ENSEMBLE METHODS FOR CONNECTIONIST ACOUSTIC MODELLING

G.D. Cook   S.R. Waterhouse   A.J. Robinson

Cambridge University Engineering Department
Trumpington Street, Cambridge, UK.

ABSTRACT

In this paper we investigate a number of ensemble methods for improving the performance of connectionist acoustic models for large vocabulary continuous speech recognition. We discuss boosting, a data selection technique which results in an ensemble of models, and mixtures-of-experts. These techniques have been applied to multi-layer perceptron acoustic models used to build a hybrid connectionist-HMM speech recognition system. We present results on a number of ARPA benchmark tasks, and show that the ensemble methods lead to considerable improvements in recognition accuracy.

1. INTRODUCTION

When developing a classification or prediction system it is common practice to train a number of different models, and to retain the model which exhibits the best performance on a cross-validation data set. However, reports in the statistics and neural network literature suggest that improved performance can be achieved by combining the estimates of all the available models [1, 2, 3, 4]. Systems that form their estimate from the estimates of a number of different models are termed *ensembles* or *committee machines*. Ensembles of connectionist models have been applied to tasks ranging from optical character recognition [5] to analysis of satellite images [6], and have been shown to provide improvements in out-of-sample accuracy.

It has been shown that for best results the individual models which comprise an ensemble should not only demonstrate as low an error rate as possible, but should also make their errors in different regions of the input space [9, 10]. A simple method for building an ensemble is *bagging* [11] in which the training data is uniformly sampled with replacement to produce different training sets for the ensemble models. The expectation is that because the models are trained on different data sets they will pick out different properties present in the data, thus improving the performance when their outputs are combined.

This paper compares the performance of two types of ensemble techniques when used for connectionist acoustic modelling. The two techniques described are boosting [7] and mixtures-of-experts [8]. Boosting and mixtures-of-experts differ from simple ensemble methods. In boosting, each member of the ensemble is trained on patterns that have been filtered by previously trained members of the ensemble. In mixtures, the members of the ensemble, or *experts*, are trained on data that is stochastically selected by a gate which additionally learns how to best combine the outputs of the experts.

The boosting algorithm has been applied successfully to speech recognition for a small isolated digit task [12], and for large vocabulary recognition, in which a modified form of boosting was used to produce ensembles of recurrent neural networks [13]. In this paper we will describe how we use a combination of boosting and mixtures to improve recognition accuracy.

We first introduce boosting. This includes some background and a detailed description of the algorithm. The mixtures-of-experts architecture and learning algorithm is then described. The ensemble methods are used to produce multi-layer perceptron (MLP) models. These MLPs are used for acoustic modelling in a large vocabulary, hybrid connectionist-HMM continuous speech recognition system, and this is described in Section 4. We then present results on a number of ARPA benchmark tasks for both boosting, and a combination of boosting and mixtures-of-experts.

2. BOOSTING

Boosting is an algorithm that, under certain conditions, allows one to improve the performance of any learning machine, and was first designed in the context of the *distribution free*, or *probably approximately correct* (PAC) model of learning [14]. In the distribution free model (also known as the *strong* learning model), the learner must be able to produce a hypothesis with an error of at most ϵ, for arbitrarily small values of ϵ. Because the learner is receiving random examples there is also the possibility that the learner will receive an outlier (an example that is highly unrepresentative). The strong learning model therefore only requires that the learner succeeds in finding a good approximation to the target function with probability at least $1 - \delta$, where δ is an arbitrarily small constant.

In a variation of the distribution free model, called the *weak learning model*, the requirement that the learner must produce hypotheses with an error rate at most ϵ is relaxed. The leaner is required to produce hypotheses with error rate slightly less than 0.5. Thus the weak learning model requires that the learner be able to produce hypotheses that perform only slightly better that random guessing.

The main result of [14] is a proof that the strong and weak learning models are actually equivalent. A provably correct technique is given for converting any learning algorithm that performs only slightly better than random guessing into one that produces hypotheses with arbitrarily small error rates. The technique creates an ensemble hypothesis from three sub-hypotheses each trained on different distributions. Applying the technique recursively allows the error rate to be made arbitrarily small. In practice this is impossible because the algorithm quickly runs out of training data.

The boosting procedure used here is as follows: train a network on a randomly chosen subset of the available training data. This network is then used to filter the remaining training data to produce a training set for a second network. Flip a fair coin. If heads, examples are passed through the first network until it misclassifies a pattern, and this pattern is added to the second training set. If tails, examples are passed through the first network until it correctly classifies a pattern, and this pattern is then added to the second training set. This process is continued until enough patterns have been collected to train the second network. The coin flipping ensures that if the training set for the second network were passed through the first network it would have an error rate of 50%. After training the second network, the first and second networks are used to produce a training set for a third network in the following manner. The remaining training data is passed through the first two networks. If these two networks disagree on the classification of a pattern, this pattern is added to the training set for the third network. If the first two networks agree, the pattern is discarded. This process is continued until enough patterns have been collected to train the third network. The boosted networks are then combined using a simple linear merge, in which the ensemble output is formed by taking the average of the member network's outputs.

3. MIXTURES-OF-EXPERTS

The mixture of experts [8] is a different type of ensemble. The ensemble members or *experts* are trained with data which is stochastically selected by a *gate*. The gate in turn learns how to best combine the experts given the data. The training of the experts, which are typically single or multi-layer networks, proceeds as for standard networks, with an additional weighting of the output error terms by the posterior probability of selecting an expert given the current data point. In the case of classification, considered here, the experts use softmax output units. The gate, which is typically a single or multi-layered network with softmax output units is trained using the posterior probabilities as targets. The overall output of the mixture of experts is given by the weighted combination of the gate and expert outputs.

The mixture of experts is based on the principle of divide and conquer, in which a relatively hard problem is broken up into a series of easier to solve problems. By using the posterior probabilities to weight the experts and provide targets for the gate, we allow the effective data sets used to train each expert to overlap. This technique has already proved useful in speech recognition using mixtures-of-recurrent-networks [16]. In this paper we consider the use of mixtures of MLPs for acoustic modelling.

4. SYSTEM DESCRIPTION

The ensemble algorithms described in Sections 2 and 3 were used to produce an ensemble of MLP acoustic models. The MLP ensemble formed the acoustic model of the Cambridge University Engineering Department connectionist speech recognition system, ABBOT. In this system, the acoustic model is used to map each frame of acoustic data to a set of posterior phone probabilities. A Viterbi based training procedure is used to train the acoustic model. In this procedure, each frame of training data is assigned a phone label based on an utterance orthography and the current model. The back-propagation algorithm [17] is then used to train the MLP to map the acoustic input vector sequence to the phone label sequence. When this training has converged, the labels are reassigned using the Viterbi algorithm and the process iterated.

The posterior phone probabilities estimated by the acoustic model are used as estimates of the observation probabilities in an HMM framework. Given new acoustic data and the connectionist-HMM framework, the maximum *a posteriori* word sequence is extracted using the NOWAY decoder. NOWAY is a single pass, start synchronous stack decoder designed to exploit the features of the hybrid connectionist-HMM approach [18]. A more complete description of the system can be found in [19].

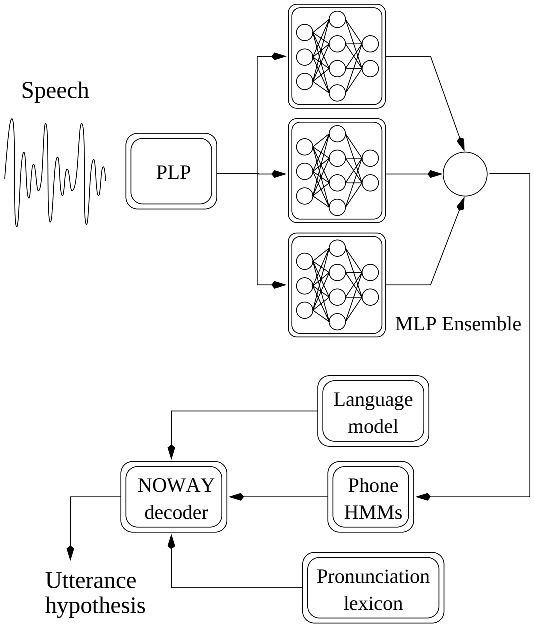

Figure 1: The ABBOT hybrid connectionist-HMM speech recognition system with an MLP ensemble acoustic model

The acoustic representation used is twelfth order perceptual linear prediction cepstral coefficients [20] plus energy. To capture dynamic information the acoustic features are augmented with first order difference parameters. The combination of acoustic features and difference parameters forms a *frame*. An input window of nine contiguous frames is used to capture contextual information, with the central frame of the window given by the current observation. The hidden units of the network use a logistic activation functions, while the output units use a softmax activation function. The cross-entropy error criterion is used during training.

5. EXPERIMENTS AND RESULTS

This section describes the experiments and presents results. The data used for both training and testing is first described. The results of experiments to evaluate boosting are then presented. We then asses the performance of a mixtures-of-experts architecture which is bootstrapped from a boosting ensemble.

The training data used for the experiments is the short term speakers from the Wall Street Journal corpus. This consists of approximately 36,400 sentences from 284 different speakers (SI284). The ensemble methods have been evaluated on a number of different ARPA benchmark tasks. A brief description of these tasks is given below.

dt_s5_93

November 1993 Spoke 5 development test set. This is a 5000 word, closed vocabulary test. The system uses the standard ARPA bigram language model. The results reported here are for the close-talking Sennheiser microphone only.

dt_s6_93
November 1993 Spoke 6 development test set. This is also a 5000 word, closed vocabulary test, and the system uses the standard ARPA bigram language model.

et_h2_93
November 1993 Hub 2 evaluation test set. Also a 5000 word closed vocabulary task. The system uses the standard ARPA trigram language model.

et_h1_93
November 1993 Hub 1 evaluation test set. This is a large vocabulary task, the prompting texts for which are from the Wall Street Journal 64k word texts pools. The system uses a 20k-word vocabulary, and a trigram language model built using 35 million words of text from the WSJ0 corpus.

et_h1_94
November 1994 Hub 1 evaluation test set. This is an open-vocabulary task, which includes prompts from various newspaper sources, including the Wall Street Journal, the Los Angeles Times, the Washington Post, and the New York Times. The system uses a 64k-word vocabulary and a language model built using 237 million words from the CSR-LM-1 corpus [21].

5.1. Boosting Evaluation

The boosting algorithm described in Section 2 has been used to produce an ensemble of MLP acoustic models. The first network is trained on 1.5 million frames randomly selected from the available training data (approximately 15 million frames). This is then used to filter the unseen training data to select frames for training the second network. The first and second networks are then used to select data for the third network as described in Section 2.

The distribution of phone classes in the training data for an acoustic model should be representative of the distribution of phone classes in the test data. This ensures that the network produces accurate posterior probability estimates. The data selection process alters the prior distribution of phone classes in the training data. Therefore the network estimates must be modified to account for the altered priors. This can be achieved using Bayes' theorem:

$$\mathrm{P}'(q_t^i|\mathbf{u}_t) = \frac{\mathrm{P}(q_t^i|\mathbf{u}_t)\mathrm{P}_{\mathrm{test}}(q^i)}{\mathrm{P}_{\mathrm{train}}(q^i)}, \qquad (1)$$

where $\mathrm{P}(q_t^i|\mathbf{u}_t)$ is the estimated posterior probability of phone class i at time t, $\mathrm{P}_{\mathrm{train}}(q^i)$ is the prior for phone class i in the training data, and $\mathrm{P}_{\mathrm{test}}(q^i)$ is the prior for phone class i in the test data. Of course $\mathrm{P}_{\mathrm{test}}(q^i)$ is dependent on the test set, and is not known. $\mathrm{P}_{\mathrm{test}}(q^i)$ is therefore estimated form the entire SI284 training set. $\mathrm{P}'(q_t^i|\mathbf{u}_t)$ is then normalised to ensure that it is a proper distribution.

The boosted acoustic models have been combined using two different methods. In the first of these, the ensemble output is formed as follows: if networks one and two classify the input frame as the same phone, the output of network one is used as the observation probabilities in the HMM as described in Section 4. If the networks disagree on the classification of the input frame, the output of network three is used as the observation probabilities. This method is denoted `vote`. The second method, denoted `average`, forms a simple average of the network outputs, and this average is used as the observation probabilities.

Model	**WER**	**WER Red$^{\underline{n.}}$**
Single MLP	16.0%	—
Boosted (vote)	14.6%	8.75%
Boosted (average)	12.9%	19.38%

Table 1: Performance of boosted MLP acoustic models on the 1993 Hub 2 evaluation test set.

Table 1 shows results on the November 1993 Hub 2 evaluation test set. As can be seen boosting has considerably reduced the word error rate when compared to a single MLP acoustic model. The word error rate reductions are statistically significant at p= 0.05, using a two-tailed t-test with the null hypothesis that there is no performance difference between the single MLP and the boosting methods. The ensemble which uses the linear average to combine the models is also significantly better that the voting scheme at p=0.05. All the significance tests were performed using the NIST package score v3.6.2, and the matched pair sentence segment (word error) test[1].

Test Set	Word Error Rate		WER Red$^{\underline{n.}}$
	Single MLP	Boosting	
dt_s5_93	20.4%	16.5%	19.1%
dt_s6_93	17.7%	14.8%	16.4%
et_h1_93	25.2%	21.8%	13.5%
et_h1_94	24.7%	20.7%	16.2%

Table 2: Evaluation of the performance of boosting MLP acoustic models.

The performance of the boosted MLP ensemble and the linear average combination scheme has also been evaluated on a number of ARPA benchmark tests. The results are summarised in Table 5.1. As can be seen, boosting has resulted in considerable improvements in performance for all of the test sets. In each case the system with boosted ensemble acoustic models performs significantly (at p=0.05) better than the system which uses a single MLP acoustic model.

5.2. Combining Boosting and Mixtures-of-Experts

In order to improve the performance of the ensembles further, three methods were investigated. Each of these focussed on the use of mixtures-of-experts to combine the ensemble members.

In the first experiment, the boosted models were combined using a gating network which was retrained on the training data, with the boosted models held fixed. Two methods of training the gate were investigated: using winner take all (WTA) in which the best performing ensemble is assigned probability 1.0 and the rest 0.0, and a soft assignment scheme [8]. It was found, however that neither method offered an advantage over a simple linear combination (results not shown). The reason for this may be that the set of combination rules to be learnt by the

[1] The NIST testing software is available via anonymous ftp from `ftp://jaguar.ncsl.nist.gov/pub/score_3.6.2.tar.Z`.

gate is too complex, or that the assumption of veridical responsibilities [22] is violated in the ensemble members.

The final pair of experiments compared two techniques, training a mixture of experts from a flat start on the entire training data (denoted Mixed in Table 5.1) and using the boosted models as a bootstrap for a mixture-of-experts (denoted Boost+Mixed in Table 5.1) and retraining on the entire training data. As can be seen from the table, the performance of the mixtures trained from a flat start (Mixed) was comparable with the Boosting method. However, the Boost+Mixed method gave an additional improvement over the Boosting, which was significant at the p=0.05 level. The reason for the failure of the flat start mixtures is unclear. However, it is clear that the use of an appropriate bootstrap, such as the boosted ensemble is important in obtaining well trained models for the mixtures. This result is consistent with the concept of using k-means as a bootstrap for Gaussian mixtures trained with EM.

Test Set	Word Error Rate		
	Boosting	Mixed	Boost+Mixed
dt_s5_93	16.5%	17.3%	14.2%
dt_s6_93	14.8%	14.5%	11.5%
et_h2_93	12.9%	13.1%	10.9%
et_h1_94	20.7%	20.5%	16.7%

Table 3: Evaluation of the performance of mixture-of-experts acoustic models

6. SUMMARY

This paper has described two ensemble methods, boosting, and mixture-of-experts. These methods have been used to produce connectionist acoustic models, and have been shown to improve recognition performance by up to 35%.

7. ACKNOWLEDGEMENTS

This work was partially funded by ESPRIT project 20007 SPRACH. One of the authors (S.R. Waterhouse) is funded by an EPSRC grant.

References

[1] M. Stone. Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society Series B*, 36:111–147, 1974.

[2] L. Breiman. Stacked Regressions. Technical Report 367, Department of Statistics, University of California Berkeley, 1992.

[3] D. Wolpert. Stacked Generalization. *Neural Networks*, 5:241–259, 1992.

[4] M.P. Perrone and L.N. Cooper. When Networks Disagree: Ensemble Methods for Hybrid Neural Networks. In R.J. Mammone, editor, *Neural Networks for Speech and Image Processing*. Chapmann-Hall, 1993.

[5] L. Bottou, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, L.D. Jackel, Y. LeCun, U.A. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of Classifier Methods: A Case Study in Handwritten Digit Recognition. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume II, pages 77–82, Jerusalem, Israel, October 1994.

[6] P. Smyth, U. Fayyad, M. Burl, and P. Perona. Inferring Ground Truth from Subjective Labelling of Venus Images. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.

[7] H. Drucker, R. Schapire, and P. Simard. Improving Performance in Neural Networks Using a Boosting Algorithm. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Neural Information Processing Systems*, pages 42–49. Morgan Kauffmann, 1993.

[8] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 1991.

[9] L.K. Hansen and P. Salamon. Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001, 1990.

[10] A. Krogh and J. Vedelsby. Neural Network Ensembles, Cross Validation, and Active Learning. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1995.

[11] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

[12] S.R. Waterhouse and G.D. Cook. Ensembles for Phoneme Classification. *Advances in Neural Information Processing Systems*, 9, 1996.

[13] G.D. Cook and A.J. Robinson. Boosting the Performance of Connectionist Speech Recognition. *International Conference in Spoken Language Processing*, 3:1305–1308, 1996.

[14] R.E. Schapire. The Strength of Weak Learnability. *Machine Learning*, 5:197 – 227, 1990.

[15] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, and V. Vapnik. Boosting and Other Ensemble Methods. *Neural Computation*, 6:1289–1301, 1994.

[16] S. R. Waterhouse, D. J. Kershaw, and A. J. Robinson. Smoothed Local Adaptation of Connectionist Systems. In *International Conference in Spoken Language Processing*, October 1996.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume I: Foundations, pages 318–362. MIT Press/Bradford Books, Cambridge, MA, 1986.

[18] S. Renals and M. Hochberg. Decoder Technology for Connectionist Large Vocabulary Speech Recognition. Technical Report CS-95-17, Dept. of Computer Science, University of Sheffield, 1995.

[19] A.J. Robinson, M.M. Hochberg, and S.J. Renals. The Use of Recurrent Neural Networks in Continuous Speech Recognition. In C. H. Lee, K. K. Paliwal, and F. K. Soong, editors, *Automatic Speech and Speaker Recognition – Advanced Topics*, chapter 19. Kluwer Academic Publishers, 1995.

[20] H. Hermansky. Precpetual Linear Prediction (PLP) Analysis of Speech. *Joint Acoustic Society of America*, 87(4):1738–1752, April 1990.

[21] F. Kubala. Design of the 1994 CSR Benchmark Tests. In *Spoken Language Systems Technology Workshop*, pages 41–6. ARPA, January 1995.

[22] R. A. Jacobs. Methods for combining experts probability assessments. *Neural Computation*, 7(5):867–888, 1995.