A Study of Ensemble of Hybrid Networks with Strong Regularization

Shimon Cohen and Nathan Intrator*

School of Computer Science, Tel Aviv University
Ramat Aviv 69978, Israel
shimon_cohen2000@yahoo.com, nin@cs.tau.ac.il
www.cs.tau.ac.il/~nin

Abstract. We study various ensemble methods for hybrid neural networks. The hybrid networks are composed of radial and projection units and are trained using a deterministic algorithm that completely defines the parameters of the network for a given data set. Thus, there is no random selection of the initial (and final) parameters as in other training algorithms. Network independent is achieved by using bootstrap and boosting methods as well as random input sub-space sampling. The fusion methods are evaluated on several classification benchmark data-sets. A novel MDL based fusion method appears to reduce the variance of the classification scheme and sometimes be superior in its overall performance.

1 Introduction

Hybrid neural networks that are composed of radial basis functions and perceptrons have been recently introduced [5,4]. Such networks employ a deterministic algorithm that computes the initial parameters from the training data. Thus, networks that have been trained on the same data-set produce the same solution and therefore, a combination of such classifiers can not enhance the performance over a single one.

Fusion of experts has been studied extensively recently. One of the main results is that experts have to be partially independent for the fusion to be effective [13,14]. The bagging algorithm [1] can be used to de-correlate between classifiers as well as to obtain some performance measure on the accuracy of the classifiers using the "out of bag" sub-set of the data. Another technique Arcing – adaptive re-weighting and combining – refers to reusing or selecting data in order to improve classification [2]. One popular arcing procedure is AdaBoost [10], in which the errors on the training data-sets are used to train more specific classifiers. Sub-sampling of the input space as well as the training patters is extensively used in the random forest algorithm [3]. A different flavor of combination of classifiers use dynamic class combination (DCS) [11] and Classifiers Local Accuracy (CLA) in order to select the best classifier when making a predication.

* Corresponding author's address: Institute for Brain and Neural Systems, Box 1843, Brown University, Providence, RI 02912, USA

T. Windeatt and F. Roli (Eds.): MCS 2003, LNCS 2709, pp. 227–235, 2003.

This is done at the cost of saving the whole training set and then selecting the predication of the best classifier at the vicinity of a given pattern.

The hybrid Perceptron Radial Basis Function Network (PRBFN) is constructed with strong regularization and with initial parameters that are estimated from the data and not random. The strong regularization and excellent approximation properties of a hybrid of projection and radial units leads to a relatively small architecture, which, in addition to the strong regularization leads to an estimator with low variance. Thus, ensemble combination, which is known to reduce the variance portion of the error is more challenging. In this paper, we investigate the use of ensemble fusion methods on a collection of low variance classifiers with a deterministic training algorithm. Several ways to increase the classifiers' independence are studied as well as different combination strategies.

In addition, we use the MDL approach for expert fusion, and estimate the accuracy of each classifier by using its description length. The description length, is then used as a weight in a convex combination of the experts, where a shorter description length, gives higher weight.

2 Training an Ensemble

Training of individual elements in an ensemble for improved independence can be done in several ways. The random forest algorithm [3] uses sub-space re-sampling for each node in the tree. AdaBoost [10] uses a fraction of the data (which earlier classifiers performed poorly on) to train a classifier, thus different classifiers train on different data-sets. We use both techniques to increase classifier's independence.

The output of the ensemble is given by:

$$f(x) = \sum_{k=1}^{M} a_k f_k(x) \qquad a_k \geq 0, \quad \sum_{k=1}^{M} a_k = 1, \tag{1}$$

where a_k is the weight of the kth expert. Other forms of combination will be discussed below.

2.1 Ensemble Generation

We have used "boosting" to generate data sets for the classifiers in the ensemble. In boosting, the first classifier is created with accuracy on the training set greater than chance, and then add new component classifiers to form an ensemble whose joint decision rule has arbitrary high accuracy on the training set. This technique trains successive components classifiers with the subset of data that is most informative. Given a data set $D = \{x_i, y_i\}_{i=1}^{N}$ where $x_i \in R^d$ and y_i is the class label. The input to the algorithm includes the maximum number of classifiers k_{max}, the size of re-sampled sub set of D $n < N$, and $\gamma \in [0, 1]$ the fraction of features for the random subspace selection. We use the following boosting algorithm:

- Initialize: empty ensemble, k=0
- while $k \leq k_{max}$
  - test the ensemble on the full training-set.
  - add to the current dataset D_k the misclassified patterns
  - select randomly $n - |D_k|$ from $D - D_k$ and add them to D_k
  - re-sample D_k on the features by using $\gamma * d$ features.
- end-loop

The above algorithm differs from the AdaBoost algorithm [10], as **all** the misclassified patterns are added to the next subset (with probability 1). Each classifier receives a different subsample of the training data and a different subsample of the input variables as in Random Forests. Thus, dependency between experts is greatly reduced.

2.2 Using MDL for Experts Fusion

The minimum description length (MDL) concept is typically used for model evaluation and selection. It is used here for weighting the different experts for optimal combination. In the MDL formulation, the coding of the data is combined with the coding of the model itself to provide the full description of the data [16]. MDL can be formulated for an imaginary communication channel, in which a sender observes the data D and thus can estimate its distribution, and form an appropriate model for that distribution. The sender then transmits the data using the code that is appropriate for the observed distribution (data model) but since the receiver does not observe the full data, the model has to be transmitted to the receiver (in a predefined coding). Noise and insufficient data to estimate the correct model lead to modeling errors. MDL has been constructed for the purpose of reducing such modeling errors. The MDL principle asserts that the best model of some data is the one that minimizes the combined cost of describing the model and the misfit between the model and the data.

This approach is formulated as follows: The sender composes a message which is consists of the model description with the length $\ell(M)$, and $\ell(D|M)$ specifies the length of the data given the model. The goal of the sender is to find the model that minimizes the length of this encoded message $\ell(M, D)$, called the description length:

$$\ell(M, D) = \ell(D|M) + \ell(M). \tag{2}$$

According to Shannon's theory, to encode a random variable X with a known distribution by the minimum number of bits, a realization of x has to be encoded by $-\log(p(x))$ bits [18,6]. Thus, the description length is:

$$\ell(M, D) = -\log(p(D|M)) - \log(p(M)), \tag{3}$$

where $p(M|D)$ is the probability of the output data given the model, and $p(M)$ is $a - priori$ model probability. Typically the MDL principle is used to select the model with the shorter description length. In this work we combine the experts by using the description length as a weight for the convex combination

in Eq. (1). Hinton and Camp [12] used zero-mean Gaussian distribution for the neural network weights. We follow this idea, and define the simplest Gaussian model prior,

$$p(M) = \frac{1}{(2\pi)^{1/2}\beta^d} exp(-\frac{\sum_{i=1}^{d} w_i^2}{2\beta^2}), \tag{4}$$

where d is the number of weights in the second layer and β is the standard deviation. Hinton and Camp [12] used a Gaussian with standard deviation α for encoding the output errors. In addition, we assume that the errors that the model makes are i.i.d with normal distribution. Clearly, a better assumption is that the error are binomial, but for purpose of estimating the relative probability of different methods the Gaussian assumption is good enough and easier to handle mathematically. We also assume that the patterns in the training set are independent.

Thus, the likelihood of the data given the model is:

$$p(D|M) = \frac{1}{(2\pi)^{\frac{NC}{2}}\alpha^{NC}} \exp(-\frac{\sum_{n=1}^{N}\sum_{k=1}^{C}(y_{kn} - t_{kn})^2}{2\alpha^2}), \tag{5}$$

where t_{kn} is the target value for the nth pattern at the k$'th$ class, y_{kn} is the respected output of the expert and α is the standard deviation. Under these assumptions the description length of the model is:

$$\ell(M, D) = \frac{NC}{2}\log(2\pi) + NC\log(\alpha) + \frac{\sum_{n=1}^{N}\sum_{k=1}^{C}(y_{kn} - t_{kn})^2}{2\alpha^2} + \frac{d}{2}\log(2\pi) + d\log(\beta) + \sum_{i=1}^{d}\frac{w_i^2}{2\alpha^2}. \tag{6}$$

Differentiating Eq. (6) with respect to α and equating to zero, we obtain:

$$\alpha^2 = \frac{1}{NC}\sum_{n=1}^{N}\sum_{k=1}^{C}(y_{kn} - t_{kn})^2. \tag{7}$$

Differentiating Eq. (6) with respect to β and equating to zero, we obtain:

$$\beta^2 = \frac{1}{d}\sum_{i=1}^{d} w_i^2. \tag{8}$$

Substituting Eq. (7) and Eq. (8) into Eq. (6) and discarding the constant terms we arrive at:

$$\ell(M, D) = NC\log(\alpha) + d\log(\beta) + \frac{d}{2}(1 + \log(2\pi)). \tag{9}$$

Equation (9) shows that the description length of the model is a tradeoff between the errors and the number of parameters d and their average value. Considering

description length as an energy and following Gibbs distribution formulation, we use the description length as a weight for each classifier in the convex combination as follows:

$$a_k = \frac{\exp(-\ell_k(M, D))}{\sum_{k=1}^{M} \exp(-\ell_k(M, D))}, \tag{10}$$

where M is the number of classifiers. Thus, a classifier with a shorter description length gets higher weight when combining the output of the ensemble. This is in contrast to other fusion methods when only the error is considered.

2.3 Other Expert Fusion Methodologies

In addition to the MDL fusion method, we have used five more classifier combination rules. They are described in this section.

I. Majority Rule: The first is the familiar majority vote; Here, the final decision is made by selection of the class with maximum number of votes in the ensemble.

II. Convex Combination: The second strategy relies on a convex combination using the error values from the first stage of training [17]. Let e_i be the classification error of the $i'th$ classifier. We set the weight of this classifier as follows:

$$a_k = \frac{1/e_k}{\sum_{i=1}^{M} 1/e_i}, \tag{11}$$

where M is the number of classifiers in the ensemble. The output of the ensemble is define as in Eq. (1).

III. Convex Combination: The third strategy relies on a convex combination using the error values from the first stage of training. Let e_i be the classification error of the $i'th$ classifier. To maximize the entropy of the ensemble, we set the weight associated with this classifier in accordance with Gibbs distribution as follows:

$$a_k = \frac{\exp(-e_k)}{\sum_{i=1}^{M} \exp(-e_i)}, \tag{12}$$

where M is the number of classifiers in the ensemble. The output of the ensemble is define as in Eq. (1).

IV. Dynamic Selection: The forth strategy involves dynamic selection of the best classifier for prediction of the output value when a novel pattern is given [11]. When the confidence of the best classifier (to be explained below) is below a given threshold, we use a dynamic combination of the classifiers to produce the output of the ensemble. We define a local accuracy for each classifier as follows. Let $k > 0$ and $x \in R^d$ be a novel pattern. Let $D_k(x)$ be the $k - nearest$ patterns

in the training set to x. Set the local accuracy of the current classifier on x the to be:

$$l(x) = \frac{\sum_{x_j \in D_k(x)} \delta(\arg\max_i p(y_i|x_j) - \arg\max_j(t_j))}{k}, \tag{13}$$

where $\delta(x)$ is one for $x = 0$ and zero otherwise, and t_j is the target for pattern x_j. Thus, the local accuracy is the number of correct classified patterns in the $k - neighborhood$ of x. Let $l_1(x)$ be the maximum local accuracy and let $l_2(x)$ be the next highest accuracy. Define the confidence level as follows:

$$cl(x) = \frac{l_1(x) - l_2(x)}{l_1(x)}. \tag{14}$$

We further define the weights for each classifier as follows:

$$a_k(x) = \frac{\exp(l_k(x))}{\sum_{i=1}^{M} \exp(l_i(x))}, \tag{15}$$

The combination rule in this case is given by:

- Compute the local accuracy for each classifier as in Eq. (13).
- Compute the confidence level $cl(x)$ from Eq. (14).
- If $\max cl(x) > threshold$, select the output of the best classifier, otherwise use Eq. (1).

V. Adaptive Boosting AdaBoost: The fifth strategy uses the AdaBoost algorithm [10]. The boosting algorithm AdaBoost - from adaptive boosting - allows the designer to continue adding weak learners until some desired low training error has been achieved. In AdaBoost each training pattern receives a weight which determines its probability of being selected for a training set for an individual component classifier. If a pattern is accurately classified then its chance to be selected again in a subsequent component classifier is reduced. In this way AdaBoost focuses on the difficult-to-classify patterns. We start by initialize the weights to be uniform. On each iteration we draw a training set at random according to these weights. Next we increase weights of misclassified patterns and decrease weights of patterns correctly classified. The new distribution of patterns is used to train the next classifier.

3 Results

The following methods of combination were used:

- ENS1-PRBFN Ensemble using a majority vote strategy.
- ENS2-PRBFN Ensemble using a convex combination (II) as in [17].
- ENS3-PRBFN Ensemble using a convex combination (III) of classifiers where the errors affect the weight of the different classifiers in the ensemble.

Table 1. Comparison of correct classification (percentage) of several ensemble fusion methods using 10 folds cross validation. Ensemble training is done by boosting.

Method	Breast-cancer	Glass	Iris	Vowel	Pima	Image
ENS1-PRBFN	96.5±1.4	96.2±3.5	95.3±4.5	85.2±3.2	77.4±3.2	91.2±6.5
ENS2-PRBFN	96.7±1.4	94.8±4.7	96.7±5.3	86.7±3.9	77.0±3.3	89.4±6.9
ENS3-PRBFN	96.8±1.8	94.2±4.6	96.0±4.5	87.1±4.0	74.9±4.4	90.4±5.6
ENS4-PRBFN	96.4±2.3	94.2±5.2	95.3±5.3	89.3±4.5	76.7±3.7	90.6±5.8
ENS5-PRBFN	97.0±1.5	95.2±3.8	96.0±4.5	87.3±2.7	78.8±2.8	91.9±3.8
ENS6-PRBFN	96.7±1.7	95.0±4.2	96.3±4.5	86.4±2.8	75.8±3.7	92.3±5.6
PRBFN	96.0±2.0	92.8±3.9	95.3±4.6	81.8±2.6	76.6±3.4	88.6±5.4

- ENS4-PRBFN The ensemble using k nearest neighbors (IV) to select the best classifier [11].
- ENS5-PRBFN Ensemble using MDL to set the weight of each classifier in the convex combination.
- ENS6-PRBFN Ensemble using AdaBoost algorithm (V) as describe in [10].
- PRBFN the single classifier as described in [4].

Data Sets Description

The Breast-cancer dataset from the UCI repository was obtained from Dr. William H. Wolberg at the University of Wisconsin Hospitals. This dataset has 9 attributes and two classes and the number of training patterns is 699. The task is to classify the patterns to Benign or Malignant.

The Glass dataset from the UCI repository has 10 attributes and 7 types of glasses. The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence if it is correctly identified! Ripley's best result on this data-set is 80% correct classification [15].

The Iris data-set [8] contains three classes, each with 50 instances. The classes refer to a type of iris plant. Each pattern is composed of four attributes. We used ten folds of cross validation in order to estimate the performance of the different classifiers.

The Deterding vowel recognition data [7,9] is a widely studied benchmark. This problem may be more indicative of a real-world modeling problem. The data consists of auditory features of steady state vowels spoken by British English speakers. There are 528 training patterns and 462 test patterns. Each pattern consists of 10 features and belongs to one of 11 classes that correspond to the spoken vowel. The speakers are of both genders. This data, unlike the other data-sets that have been studied, has a fixed training and test set. We provide results with cross validation in Table 1, where we compare experts on cross validated test set. Previous best score on the fixed test set was reported by Flake using SMLP units. His average best score was 60.6% [9] and was achieved with 44 hidden units. The single PRBFN network surpasses this result and achieves 68.4% correct classification with only 22 hidden units [4]. Thus, the additional

improvement that is obtained here using ensemble, puts this result at the top of performance for the vowel data set.

The Image Segmentation data from the UCI repository is composed of 210 instances for train. The instances were drawn randomly from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel. Each instance is a 3x3 region and has 19 continuous attributes. The task is to classify to one of the seven classes: brickface, sky, foliage, cement, window, path and grass.

The results in Table 1 are the average of three to ten times cross validation tests of ten folds on each of the data sets. The average performance is given in each entry as well as the variance of each predictor.

4 Discussion

The performance of ensemble methods on a tight architecture, which has been shown to have a low variance portion of the error, was evaluated on several benchmark data-sets. Partial independence of the experts was achieved via boosting or cross validation, and several methods were used for expert fusion. PRBFN is a deterministic classifier with a tightly controlled variance, therefore, simple fusion methods do not improve its performance. For instance, the best known result on the Glass data set is 80% accuracy [15], while PRBFN obtained 92.8% accuracy. We considered few approaches, to enhance the independence of several PRBFN, on the same data set. We note that the improvement of ensemble of such architectures is smaller than improvement that can be achieved on other architectures which posses higher variance, nevertheless, improvement still exists, and is sometimes quite significant.

Most of the fusion methods we have studied, do not appear to be significantly different in their improvement over a single expert. The key factor affecting the improvement is the degree of decorrelation of experts, which in this case, due to the deterministic nature of the architecture, depends on data re-sampling methods. The DCS fusion (ENS4) achieved a noticeable improvement on the Vowel data set. However, we note that this fusion method has large variance. This is due to the fact that quite often, prediction of a single expert (the best classifier) is selected, and thus there is no averaging that reduces the variance. The fusion based on the MDL principle (ENS5-PRBFN) appears to have a lower variance compared with other fusion methods. This is due to the higher emphasis that the MDL approach gives to lower description length and, thus, to simple models with a lower variance. The MDL fusion does not have to store the training data for future prediction and is thus faster in recognition compared with the DCS method.

References

1. L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
2. L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

3. L. Breiman. Random forests. Technical Report, Statistic Department University of California, Berkeley, 2001.
4. S. Cohen and N. Intrator. Automatic model selection in a hybrid perceptron/radial network. *Information Fusion Journal*, 3(4), December 2002.
5. S. Cohen and N. Intrator. A hybrid projection based and radial basis function architecture: Initial values and global optimization. *Pattern Analysis and Applications special issue on Fusion of Multiple Classifiers*, 2:113–120, 2002.
6. T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.
7. D.H. Deterding. *Speaker Normalisation for Automatic Speech Recognition*. PhD thesis, University of Cambridge, 1989.
8. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.
9. G.W. Flake. Square unit augmented, radially extended, multilayer percpetrons. In G. B. Orr and K. Müller, editors, *Neural Networks: Tricks of the Trade*, pages 145–163. Springer, 1998.
10. Y. Freund and R.E. Schapire. A decision theorethic generalization of on-line learning and application to boosing. *Journal of Computer and System Sciences*, 55(1):119–139, 1995.
11. G. Giacinto and F. Roli. Dynamic classifier selection. In *First International workshop on Multiple Classifier Systems*, pages 177–189, 2000.
12. G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Sixth ACM conference on Computational Learning Theory*, pages 5–13, July 1993.
13. M. P. Perrone and Leon N Cooper. When networks disagree: Ensemble method for neural networks. In R. J. Mammone, editor, *Neural Networks for Speech and Image processing*. Chapman-Hall, 1993.
14. Y. Raviv and N. Intrator. Bootstrapping with noise: An effective regularization technique. *Connection Science, Special issue on Combining Estimators*, 8:356–372, 1996.
15. B. D. Ripley. *Pattern Recognition and Neural Networks*. Oxford Press, 1996.
16. J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11:416–431, 1983.
17. F. Roli and G. Fumera. Analysis of linear and order statistic for combiners for fusion of imbalanced classifiers. In *Third International workshop on Multiple Classifier Systems*, pages 252–261, 2002.
18. C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423 and 623–656, 1948.