination process of the final layer weights. In this sense, feed-forward layered networks operating as classifiers succeed because they perform a specific discriminant analysis by exploiting subspace methods.

2.1 Specific coding scheme.

Although the generic expressions of the matrices $\boldsymbol{S}_B$, $\boldsymbol{S}_T$ are given in Appendix A it is instructive to consider specific forms for different prototype target coding schemes. Consider a c class problem where it is assumed that there are n_k training patterns in class k: $k = 1, 2, \ldots, c$.

- **Example 1:** $d_p = 1$ and 1-from-c target coding.
  The desired prototype output target values are $t_k = 1$ if the input pattern belongs to class k and zero otherwise. This is the most common form of assumed target coding scheme and error weighting used in adaptive network training. The matrices $\boldsymbol{S}_B$ $\boldsymbol{S}_T$ may be expanded to :

$$\begin{aligned} \boldsymbol{S}_T &= \frac{1}{P}\sum_{p=1}^{P}\left(\boldsymbol{\phi}^p - \boldsymbol{m}^H\right)\left(\boldsymbol{\phi}^p - \boldsymbol{m}^H\right)^* \\ \boldsymbol{S}_B &= \sum_{k=1}^{c}\left(\frac{n_k}{P}\right)^2\left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)\left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)^* \end{aligned} \tag{5}$$

  where $\boldsymbol{x}^*$ denotes the transpose of vector $\boldsymbol{x}$, $\boldsymbol{\phi}^p$ denotes the output vector of the final hidden layer for the p-th pattern, $\boldsymbol{m}^H \triangleq \sum_{p=1}^{P}\boldsymbol{\phi}^p/P$ is the overall mean of the

training set and $\boldsymbol{m}_k^H \triangleq \sum_{\boldsymbol{\phi}^p \in \boldsymbol{k}} \boldsymbol{\phi}^p / n_k$ is the mean over all patterns in class $\boldsymbol{k}$ *evaluated in the space spanned by the outputs of the hidden units.* It is assumed that there are n_k patterns in the k–th class, so that $n_k/P \equiv P_k$ is the prior probability of the k–th class.

It is clear that the expression for $\boldsymbol{S}_T$ in equation (5) is the total covariance matrix of patterns $\boldsymbol{\phi}^p$ at the outputs of the hidden layer and $\boldsymbol{S}_B$ is the *weighted* between class covariance matrix. The weighting is determined by the square of the number of patterns in each class in the training set which skews the feature extraction towards those classes with more patterns. This indicates why networks trained on classification problems where there is an uneven distribution of patterns between classes will bias strongly towards those classes with largest membership. In actual operation (generalisation mode) classes with smallest membership will tend to be ignored.

- **Example 2:** Targets weighted by priors.
In this case the desired output prototype target value t_k will be equal to $1/\sqrt{P_k}$ if the input pattern belongs to class $\boldsymbol{k}$ and zero otherwise. Thus the *gain* in achieving correct classification is inversely proportional to the number of samples occuring in the correct class. The total covariance matrix, $\boldsymbol{S}_T$ remains the same, but $\boldsymbol{S}_B$ becomes

$$\boldsymbol{S}_B = \sum_{k=1}^{c} P_k \left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)\left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)^* \tag{6}$$

the *conventional* between class covariance matrix (where the classes are weighted linearly according to the number of patterns in that class.)

- **Example 3:** $d_p = 1$ and arbitrary loss factors for targets.
The desired prototype target vector for the p–th pattern $\boldsymbol{p}$ will have components t_k which represent the *loss* to be expected from classifying pattern $\boldsymbol{p}$ in class $\boldsymbol{k}$. Again, the total covariance matrix remains the same but $\boldsymbol{S}_B$ is substantially modified to

$$\boldsymbol{S}_B = \sum_{j=1}^{c}\left[\sum_{k=1}^{c} l_{jk} P_k (\boldsymbol{m}_k^H - \boldsymbol{m}^H)\right]\left[\sum_{k=1}^{c} l_{jk} P_k (\boldsymbol{m}_k^H - \boldsymbol{m}^H)^*\right] \tag{7}$$

where l_{jk} is the loss incurred by ascribing to class $\boldsymbol{j}$ a pattern belonging to class $\boldsymbol{k}$. Note that if $l_{jk} = \delta_{jk}$ then this expression reduces to the usual weighted between class covariance matrix in (5).

- **Example 4:** Weight each pattern residual of the training error according to the *a priori* class probabilities and the number of patterns in each class according to

$$d_p = \frac{P(k)}{P_k} \quad \text{for the p–th pattern in the } k\text{–th class}$$

where $P(k)$ is the actual class importance or frequency of occurrence in operation, coded as a probability, and recall that P_k is the prior probability of the k–th class in the training set. In this case both $\boldsymbol{S}_T$ and $\boldsymbol{S}_B$ change. $\boldsymbol{S}_T$ is independent of the particular target coding scheme and becomes

$$\boldsymbol{S}_T = \sum_{k=1}^{c} \frac{P(k)}{n_k} \sum_{\boldsymbol{\phi}^p \in \boldsymbol{k}} \left(\boldsymbol{\phi}^p - \boldsymbol{m}^H\right)\left(\boldsymbol{\phi}^p - \boldsymbol{m}^H\right)^* \tag{8}$$

where the sample based estimate of the population mean $\boldsymbol{m}^H$ now becomes

$$\begin{aligned} \boldsymbol{m}^H &\equiv \sum_{k=1}^{c} \frac{P(k)}{n_k} \sum_{\boldsymbol{\phi}^p \in k} \boldsymbol{\phi}^p \\ &= \sum_{k=1}^{c} P(k) \boldsymbol{m}_k^H \end{aligned} \tag{9}$$

Thus, equation (8) is a sample based estimate of the *expected* total covariance matrix. The form of $\boldsymbol{S}_B$ remains the same as before, *except* that the weighting factors are determined by the expected class importance, $P(k)$ instead of the actual class prior in the training set, P_k. Thus, for instance for the 1-from-c target coding scheme (Example 1), the weighted between class covariance matrix becomes the sample based estimate of the expected weighted between class covariance matrix:

$$\boldsymbol{S}_B = \sum_{k=1}^{c} P(k)^2 \left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)\left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)^* \tag{10}$$

and where $\boldsymbol{m}^H$ is the expected sample based mean given in (9)

A brief summary of the results obtained in this section is justified. It has been illustrated that feed forward deterministic networks perform well as pattern classification devices because the optimum subspace representation formed by the hidden layer executes a specific feature extraction criterion allowing an optimised discriminant analysis. The nature of the transformation is such that, for a 1-from-c target coding scheme, the optimum network solution is obtained by biasing very strongly in favour of those classes with largest pattern membership, *irrespective of the significance of that class*. This is primarily an assumption that the expected occurrence of patterns in the test set is the same as in the training set, which is the best assumption one can make without any prior knowledge. However, it might be known that the occurrence or relative importance of classes in the test set is *not* reflected by the frequency of occurrence in the training set. In this case, one can impose expected class distributions by appropriately weighting the error for patterns in each class. For instance, if in a c-class classification problem the occurrence of each class in operation is considered equally likely but the number of patterns in each class in the training set is distributed with n_k in class $\boldsymbol{k}$, then one should weight each training pattern according to

$$d_p = \frac{1}{c \times P_k} \text{ for pattern } \boldsymbol{p} \text{ in class } \boldsymbol{k}. \tag{11}$$

This will give an equal importance to each class *in the test set*. In addition one may have knowledge regarding the relative costs of misclassification. These effects may be accommodated by incorporating off-diagonal components in the prototype target matrix. The effect of modifying the error weighting and the target matrix has been shown to be effective by influencing the feature extraction criterion, which effectively distorts the between class and total covariance matrices of the transformed patterns.

The 'standard' model (Example 1) is reproduced in the limit of the expected prior probabilities being proportional to the numbers in each class. The analysis has indicated how a combination of gain factors in the prototype target matrix and error weighting by

the inclusion of expected probabilities in the training scheme, induces a nonlinear transformation equivalent to maximising a specific feature extraction criterion which is capable of compensating for uneven inter-class significance. These points will be illustrated later with regard to a specific medical prognosis problem. First, however, it will be useful for subsequent purposes to point out further advantages of training networks by an optimised least mean square error analysis.

2.2 Networks and Probabilities

It is not widely appreciated that there are intimate links between optimally trained networks and traditional Bayesian inference with regard to statistical pattern recognition. This subsection states a few properties which relate to the particular feature extraction discussed in the previous subsection. We discuss how the outputs of a network may be interpreted as probabilities. This motivates the initialisation of a multilayer perceptron as a statistical independence model.

First, note that an optimally trained network with linear output units trained on 1-from-c prototype targets satisfies an interesting property: the outputs of the network *for arbitrary input* are guaranteed to sum to unity[1] (see Appendix B). This is interesting if one wishes to interpret the network outputs as probabilities. However, there is no guarantee of positivity at all the network outputs, thus invalidating any strict classical probabilistic interpretation. Although one can introduce a nonlinear normalisation subsequent to the network output to force the outputs to be all positive and sum to unity [4] regardless of the method of training the network, we will remain within the confines of our assumed model and impose structure through training.

One may motivate the idea that an appropriately trained network can approximate the Bayes decision vector (such that the output of each node represents the probability of that class given the input pattern) by the following example. Consider a network trained on a finite number, P, of input patterns where there are c prototype target vectors. We also assume duplicity of class membership: pattern $\boldsymbol{p}$ occurs n_{pk} times in class $\boldsymbol{k}$. Then the uniform (i.e. $d_p = 1$) error may be rearranged to give

$$\begin{aligned} E &= \frac{1}{P}\sum_{p=1}^{P}\sum_{k=1}^{c}(o_k^p - t_k^p)^2 \\ &\equiv \frac{1}{P}\sum_{p=1}^{\bar{P}}\sum_{k'=1}^{c}\sum_{k=1}^{c} n_{pk'}\left(o_k^p - \bar{t}_k^{k'}\right)^2 \end{aligned} \tag{12}$$

where $\bar{P}$ is the number of distinct patterns, and $\bar{t}_k^{k'}$ is the k' component of the k-th prototype target. The optimum network output which minimises this error is

$$o_k^p = \frac{\sum_{k'=1}^{c} n_{pk'}\bar{t}_k^{k'}}{\sum_{k'=1}^{c} n_{pk'}} \tag{13}$$

which, for a 1-from-c target coding scheme reduces to $P(\boldsymbol{k}|\boldsymbol{p}) = n_{pk}/\sum_{k=1}^{c} n_{pk}$, the probability of the class $\boldsymbol{k}$ occurring given input pattern $\boldsymbol{p}$ where the probability density functions

[1] Actually a more general sum rule holds: if a set of target vectors satisfy several linear constraints simultaneously, then so will the general network output.

have been approximated by the histograms of the data distribution. For a 1–from–c output coding scheme, the best network output (even assuming that the network structure was capable of capturing this relation in principle) attempts a sample estimate to the Bayes probabilities. This result is a consequence of the coding scheme and least mean square minimisation. It is *not* a special property of networks. However, it does justify the slightly unorthodox approach of setting up a network to perform posterior probability density estimation based on an assumption of statistical independence, which we now outline.

Previous work [18] on the medical data discussed in the next section obtained the best *overall classification* performance using a statistical independence model. This model assumes that the probability of a pattern, $\boldsymbol{p}$ occurring in the i–th class i is proportional to the product of the estimates of the marginal densities. Since the data is categorical (the value of each channel, r, representing one of the R features of the input pattern is allowed only one value y_r of a finite number of Y_r values), the probability of the pattern given the class is given by

$$P(\boldsymbol{p}|i) \propto \left\{ \prod_{r=1}^{R} \frac{n_i(y_r) + 1/Y_r}{N_i(r) + 1} \right\}^B \tag{14}$$

where the value y_r in channel r occurs $n_i(y_r)$ times in class i and $N_i(r)$ is the total number of times any non zero value occurs in channel r for class i. Note that $N_i(r)$ is not equal to the number of patterns in class i due to missing data. B is a smoothing factor [13]. By Bayes theorem, the probability of the class given the pattern is determined by multiplying the above probability by the prior probability of the class.

A network which approximates this probabilistic model may be constructed as follows. Since there are Y_r levels for each feature, we have $n = \sum_{r=1}^{R} Y_r$ inputs in total, with the first Y_1 inputs corresponding to feature 1, the next Y_2 to feature 2 and so on. The input coding for the r–th group is '1' in the position corresponding to the feature value y_r and zero otherwise. There is a hidden unit for each class, and there are c linear output nodes. Let s denote the position in the input coding corresponding to the y_r–th value of variable r. Then the weight from input unit s to hidden unit i is given by

$$weight(s,i) = B \log \left[\frac{n_i(y_r) + 1/Y_r}{N_i(r) + 1} \right] \tag{15}$$

and there is a bias term denoted by $\log A$ where A has yet to be specified. It is clear that using these weights, the scalar product between an input pattern, $\boldsymbol{p}$ and these weights gives

$$x = \sum_{s=1}^{n} weight(s,i) \equiv \log P(\boldsymbol{p}|i) \tag{16}$$

where the last equality follows from (14). Thus for a logistic transfer function the output of node i may be expressed as

$$f_i(\boldsymbol{p}) = 1/[1 + \exp -(\log A + \log P(\boldsymbol{p}|i))] \equiv 1/[1 + AP(\boldsymbol{p}|i)] \tag{17}$$

Now the easiest way to ensure that the output of the network approximates the probability of the class given the pattern is to have a single direct connection from each hidden node to the output node with a weight given by $-p_i/A$ (all other connection strengths set to zero) and a bias term given by p_i/A where p_i is the prior probability associated with class

i. Then, if A is sufficiently small we may approximate the output of the hidden node by $f_i \approx 1 - AP(\boldsymbol{p}|i)$ which subsequently gives the output of the network as an approximation to the probability of the class given the pattern. This procedure of initialising a network to give numbers which mimic posterior probabilities at the outputs assuming independence will be used in the discussion of the medical data. Note that this interesting relationship is a consequence of an appropriate coding of the input data.

There are additional interesting relationships between optimally trained networks and Bayesian inference [22, 8, 15] in the limit of an infinite amount of data. In particular the solution of the parameters which minimises the sum square error (3) using 1-from-c targets has minimum variance from the optimal Bayes discriminant function, and making a decision on the basis of the nearest target vector to an output is the same as picking the class with largest output. This approximates the optimum Bayes solution for minimum error: i.e. maximises the likelihood of the class given the pattern. If the prototype target matrix is the 'equal cost' matrix, it costs nothing to classify a pattern correctly (zero diagonal) and always costs unity for an incorrect classification (unity off-diagonal), then picking the nearest target vector to an output approximates the Bayes decision rule for minimum risk. For an arbitrary loss matrix, such a useful relationship does not hold.

The point is made that appropriately trained networks can approximate traditional statistical inference. How this is achieved depends on the combination of input and output coding and error weighting. The specific choices are determined by prior knowledge of the data, and the effects of these choices have been outlined above in terms of feature extraction criteria. We now illustrate some of the previous results by considering a specific real pattern processing problem taken from the medical sciences.

3 A Medical Prognosis Problem.

3.1 Discussion of the data.

The problem[2] is to attempt to predict the future outcome (prognosis) of patients suffering from severe head injury on the basis of data collected shortly after injury. Given that patients suffering from this type of injury tend either to die very soon after injury, or not to progress significantly after a period of the order of six months, this problem may be considered as a static pattern classification task, *i.e. given* data from a patient, what class of recovery is he likely to be in after a period of six months. For the purpose of this study, only three classes are considered. A class is chosen depending upon whether after six months the patient

Class 1: is dead or vegetative

Class 2: has severe disability

Class 3: has moderate disability, or shows good recovery.

The information on which such a decision is based comes from a limited amount of data which may be obtained from a coma victim such as the patient's age, pupils' sensitivity to light, motor response in all four limbs, change in neurological function over 24 hours, and eye movements. For this experiment, the following six feature variables, or indicants were used, along with their coding schemes:

Feature	Coding Scheme
AGE (in decades)	$1 \leftarrow 0-9, 2 \leftarrow 10-19, \ldots, 8 \leftarrow 70+$
EMV (Glasgow Coma Sum)	$0 \leftarrow missing, 1 \leftarrow 3, 2 \leftarrow 4, \ldots, 6 \leftarrow 8, 7 \leftarrow 9-15$
MRP (Motor Response Pattern)	$0 \leftarrow missing, 1 \leftarrow bad, \ldots, 7 \leftarrow good$
CHANGE	$0 \leftarrow missing, 1 \leftarrow bad, \ldots, 3 \leftarrow good$
EYE INDICANT	$0 \leftarrow missing, 1 \leftarrow bad, \ldots, 3 \leftarrow good$
PUPILS	$0 \leftarrow missing, 1 \leftarrow nonreacting, 2 \leftarrow reacting$

These six features constitute a six dimensional feature vector. Each component is either binary or ordered and thus may be viewed as quantised levels of continuous variables, or as discrete binary values where the precise ordering of the values is ignored. Both input coding schemes will be considered subsequently. For the purposes of our experiments the data was scaled so that each feature was in the range $[0, 1]$ inclusive.

The *EMV* score is a coding for the sum of three separate indicants: the *E*ye opening response to stimulation graded 1 to 4 (normal); the *M*otor response of the best limb to stimulation graded 1 to 6 (normal); and the *V*erbal response to stimulation graded 1 to 5 (normal).

The information constituting the data used in this experiment was collected prospectively from 1000 patients who had been in coma for at least six hours. The data collection study lasted over a period of 8 years beginning in 1968, with data obtained primarily by clinicians at the Institute of Neurological Sciences, Glasgow, and also from two Netherlands

[2] See [18] for an excellent presentation of this problem and data.

centres at Rotterdam and Groningen. Los Angeles subsequently provided additional data. The clinicians involved in this data collection decided that the above indicants were suitable to be recorded reliably by different clinicians in different circumstances in different countries (see [18] and references therein for details). Nevertheless, it was still not always possible to obtain a full set of test results on each patient. The occurrence of '0' in any feature accounts for those instances when that particular piece of information is missing. This is an added 'interesting feature' of this data set. We have performed experiments where the missing data was treated as part of the feature vector, although they will not be discussed here. The question of what to do with incomplete data sets is an interesting statistical problem, and to what extent such an anomaly affects network performance has still to be properly analysed. The 1000 feature vectors were randomly split into training and test sets representing 500 patients in each group [3]. This division of the data produced differences in the distribution of patterns in the classes between the training and test sets as Table 1 shows.

	Frequencies	
	Training Set	*Test Set*
Class 1	259	250
Class 2	52	48
Class 3	189	202

Table 1: Pattern distribution between classes in the training and test sets.

3.2 Problems with the data set.

This is a particularly interesting data set to work with for a variety of reasons, most of which create difficulties for many techniques. However, the problems occurring in this data set reflect trends which are observed in many distinct real-world pattern processing tasks, and therefore it is valid to emphasise the more obvious features.

- *Size.* The first point to note is that it constitutes an unusually large data set in relation to the input dimensionality and the number of classes. This is a good aspect of this data set and particularly rare in medical applications. Very often in applications of adaptive networks the size of the training set in terms of the number of constraints imposed is far too small when compared to the dimensionality of the problem reflected in the number of adjustable parameters in the network. This leads to an undesirable trend of a network overtraining on a specific data set at the expense of good generalisation performance: there are not enough constraints to estimate the adjustable parameters reliably and the problem is underspecified leading to a network solution with zero error on the training set. It is almost always possible to obtain zero error on a finite training set by increasing the complexity of the network, but this is not interesting.

- *Features:* The observations constituting the feature vector of an individual patient are a combination of specific, well-categorised variables (e.g. AGE) and interdependent subjective variables (e.g. CHANGE). An arbitrary coding scheme has been

[3] We would like to thank Dr. Murray for supplying us with the same data, including the division into train and test sets, as used in [18]

introduced to quantify the degree of response for several observations. It is not clear how critical this choice of coding is for classification. A significant amount of prior knowledge and nonlinear feature extraction has already been performed on the raw data to produce the feature vectors which are to be presented to a classifier in a manner which is almost certainly suboptimal. This is a common procedure for almost all real pattern processing tasks, in that a significant amount of manual nonlinear feature extraction is performed generally prior to automatic pattern discrimination.

- *Missing data:* Previously, it was mentioned that in spite of attention to creating this database, occasionally some observations were not made. Thus, several vectors have at least one of the features totally absent. This is a major aspect of this data since 206 patterns out of 500 training patterns and 199 out of 500 test patterns have *at least* one observation missing. The correct treatment of missing data is an important area of study by itself [17, 12]. Although it is possible to initialise networks to accommodate missing data by working with the class–conditional distributions and assuming statistical independence (this will be used later), a trained network loses this independence assumption. In spite of the importance of the correct procedure for dealing with missing data, its detailed discussion would detract from the main aims of this paper. Therefore, for simplicity the (suboptimal) method adopted in the experiments is to replace the missing data in the training set by values selected randomly according to the distributions as determined from the observed components, and in the test set by the means estimated from the training set.

- *Uneven class distribution:* Another major aspect of this data set is the fact that the distribution of patterns between the classes is strongly nonuniform (Table 1). Since least mean square error training of networks forces the hidden unit space to maximise a feature selection criterion which employs a (squared) weighted between class covariance matrix, the optimum network solution tends to ignore the least represented class (class 2 in this case). This is a generic result which could lead to undesirable consequences in problems where it was required that the *classes* were uniformly weighted. In addition, there is a slight unevenness between the train and test set distributions, but this would not be expected to be significant in this problem. However it is usually assumed in network training schemes that the distribution of patterns in the training set is *representative* of the distribution of the patterns in the test set. In problems where it is very difficult to obtain information on one particular class, but it is very easy to obtain data on the remaining classes, and it is desirable to optimise the correct classification of the difficult class, one has to compensate for the *expected* error by exploiting knowledge of the *a priori* probabilities.

- *Uneven class importance:* This is related to the previous comment. Most medical prognosis/diagnostic problems have an uneven *importance* attached to each class. It is more important to diagnose serious ailments correctly than to diagnose psychosomatic problems correctly. If the data set is loaded against diagnosing the serious complaints correctly (by not having sufficient patterns in that class in the training set) it is important to increase the priors associated with that class artificially. In this particular case, it is most important to obtain a correct prognosis on those patients in class 2, since they are the ones requiring long term rehabilitation. Although we can assume that the distribution of patterns in the training set is representative of the patterns occurring in practice, the fact that class 2 has the least membership implies that network training will need to be biased by exploiting the priors in order to increase the recognition of patterns in that class.

- *Cross-Class Penalties:* In addition to class-conditional priors, there is often an associated cross-class penalty index. There is a larger penalty associated with diagnosing a serious ailment as psychosomatic, rather than the other way around. This is an effect which cannot be compensated for by altering the priors, and an explicit *cost matrix* has to be used to modify the target prototype vectors. For this particular data set, a cost matrix has been devised on subjective grounds by neurosurgeons involved with this study. This matrix reflects how serious the neurosurgeons would judge the different misclassification errors. The matrix has the specific form

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \begin{array}{ccc} \multicolumn{3}{c}{\textit{Predicted}} \end{array} \\ \begin{pmatrix} 0 & 10 & 75 \\ 10 & 0 & 90 \\ 750 & 100 & 0 \end{pmatrix} \end{array} \tag{18}$$

  For instance, it is ten times more serious (clinically) to make a prognosis that a patient who recovers (class 3) is predicted to die (class 1) rather than predict incorrectly that a patient who would die should recover. Later the effects of exploiting this cost matrix on a network's classification performance will be illustrated by modifying the targets as outlined in the previous section.

- *Ambiguous data:* Another aspect of this data set is that the three classes are extremely overlapping: the three classes are not easily separable into distinct clusters. Indeed, a few percent of the patterns occur more than once but with correspondingly different target vectors (22 distinct vectors in the training set and 24 in the test set belong to more than one distinct class). This implies ambiguous data leading to intrinsically inconsistent training (technically, the actual mapping implied by the data is not a function at all, and networks are only capable of functional interpolation). The best performance one could expect in such circumstances is to reproduce the likelihood of the class given the pattern (see section 2.2).

- *Multi-centre:* Finally, although the collection of this data represents an admirable collaboration between establishments in different countries, the geographic, social and cultural differences in the behaviour of the populations should lead to biases in the data collected from each centre. The *type* and severity of head injury sustained is likely to be different from country to country, for instance because of the different regulations governing the wearing of crash helmets. As the data was pooled, these trends have been dissipated in the data set, potentially leading to anomalies. Such anomalies are likely to arise in other real pattern processing problems when data is collected over a period of time, at different locations, by different people using different experimental techniques or different monitoring equipment etc.

These points illustrate that this is a particularly interesting data set, not because it is too difficult to obtain good performance by network techniques, but because there are several aspects in the data set which are likely to be reflected in almost any real pattern processing problem. The first rule of applying adaptive network techniques to a given application is, before anything else *study and understand the limitations of the data*. This emphasises the need for baseline performance figures on a given data set, for instance by other traditional pattern classifiers.

3.3 Standard Classification Results

As with all other techniques of statistical pattern recognition, it is important to assess the capabilities of adaptive networks against a baseline performance expectation. In this paper the expectation is provided by the classification abilities of a range of standard techniques applied to the same medical data set. The specific methods used are: Distance to class mean (DCM), Gaussian classifier (GC), nearest neighbour (NN), K-nearest neighbour (KNN), Optimum Linear Transformation (OLT) and a Statistical Independence (SI) model. A brief discussion of each of these methods is presented in Appendix C. For a different range of techniques applied to this data set see [18]. The statistical independence model is the same as employed in [18] and has been included here to illustrate the reproducibility of the results obtained in that paper. For instance, using the original data (unscaled and unsubstituted), the best result obtained *on the test set* using the statistical independence model with $B = 1$ returned 75.2% correctly classified patterns (with 74.6% on the training set). This is the same as reported in [18]. However, the best result on the training set gave worse results on the test set. Thus, these figures are biased estimates of the statistical independence model performance. It is interesting to note that in this solution, the statistical independence model classified correctly only 3 patterns out of 48 in class 2 on the test set (compared to 203 out of 250 for class 1 and 170 out of 202 from class 3).

Figure 1 depicts the classification performance of each of the standard techniques on the zero substituted data set. Note that the nearest neighbour classifier does not achieve 100% on the training set as would generally be expected. This is due to the inconsistent nature of the data leading to unresolvable ties for correct classification. Interestingly (and anomalously) the Optimum Linear Transformation performs well on this data. This technique has the closest relationship to the network models, although networks may be initialised according to the statistical independence model (as has already been discussed) and to a nearest distance-to-class-mean classifier [2]. Also note that the statistical independence model which gave the best test set results overall only ranked fourth on the training set.

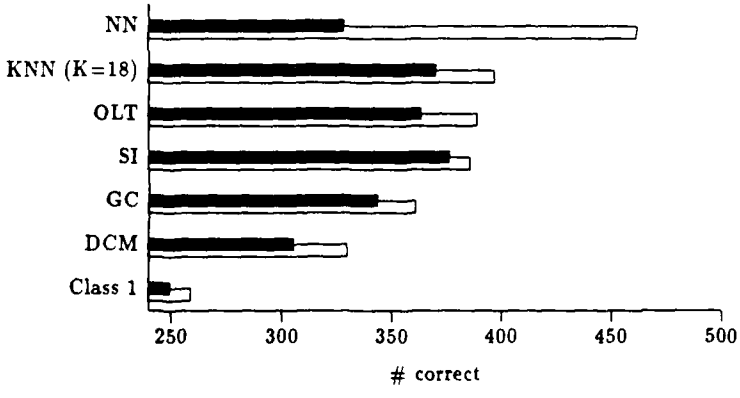


Figure 1: Graph depicting the total number of correctly classified patterns on the training set (open bars) and the test set (solid bars) for a range of classification techniques. The Class 1 bars denote the performance obtained by classifying everything as class 1.

However, these performance figures do not convey the accuracy with which each class was separately classified. This information may be observed in the *confusion matrices* of each classifier. Figure 2 and Figure 3 show the confusion matrices produced on the training and test sets by two of the best techniques the optimum linear transformation model and the K–nearest neighbour model. Note that the criterion for 'best' is simply the one which gave the maximum number correct overall on the test set. This is *not* the criterion that will be used to assess network performance. Neither is it a particularly useful criterion for this medical data problem since it takes no account of costs. This is reflected in the fact that the 'better' techniques achieve their superior performance at the expense of totally misclassifying class two, as is evident by examining the confusion matrices. This same trend will be observed in the naïve application of network techniques where minimising the residual error is achieved by ignoring the class least represented. Figure 4 shows the confusion matrices as obtained by the statistical independence model on this data. Again, there is a dominant trend to misclassify class 2 patterns at the expense of patterns from classes 1 and 3. In each figure the 'average loss' is displayed below each confusion matrix. This loss is obtained by multiplying the confusion matrix by the clinicians' loss matrix element by element, summing and dividing by the total number of examples.

$$\begin{array}{c} \\ Actual \end{array} \begin{array}{c} Predicted \\ \begin{pmatrix} 220 & 0 & 39 \\ 27 & 0 & 25 \\ 20 & 0 & 169 \end{pmatrix} \\ Train \end{array} \qquad \begin{array}{c} \\ Actual \end{array} \begin{array}{c} Predicted \\ \begin{pmatrix} 199 & 0 & 51 \\ 21 & 0 & 27 \\ 37 & 0 & 165 \end{pmatrix} \\ Test \end{array}$$

$Average\ loss = 40.89$ $\qquad$ $Average\ loss = 68.43$

Figure 2: The confusion matrices of the Optimum Linear Transformation classifier.

$$\begin{array}{c} \\ Actual \end{array} \begin{array}{c} Predicted \\ \begin{pmatrix} 224 & 0 & 35 \\ 23 & 0 & 29 \\ 16 & 0 & 173 \end{pmatrix} \\ Train \end{array} \qquad \begin{array}{c} \\ Actual \end{array} \begin{array}{c} Predicted \\ \begin{pmatrix} 197 & 0 & 53 \\ 20 & 0 & 28 \\ 28 & 0 & 174 \end{pmatrix} \\ Test \end{array}$$

$Average\ loss = 34.93$ $\qquad$ $Average\ loss = 55.39$

Figure 3: The confusion matrices produced by the best (as determined on the test set!) KNN model (using $K = 18$).

3.4 Network Results.

The adaptive network used in the simulations has a standard feed forward architecture with logistic nonlinearities at the hidden nodes but with linear transfer functions at the output nodes. However the training method employed is not standard and consequently

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 217 & 2 & 40 \\ 24 & 4 & 24 \\ 21 & 3 & 165 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 205 & 5 & 40 \\ 23 & 4 & 21 \\ 25 & 9 & 168 \end{pmatrix} \\ \textit{Test} \end{array}$$

Average loss = 42.94 *Average loss* = 49.64

Figure 4: The confusion matrices produced by the best (as determined on the test set!) Statistical Independence model (using $B = 1$).

warrants a brief discussion. The function evaluated by such a network is continuous and differentiable. Since the residual error to be minimised is analytic, any standard nonlinear least mean squares optimisation technique which employs knowledge of a function and its first derivative may be used. Since the total number of weights in a network for this medical problem is likely to be small (< 100) we have previously found [21, 20] that a quasi-Newton method is an efficient procedure for obtaining a local minimum of the error function. In addition it is natural to consider the various layers of a feed forward network as evolving on different time scales: the final layer weights adapting rapidly to the varying first layer weights. In this sense the final layer weights are slaved to the behaviour of the first layer weights. Since the output nodes are linear, *given* the set of patterns at the output of the hidden layer the final layer weights may be obtained *instantaneously* (on the time scale of the previous layer) by a pseudo-inverse method. Thus, the training method adopted for this network is as follows. The network is initialised with random weights in an appropriate range. The first layer weights are incrementally varied in a direction determined by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton nonlinear optimisation scheme [6]. As the (slow) search for a local minimum continues in the weight space spanned by the first layer weights, the final layer weights are adapted instantly to any change so as to always remain in a (global) minimum in the weight space spanned by the final layer weights. This ensures that the network as a whole is guaranteed to be in a local minimum once the first layer weights have found a locally optimum position. We have already seen that this optimum position corresponds to a nonlinear transformation which maximises a specific feature selection criterion. The multiple time scale description implies that feature selection emerges slowly compared to the (linear) automatic classification part of the network.

The choice of network complexity also warrants a brief discussion. As the number of hidden nodes increases, it is possible to fit the training data more and more accurately. Beyond a certain number, the noise on the data as well as the hidden structure of the data is also being fitted. This is overtraining and the generalisation error will begin to fluctuate. For this problem, it was decided (on the criterion of minimum generalisation error) that three hidden units was sufficient to model the structure in the data adequately. The specific choice is not crucial for the purposes of this paper, as the issues we are concerned with are *given* a network, what is the effect on the feature extraction mechanism and subsequent classification of changing the priors and the targets. Thus, we choose to fix the number of hidden units for each set of experiments to allow comparisons to be made across experiments.

3.4.1 The 'standard' result.

The standard experiment consists of using 1-from-c target coding and no extra priors ($a_p = 1 \; \forall \; p$). From 100 different random weight starts, the classification results of the trained network with smallest error *on the training set* are presented in Figure 5. Note that, as predicted, the network has achieved the minimum error solution by totally ignoring the class with smallest membership. The figures also present the *average loss* associated with the experiment. The significance of the average loss will be apparent only when we incorporate the cost matrix as part of the training phase of the networks.

$$\begin{array}{c} \\ Actual \end{array} \begin{array}{c} Predicted \\ \begin{pmatrix} 229 & 0 & 30 \\ 21 & 0 & 31 \\ 11 & 0 & 178 \end{pmatrix} \\ Train \end{array} \qquad \begin{array}{c} \\ Actual \end{array} \begin{array}{c} Predicted \\ \begin{pmatrix} 195 & 0 & 55 \\ 25 & 0 & 23 \\ 31 & 0 & 171 \end{pmatrix} \\ Test \end{array}$$

$Average\ loss = 27.0$ $\qquad$ $Average\ loss = 59.39$

Figure 5: The confusion matrices produced by the best (as determined by the smallest training error) 'standard' network (6-3-3).

3.4.2 1-from-c targets, equal priors

In this experiment, the prototype target matrix is the identity matrix, but the priors are adjusted to be equal on the test set (thus $d_p \equiv P(i)/(P_i) \propto 1/P_i$ if the p-th pattern is in the i-th class). Strictly speaking, this compensation is not necessary for this data set since the distribution of patterns between the classes is approximately the same between the train and the test sets. Nevertheless it does reveal how the relative importance of classifying correctly class 2 can be increased. Figure 6 shows the misclassification matrices obtained from the network solution with smallest training error. It is noted that the network is recognising correctly several patterns from class 2, although there is a high misclassification rate from class 1 into class 2. The training error is slightly worse and the *average loss* is slightly worse than the standard case.

3.4.3 Targets taken from the loss matrix, no priors.

In this experiment, the prototype target matrix is the loss matrix subjectively obtained by the group of clinicians involved in the study. The priors are not compensated for (thus $d_p \equiv$ (priors on the test set)/(priors on the training set) $= 1 \; \forall \; p$). Figure 7 shows the misclassification matrices obtained from the network solution with smallest training error. Now, as well as raising the recognition performance of the minority class, the average loss is significantly reduced. This has been at the expense of overall recognition performance. However it does indicate the effects of minimising a cost function which incorporates misclassification costs.

	Predicted		
Actual	196	22	41
	7	18	27
	12	3	174

Train

Average loss = 31.54

	Predicted		
Actual	165	24	61
	10	9	29
	9	18	175

Test

Average loss = 32.15

Figure 6: The confusion matrices produced by the best (as determined by the smallest training error) network with equalising the priors.

	Predicted		
Actual	137	95	27
	7	21	24
	2	15	172

Train

Average loss = 16.41

	Predicted		
Actual	126	79	45
	7	15	26
	6	33	163

Test

Average loss = 28.75

Figure 7: The confusion matrices produced by the b t (as determined by the smallest training error) network incorporating the misclassification costs in the training process.

3.4.4 Targets taken from the loss matrix, equal priors.

In this experiment, again the prototype target matrix is the loss matrix. In addition the priors are equalised on the test set (thus $d_p \propto 1/P_t$). Figure 8 shows the misclassification matrices for this experiment (choosing the smallest training error over 100 separate trials). This network has a predominant tendency to misclassify class 1 patterns as class 2. This is an inappropriately trained network since equalising the priors and incorporating misclassification costs into the training unfairly biases against the high cost class since the test and train distributions are approximately equal. However, Figure 8 does make the point that class 3 patterns are not very likely to be classified incorrectly as class 1 (i.e. it is important not to make the prognosis that a patient who will recover is likely to die — particularly if this decision affects the operation of a life-support system on that patient). Alternatively, it is comparatively easy to misclassify patterns from class 1 as from class 3 (it is not too serious to predict a patient will recover if he actually dies, as far as the health of the patient is concerned).

3.4.5 Class-weighted targets, no priors

In this experiment, the target vector for a training pattern is $1/\sqrt{P_i}$ if the pattern is in class i, and zero otherwise. We have seen previously that this forces the feature selection criterion to employ the *conventional* between class covariance matrix which does not bias

	Predicted		
Actual	71	147	41
	5	18	29
	0	5	184

Train

Average loss = 15.41

	Predicted		
Actual	57	126	67
	6	10	32
	7	20	175

Test

Average loss = 32.95

Figure 8: The confusion matrices produced by the best (as determined by the smallest training error) network incorporating the misclassification costs and compensating for the priors.

so heavily in favour of the most represented class. The decision regions are still influenced by the numbers in each class though. It is seen in Figure 9 that a small proportion of class 2 are now recognised correctly but the minimum error solution is still predominantly to misclassify class 1 as class 3 and class 3 as class 1.

	Predicted		
Actual	221	1	37
	21	8	23
	22	0	167

Train

Average loss = 42.13

	Predicted		
Actual	201	3	46
	19	2	27
	33	4	165

Test

Average loss = 62.50

Figure 9: The confusion matrices produced by the best (as determined by the smallest training error) network with class-weighted targets, not compensating for the priors.

3.5 Network results using binary input coding.

The previous results have employed a coding scheme for the input patterns which assumed quantised, ordered features leading to a six dimensional continuous input. An obvious alternative input coding scheme is to assume that the value of each variable is independent, leading to a 30 dimensional binary input pattern for this medical data. An advantage of this input coding scheme for networks is that the structure may be mapped onto the statistical independence model as discussed in section 2.2. This allows a network to be initialised with a 'sensible' weight configuration prior to optimisation. The following results present the various confusion matrices using the binary input coding and the various combinations of error weighting and cost matrices.

3.5.1 The 'standard' result.

Employing 1-from-c targets, $d_p = 1 \; \forall \; p$ and a $30 - 3 - 3$ network, the best (i.e. the one returning the smallest residual training error) experiment from 100 different initial random weight starts gave the confusion matrix in Figure 10. Generally, the performance is better than the corresponding result using continuous input coding. Specifically, using binary input coding gave more correct on training, the normalised training error was slightly less, the average loss was less and it also succeeded in correctly classifying many of the class 2 patterns. However the average loss on test is significantly higher.

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 214 & 10 & 35 \\ 8 & 24 & 20 \\ 7 & 6 & 176 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 162 & 28 & 60 \\ 15 & 12 & 21 \\ 23 & 22 & 157 \end{pmatrix} \\ \textit{Test} \end{array}$$

$$\textit{Average loss} = 20.91 \qquad \textit{Average loss} = 52.54$$

Figure 10: The confusion matrices produced by the best (as determined by the smallest training error) network on the standard network using binary input coding.

3.5.2 The 'standard' network initialised as a statistical independence model.

This experiment used the same input and output coding as the previous case. In this example, the network weights were initially configured to reproduce the results as obtained by a statistical independence model. This does not correspond to a minimum of the network's residual error and so when optimisation proceeds, the weights adapt away from the preset values to find a smaller error configuration. The confusion matrices are depicted in Figure 11. In terms of training error, this configured network does not achieve as low a training and test error as the best of the random configurations presented in Figure 10. It does return slightly better classification performance figures. However, the differences are so small that one may take Figure 11 as indicative that it is useful to exploit prior knowledge to initialise a network in a particular configuration prior to the optimisation scheme. This is particularly so if the optimisation experiments are likely to be extensive and time consuming to find a suitable initial random start configuration for the weights

3.5.3 1-from-c targets, equal priors

The prototype target matrix is the identity matrix, and the priors are adjusted to be equal on the test set. Comparing the results in Figure 12 with those of the corresponding continuous case (see Figure 6), the training performance is better in this experiment but the generalisation is worse.

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 223 & 1 & 35 \\ 19 & 12 & 21 \\ 8 & 0 & 181 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 184 & 6 & 60 \\ 20 & 4 & 24 \\ 32 & 7 & 163 \end{pmatrix} \\ \textit{Test} \end{array}$$

$$\textit{Average loss} = 21.43 \qquad \textit{Average loss} = 63.22$$

Figure 11: The confusion matrices produced by the standard network initialised as a statistical independence model.

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 178 & 56 & 25 \\ 0 & 50 & 2 \\ 4 & 51 & 134 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 140 & 70 & 40 \\ 10 & 26 & 12 \\ 10 & 77 & 115 \end{pmatrix} \\ \textit{Test} \end{array}$$

$$\textit{Average loss} = 21.43 \qquad \textit{Average loss} = 40.16$$

Figure 12: The confusion matrices produced by the network with binary inputs, the identity as prototype target vectors and the priors adjusted to equalise expectation on the test set.

3.5.4 Targets taken from the loss matrix, no priors.

The priors are not compensated for, and the prototype target vectors are taken from the clinicians' cost matrix. The confusion matrices of the best experiment out of 100 random starts are depicted in Figure 13 which should be compared with Figure 7. Because the costs are being incorporated into the training phase there is a tendency to misclassify class 1 patterns significantly. This gives a small overall loss on the training and test sets, but also returns poor overall classification results. These results are not as good as those using continuous features.

3.5.5 Targets taken from the loss matrix, equal priors.

The priors are compensated for so that the test set expectation is equal, and the prototype target vectors are taken from the clinicians' cost matrix. The confusion matrices of the best experiment out of 100 random starts are depicted in Figure 14. The corresponding results for the continuous input coding scheme are shown in Figure 8. The effect on class 1 classification of equalising the priors is even more drastic. All class 1 patterns are incorrectly classified in either the train or test sets. However, the overall loss is significantly small in both instances.

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 109 & 107 & 43 \\ 6 & 21 & 25 \\ 0 & 37 & 152 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 90 & 105 & 55 \\ 4 & 21 & 23 \\ 4 & 40 & 158 \end{pmatrix} \\ \textit{Test} \end{array}$$

Average loss = 20.61 *Average loss* = 28.51

Figure 13: The confusion matrices produced by the network with binary inputs using the clinicians' cost matrix as prototype target vectors and not compensating for the priors.

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 0 & 230 & 20 \\ 0 & 44 & 8 \\ 0 & 9 & 158 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 0 & 190 & 60 \\ 0 & 25 & 23 \\ 0 & 49 & 153 \end{pmatrix} \\ \textit{Test} \end{array}$$

Average loss = 12.19 *Average loss* = 26.74

Figure 14: The confusion matrices produced by the network with binary inputs using the clinicians' cost matrix as prototype target vectors as well as equalising for the priors on the test set.

3.5.6 Class-weighted targets, no priors

The target vector for a training pattern is $1/\sqrt{P_i}$ if the pattern is in class i, and zero otherwise. The priors are not compensated for. Since the costs are not incorporated into the training, this experiment misclassifies class 2 to achieve a minimum error as observed in the results of Figure 15. These results are slightly better than those in the corresponding experiment using an ordered input coding (Figure 9).

3.5.7 Targets taken from the *scaled* loss matrix, no priors.

It is interesting to consider the effects on training performance that scaling the cost matrix has. In principle the absolute values of the elements constituting the cost matrix do not affect the network's performance. It is only the *relative* values which are important. However in practice the absolute values of the targets influence the iterative optimisation scheme due to the step lengths taken. The results in Figure 16 depict an experiment where the priors have not been compensated for and the target coding is taken from a scaled version of the clinicians' cost matrix where each loss is scaled down by a factor of 100. It is interesting to note that the training error in this case is significantly less than the corresponding case working with the cost matrix directly. In addition, it took many more iterations to reach the minimum in the scaled case indicating that the minimum

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 212 & 0 & 47 \\ 23 & 0 & 29 \\ 10 & 0 & 179 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 180 & 0 & 70 \\ 18 & 0 & 30 \\ 21 & 0 & 181 \end{pmatrix} \\ \textit{Test} \end{array}$$

$\textit{Average loss} = 27.73$ $\textit{Average loss} = 47.76$

Figure 15: The confusion matrices produced by the network with binary inputs. The prototype target vectors are class weighted by the square root of the numbers in that class and the priors are not compensated for.

found without scaling the cost matrix was a poor local minimum. The effect of the better minimum is to find a solution with a much smaller average cost (10.74 compared to 20.61) and a much better overall classification performance (372 correct on training and 315 correct on test compared to 282 and 269 correct respectively *in the unscaled case*). The normalised test error was slightly worse than the unscaled case.

$$\begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 185 & 54 & 20 \\ 22 & 21 & 9 \\ 0 & 23 & 166 \end{pmatrix} \\ \textit{Train} \end{array} \qquad \begin{array}{c} \\ \textit{Actual} \end{array} \begin{array}{c} \textit{Predicted} \\ \begin{pmatrix} 152 & 55 & 43 \\ 12 & 22 & 14 \\ 7 & 54 & 141 \end{pmatrix} \\ \textit{Test} \end{array}$$

$\textit{Average loss} = 10.74$ $\textit{Average loss} = 31.61$

Figure 16: The confusion matrices produced by the network with binary inputs using the clinicians' cost matrix *scaled down* by a factor of 100 as prototype target vectors. The priors were not compensated for.

4 Discussion

The intention of this paper has been to expand and explore relationships between adaptive networks trained 'optimally' and feature extraction criteria, and the relevance of these links in the context of a real-world classification problem (medical prognosis). In particular it was stressed that networks perform feature extraction and classification simultaneously, and the specific feature extraction criterion depends on the classification. In addition, it was necessary to emphasise the significance of input/output coding schemes for network training and what this implied for the interpretation of actual network outputs in a probabilistic context. This led to a novel scheme where, under appropriate circumstances, the set of network parameters may be initialised so that the network performs the same task as a standard statistical independence model. Subsequent optimisation adapted these parameters to reduce the error, but at the expense of classification for the medical prognosis problem considered. Numerical comparisons were made between network results and a range of standard pattern classification techniques on the medical data. Finally, a comparison of results obtained by a fixed network structure but utilising various combinations of coding schemes and error weightings was presented.

The main theme of the paper has been to discuss that group of classification problem where the relative importance of the classes follows a different distribution to their prior occurrence in the training set. In this general theme, two connected issues were discussed. In one case, the distribution of patterns amongst the various classes may be the same between the training and the test sets, but there may be a nonuniform penalty associated with misclassification. This is particularly so for the medical data used, since it is more important to try and diagnose that class of patient who is likely to survive but will require long term care. With prior knowledge of the loss matrix, the distribution of patterns in recognition mode may be varied by incorporating the loss matrix in the training process itself. This redistribution of patterns is produced by a feature extraction transformation which attempts to allow a decision to be made which relates to minimising the overall expected loss. This redistribution is evident in the confusion matrices. The anomaly with the second group of problems is that the expected occurrence of classes in the test set may be different to the distribution of patterns between the classes in the available training set. Since minimising the uniform error maximises a feature extraction criterion which weights in favour of those classes most represented, the error has to be weighted according to the *expected* probability distribution of the classes *in operation*. Using appropriate error weighting factors, minimising the error distorts the effective between class covariance matrix of the nonlinearly transformed patterns so as to produce classification results which reflect the expected class distribution.

For certain choices of coding scheme it was shown that the optimum network solution attempted to reproduce the Bayes decision vector and choosing-the-nearest decision rule corresponded to maximising the likelihood, or minimising the expected risk in a Bayesian sense. When one is not in this optimum situation, the outputs of the model network do not reflect probabilities (although they may still be employed as calculational devices in any subsequent decision-making process) and it is not obvious what is the best decision rule to apply. For consistency with the least-mean square approach, we have chosen to always use the pick-the-closest decision rule. The precise interpretation of the outputs of an approximate network model and how these outputs should be employed in a decision rule is an area of research to be continued.

A discussion of the peculiar network optimisation strategy that is preferred in our simulations was given. This introduces the concept of the different dynamics of the various weights, depending on the layer. Specifically, the weights performing the feature extraction transformation evolve on a slow time scale compared to the speed of adaptation of the final layer weights performing the classification.

Having established a sound framework for the network methodology, the techniques were applied to a specific real world pattern processing problem: medical prognosis of head-injured coma patients. This is a particularly useful data set with many problems and advantages as discussed in the text. In particular, a standard network trained on this data minimised the overall error by misclassifying most of the patterns in class 2, in common with many other standard pattern recognition techniques. It was numerically demonstrated that the distribution of predicted classes in the test set may be manipulated by an appropriate choice of target coding and error weighting (although for this data set the distribution of patterns between classes is approximately uniform between the train and test sets). The numerical results corroborate the theoretical expectations given in the earlier sections, in that incorporating the loss matrix allows a solution to be obtained with a smaller overall risk by forcing a redistribution of elements in the confusion matrix. This smaller risk is achieved generally at the expense of worse overall classification accuracy.

It was not our intention to find the best possible solution for medical prognosis of head injured patients. In fact the technique with best classification performance remains the standard statistical independence model on the range of classifiers that we have considered, although the optimum linear transformation gave better performance in terms of the overall loss. We have not examined different types of adaptive networks or methods to determine the optimum network topology for this particular data set. Neither have we managed to solve the problem of what is the best strategy of treating missing data which is optimally matched to the network model. All these must be considered to be open problems.

A The Generalised Network Feature Extraction Criterion

This appendix shows how minimising the error at the output of a network with linear output units is equivalent to maximising a *specific* feature extraction criterion at the outputs of the final hidden layer.

The error to be minimised is

$$\begin{aligned} E &= \frac{1}{P}\sum_{p=1}^{P} d_p \| \boldsymbol{t}^p - \boldsymbol{o}^p \|^2 \\ &= \frac{1}{P}\| [\boldsymbol{T} - (\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{H})] \boldsymbol{D} \|^2 \end{aligned} \tag{19}$$

where $\boldsymbol{T}$ is the $c \times P$ matrix of target patterns, $\boldsymbol{\lambda}_0$ is the $c \times 1$ vector of output biases, $\boldsymbol{\Lambda}$ is the $c \times n_0$ matrix of weights between the n_0 hidden nodes and the c output nodes, $\boldsymbol{H}$ is the $n_0 \times P$ matrix of output patterns at the hidden layer, and $\boldsymbol{D}$ is the $P \times P$ diagonal matrix of error weightings, $\sqrt{d_p}$.

Minimising (19) with respect to the bias vector gives the optimum solution for the biases as

$$\boldsymbol{\lambda}_0 = \bar{\boldsymbol{t}} - \boldsymbol{\Lambda}\boldsymbol{m}^H \tag{20}$$

where $\bar{\boldsymbol{t}}$ is the weighted target mean:

$$\bar{\boldsymbol{t}} \triangleq \frac{\boldsymbol{T}\boldsymbol{D}^2\boldsymbol{1}}{\sum_{p=1}^{P} d_p}$$

and $\boldsymbol{m}^H$ is the weighted mean pattern evaluated on the training set at the outputs of the hidden nodes:

$$\boldsymbol{m}^H \triangleq \frac{\boldsymbol{H}\boldsymbol{D}^2\boldsymbol{1}}{\sum_{p=1}^{P} d_p}$$

In this equation, $\boldsymbol{1}$ is a $(P \times 1)$ vector of 1's. Thus, the error to be minimised may be expressed as

$$E = \frac{1}{P}\| \left[\hat{\boldsymbol{T}} - \boldsymbol{\Lambda}\hat{\boldsymbol{H}}\right] \boldsymbol{D} \|^2 \tag{21}$$

where $\hat{\boldsymbol{T}} = \boldsymbol{T} - \bar{\boldsymbol{t}}\boldsymbol{1}^*$ is the mean-shifted target matrix and $\hat{\boldsymbol{H}} = \boldsymbol{H} - \boldsymbol{m}^H\boldsymbol{1}^*$ is the matrix of mean-shifted hidden unit output patterns. The weight matrix $\boldsymbol{\Lambda}$ which minimises (20) with minimum norm is

$$\boldsymbol{\Lambda} = \hat{\boldsymbol{T}}\boldsymbol{D}\left(\hat{\boldsymbol{H}}\boldsymbol{D}\right)^+ \tag{22}$$

where $\boldsymbol{A}^+$ is the Moore-Penrose pseudo-inverse of matrix $\boldsymbol{A}$. Using (22) in (21) and exploiting the properties of the pseudo-inverse gives the error in the form

$$E = \frac{1}{P} Tr\left\{\hat{\boldsymbol{T}}\boldsymbol{D}^2\boldsymbol{T}^* - \hat{\boldsymbol{T}}\boldsymbol{D}^2\hat{\boldsymbol{H}}^*(\hat{\boldsymbol{H}}\boldsymbol{D}^2\hat{\boldsymbol{H}}^*)\hat{\boldsymbol{H}}\boldsymbol{D}^2\boldsymbol{T}^*\right\} \tag{23}$$

where Tr is the trace operation. Since the targets and weights are fixed, minimising the error is equivalent to maximising the function

$$J = Tr\left\{\boldsymbol{S}_B\boldsymbol{S}_T^+\right\} \tag{24}$$

where the matrices S_T and S_B are specified entirely in terms of the targets and the distribution of patterns at the outputs of the hidden nodes:

$$\begin{aligned} S_T &\triangleq \frac{1}{P}\widehat{H}D^2\widehat{H}^* \\ S_B &\triangleq \left(\frac{1}{P}\right)^2 \widehat{H}D^2\hat{T}^*\hat{T}D^2\widehat{H}^* \end{aligned} \tag{25}$$

These matrices may be interpreted as weighted total and between class covariance matrices of the nonlinearly transformed input patterns. This allows (24) to be interpreted as a feature extraction criterion for discriminant analysis, where the criterion is optimum for the subsequent linear classification scheme to the output targets. The specific feature extraction criterion above is similar and has an equivalent interpretation to other suggested cost functions used in traditional discrimininant analysis [10, 7]. The difference is that this particular feature extraction has been optimised to achieve the best performance consistent with a subsequent linear classification of the patterns. Thus, such networks are performing optimised feature extraction and classification simultaneously.

B Sum Rules

This appendix proves the result that optimally trained networks with 1-from-c coding schemes satisfy the sum rule that the *general* outputs of the network for arbitrary input sum to unity.

Theorem.

Consider a network having linear output units. Let the weights associated with the connections to these units be determined by linear minimum norm least squares optimisation. Then if there exists an arbitrary linear constraint of the form

$$\mathbf{u}^*\mathbf{t}^p = \mathbf{u}^*\bar{\mathbf{t}} \;\forall\; p = 1, 2, \ldots, P$$

with $\mathbf{u}$ *a constant vector,* $\mathbf{t}^p$ *the prototype target vector for the p–th pattern and* $\bar{\mathbf{t}}$ *the mean target vector, then the general output* $\mathbf{o}$ *of the network also satisfies:*

$$\mathbf{u}^*\mathbf{o} = \mathbf{u}^*\bar{\mathbf{t}}$$

Proof

The general output of the network from Appendix A may be expressed as:

$$\mathbf{o} = \bar{\mathbf{t}} + \hat{\mathbf{T}}\hat{\mathbf{H}}^+ \left(\mathbf{h} - \mathbf{m}^H\right)$$

where $\mathbf{h}$ is the vector of outputs of the final hidden layer for a given input pattern. Therefore

$$\mathbf{u}^*\mathbf{o} = \mathbf{u}^*\bar{\mathbf{t}} + \mathbf{u}^*\hat{\mathbf{T}}\hat{\mathbf{H}}^+ \left(\mathbf{h} - \mathbf{m}^H\right)$$

But

$$\mathbf{u}^*\hat{\mathbf{T}} = \mathbf{u}^*\mathbf{T} - \mathbf{u}^*\bar{\mathbf{t}}\mathbf{1}^*$$

By hypothesis, $\mathbf{u}^*\mathbf{T} = \mathbf{u}^*\bar{\mathbf{t}}\mathbf{1}^*$. Therefore

$$\mathbf{u}^*\mathbf{o} = \mathbf{u}^*\bar{\mathbf{t}} \quad \blacksquare$$

Remark 1: If the set of target vectors satisfy several linear constraints simultaneously, then so will the general network outputs.

Remark 2: If $\mathbf{u}$ is a vector with unity for each component and the prototype target matrix is the identity matrix (1-from-c coding) then the general network output sums to unity.

C Standard Classifiers.

This appendix gives a brief discussion of the standard classifiers employed in this paper. The techniques employed were:

- *Euclidean distance to the class mean (DCM).* For each pattern choose the class which has the closest average pattern vector as determined on the training set.
- *Gaussian classifier (GC).* Assume each pattern in each class is drawn from a full Gaussian distribution where each class may be characterised by a mean vector, $\boldsymbol{\mu}_i$ and a (full) covariance matrix, $\boldsymbol{\Sigma}_i$. These are approximated by the training set samples. Then given the i-th class, i, the probability of any given pattern $\boldsymbol{p}$ belonging to that class may be expressed as

$$P(\boldsymbol{p}|i) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_i|} \exp -(\boldsymbol{p} - \boldsymbol{\mu}_i)^* \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{p} - \boldsymbol{\mu}_i)$$

  Knowing the prior probabilities p_i of the occurrence of each class allows the determination of the probability of that class given the pattern as $p_i P(\boldsymbol{p}|\boldsymbol{c})$. Thus, the decision is to choose that class which gives the largest pattern conditional probability.
- *Nearest neighbour (NN):* For each pattern in a given set, choose the class which is associated to the (Euclidean) *nearest* pattern in the training set.
- *K-nearest neighbour (KNN):* For each pattern in a given set, examine the classes associated with the K nearest patterns in the training set and choose the class which occurs most often. Ties are broken by choosing the nearest class mean using the K nearest neighbours. The optimum value of K is chosen as the one which gives best performance on the **test** set and is therefore a *biased* estimate of the model order. This technique usually gives good results but is computationally expensive on test.
- *Optimum Linear Transformation (OLT).* For the $n \times P$ matrix $\mathbf{X}$ of input patterns and the corresponding $c \times P$ matrix $\mathbf{T}$ of target patterns on the training set, find the optimum $c \times n$ matrix, $\mathbf{A}$ and $c \times 1$ vector $\boldsymbol{b}$ which satisfy the equation

$$\mathbf{AX} + \boldsymbol{b}\mathbf{1}^* = \mathbf{T}$$

  with minimum residual error. The solution with minimum Frobenius norm may be found by pseudo-inverse methods. Once $\mathbf{A}$ and $\mathbf{b}$ have been determined, an arbitrary input pattern is linearly transformed into a pattern $\tilde{\boldsymbol{t}}$ in the target pattern space. The class associated with that input pattern is determined by the closest target vector to the transformed pattern, $\tilde{\boldsymbol{t}}$.
- *Statistical Independence (SI).* This model was discussed in the text (14). It assumes that the density estimates of each feature are independent and so the conditional probability density is given by the product of the marginal densities. The class associated with a pattern is determined from the largest class-conditional density.

References

[1] W.D. Beastall. Recognition of radar signals by neural networks. In *1st IEE International Conference on Artificial Neural Networks*, pages 139–142, IEE, 1989.

[2] M.D. Bedworth. *Improving upon Standard Pattern Classification Algorithms by Implementing them as Multi-Layer Perceptrons.* R.S.R.E. Memorandum 4346, Royal Signals and Radar Establishment, St Andrews Rd., Great Malvern, Worcestershire, WR14 3PS, U.K., December 1989. Unlimited.

[3] David G. Bounds, Bruce Mathew, and Gordon Waddell. A Multi-layer Perceptron Network for the Diagnosis of Low Back Pain. In *IEEE Int. Conf. on Neural Networks*, pages II–481–II–489, 1988.

[4] J.S. Bridle. Probabilistic scoring for Back-Propagation Networks, with Relationships to Statistical Pattern Recognition. In *Neural Networks for Computing*, page , Publisher, Snowbird, 1989.

[5] D.S. Broomhead and David Lowe. Multi-variable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2(3):269–303, 1988.

[6] C.G. Broyden. Quasi-newton methods and their application to function minimisation. *Math. Comp.*, 21:368–381, 1967.

[7] P.A. Devijver and J. Kittler. *Pattern Recognition.* Prentice-Hall International, Inc, London, 1982.

[8] Pierre A. Devijver. Relationships between statistical risks and the least mean square error design criterion. In *Proceedings First International Conference on Pattern Recognition, Washington*, pages 139–148, 1973.

[9] Ronald A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

[10] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, 1972.

[11] R. Paul Gorman and Terrence J. Sejnowski. Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets. *Neural Networks*, 1(1):75–90, 1989.

[12] D.J. Hand. *Discrimination and Classification.* John Wiley & Sons, New York, 1981.

[13] J. Hilden and B. Bjerregaard. Computer aided diagnosis and the atypical case. In F.T. DeDombal and F. Gremy, editors, *Decision Making and Medical Care: Can Information Science Help?*, pages 365–378, North-Holland, Amsterdam, 1976.

[14] K Hornik, M. Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[15] D. Lowe and A.R. Webb. On Networks, Optimised Feature Extraction and the Bayes Decision. *submitted to Pattern Analysis and Machine Intelligence (also available as RSRE Memorandum 4342, St. Andrews Rd., Great Malvern, Worcs WR14 3PS., U.K.)*, 1989.

[16] S.M. Peeling, R.K. Moore, and M.J. Tomlinson. The multi-layer perceptron as a tool for speech pattern processing research. In *Proceedings IoA Autumn Conference on Speech and Hearing*, pages 307–314, 1986.

[17] Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[18] D.M. Titterington, G.D. Murray, L.S. Murray, D.J. Spiegelhalter, A.M. Skene, J.D.F. Habbema, and G.J. Gelpke. Comparison of Discrimination Techniques Applied to a Complex Data Set of Head Injured Patients. *Journal of the Royal Statistical Society*, 144:145–175, 1981.

[19] A.R. Webb and D. Lowe. The Optimised Internal Representation of Multilayer Classifier Networks Performs Nonlinear Discriminant Analysis. *Neural Networks (to appear)*, 1989.

[20] A.R. Webb and David Lowe. *A Hybrid Optimisation Strategy for Adaptive Feed-forward Layered Networks.* Memorandum 4193, Royal Signals and Radar Establishment, St Andrews Rd., Great Malvern, Worcestershire, WR14 3PS, U.K., September 1988. Unlimited.

[21] A.R. Webb, David Lowe, and M.D. Bedworth. *A Comparison of Nonlinear Optimisation Strategies for Feed-forward Adaptive Layered Networks.* Memorandum 4157, Royal Signals and Radar Establishment, R.S.R.E., St Andrews Road, Great Malvern, Worcs WR14 3PS, July 1988. Unlimited.

[22] William G. Wee. Generalised Inverse Approach to Adaptive Multiclass Pattern Classification. *IEEE Transactions on Computers*, C-17(12):1157–1164, December 1968.

THIS PAGE IS LEFT BLANK INTENTIONALLY

DOCUMENT CONTROL SHEET

Overall security classification of sheet ..... UNCLASSIFIED .....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification, eg (R), (C) or (S))

1 DRIC Reference (if known)	2 Originator's Reference	3 Agency Reference	4 Report Security Classification
	MEMO 4343		UNCLASSIFIED

5 Originator's Code (if known)	6 Originator (Corporate Author) Name and Location
7784000	ROYAL SIGNALS & RADAR ESTABLISHMENT ST ANDREWS ROAD, GREAT MALVERN WORCESTERSHIRE WR14 3PS

5a Sponsoring Agency's Code (if known)	6a Sponsoring Agency (Contract Authority) Name and Location

7 Title

INCORPORATING PRIOR PROBABILITIES AND MISCLASSIFICATION COSTS INTO NETWORK TRAINING: AN EXAMPLE FROM MEDICAL PROGNOSIS

7a Title in Foreign Language (in the case of Translations)

7b Presented at (for Conference Papers). Title, Place and Date of Conference

8 Author 1 Surname, Initials	9a Author 2	9b Authors 3, 4	10 Date	pp. ref
LOWE D	WEBB A R		1989.11	31

11 Contract Number	12 Period	13 Project	14 Other Reference

15 Distribution Statement

UNLIMITED

Descriptors (or Keywords)

Continue on separate piece of paper

Abstract Feed-forward layered networks trained on a pattern classification task in which the number of training patterns in each class is non-uniform, bias strongly in favour of those classes with largest membership. This is an unfortunate property of networks when the relative importance of classes with smaller membership is much greater than that of classes with many training patterns. In addition, there are many pattern classification tasks where different penalties are associated with misclassifying a pattern belonging to one class as another class. It is not generally known how to compensate for such effects in network training. This paper discusses an analytical regularisation scheme whereby prior expectations of class importance occurring in the generalisation data and misclassification costs may be incorporated into the training phase, thus compensating for the uneven and unfair class distributions occurring in the training set. The effects of the proposed scheme on the feature extraction criterion employed in the hidden layer of the network is discussed. An illustration of the results is presented by considering a real medical prognosis problem concerning data collected from head-injured coma patients. Relationships between least mean square error minimisation and Bayesian minimum risk estimation is mentioned and the importance and relevance of input/output coding schemes for network performance is considered.

THIS PAGE IS LEFT BLANK INTENTIONALLY