Joint Estimation of Parameters in Hidden Neural Networks

Søren Kamaric Riis[1] and Anders Krogh[2]

[1]Department of Mathematical Modelling
Technical University of Denmark, B305
DK-2800 Lyngby, Denmark
Email: riis@ei.dtu.dk

[2]The Sanger Centre
Hinxton Hall, Hinxton
Cambs CB10 1RQ, UK.
Email: krogh@sanger.ac.uk

ABSTRACT

It has been proven by several authors that hybrids of Hidden Markov Models (HMM) and Neural Networks (NN) yield good performance in speech recognition. However, in many of the current hybrids the HMM and neural networks are trained separately and only combined during decoding. In this paper we propose a new hybrid called Hidden Neural Networks (HNN) where all parameters are trained discriminatively at the same time by maximizing the probability of correct classification. The probability parameters in the HMM are replaced by neural network outputs, and instead of the local normalization of parameters used in standard HMMs the HNN is normalized globally. On the task of classifying TIMIT phonemes into five broad classes the new hybrid obtains a recognition accuracy of 83.7%, whereas a standard HMM obtains 76.1%.

1. INTRODUCTION

It is well known that standard HMMs are based on a number of assumptions which limit their static classification abilities. First of all, it is usually assumed that the Markov process in HMMs is first order and that the observations are independent in the sense that emission probabilities only depend on the current state. Furthermore, the transition probabilities in standard HMMs are time independent and contextual dependencies between observations can only be handled by explicit construction of context dependent models. Some of these assumptions can be relaxed by introducing neural networks to estimate the probability parameters in the HMM. Recently several approaches for combining HMMs and neural networks have been proposed, see *e.g.* [2, 3, 4, 6, 11, 12, 13]. Here we present a new hybrid called Hidden Neural Networks where the usual HMM probabilities are estimated by small neural networks. In the HNN it is possible to assign up to two networks to each state: 1) a *match network* estimating the probability that the current observation matches a given state and 2) a *transition network* that estimates transition probabilities conditioned on observations. One of the two types of networks can be omitted and replaced by standard HMM parameters. In fact all sorts of combinations with standard HMM states are possible.

One of the main ideas in this work is to train the whole HNN supervised, by a joint optimization of parameters (see *e.g.* [3, 4, 5] for similar joint methods). Another important idea is a new way of normalizing the model. Instead of normalizing the model locally, *e.g.*, by using softmax on neural network outputs or by normalizing with class priors, the HNN is normalized globally.

2. THE MODEL

The basic idea of the HNN is to replace the probability parameters of the HMM by neural network outputs that can depend on the context of the observation vector x_l at time l. The emission probability $\phi_i(x_l)$ of observation vector x_l in state i is replaced by a match network $\phi_i(s_l; w^i)$, which is a feed-forward neural network parameterized by weights w^i with input s_l and only one output. The network input s_l corresponding to x_l will usually be a window of context around x_l, *e.g.*, a symmetrical context window of $2K+1$ observation vectors, $x_{l-K}, x_{l-K+1}, \ldots, x_{l+K}$. It can however be any other sort of information related to x_l or the observation sequence in general. Similarly, the probability θ_{ij} of a transition from state i to j is replaced by the output of a transition network $\theta_{ij}(s_l; u^i)$, which is parameterized by weights u^i. The transition network assigned to state i has $\mathcal{J}_i$ outputs, where $\mathcal{J}_i$ is the number of (non-zero) transitions *from* state i.

In complete analogy with the likelihood $p(x|\mathcal{M})$ of a HMM for observation sequence $x = x_1, \ldots, x_L$, we define the quantity

$$q(x|\mathcal{M}) = \sum_{\pi} q(x, \pi|\mathcal{M}) \tag{1}$$

with

$$q(x, \pi|\mathcal{M}) = \prod_{l=1}^{L} \theta_{\pi_{l-1}\pi_l}(s_{l-1}; u^{\pi_{l-1}})\phi_{\pi_l}(s_l; w^{\pi_l}) \tag{2}$$

where $\mathcal{M}$ denotes the whole model, *i.e.*, all the network parameters, and the state sequence $\pi = \pi_1, \ldots, \pi_L$ is a particular path through the model. We define $\pi_0 = 0$ and $\theta_{0i}(s_0; u^0)$ is the probability of initiating a path in state i. s_0 is the context we choose to associate with the beginning of the sequence. The global normalization is insured by explicit normalization of q,

$$p(x|\mathcal{M}) = \frac{q(x|\mathcal{M})}{\int_{x' \in \mathcal{X}} q(x'|\mathcal{M})dx'} \tag{3}$$

The integration in the denominator is taken over the space of observation vector sequences $\mathcal{X}$, but as we shall see below, we will never need to calculate this normalization. Apart from making this model much more elegant from a mathematical and computational point of view, we also believe that it may be beneficial to give up local normalization of parameters even for standard HMMs. In fact, non-normalizing parameters are already used frequently in speech recognition by introducing so-called "transition biases" and "stream exponents", see *e.g.* [6, 8, 13]. These heuristic approaches are used in order to reduce the mismatch between transition and emission probabilities in standard HMMs. In [10, 13] these issues are discussed in greater detail.

2.1. Training and decoding

To train the model we assume that the *complete labeling* is available[1] , *i.e.*, that each observation x_l has an associated label y_l corresponding to the class to which it belongs. In order to maximize the prediction accuracy we choose parameters so as to maximize,

$$P(y|x,\mathcal{M}) = \frac{p(x,y|\mathcal{M})}{p(x|\mathcal{M})} \tag{4}$$

as we have previously proposed in [9] (where it was called CHMM for 'class HMM'). This has also been called Conditional Maximum Likelihood (CML) and is equivalent to Maximum Mutual Information estimation (MMI) [1, 7] if the language model is fixed. $p(x,y|\mathcal{M})$ is calculated as a sum over all paths consistent with the labeling, *i.e.*, if observation l is labeled by f only paths in which the l-th state has label f are allowed. If the set of these consistent paths is called $\mathcal{A}(y)$ we have,

$$p(x,y|\mathcal{M}) = \frac{q(x,y|\mathcal{M})}{\sum_{y'}\int_{x'\in\mathcal{X}} q(x',y'|\mathcal{M})dx'} \tag{5}$$

with,

$$q(x,y|\mathcal{M}) = \sum_{\pi\in\mathcal{A}(y)} q(x,\pi|\mathcal{M}). \tag{6}$$

Since,

$$\sum_{y'}\int_{x'\in\mathcal{X}} q(x',y'|\mathcal{M})dx' = \int_{x'\in\mathcal{X}} q(x'|\mathcal{M})dx' \tag{7}$$

equation (4) becomes

$$P(y|x,\mathcal{M}) = \frac{q(x,y|\mathcal{M})}{q(x|\mathcal{M})} \tag{8}$$

and the normalizing factor has conveniently disappeared. Both $q(x|\mathcal{M})$ and $q(x,y|\mathcal{M})$ can be calculated by a straight-forward extension of the forward algorithm, see [9, 10].

[1] In speech recognition this is not always the case. Often only the sequence of utterance symbols, *e.g.* the phoneme transcription, is known (*incomplete labeling*). However, the framework presented here can easily be adapted to the case of incomplete labeling, see [7, 10].

To optimize (8) we use gradient descent. Calculating the derivative of $\log P(y|x,\mathcal{M})$ w. r. t. a weight in the match or transition networks, yields back-propagation training of the neural networks based on an error signal calculated by the forward–backward algorithm, see [10] for details. This has also been observed by several other authors.

In agreement with [6] we have found that Viterbi decoding does not perform well for discriminatively trained models. The reason is that the model is optimized so as to maximize the probability of the labeling, which need not correspond to the most probable path found by Viterbi decoding. Instead one can select the most probable label y_l^* at time l by assuming that consecutive labels are independent. This is done by calculating the sums of state posterior probabilities $P(\pi_l|x,\mathcal{M})$ at time l for those states that carry the same label. The largest of these sums then identify the most probable label. Since the posteriors $P(\pi_l|x,\mathcal{M})$ are readily found using the forward-backward algorithm [7], we call it forward-backward decoding. This type of decoding, however, sometimes results in label sequences that do not correspond to possible paths through the model, and the assumption of independent labels is obviously not good. A better type of decoding which gives label sequences consistent with legal paths in the model is N-best decoding [6, 14]. Results will be reported using both forward-backward and 1-best decoding (in the latter case a maximum of 10 active hypotheses is allowed in each state during decoding).

3. EXPERIMENTS

We have selected the task of predicting five broad phoneme classes in the TIMIT database: Vowels (V), Consonants (C), Nasals (N), Liquids (L) and Silence (S). The five classes are highly confusable and covers all phonetic variations in American English.

We used one sentence from each of the 462 speakers in the TIMIT training set as training set, and the results are reported for the recommended TIMIT core test set. These are also the training and test sets used in [5, 6] for the same task. An independent crossvalidation set is used to monitor the performance of the model during training.

Our preprocessor outputs an observation vector every 10ms consisting of 26 features: 12 mel scaled cepstral coefficients, 1 log energy coefficient and the corresponding derivatives. The observation vectors are normalized to zero mean and unit variance in order to speed up training of the HNN. For the discrete HMMs we used a codebook of 256 prototype feature vectors.

The models used are simple three state left-to-right models for each of the five classes, and only the middle state has a selfloop. In the HNN we only use match networks and let the transitions be the

Table 1*: Test set recognition accuracies for the five broad classes obtained by aligning the predicted and observed phoneme transcriptions. The accuracy is given by:* $\%Acc = 100\% - \%Ins - \%Del - \%Sub$*, where* $\%Ins$*,* $\%Del$ *and* $\%Sub$ *are the number of insertions, deletions and substitutions respectively. Results for both forward-backward (F-B) and 1-Best decoding are shown.*

Model	#Par.	F-B	1-Best
HMM (ML)	3856	75.5	76.1
HMM (CML)	3856	78.6	79.0
No hidden units			
HNN, $K = 0$	436	77.3	77.3
HNN, $K = 1$	1216	78.4	78.9
HNN, $K = 2$	1996	78.8	79.3
10 hidden units			
HNN, $K = 0$	4246	81.0	83.0
HNN, $K = 1$	12046	82.8	83.7
HNN, $K = 2$	19846	82.1	83.0

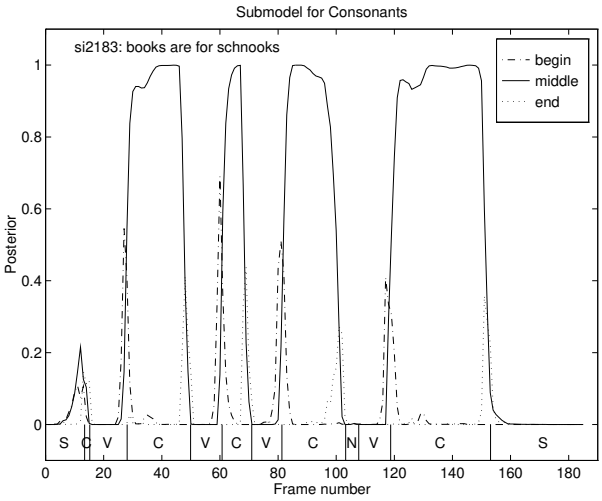

Fig. 1*: State posterior probabilities* $P(\pi_l|x, \mathcal{M})$ *for the three states in the consonant submodel for test sentence "si2183".*

usual time independent transitions. Note however, that the transitions are also trained discriminatively in the HNN as opposed to MMI trained HMMs. The match networks have a symmetric input window of $2K + 1$ frames and a sigmoid output function. With this model setup we found that transition networks did not increase performance. The match networks are initially trained by standard backpropagation to classify the observations into the five broad classes. This speeds up training of the HNN considerably and the models are less prone to getting stuck in local minima. Thus, the performance on the crossvalidation set reaches a maximum within less than 30 epochs (full sweeps through the training set) for all tested models and training times are therefore less than two days on a HP735 workstation. All models except the ML estimated HMM are trained discriminatively by minimizing the negative logarithm of (8) with standard gradient descent.

From table 1 it is observed that 1-best decoding gives slightly higher recognition accuracies compared to forward-backward decoding. This is to be expected since the 1-best decoder selects the transcription that maximizes the full model likelihood (at least approximatively), whereas the forward-backward decoder is based only on "local" label posterior probabilities for each frame.

Compared to the ML trained discrete HMM a gain of about 3% in accuracy is obtained by using discriminative training, see table 1. In [6] an accuracy of 69.3% is reported for a ML trained continuous density HMM with a mixture of six diagonal covariance gaussians per state. Thus, for approximately the same number of parameters the ML trained discrete HMM outperforms the continuous density HMM by more than 6%. For a MMI trained model with a single diagonal covariance gaussian per state they report an accuracy of 72.4%, compared to our 79.0% for the CML trained discrete HMM.

Eventhough the HNN with zero hidden units and no context ($K = 0$) contains almost ten times less parameters than the ML trained HMM it performs considerably better. By increasing the input window width, the HNN performance improves above that of the CML trained HMM, see table 1. No further improvement was observed for contexts larger than $K = 2$. It is interesting to note that the match networks in the HNN without hidden units actually just implements linear weighted sums of input features. Adding hidden units to the HNN drastically increases performance even for small input windows, see table 1. Thus, for approximately the same number of parameters, the HNN with no context ($K = 0$) outperforms the CML trained HMM by 4% in accuracy. The best HNN with a context of one frame ($K = 1$) yields a recognition accuracy of 83.7% which is more than 4% better than the CML trained HMM, and almost 8% better than the ML trained HMM. For comparison, [6] uses a multi-layer perceptron to transform the feature vectors for the continuous density HMM with a single diagonal covariance gaussian per state. This hybrid is trained by MMI and 1-best decoding gives an accuracy of 78.5%. The results in [6] have later been improved to 81.3% by using a linear encoding instead of the multi-layer perceptron [5], and by using multiple CML training passes and non–normalizing transition probabilities.

For a chosen sentence the state posterior probabilities provided by the HNN with 10 hidden units and $K = 1$ are plotted for the the consonant submodel states in fig. 1. It is observed, that the posteriors for the "begin" and "end" states are only large at the class boundaries, whereas the posterior for the "middle" state is large only between these boundaries. Hence, the model is very good at discriminating between different classes. This is also verified in

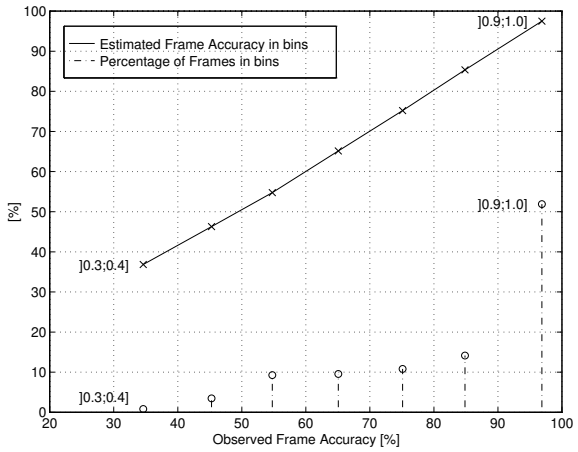

Fig. 2*: The solid line shows the average label posterior probability of the winning class* $\bar{P}(y_l^*|x,\mathcal{M})$ *ranked into 7 equally spaced bins and plotted as a function of the observed frame accuracy in these bins. The dashed lines show the percentage of frames falling into these bins, and are located at the corresponding observed accuracies.*

fig. 2 where it is observed that more than 50% of the frames have a winning-label posterior probability $P(y_l^*|x,\mathcal{M})$ larger than or equal to 0.9. This corresponds to an observed frame recognition rate of more than 95%. Furthermore, there is an almost perfect correlation between the observed frame accuracy and the average label posterior of the winning class. Thus, the larger $P(y_l^*|x,\mathcal{M})$ the more confident is the prediction.

4. CONCLUSION

It has been shown that the HNN combination of neural networks and a hidden Markov model yields a better recognition ability than a standard HMM on the task of classifying TIMIT phonemes into five broad classes. In addition it was illustrated that the HNN provides very accurate estimates of label posterior probabilities. Hence, regions of speech that are predicted with high confidence can easily be identified. The particular architecture introduced here is characterized by: 1) All parameters are trained discriminatively at the same time 2) Proper probability distributions are obtained by global normalization as opposed to local normalization, and 3) Large flexibility in that not all of the HMM parameters need to be replaced by neural networks. In the future we will test the method on larger speech recognition tasks and on problems in molecular biology.

ACKNOWLEDGMENTS

The authors would like to thank Steve Renals for valuable comments and suggestions to this work. The Sanger Centre is supported by the Wellcome Trust.

REFERENCES

[1] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proceedings of ICASSP'86*, pp. 49–52, 1986.

[2] P. Baldi and Y. Chauvin, "Protein modeling with hybrid hidden Markov model/neural network architectures," in *Proceedings of the 3rd ISMB'95*, pp. 39–47, 1995.

[3] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden Markov model hybrid," *IEEE Trans. on Neural Networks*, vol. 3, no. 2, pp. 252–9, 1992.

[4] P. Frasconi and Y. Bengio, "An EM approach to grammatical inference: input/output HMMs," in *Proceedings of the 12th ICPR'94*, pp. 289–94, 1994.

[5] F. Johansen, "Global optimisation of HMM input transformations," in *Proceedings of ICSLP'94*, pp. 239–42, 1994.

[6] F. Johansen and M. Johnsen, "Non-linear input transformations for discriminative HMMs," in *Proceedings of ICASSP'94*, pp. 225–28, 1994.

[7] B. Juang and L. Rabiner, "Hidden Markov models for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–72, 1991.

[8] S. Kapadia, V. Valtchev, and S. Young, "Mmi training for continuous phoneme recognition on the timit database," in *Proceedings of ICASSP'93*, pp. 491–4, 1993.

[9] A. Krogh, "Hidden Markov models for labeled sequences," in *Proceedings of the 12th ICPR'94*, pp. 140–4, 1994.

[10] A. Krogh and S. Riis, "Hidden neural networks,". In preparation.

[11] E. Levin, "Word recognition using hidden control neural architecture," in *Proceedings of ICASSP'90*, pp. 433–6, 1990.

[12] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in hmm speech recognition," *IEEE Trans. on Speech and Audio Proc.*, vol. 2, no. 1, pp. 161–74, 1994.

[13] T. Robinson, M. Hochberg, and S. Renals, *Automatic speech and speaker recognition – Advanced topics*. Dordrecht, Netherlands: Kluwer, 1995.

[14] R. Schwarz and Y.-L. Chow, "The n-best algorithm: An efficient and exact procedure for finding the n most likely hypotheses," in *Proceedings of ICASSP'90*, pp. 81–84, 1990.