BOOSTING THE PERFORMANCE OF CONNECTIONIST LARGE VOCABULARY SPEECH RECOGNITION

Gary Cook *Tony Robinson*

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, England.
Tel : [+44] 1223 332754 Fax : [+44] 1223 332662
email : gdc,ajr@eng.cam.ac.uk

ABSTRACT

Hybrid connectionist-hidden Markov model large vocabulary speech recognition has, in recent years, been shown to be competitive with more traditional HMM systems [4]. Connectionist acoustic models generally use considerably less parameters than HMM's, allowing real-time operation without significant degradation of performance. However, the small number of parameters in connectionist acoustic models also poses a problem — how do we make the best use of large amounts of training data? This paper proposes a solution to this problem in which a "smart" procedure makes selective use of training data to increase performance.

1. INTRODUCTION

Traditional HMM systems show a considerable improvement in performance when more training data is available [9]. However, blindly throwing more training data at connectionist models results in increased training times, without significant improvements in performance. This is most probably due to the compact nature of connectionist acoustic models. We are able to accurately estimate the network parameters from relatively small amounts of data. This effect can be seen in the results presented in Section 5.

The aim of the work presented here is to develop a principled method for using large amounts of training data to improve performance. The method is based on *boosting*, a procedure which results in an ensemble of networks [2]. Unlike many ensembles in which each network is trained on the same data [3], the networks in a boosting ensemble are trained sequentially on data that has been filtered by the previously trained networks in the ensemble. This ensures only data that is likely to result in improved generalisation performance is used for training. This is particularly important for speech data, as a considerable amount of computation is required to train networks for large vocabulary speech recognition, and we do not want to waste resources training on data that will not result in an increase in performance.

We first describe ABBOT, an hybrid connectionist-HMM speech recognition system developed at Cambridge University Engineering Department. We then describe the original boosting procedure, and briefly dicuss the use of this procedure with neural networks. Next we present a novel boosting procedure suitable for use with temporal data such as speech. Results are presented on two large vocabulary continuous speech tasks for both context-independent and context-dependent acoustic models.

2. SYSTEM DESCRIPTION

The Cambridge University Engineering Department connectionist speech recognition system (ABBOT) uses a hybrid connectionist - HMM approach. A recurrent network acoustic model is used to map each frame of acoustic data to posterior phone probabilities. The recurrent network provides a mechanism for modelling the context and the dynamics of the acoustic signal. The network has one output node per phone, and generates all the phone probabilities in parallel.

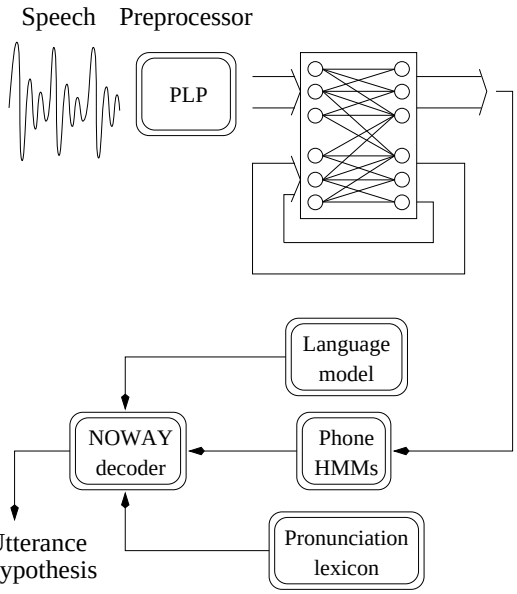

Figure 1: Hybrid Connectionist-HMM Speech Recognition System

A Viterbi based training procedure is used to train the acoustic model. Each frame of training data is assigned a phone label based on an utterance orthography and the current model. The back-propagation through time algorithm is then used to train the recurrent network to map the acoustic input vector sequence to the phone label sequence. The labels are then reassigned and the process it-

erated. A complete description of the acoustic training is given in [11].

The posterior phone probabilities estimated by the acoustic model are then 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 then extracted using the NOWAY decoder. NOWAY is a single pass, start synchronous decoder designed to exploit the features of the hybrid connectionist-HMM approach [10]. A more complete description of the system can be found in [5].

3. 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 [13]. 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 [13] 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 produces an ensemble hypothesis from three sub-hypotheses trained on different distributions. If the three sub-hypotheses have an error rate of $\alpha < 0.5$ with respect to the distribution on which they were trained, then the resulting ensemble hypothesis will have an error rate of $3\alpha^2 - 2\alpha^3$, which may be significantly less than α.

3.1. Boosting Neural Networks

The first practical application of a boosting procedure was for the optical character recognition task [2]. An ensemble of feedforward neural networks was trained using supervised learning. The boosting procedure was 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, pass examples through the first network until it misclassifies a pattern, and add this pattern to the second training set. If tails, pass example through the first network until in correctly classifies a pattern, and add this pattern to the second training set. Continue until enough patterns have been collected to train the second network.

After training the second network, the first and second networks are used to produce a training set for a third network. Pass the remaining training data through the first two networks. If the two disagree on the classification of a pattern add this to the training set for the third network. If the first two networks agree, discard the pattern. Continue until enough patterns have been collected to train the third network. Using this method the authors reported a reduction in error rate on ZIP codes from the United States Postal Service of 28% compared to a single network. Further experiments have shown that boosting also results in a significant reduction in error rate compared to an ensemble of networks, each trained on the same data [1].

4. BOOSTING ACOUSTIC MODELS

The original boosting procedure is designed for static pattern recognition problems — it is necessary to select frames from the pool of available training data. However, speech is temporal, often with high degrees of correlation between successive samples due to coarticulation effects. There are two basic approaches used to model the dynamic nature of speech signals with neural networks: either window the input and treat the time domain data like any other data, or use some internal storage to maintain a current state. In both cases the order of the input frames is important so we cannot select frames from the pool training data.

To overcome this problem we have developed a number boosting procedures for use with speech data.

Boost 1 : Train a network on a randomly selected subset of the training *sentences*. Use this network to filter the remaining data to produce a training set for a second network. Compute the frame recognition rate of the first network on new sentences from the training data. Select training data for a second network such that half are the sentences on which the frame recognition rate of the first network is lowest, and the other half are the sentences on which the frame recognition rate of the first network is the highest.

Boost 2 : Train a network on a randomly selected subset of the training sentences. Compute the frame recognition rate of the first network on new sentences from the training data. Select those sentences on which the frame recognition rate of the first network is lowest to train a second network.

Boost 3 : Train a network on a randomly selected subset of the training sentences. Use this network to produce posterior phone probabilities. These posterior phone probabilities are then used as observation probabilities within an HMM framework. Decode the new sentences from the training data, and compute the per sentence word error rate. Select those sentences for which the word error rate is highest to train a second network.

In all cases the number of sentences chosen for training the second network is the same as were used to train the first network.

The evaluation of the performance of the boosted networks is the same for each boosting procedure. The original boosting procedure combines the outputs of three networks using a simple voting scheme. However, since we wish to use the network outputs as posterior phone probabilities, this is not possible in this case.

A simple method to merging the outputs of several networks is to form a linear combination,

$$y_i = \sum_{k=1}^{K} \beta_{ik} y_i^{(k)} \quad (1)$$

where $y_i^{(k)}$ is the i^{th} output of the k^{th} network. In order to maintain a probabilistic interpretation of the combined network outputs the βs must be *tied* (i.e. $\beta_{ik} = \beta_k$), *sum-to-one* (i.e. $\sum_{k=1}^{K} \beta_{ik} = 1$) and *non-negative* (i.e. $\beta_{ik} \geq 0$). In this work we have used a simple average with $\beta_{ik} = 1/K$.

An alternative is to combine the outputs after conversion to the log domain,

$$\log y_i = \frac{1}{K} \sum_{k=1}^{K} \log y_i^{(k)} - Z \quad (2)$$

where Z is a normalisation constant such that $\mathbf{y}$ is a probability distribution. Probability distribution combination in the log domain can be interpreted as choosing the distribution which minimises the average Kullback-Leibler information [12].

Combining the outputs of the networks in this manner also allows the boosting procedures described above to be extended to three or more networks. New training data can be filtered by scoring the combined output of previous networks.

5. EXPERIMENTS & RESULTS

The boosting procedures introduced in section 4 have been evaluated on the November 1993 Hub 2 evaluation test data [7]. The test utterances are from a closed 5,000 word, non-verbalised punctuation vocabulary, using a standard trigram language model. The training data we have used is the short-term speakers from the Wall Street Journal corpus. This consists of 36309 sentences from 284 different speakers. Our standard acoustic model is trained on the short term speakers from WSJ0 which consists of 7200 sentences from 84 speakers.

Training Data	No. Sentences	Word Error Rate
WSJ0	7200	11.2%
WSJ0 + WSJ1	7200	11.5%
WSJ0 + WSJ1	14400	11.5%

Table 1: Effect of training data on error rates for the November 1993 Hub 2 evaluation test data

We first evaluated the effect of increasing the amount of training data. All of the training data from WSJ0 has been used to train the baseline acoustic model. When using data from WSJ0 and WSJ1 we randomly select sentences from the available training data. As can be seen from the results in table 1, when randomly selecting data the error rate increases slightly. Training on twice the amount of data does not result in any improvement in performance. We believe this is due to the compact representation offered by the recurrent neural network acoustic model. The parameters of the model can be accurately estimated from 7200 sentences, and so increasing the amount of training data has no effect.

Each of the boosting procedures has been evaluated on the November 1993 Hub 2 evaluation data. The results can be seen in table 2. The row random indicates the results achieved by combining the output of two networks, each trained on different randomly selected data sets. The network outputs have been combined using the simple weighted average described in section 4. The best result is achieved using boosting procedure **boost2** in which the second network is trained on those sentences for which the first network's frame recognition rate is lowest.

Boosting Procedure	Word Error Rate	Improvement
baseline	11.2%	—
random	10.4%	7.1%
boost1	10.1%	9.8%
boost2	9.5%	15.2%
boost3	10.4%	7.1%

Table 2: Comparison of different boosting procedures on the November 1993 Hub 2 evaluation test data

We believe the poor performance of the **boost3** algorithm is due to influence of the language model used for decoding[1]. The word error rate for each sentence is strongly influenced by the language model score. The out-of-vocabulary rate and degree of mismatch between the training sentence text and the language model will be differ for each sentence. The word error rate reflects not only the performance of the acoustic model but also that of the language model. Thus the boosting procedure selects training data on which the combined acoustic and language model scores are lowest, and not those on which the performance of the acoustic model is poorest.

The original motivation for combining the outputs of two or more acoustic models came from analysis of the recurrent network. The recurrent network structure is time asymmetric, and training a network to recognise forward in time will result in different dynamics than training to recognise backwards in time. As a result, ABBOT uses the combined outputs of networks trained both forward and backward in time. We have used the boosting procedure **boost2** to produced boosted acoustic models trained backwards in time.

Acoustic Models	Word Error		Improvement
	Standard	Boosted	
forward	11.2%	9.5%	15.2%
backward	11.5%	10.0%	13.0%
fwd+bkwd	9.8%	8.4%	14.3%

Table 3: Comparison of standard and boosted systems for different acoustic models on the November 1993 Hub 2 evaluation test data

As can be seen from table 3 boosting results in significant reductions in error rates for forward, backward, and combined forward and backward acoustic models.

We have extended the boosting procedure to three networks for

[1]The language model has a 20,000 word vocabulary and was constructed from verbalised punctuation texts which comprise the standard ARPA 1994 language model

forward in time acoustic models. New training data was passed through each of the first two networks. The outputs of each network were then combined using a simple average. The frame recognition rate of the combined networks was then evaluated. The third boosting network was trained on sentences on which the performance of the first two networks was lowest. We found that a third network resulted in no improvement in performance.

The first network filters new training data and selects those sentences on which its performance is poor. We then use a second network to learn to classify this data. The combination of these two networks performs much better on unseen data. This is because those data that are outliers for the first network have been learnt by the second network. Thus a combination of the networks is able to correctly classify data from a much wider distribution than a single network. We believe the addition of a third network has no effect on overall performance because filtering selects data on which the performance of the first two networks is already relatively good. The performance of a trained network on this data is only marginally better than the combination of the first two networks.

We have also evaluated the boosting procedure **boost2** on the 1995 Hub 3 evaluation utterances, which are from an open, 60,000 word, non-verbalised punctuation vocabulary, using a standard trigram language model [8]. The results presented are for the contrast C0 Sennheiser microphone data. For this task we have used both context independent, and limited context dependent acoustic models. For more details of the context dependent system, and changes made to accommodate the Hub 3 task, see [6].

Acoustic Model	Word Error		Improvement
	Standard	Boosted	
CI	15.3%	13.4%	12.4%
CD	12.5%†	10.8%	13.6%

Table 4: Results for both context independent and context dependent boosted acoustic models on the 1995 Hub 3 evaluation test data. Note that the CD system uses speaker adaptation. † indicates the official CU-CON entry for the 1995 ARPA evaluations

The boosting procedure has again resulted in a considerable reduction in error rates. Forward and backward boosted acoustic models, with log domain merging were used for both the context independent and context dependent experiments.

6. CONCLUSIONS

This paper has presented three novel boosting procedures for use with neural network acoustic models. The best of these has been shown to result in a reduction of word error rate of 15.2% over a single network, and 8.7% over an ensemble of networks trained on random data. In addition, the boosting procedure has been shown to work with both context independent and context dependent acoustic models.

7. ACKNOWLEDGEMENTS

This work was partially funded by ESPRIT project 6487 Wernicke. The authors would like to acknowledge MIT Lincoln Laboratory and CMU for providing the language model, and LIMSI-CNRS and ICSI for providing the pronunciation lexicon for the Hub 3 experiments. We also acknowledge Dan Kershaw for his help in obtaining results on the Hub 3 data.

8. REFERENCES

1. H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, and V. Vapnik. Boosting and Other Ensemble Methods. *Neural Computation*, 6:1289–1301, 1994.
2. 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.
3. L.K. Hansen and P. Salamon. Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001, 1990.
4. M.M. Hochberg, G.D.Cook, S.J. Renals, A.J. Robinson, and R.S. Schechtman. The 1994 ABBOT Hybrid Connectionist-HMM Large-Vocabulary Recognition System. *Proc. of Spoken Language Systems Technology Worshop, ARPA*, 1995.
5. M.M. Hochberg, S.J. Renals, and A.J. Robinson. ABBOT: The CUED hybrid connectionist-HMM large-vocabulary recognition system. *Proc. of Spoken Language Systems Technology Worshop, ARPA*, March 1994.
6. D.J. Kershaw and A.J. Robinson. The 1995 ABBOT LVCSR System for Multiple Unknown Microphones. To appear in ICSLP, 1996.
7. D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo, B.A. Lund, and M.A. Pryzbocki. 1993 Benchmark Tests for the ARPA Spoken Language Program. *ARPA Workshop on Human Language Technology*, pages 51–73, March 1994. Merrill Lynch Conference Center, Plainsboro, NJ.
8. D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garoforo, A.F. Martin, and M.A. Przybocki. 1995 Hub-3 NIST Multiple Microphone Corpus Benchmark Tests. *ARPA Speech Recognition Workshop*, February 1996. The Arden Conference Center, Columbia University, NY.
9. D. Pye, P.C. Woodland, and S.J. Young. Large Vocabulary Multilingual Speech Recognition uisng HTK. *European Conference on Speech Communication and Technology*, pages 181–184, September 1995.
10. 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.
11. A.J. Robinson. An application of Recurrent Nets to Phone Probability Estimation. *IEEE Transactions on Neural Networks*, 5(2):298 – 305, March 1994.
12. Tony Robinson, Mike Hochberg, and Steve 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.
13. R.E. Schapire. The Strength of Weak Learnability. *Machine Learning*, 5:197 – 227, 1990.