Ensemble MLP Classifier Design

Terry Windeatt

Centre for Vision, Speech and Signal Proc., Department of Electronic Engineering,
University of Surrey, Guildford, Surrey, United Kingdom GU2 7XH
t.windeatt@surrey.ac.uk

Abstract. Multi-layer perceptrons (MLP) make powerful classifiers that may provide superior performance compared with other classifiers, but are often criticized for the number of free parameters. Most commonly, parameters are set with the help of either a validation set or cross-validation techniques, but there is no guarantee that a pseudo-test set is representative. Further difficulties with MLPs include long training times and local minima. In this chapter, an ensemble of MLP classifiers is proposed to solve these problems. Parameter selection for optimal performance is performed using measures that correlate well with generalisation error.

1 Introduction

The topic of this chapter concerns solving problems in pattern recognition using a combination of neural network classifiers. Pattern classification involves assignment of an object to one of several pre-specified categories or classes, and is a key component in many data interpretation activities. Here we focus on classifiers that learn from examples, and it is assumed that each example pattern is represented by a set of numbers, which are known as the pattern features. In the case of face recognition (Section 5), these features consist of numbers representing different aspects of facial features. In order to design a learning system it is customary to divide the example patterns into two sets, a training set to design the classifier and a test set, which is subsequently used to predict the performance when previously unseen examples are applied. A problem arises when there are many features and relatively few training examples, and the classifier can learn the training set too well, known as over-fitting so that performance on the test set degrades.

Automating the classification task to achieve optimal performance has been studied in the traditional fields of pattern recognition, machine learning and neural networks as well as newer disciplines such as data fusion, data mining and knowledge discovery. Traditionally, the approach that has been used in the design of pattern classification systems is to experimentally assess the performance of several classifiers with the idea that the best one will be chosen. Ensemble classifiers, also known as Multiple Classifier Systems (MCS), were developed to address the problem of reliably designing a system with improved accuracy. Recognising that each classifier may make different and perhaps complementary errors, the idea is to pool together the results from all classifiers to find a composite system that outperforms any individual (base) classifier. In this way a single complex classifier may be replaced by a set of relatively simple classifiers.

L.C. Jain et al. (Eds.): Comp. Intel. Para.: Innov. Applications, SCI 137, pp. 133–147, 2008.
springerlink.com

Even though an ensemble is less likely to over-fit, there is still the difficulty of tuning individual classifier parameters. Multi-layer perceptrons (MLP) make powerful classifiers that may provide superior performance compared with other classifiers, but are often criticized for the number of free parameters. The common approach to adjusting parameters is to divide the training set into two to produce a validation set. When the number of examples is in short supply, cross-fold validation may be used. For example, in ten-fold cross-validation, the set is randomly split into ten equal parts with nine parts used for training and one part used as a validation set to tune parameters. Training is repeated ten times with a different partition each time, and the results averaged. However, it is known that these approaches to validation are either inappropriate or very time-consuming. Ideally all the training set should be used for training, so that there is no need for validation. However, this requires that over-fitting be detected by looking at performance on only the training set, which is a difficult problem.

It is known that certain conditions need to be satisfied to realise ensemble performance improvement, in particular that the constituent classifiers be not too highly correlated. If each classifier solves a slightly different problem, then composite performance may improve. Various techniques have been devised to reduce correlation by injecting randomness. For example, two techniques that are used in this chapter are Bagging [1] (Section 2), which resamples the training set and Error-Correcting-Output-Coding (ECOC Section 4), which solves multi-class problems by randomly decomposing into two-class problems.

Although it is known that diversity among base classifiers is a necessary condition for improvement in ensemble performance, there is no general agreement about how to quantify the notion of diversity among a set of classifiers. Diversity Measures can be categorised into pair-wise (Section 3) and non-pair-wise, and to apply pair-wise measures to finding overall diversity it is necessary to average over the classifier set. These pair-wise diversity measures are normally computed between pairs of classifiers independent of target labels. As explained in [14], the accuracy-diversity dilemma arises because when base classifiers become very accurate their diversity must decrease, so that it is expected that there will be a trade-off.

A possible way around this dilemma is to incorporate diversity and accuracy to produce a single class separability measure, as suggested in Section 3. The measure is based on a spectral representation that was first proposed for two-class problems in [2], and later developed in the context of Multiple Classifier Systems in [3]. It was shown for two-class problems in [4] that over-fitting of the training set could be detected by observing the separability measure as it varies with base classifier complexity. Since realistic learning problems are in general ill-posed [5], it is known that any attempt to automate the learning task must make some assumptions. The only assumption required here is that a suitable choice be made for the range over which base classifier complexity is varied.

2 MLP Classifiers and Ensembles

There are many text books that contain an introduction to MLP classifiers. Since the topic of neural networks is multi-disciplinary, it is useful to find a text that is written from a stand-point similar to the reader. For students of engineering and computer science, reference [6] is recommended.

A multi-layer rather than a single layer network is required since a single layer perceptron (SLP) can only compute a linear decision boundary, which is not flexible enough for most realistic learning problems. For a problem that is linearly separable, (that is capable of being perfectly separated by linear decision boundary), the perceptron convergence theorem guarantees convergence. In its simplest form, SLP training is based on the simple idea of adding or subtracting a pattern from the current weights when the target and predicted class disagrees, otherwise the weights are unchanged. For a non-linearly separable problem, this simple algorithm can go on cycling indefinitely. The modification known as least mean square (LMS) algorithm uses a mean squared error cost function to overcome this difficulty, but since there is only a single perceptron, the decision boundary is still linear.

An MLP is a universal approximator [6] that typically uses the same squared error function as LMS. However, the main difficulty with the MLP is that the learning algorithm has a complex error surface, which can become stuck in local minima. There does not exist any MLP learning algorithm that is guaranteed to converge, as with SLP. The popular MLP back-propagation algorithm has two phases, the first being a forward pass, which is a forward simulation for the current training pattern and enables the error to be calculated. It is followed by a backward pass, that calculates for each weight in the network how a small change will affect the error function. The derivative calculation is based on the application of the chain rule, and training typically proceeds by changing the weights proportional to the derivative. For the back-propagation equations see reference [6].

Although there are different ways of trying to overcome the local minima problem [6], in this chapter an ensemble of MLPs is proposed. An MLP with random starting weights is a suitable base classifier since randomisation has shown to be beneficial in the MCS context. Problems of local minima and computational slowness may be alleviated by the MCS approach of pooling together the decisions obtained from

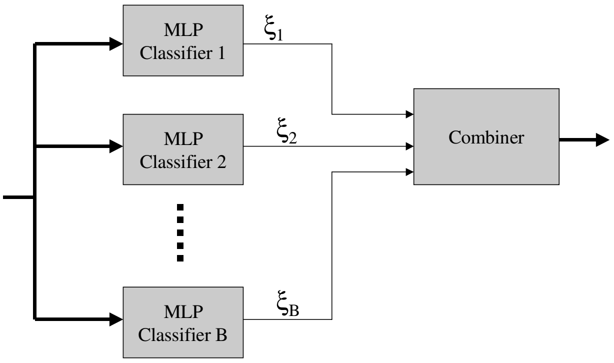

Fig. 1. Ensemble MLP architecture, showing base MLP classifier decisions

locally optimal classifiers. The architecture envisaged is a simple MCS framework in which there are B parallel MLP base classifiers, as shown in figure 1.

Bootstrapping [7] is a popular ensemble technique that provides a beneficial increase in diversity among base classifiers. The implication in bootstrapping is that if μ training patterns are randomly sampled with replacement, $(1\text{-}1/\mu))^{\mu} \cong 37\%$ are removed with remaining patterns occurring one or more times. The base classifier out-of-bootstrap (OOB) estimate uses the patterns left out, and should be distinguished from the ensemble OOB. For the ensemble OOB, all training patterns contribute to the estimate, but the only participating classifiers for each pattern are those that have not been used with that pattern for training (that is, approximately thirty-seven percent of classifiers). Note that OOB gives a biased estimate of the absolute value of generalization error [8], but here the estimate of the absolute value is not important. The OOB estimate for the ECOC ensemble is given in Section 4. In Section 5 we demonstrate by example that the OOB estimate can predict the optimal base classifier complexity.

3 Diversity/Accuracy and MCS

Attempts to understand the effectiveness of the MCS (ensemble) framework have prompted the development of various diversity measures with the intention of determining whether they correlate with ensemble accuracy. However, the question of whether the information available from these measures can be used to assist MCS design is open. Most commonly, ensemble parameters are set with the help of either a validation set or cross-validation techniques [9]. In this section, we review the definition of a popular pair-wise diversity measure (equation (6)), and show that diversity and accuracy can be incorporated within a single class separability measure.

Classical class separability measures refer to the ability to predict separation of patterns into classes using original features and rely on a Gaussian assumption [10]. The problem with applying classical class separability measures to the binary mapping associated with MCS (equation ((1))) is that the implicit Gaussian assumption is not appropriate [11]. In [4], [12] a class separability measure is proposed for MCS that is based on a binary feature representation, in which each pattern is represented by its binary ensemble classifier decisions. It is restricted to two-class problems and results in a binary-to-binary mapping.

Let there be μ training patterns with the label ω_m given to each pattern $\boldsymbol{x_m}$ where $m = 1,\dots \mu$. In the MCS framework shown in figure 1, the mth pattern may be represented by the B-dimensional vector formed from the B base classifier decisions given by

$$x_m = (\xi_{m1}, \xi_{m2}, \dots, \xi_{mB}) \qquad \xi_{mi},\ \omega_m \in \{0,1\}, \quad i = 1 \dots B \tag{1}$$

In equation (1) $\omega_m = f(\boldsymbol{x_m})$ where f is the unknown binary-to-to binary mapping from classifier decisions to target label. Following [13] the notation in equation (1) is modified so that the classifier decision is 1 if it agrees with the target label and 0 otherwise

$$x_m = (y_{m1}, y_{m2}, \dots, y_{mB}) \qquad y_{mi}\ ,\ \omega_m = \in \{0,1\},\ \ y_{mi}=1 \text{ iff } \xi_{mi} = \omega_m \tag{2}$$

Pair-wise diversity measures, such as Q statistic (equation (6)), Correlation Coefficient, Double Fault and Disagreement measures [13] take no account of class assigned to a pattern. In contrast, class separability [14] is computed between classifier decisions (equation (2)) over pairs of patterns of opposite class, using four counts defined by logical AND ($\wedge$) operator

$$\tilde{N}^{ab}_{mn} = \sum_{j=1}^{B} \psi^{a}_{mj} \wedge \psi^{b}_{nj}, \quad \omega_m \neq \omega_n \quad a,b\in\{0,1\},\ \psi^1 = y, \psi^0 = \overline{y} \tag{3}$$

The *nth* pattern for a two-class problem is assigned

$$\sigma'_n = \frac{1}{\tilde{K}_\sigma}\left(\frac{\tilde{N}_n^{11}}{\sum\limits_{m=1}^{\mu}\tilde{N}_m^{11}} - \frac{\tilde{N}_n^{00}}{\sum\limits_{m=1}^{\mu}\tilde{N}_m^{00}}\right) \tag{4}$$

where $$\tilde{K}_\sigma = \left(\frac{\tilde{N}_n^{11}}{\sum\limits_{m=1}^{\mu}\tilde{N}_m^{11}} + \frac{\tilde{N}_n^{00}}{\sum\limits_{m=1}^{\mu}\tilde{N}_m^{00}}\right), \tilde{N}_n^{ab} = \sum_{m=1}^{\mu}\tilde{N}_{mn}^{ab}$$

The motivation for σ'_n in equation (4) comes from estimation of the first order spectral coefficients [4] of the binary-to-binary mapping defined in equation (1). Each pattern is compared with all patterns of the other class, and the number of jointly correctly ($\tilde{N}_n^{11}$) and incorrectly ($\tilde{N}_n^{00}$) classified patterns is counted. Note that a classifier that correctly classifies one pattern but incorrectly classifies the other does not contribute. The two terms in equation (4) represent the relative positive and negative evidence that the pattern comes from the target class. A simple example for nine classifier pairs demonstrates the idea.

classifier pair	1 2 3 4 5 6 7 8 9	
	0 0 1 1 1 0 0 1 1	class ω_1
	1 1 1 0 1 0 0 1 1	class ω_2

If 1 indicates correct classification, the contradictory evidence from classifier pair 1, 2, 4 is ignored, evidence from classifiers 3, 5, 8, 9 is for positive correlation with change of class and the remaining classifiers give evidence for negative correlation. The evidence is summed over all pairs, as given in equations (3) and (4), where a pair must contain a pattern from each class.

Furthermore, we sum over patterns with positive coefficient to produce a single number between 0 and 1 that represents the separability of a set of patterns

$$\sigma' = \sum_{n=1}^{\mu}\sigma'_n, \sigma'_n > 0 \tag{5}$$

In our examples in Section 5 we compare the Q diversity measure, as recommended in [13]. Diversity Q_{ij} between *ith* and *jth* classifiers is defined as

$$Q_{i\,j} = \frac{N_{ij}^{\ 11} N_{ij}^{\ 00} - N_{ij}^{\ 01} N_{ij}^{\ 10}}{N_{ij}^{\ 11} N_{ij}^{\ 00} + N_{ij}^{\ 01} N_{ij}^{\ 10}} \tag{6}$$

where $N_{ij}^{ab} = \sum_{m=1}^{\mu} \psi_{mj}^{a} \wedge \psi_{mj}^{b}$ with a,b, Ψ defined in equation (3). The mean is taken over B base classifiers $Q = \frac{2}{B(B-1)} \sum_{i=1}^{B-1} \sum_{j=i+1}^{B} Q_{ij}$.

4 Error Correcting Output Coding (ECOC) and Multi-class Problems

A single MLP with multiple outputs can handle directly a K-class problem, K > 2. The standard technique is to use a K-dimensional binary target vector that represents each one of K classes using a single binary value at the corresponding position, for example [0,…,0,1,0,…..0]. In practice the target value is often specified as 0.9 and 0.1 rather than 1 and 0, to stop the MLP from saturating at the extremities of the sigmoid activation function, for which the gradient is zero [6]. The reason that a single multi-class MLP is not a suitable candidate for use with a more elaborate coding scheme is that all nodes share in the same training, so errors are far from independent and there is not much benefit to be gained from combining.

There are several motivations for decomposing a multi-class problem into complementary two-class problems. The decomposition means that attention can be focused on developing an effective technique for the two-class MLP classifier, without having to consider explicitly the design of the multi-class case. Also, it is hoped that the parameters of an MLP base classifier run several times may be easier to determine than a single complex MLP classifier run once, and perhaps facilitate faster and more efficient solutions. Finally, solving different two-class sub-problems repeatedly with random perturbation may help to reduce error in the original problem.

The ECOC method [15] is an example of distributed output coding [16], in which a pattern is assigned to the class that is closest to a corresponding code word. Rows of the ECOC matrix act as the code words, and are designed using error-correcting principles to provide some error insensitivity with respect to individual classification errors. The original motivation for encoding multiple classifiers using an error-correcting code was based on the idea of modelling the prediction task as a communication problem, in which class information is transmitted over a channel. Errors introduced into the process arise from various stages of the learning algorithm, including features selected and finite training sample. From error-correcting theory, we know that a matrix designed to have *d* bits error-correcting capability implies that there is a minimum Hamming Distance *2d+1* between any pair of code words. Assuming each bit is transmitted independently, it is then possible to correct a received pattern having fewer than *d* bits in error, by assigning the pattern to the code word closest in Hamming distance. Clearly, from this perspective it is desirable to use a matrix containing code words having high minimum Hamming distance between any pair.

To solve a multi-class problem in the ECOC framework we need a set of codes to decompose the original problem, a suitable two-class base classifier, and a decision-making framework. For a K-class problem, each row of the *K x B* binary ECOC matrix Z acts as a code word for each class. Each of the B columns of Z partitions the training data into two 'superclasses' according to the value of the corresponding binary element. Consider the following 6 x B code matrix for a six-class problem.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & \cdots \\ 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 1 & 0 & 1 & 1 & \cdots \\ 1 & 1 & 0 & 1 & \cdots \\ 0 & 0 & 1 & 1 & \cdots \end{bmatrix}$$

The first column would naturally be interpreted as putting patterns from class 2, 4, 5 into one 'superclass' and remaining patterns in the second 'superclass'. Columns 2,3,4 … correspond to different classifier runs and represent different two-class decompositions.Consider the decoding step for a three class problem, represented in figure 2. To classify a pattern $\boldsymbol{x}_m$, it is applied to the B trained base classifiers as shown in figure 2, forming vector $[x_{m1}, x_{m2}, ..., x_{mB}]$ where x_{mj} is the output of the *jth* base classifier. The L^1 norm distance L_i (where i = 1.… K) between output vector and code word for each class is computed

$$L_i = \sum_{j=1}^{b} \left| Z_{ij} - x_{mj} \right| \tag{7}$$

and $\boldsymbol{x}_m$ is assigned to the class ω_m corresponding to closest code word .

To use the ensemble OOB estimate, pattern $\boldsymbol{x}_m$ is classified using only those classifiers that are in the set OOB_m, defined as the set of classifiers for which $\boldsymbol{x}_m$ is OOB. For the OOB estimate, the summation in equation (7) is therefore modified to $\sum_{j \in OOB_m}$. Therefore only a subset of the columns of Z is used in the decoding step in figure 2, so that approximately B/3 columns are used for each pattern.

The topic of designing suitable code matrices has received much attention. In [17] it is shown that any variation in Hamming distance between pairs of code words will reduce the effectiveness of the combining strategy. In [18] it is shown that maximising the minimum Hamming Distance between code words implies minimising upper bounds on generalisation error. In classical coding theory, theorems on error-correcting codes guarantee a reduction in the noise in a communication

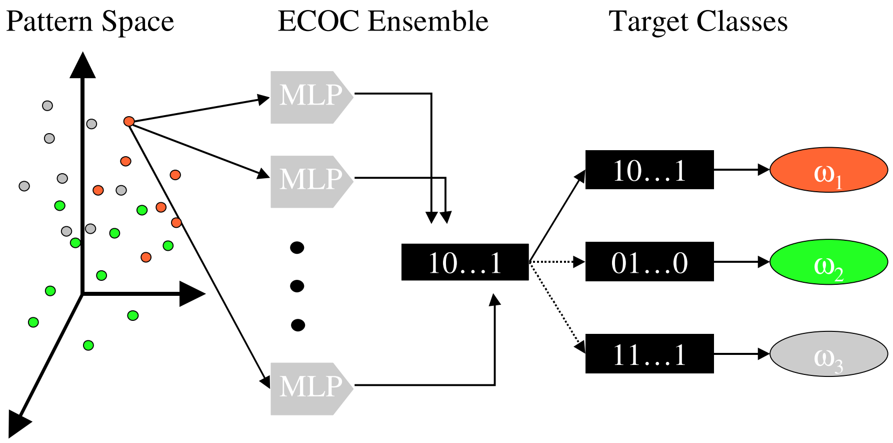

Fig. 2. ECOC Ensemble showing code words for target classes of 3 class problem

channel, but the assumption is that errors are independent. When applied to machine learning the situation is more complex, in that error correlation depends on the data set, base classifier as well as the code matrix Z. In the original ECOC approach [15], heuristics were employed to maximise the distance between the columns of Z to reduce error correlation. Random codes, provided that they are long enough, have frequently been employed with almost as good performance [17]. It would seem to be a matter of individual interpretation whether long random codes may be considered to approximate required error-correcting properties. In this chapter, a random code matrix with near equal split of classes (approximately equal number of 1's in each column) is chosen, as proposed in [19]. Note that some papers prefer to refer to the method as Output Coding, in recognition of the discussion over whether error-correcting properties are significant.

5 Examples

The following examples use natural benchmark data that demonstrate the usefulness of the measures proposed in this chapter. The main purpose of these examples is to demonstrate how well the measures correlate with test error as the number of training epochs of single hidden-layer MLP base classifiers are systematically varied. All the measures are computed on the training data and the datasets use random training/testing split, respecting the class distribution as closely as possible. Experiments are repeated ten times, except for the face database which is repeated twenty times. The split is 50/50 for the ORL face database (described in Section 5.2) and implies five training images per class, while other datasets are split 20/80. The ensemble MLP architecture is shown in figure 1. Each base classifier run uses identical classifier parameters, with variation arising from three sources, (i) random

Table 1. Benchmark Datasets showing numbers of patterns, classes, continuous and discrete features

DATASET	#pat	#class	#con	#dis
diabetes	768	2	8	0
dermatology	366	6	1	33
ecoli	336	8	5	2
glass	214	6	9	0
iris	150	3	4	0
segment	2310	7	19	0
soybean	683	19	0	35
vehicle	846	4	18	0
vowel	990	11	10	1
wave	3000	3	21	0
yeast	1484	10	7	1

initial weights, (ii) bootstrapping, and (iii) for multi-class, the problem decomposition defined by the respective code matrix column.

Natural benchmark problems, selected from [20] and [21] are shown in table 1 with numbers of patterns, classes, continuous and discrete features. For datasets with missing values the scheme suggested in [20] is used. The two-class dataset ‘diabetes’ uses one hundred classifiers (B = 100), multi-class problems use two hundred classifiers (K x 200 coding matrix), and the face database uses five hundred classifiers (40 x 500 code matrix). The Resilient BackPropagation (RPROP) [22] and Levenberg-Marquardt (LM) [6] algorithms are used. The only observed difference between the two algorithms is in the number of epochs required for optimal performance, indicating that the MLP ensemble is insensitive to the convergence properties of the base algorithm. In the examples presented, RPROP is used for the face database and LM for benchmark data. For further details see reference [12].

5.1 Benchmark Data

Figure 3 shows ‘diabetes’ 20/80, a difficult dataset that is known to over-fit with increasing classifier complexity. Figure 3 (a) (b) shows base classifier and ensemble test error rates, (c) (d) the base classifier and ensemble OOB estimates and (e) (f) the measures σ', Q defined in equations (5) and (6) for various node-epoch combinations. It may be seen that σ' and base classier OOB are good predictors of base classifier test error rates as base classifier complexity is varied. The correlation between σ' and test error was thoroughly investigated in [12], showing high values of correlation coefficient (0.98) that were significant (95 % confidence when compared with random chance). In [12] it was also shown that bootstrapping did not significantly change the ensemble error rates, actually improving them slightly on average.

The class separability measure σ' shows that the base classifier test error rates are optimised when the number of epochs is chosen to maximize class separability. Furthermore, at the optimal number of epochs Q shows that diversity is minimized. It

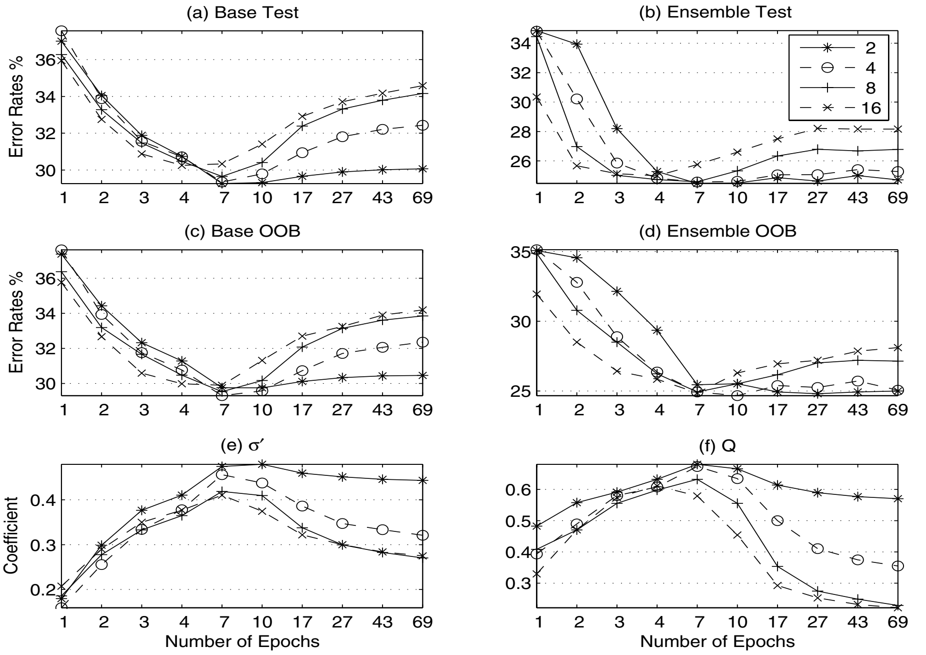

Fig. 3. Mean test error rates, OOB estimates, measures σ', Q for Diabetes 20/80 with [2,4,8,16] nodes

appears that base classifiers starting from random weights increase correlation (reduce diversity) as complexity is increased and correlation peaks as the classifier starts to over-fit the data. A possible explanation of the over-fitting behaviour is that classifiers produce different fits of those patterns in the region where classes are overlapped. From equation (4) σ' is the probability that $\left(\frac{\tilde{N}_n^{\;11}}{\sum_{m=1}^{\mu} \tilde{N}_m^{\;11}} - \frac{\tilde{N}_n^{\;00}}{\sum_{m=1}^{\mu} \tilde{N}_m^{\;00}} \right) > 0$. It provides an intuitive explanation of why we may expect that σ'correlates well with base classifier test error, and in particular peaks at the same number of training epochs. Consider the example of two overlapping Gaussians representing a two-class problem, with the Bayes boundary assumed to be placed where the probability density curves cross. Let the overlapping region be defined as the tails of the two Gaussians with respect to the Bayes boundary. If base classifiers are capable of approximating the Bayes boundary, by definition an optimal base classifier will incorrectly classify all patterns in the overlapping region and correctly classify all other patterns. Now consider the situation that the complexity of the base classifiers increases beyond optimal, so that some patterns in the overlapping region become correctly classified and some of the remaining patterns become incorrectly classified. The result is that there is greater variability in classification among patterns close to the Bayes boundary, and it is more difficult to separate them. The probability represented by σ' decreases as complexity increases since $\tilde{N}^{00}$ is more evenly distributed over all patterns, leading to a reduction in positively correlated patterns. However, if the base classifier becomes too powerful, eventually all patterns are correctly classified and $\tilde{N}^{00} \rightarrow 0$ and $\sigma' \rightarrow 1$, so it is expected that σ'would start to increase.

Note from Figure 3 that the ensemble is more resistant to over-fitting than base classifier for epochs greater than 7, and the ensemble OOB accurately predicts this trend. This experiment was performed for all datasets, and in general the ensemble test error was found to be more resistant to over-fitting. Figure 4 shows similar curves to Figure 3 averaged over all multi-class datasets (for multi-class, the correlation coefficient between $\sigma'$and test error is 0.96).

5.2 Face Data

Facial images are a popular source of biometric information since they are relatively easy to acquire. However, automated face recognition systems often perform poorly due to small number of relatively high-dimensional training patterns, which can lead to poor generalisation through over-fitting. Face recognition is an integral part of systems designed for many applications including identity verification, security, surveillance and crime-solving. Improving their performance is known to be a difficult task, but one approach to improving accuracy and efficiency is provided by the method of ECOC ensemble classifiers [23].

A typical face recognition system consists of three functional stages. In the first stage, the image of a face is registered and normalised. Since face images differ in both shape and intensity, *shape alignment* (geometric normalisation) and *intensity correction* (photometric normalisation) can improve performance. The second stage is feature extraction in which discriminant features are extracted from the face region. Finally, there is the matching stage in which a decision-making scheme needs to be designed depending on the task to be performed. In identification, the system classifies a face from a database of known individuals, (while in verification the system should confirm or reject a claimed identity).

There are two main approaches to feature extraction. In the geometric approach, relative positions and other parameters of distinctive features such as eyes, mouth and nose are extracted. The alternative is to consider the global image of the face as with methods such as Principal Component Analysis (PCA). By itself PCA is not adequate for the face recognition task since projection directions only maximise the total scatter across all classes. Therefore PCA may be combined with Linear Discriminant Analysis (LDA), which requires computation of the between-class scatter matrix, S_B and the within-class scatter matrix, S_W. The objective of LDA is to find the transformation matrix, W_{opt}, that maximises the ratio of determinants $\left|\boldsymbol{W^T S_B W}\right| / \left|\boldsymbol{W^T S_W W}\right|$. W_{opt} is known to be the solution of the following eigenvalue problem $S_B - S_W\Lambda = 0$, where Λ is a diagonal matrix whose elements are the eigenvalues of matrix $S_W^{-1}S_B$.

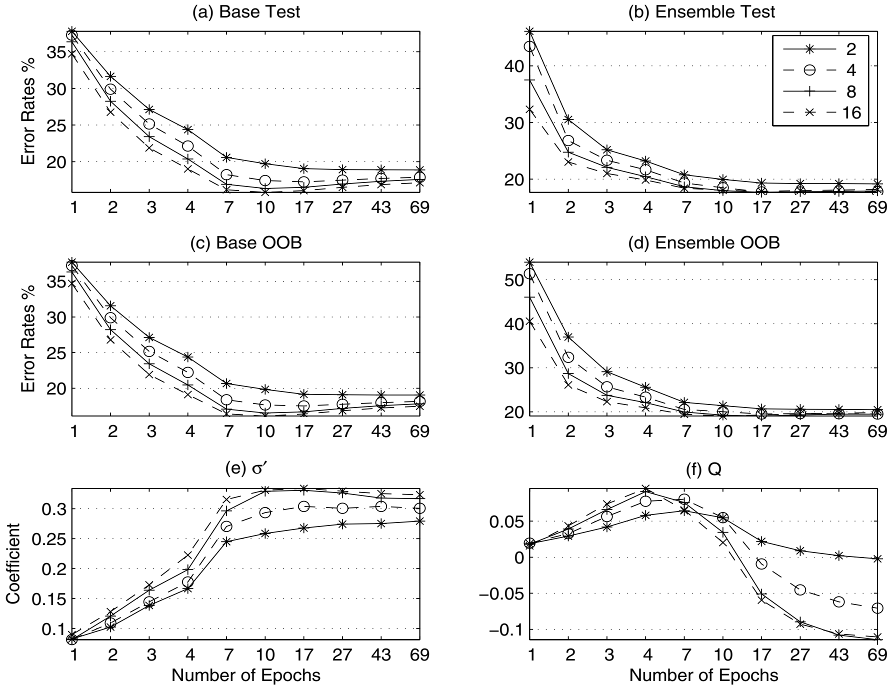

Fig. 4. Mean test error rates, OOB estimates, measures σ', Q over ten multi-class 20/80 datasets with [2,4,8,16] nodes

The face database used is the ORL (Olivetti Research Laboratory http//www.cam-orl.co.uk), consisting of four hundred images of forty individual faces with some variation in lighting, facial expression, facial hair, pose and spectacles. The background is controlled with subjects in an upright frontal position, although small variation in rotation and scale is allowed. The advantage of this database is that it can be used without need for a face detection algorithm or any other pre-processing, so that there is a fairer comparison with the results obtained by other researchers. In our examples, images have been projected to forty-dimensions using PCA and subsequently to a twenty-dimension feature space using LDA. It is treated as a forty-class face identification problem with the four hundred images randomly split into 50/50 training/testing patterns. A comparison of results on this database is given in [24], using 50/50 split, but the number of runs is smaller than used in our examples. As pointed out in [24], some researchers do not specify the split that they have used, and some only base their results on one run, so that care is needed before making any comparison.

Figure 5 shows error rates, σ', Q for 50/50 60/40 70/30 and 80/20 splits with 16 nodes and bootstrapping applied. As with two-class and multi-class benchmark problems, σ' appears to be strongly correlated with test error.

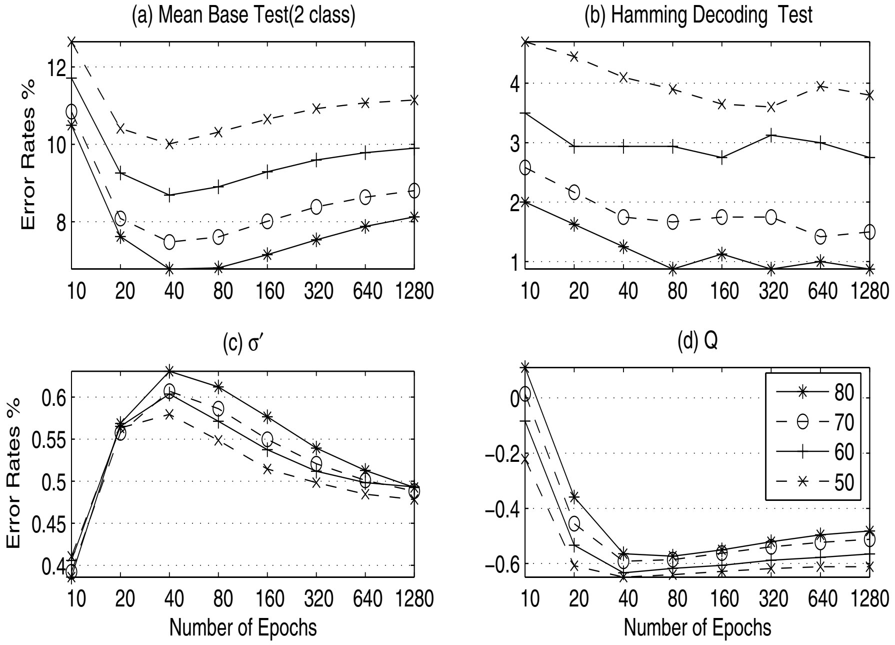

Fig. 5. Test error, σ′, Q for ORL database using 16 hidden-node bootstrapped base classifiers for [50/50,60/40,70/30,80/20] train/test splits

6 Discussion

There is another aspect of ensemble classifier design that needs to be addressed, the problem of feature selection. The aim of feature selection is to find a feature subset from the original set of features such that an induction algorithm that is run on data containing only those features generates a classifier that has the highest possible accuracy [25]. Typically, an exhaustive search is computationally prohibitive, and the problem is known to be NP-hard, so that a greedy search scheme is required. For problems with a large number of features, classical feature selection schemes are not greedy enough, and filter, wrapper and embedded approaches have been developed [26]. One-dimensional feature ranking methods consider each feature in isolation and rank the features according to a scoring function, but are disadvantaged by implicit orthogonality assumptions [26]. They are very efficient but in general have been shown to be inferior to multi-dimensional methods that consider all features simultaneously.

The measures proposed in this chapter have been used to select the optimal number of features of an ensemble of MLP classifiers [27]. In [28], a multi-dimensional feature-ranking criterion based on modulus of MLP weights identifies the least relevant features. It is combined with Recursive Feature Elimination (RFE) to recursively remove the irrelevant features until the measures indicate that test error performance degrades if further features are eliminated.

7 Conclusion

Multi-layer perceptrons (MLP) make powerful classifiers that may provide superior performance compared with other classifiers, but are often criticized for the number of free parameters. In this chapter an ensemble of relatively simple MLP classifiers is proposed, along with some measures that facilitate ensemble design. The proposed measures facilitate the detection of over-fitting as base classifier complexity is varied. Although the measures are only applicable to two-class problems, they may also be applied to multi-class via the two-class decompositions induced by Error-Correcting Output Coding.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1997)
2. Windeatt, T., Tebbs, R.: Spectral technique for hidden layer neural network training. Pattern Recognition Letters 18(8), 723–731 (1997)
3. Windeatt, T.: Recursive Partitioning for combining multiple classifiers. Neural Processing Letters 13(3), 221–236 (2001)
4. Windeatt, T.: Vote Counting Measures for Ensemble Classifiers. Pattern Recognition 36(12), 2743–2756 (2003)
5. Tikhonov, A.N., Arsenin, V.A.: Solutions of ill-posed problems. Winston & Sons, Washington (1977)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Englewood Cliffs (1999)
7. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, Boca Raton (1993)
8. Bylander, T.: Estimating generalisation error two-class datasets using out-of-bag estimate. Machine Learning 48, 287–297 (2002)
9. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Trans. Pattern Analysis and Machine Intelligence 12(10), 993–1001 (1990)
10. Fukunaga, K.: Introduction to statistical pattern recognition. Academic Press, London (1990)
11. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Trans. PAMI 24(3), 289–300 (2002)
12. Windeatt, T.: Accuracy/Diversity and ensemble classifier design. IEEE Trans. Neural Networks 17(5), 1194–1211 (2006)
13. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles. Machine Learning 51, 181–207 (2003)
14. Windeatt, T.: Diversity Measures for Multiple Classifier System Analysis and Design. Information Fusion 6(1), 21–36 (2004)
15. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
16. Sejnowski, T.J., Rosenberg, C.R.: Parallel networks that learn to pronounce english text. Journal of Complex Systems 1(1), 145–168 (1987)
17. Windeatt, T., Ghaderi, R.: Coding and Decoding Strategies for Multi-class Learning Problems. Information Fusion 4(1), 11–21 (2003)

18. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing Multi-class to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research 1, 113–141 (2000)
19. Schapire, R.E.: Using Output Codes to Boost Multi-class Learning Problems. In: 14th Int. Conf. of Machine Learning, pp. 313–321. Morgan Kaufmann, San Francisco (1997)
20. Prechelt, L.: Proben1: A Set of Neural Network Benchmark Problems and Benchmarking Rules, Tech Report 21/94, Univ. Karlsruhe, Germany (1994)
21. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
22. Riedmiller, M., Braun, H.: A Direct Adaptive Method for Faster Backpropagation Learning: The {RPROP} Algorithm. In: Proc. Intl. Conf. on Neural Networks, San Francisco, Calif., pp. 586–591 (1993)
23. Kittler, J., Ghaderi, R., Windeatt, T., Matas, J.: Face Verification via Error Correcting Output Codes. Image and Vision Computing 21(13-14), 1163–1169 (2003)
24. Er, M.J., Wu, S., Toh, H.L.: Face Recognition with RBF Neural Networks. IEEE Trans. Neural Networks 13(3), 697–710 (2002)
25. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence Journal, special issue on relevance 97(1-2), 273–324 (1997)
26. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
27. Windeatt, T., Prior, M.: Stopping Criteria for Ensemble-based Feature Selection. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472. Springer, Heidelberg (2007)
28. Windeatt, T., Prior, M., Effron, N., Intrator, N.: Ensemble-based Feature Selection Criteria. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571. Springer, Heidelberg (2007)