Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 5
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima,
Soo-Young Lee, and Xin Yao (Editors)

HYBRID HMM-NN FOR SPEECH RECOGNITION AND PRIOR CLASS PROBABILITIES

Dario Albesano, Roberto Gemello and Franco Mana

Loquendo S.p.A.
Via Nole, 55
10149 Torino Italy

ABSTRACT

During the last years, speech recognition technologies have started their migration from research laboratories to real word applications gaining market shares. Although this shows that paradigms like Neural Networks have reached a high level of accuracy in modeling speech, it must be realized that there is still room for improving recognition performances exploiting the feedbacks coming from the applicative fields. In these cases, in fact, precious application dependent speech material can be recorded, and used to train the acoustic models in order to improve the behaviour of the recognizer on target dictionaries. The best results can be achieved when an iterative, refining process is set up.

Unfortunately, speech corpora coming from the field are seldom phonetically balanced and this can cause the performances of the Neural Network to get worse, wasting the benefits of the refining process.

In this paper, the problem of Prior Probability normalization has been faced and a method for Prior Probability normalization has been investigated, with the important characteristic of being applicable simply through a modification of the biases at the end of the training phase (therefore on trained nets).

An experimentation on several languages is reported, showing the Prior Probability normalization seems quite useful to improve recognition accuracy and to get rid of some undesired effects of training data-bases not perfectly phonetically balanced.

1. INTRODUCTION

Neural Networks (NN) are playing an increasing role in the field of speech processing. In particular in the speech recognition area the refinement of hybrid Hidden Markov - Neural Networks models (HMM-NN) has led them to reach and often overcome the performances obtained with the classical Continuous Density Hidden Markov Models (CDHMM).

While CDHMM are memoryless (they look at just one speech frame) and are trained with a characterizing method (Baum-Weltch forward-backward), HMM-NN look at a temporal context and use an intrinsic discriminative training method (error backpropagation) potentially achieving a better separation of similar acoustic classes.

These NN characteristics have been studied in the last years by many research groups and have been exploited to build effective state-of-the-art recognition systems [1].

Hybrid HMM-NN models have been investigated by several research teams: [2] and [3] have introduced the Connectionist Viterbi Training to enhance HMM based continuous speech recognition; in [4] they have proposed connectionist probability estimation to overcome some limitations of HMM-only recognizers; [5] and [6] described a recurrent neural network / HMM approach to large vocabulary, speaker independent speech recognition system and the authors of this paper introduced a phoneme based hybrid HMM-NN model [7] whose training procedure employs an integrated gradual movement of bootstrap speech segmentations.

The activity described in these pages has been originally triggered by the desire to improve the performances obtained by the hybrid HMM-NN models, in particular in correspondence to training data-bases including applicative components (hence not necessarily phonetically balanced) and to deal with training data with class probabilities that are not always representative of test conditions.

As well known, training components coming from the field are very useful and can boost recognition performance in a significant way, especially when they are part of an iterative process of feedback and refinement. It is very important, then, to find out a method to overcome the drawback of the imperfect phonetical balancing, typical of this kind of data-bases. Thus, the problem of Prior Probability normalization has been faced, with the aim of finding a normalization method based only on the training set (therefore not using a validation set, not always available) and of investigating the usefulness of normalizing all the priors.

A method of Prior Probability normalization has been investigated, with the important characteristic of being implementable through a modification of the biases of the trained net, without the need of retraining the neural network.

An experimentation on several languages is reported, showing that Prior Probability normalization seems quite useful to improve recognition accuracy and to get rid of polarized prior probabilities deriving from training databases not well phonetically balanced.

2. MLPS AS PROBABILITY ESTIMATORS

MLPs may be used to estimate probabilities, and presently we use them to estimate the probabilities of the acoustical classes given a piece of input signal.
As shown by many authors [8] [9], a MLP trained to perform classification is a class-conditional posterior probability estimator. That is, after a '1-from-N' training, where there is a one-to-one correspondence between output units and classes, a MLP output value, given the input x, will be an estimate of the posterior probability of the corresponding class q_k given the input, $P(q_k|x)$. This result holds for training with various error functions (including relative entropy – the one we use currently – and sum squared) and for outputs units constrained to be non-negative and less than 1, as the common sigmoids or the softmax units (that we are currently using).

Thus the network outputs approximate Bayesian probabilities (posterior probability):

$$p(q_k \mid x_n) \;=\; \frac{p(x_n \mid q_k)p(q_k)}{p(x_n)}$$

which implicitly contains the a priori class probability $p(q_k)$.

It is thus possible to vary a priori class probability during classification without retraining, since these probabilities occur only as multiplicative terms in producing the network outputs. As a result, class probabilities can be adjusted during use of a classifier to compensate for training data with class probabilities that are not representative of actual use or test conditions [4][9].

3. PRIOR CLASS PROBABILITIES AND LIKELIHOODS

3.1 Advantages and disadvantages

The use of posterior probability (which implicitly contains the a priori class probability) can be twofold:
it can be an advantage if the priors of the test set are very coherent with those of the training set;
it can be a disadvantage in the opposite situation.

In particular, posterior probability could be used when:

- the training DB is phonetically well balanced;
- the priors of the test set are similar to those of the language.

Posterior probability should not be used when:

- the training DB contains many applicative components (i.e. it is not phonetically balanced);
- the priors of the test set are quite different from those of the language.

The influence of the priors component may be harmful at the boundaries of the classes, because if the $P(x| q_k)$ is small for several classes, the a priori component $P(q_k)$ of the classes may become predominant and mask the other term, so damaging a fine separation of the classes.

According to some authors [4] HMM decoding requires likelihoods and we need to divide each of the MLP outputs by the relative class priors to give us a scaled likelihood that can be used in HMM for recognition.
In practice the difference of using posterior probabilities or scaled likelihoods may be not so relevant for certain languages where the priors of the acoustic classes are more smoothed but can be quite important for other languages like American, where the relative frequencies of training data are much more difficult to be kept balanced and similar to the priors of the language.

3.2 Division by Priors

A direct way of transforming posterior probabilities to scaled likelihoods is to divide the network outputs by the relative frequencies of the classes q_k, estimated on the training set:

$$p(x_n \mid q_k) \;=\; \frac{p(q_k \mid x_n)p(x_n)}{p(q_k)} \propto \frac{p(q_k \mid x_n)}{p(q_k)}$$

The p(x) is constant during recognition for all classes and can be ignored.

It is possible to implement the division by priors as a change of the biases of the output units of the trained network: $B_k = b_k - \log(p(q_k))$, without the need of changing the recognition engine.

In fact:

$$\frac{p(q_k \mid x_n)}{p(q_k)} \equiv \frac{o_k}{p_k} = \frac{\dfrac{e^{x_k+b_k}}{e^{\log(p_k)}}}{\sum_j e^{x_j+b_j}} = \frac{e^{x_k+b_k-\log(p_k)}}{\sum_j e^{x_j+b_j}} \propto \frac{e^{x_k+B_k}}{\sum_j e^{x_j+B_j}}$$

where $p_k = p(q_k)$ is the prior probability of class k, $x_k = \sum_i w_{ki} hid_i$ is the weighted sum of the outputs of the hidden units incoming to unit k, b_k are the original biases, and $B_k = b_k - \log(p(q_k))$ are the new biases of the output units.

The complete algorithm for priors division is the following:

- estimate the prior probabilities of the classes by computing the relative frequencies of the training set classes;
- during this computation keep into account the patterns skipped by acceleration methods like FABP (Focused Attention Back-Propagation) (if present);
- sieve the relative frequencies under a floor value to avoid problems with void classes;
- implement the division by priors as a change of the biases of the output units of the trained network:

  $$B_k = b_k - \log(p(q_k))$$

  where b_k are the original biases and B_k are the new biases.

3.3 Priors and MLP Output Biases

Another way to normalize the prior probabilities is to relate them to the biases of the trained NN. Given the activation function of the output units (we consider in the following the softmax activation function as it is the one we actually use in the MLP involved in the hybrid HMM-NN model), we have:

$$p(q_k \mid x) = o_k = \frac{e^{x_k + b_k}}{\sum_j e^{x_j + b_j}} \propto \exp(x_k + b_k)$$

where $x_k = \sum_i w_{ki} hid_i$ is the weighted sum of the outputs of the hidden units incoming to unit k, and b_k is the bias for unit k. Then we have:

$$p(q_k \mid x) \propto \exp[\log(p(x \mid q_k) + \log(p(q_k)]$$

As suggested in [4] it is tempting to identify the weighted sum of the hidden unit outputs x_k as the data part (and so, the log likelihood $\log(p(x|q_k))$ and the bias as the prior part (the log prior, $\log(p(q_k))$. This is supported by empirical observations on the output biases of the trained networks that are correlated to the log priors of the training classes.

However this relationship is too facile, as shown by [4] and the biases are influenced by the class mean and covariance as well as the log prior term. We may expect the output biases of a trained MLP to be influenced by the acoustic data, as well as prior information.

Nevertheless, from an experimental point of view the equation between biases of output units and log priors can be considered as it were true.

In such a case:

- the complete suppression of priors is implemented by setting to zero all the output biases;
- the decrease of a single prior for class q_k (by multiplying it by a factor $0<a<1$) is implemented by decreasing the value of b_k according to the equation:

  $$b'_k = b_k + \log(\alpha)$$

4. HYBRID HMM-NN ARCHITECTURE

The recognition model we currently use in experiments and applications is a hybrid HMM-NN architecture devoted to recognize sequential patterns, named *Neural Network Automata* (NNA). A hybrid HMM-NN typical architecture is depicted in Figure 1.

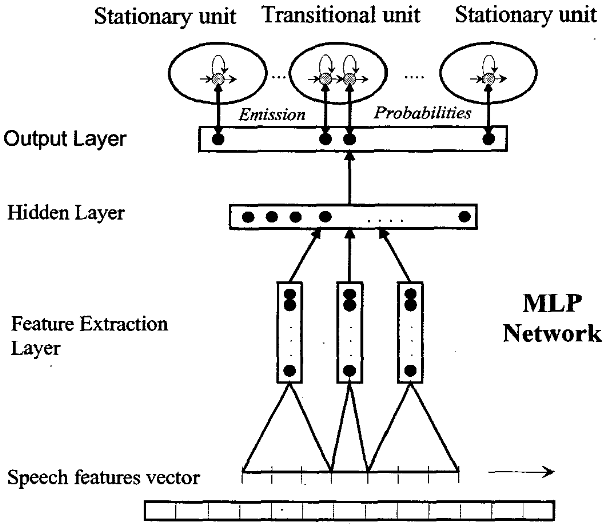

Fig. 1. Architecture of a NNA for speech recognition

Each sound class is described in terms of a left-to-right automaton (with self loops) as in the HMMs. The emission probabilities of the automata states are estimated by a Multi-Layer Perceptron (MLP) neural network, instead than by mixtures of gaussians, while the transition probabilities are not considered.

A NNA has an input window that comprises some contiguous frames of the sequence, one or more hidden layers and an output layer where the activation of each unit estimates the probability P(Q|X) of the corresponding automaton state Q given the input window X.

A NNA has many degrees of freedom: the architecture of the MLP, the input window width, the number of automaton states for the different context independent phonemes or for Stationary Transition Units (STU) [1] employed.

The MLP basic learning algorithm is the back-propagation. The error function is Cross-entropy and the output unit activation function is Softmax. Bootstrap segmentation of the spoken utterances in context-independent phonemes or in STUs is generally obtained by a forced segmentation via HMMs.

5. EXPERIMENTAL SETUP AND RESULTS

The experiments described in this paper have been carried out with corpora made of telephonic speech, collected in different languages (see Table 1, column *IDIOM* for a list of the languages involved in the experimentation).

IDIOM	*DICTIONARY CARDINALITY*	*MODE*	*ERR RED*
			(%)
Italian	475 towns	iso	12
French	78 applicative words	iso	69
French	8319 applicative words	iso	13
American	685 applicative words	iso	53
American	Spelling	cont	13
German	46 applicative words	iso	47
Brazilian	2167 applicative words	iso	10
Greek	43 Applicative words	iso	23

Table 1. Results of Class Prior Probability normalization

The signal bandwidth is 300-3400 Hz and the sampling frequency is 8 kHz. One HMM-NN model per language has separately been trained and tested. The features used as input of all the HMM-NN models are standard MFCCs with first and second order derivatives. For each language, training and test corpora are composed of a few thousands of utterances with no overlap in the speakers of the training and the test corpora. For example, for the Italian language, speakers were evenly distributed among males and females coming from many Italian regions and with different accents. Training was performed on 1136 speakers uttering a total of 4875 phonetically balanced sentences with a vocabulary of 3653 words. The test corpus consists of 14473 isolated words from 1050 speakers pertaining the 475 most common Italian city names.

As experimentations based on the method described in 3.2 have given better results than method 3.3, for the sake of brevity only the results obtained with method 3.2 has been reported and collected into Table 1. In this table, column *IDIOM* indicates the language target of the experimentation, *DICTIONARY CARDINALITY* reports cardinality and contents of the test data-base, *MODE* tells the recognition modality (isolated or continuous) and, finally, *ERR RED* indicates the error reduction percentages. Word Accuracies vary, across languages, as reference values, between 78.8% and 98.5%, and between 85.2% and 99.2% after applying the proposed normalization.

The tests show that:

- the Class Prior Probability normalization seems always useful to improve recognition performances;
- for some languages, in particular American and French, the error reduction is often very relevant (more than 50%);
- the Class Prior Probability normalization seems more effective for Anglo-Saxon languages (American, German) than for Latin languages (Italian, Brazilian);
- the Class Prior Probability normalization seems more effective for small test sets (with priors quite different from the training set) and less effective for large, balanced test sets.

6. CONCLUSIONS

Triggered by the desire to improve the performances obtained by the hybrid HMM-NN models, in particular in correspondence to training data-bases containing applicative components (hence not necessarily phonetically balanced) the topic of Prior Class Probabilities in Neural Networks for speech recognition has been analyzed theoretically and experimentally on several languages. Both the theoretical analysis and the experimentations indicate that a post-training normalization of Prior Class Probabilities can be useful to improve recognition performances.

A method of Prior Probability normalization has been described, with the important characteristic of being implementable through a modification of the biases of the trained net, without the need of software modifications in the recognition engine or, even worse, to retrain the neural network. Experimentation on several languages is reported, showing significant error reductions, especially on small test-sets with priors quite different from those of the training set.

7. REFERENCES

[1] R. Gemello, D. Albesano, F. Mana "Continuous Speech Recognition with Neural Networks and Stationary-Transitional Acoustic Units", *Proc. of IEEE International Conference on Neural Networks (ICNN-97)*, Houston, USA, 1997.

[2] M.A. Franzini, K.F. Lee and A. Waibel, "Connectionist viterbi training: a new hybrid method for continuous speech recognition", *Proc. ICASSP 90*, Albuquerque, NM, April 1990, pp.425-428.

[3] P. Haffner, M. Franzini and A. Waibel, "Integrating time alignment and neural networks for high performance continuous speech recognition", *Proc. ICASSP 91*, pp. 105-108.

[4] H. Bourlard and N. Morgan, "Connectionist speech recognition: a hybrid approach", Kluwer Academic Publishers, 1993.

[5] Anthony J. Robinson, "An application of recurrent nets to phone probability estimation", *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, March 1994, pp. 298-305.

[6] M. M. Hochberg, S. J. Renals, A. J. Robinson and G. D. Cook, "Recent improvements to the abbot large vocabulary CSR system", *Proc. ICASSP 95*, Detroit, USA, pp. 69-72.

[7] R. Gemello, D. Albesano, F. Mana, "CSELT Hybrid HMM/Neural Networks Technology for Continuos Speech Recognition", in *Proc. IJCNN '00,* Como, Italy 2000.

[8] H.A. Bourlard, N. Morgan. "Links between Morkov models and multilayer perceptrons". In *Advances in Neural Information Processing Systems 1*, volume 1, pages 502-510. Morgan Kaufmann, 1989.

[9] R.P. Lippmann, M.D. Richard. "Neural Network Classifiers estimate Bayesian a posteriori probabilities". *Neural Computation*, 3 (4):461-483, 1991