HIDDEN NEURAL NETWORKS: APPLICATION TO SPEECH RECOGNITION

Søren Kamaric Riis

Department of Mathematical Modelling
Section for Digital Signal Processing
Technical University of Denmark
2800 Lyngby, Denmark
Email: sr@imm.dtu.dk

ABSTRACT

In this paper we evaluate the Hidden Neural Network HMM/NN hybrid presented at last years ICASSP on two speech recognition benchmark tasks; 1) task independent isolated word recognition on the PHONEBOOK database, and 2) recognition of broad phoneme classes in continuous speech from the TIMIT database. It is shown how Hidden Neural Networks (HNNs) with much fewer parameters than conventional HMMs and other hybrids can obtain comparable performance, and for the broad class task it is illustrated how the HNN can be applied as a purely transition based system, where acoustic context dependent transition probabilities are estimated by neural networks.

1. INTRODUCTION

Although the HMM is good at capturing the temporal nature of processes such as speech it has a very limited capacity for recognizing complex patterns involving more than first order dependencies in the observed data. This is primarily due to the first order state process and the assumption of state conditional observation independence. Neural networks and in particular multi-layer perceptrons (MLP) are almost the opposite: they cannot model temporal phenomena very well, but are good at recognizing complex patterns in a very parameter efficient way.

The Hidden Neural Network hybrid introduced in [16] is a very flexible architecture, where the probability parameters of an HMM are replaced by the outputs of small state specific neural networks. The model is estimated by the discriminative Conditional Maximum Likelihood (CML) criterion and is normalized globally. The global normalization works at the sequence level and ensures a valid probabilistic interpretation as opposed to the often approximate local normalization enforced in many other hybrids. Furthermore, instead of training the HMM and NNs separately, all parameters in the HNN are estimated simultaneously.

2. THE HNN

The HNN is a very natural extension of the so-called class HMM (CHMM) introduced in [10], which is basically a standard HMM where each state, in addition to the emission or *match* distribution, also has assigned a distribution over labels (classes). The basic idea is to *replace* the probability parameters of the CHMM by the outputs of state specific neural networks. Thus, it is possible to assign up to three networks to each state: 1) a *match network* $\phi_i(s_l; w^i)$ estimating the "probability" that the current observation matches a given state, 2) a *transition network* $\theta_{ij}(s_l; u^i)$ that estimates transition "probabilities" conditioned on observations, and finally 3) a *label network* $\psi_{ik}(s_l; v^i)$ estimating the probability of label $y_l = k$ in state i at time l. We have put probabilities in quotes because we do not require that the network outputs normalize locally in the HNN, *e.g.* the outputs of the transition network assigned to a state need not sum to one. The match network is parameterized by weights w^i and has only one output, which replaces the usual emission probability. Similarly the transition network is parametrized by weights u^i, and has the same number of outputs as there are non-zero transitions from state i to state j. The label network is parameterized by weights v^i and has one output for each of the possible labels in this state. Note that a given state can be restricted to model only a subset of the possible labels, *i.e.*, some labels have probability zero. The input s_l to the networks will usually be a window of context around the current observation x_l, *e.g.*, a symmetrical context window of $2K+1$ observation vectors, $s_l = x_{l-K}, x_{l-K+1}, \ldots, x_{l+K}$. It can, however, be any sort of information related to x_l or even the observation sequence in general. We will call s_l the context. Each of the three types of networks in each HNN state can be omitted and replaced by standard CHMM parameters. In fact all sorts of combinations with standard CHMM states are possible. In this work we assume, that the label networks are just delta-functions, *i.e.*, each state can only model one particular label and $\psi_{ik}(s_l; v^i) = \delta_{k,c_i}$, where c_i is the label of state i. Similarly we here restrict ourselves to use MLP match and transition networks, although they can in principle be any kind of mapping defined on the space of observations.

It is well known that Maximum Likelihood (ML) estimation is not optimal when the models are used for recognition especially when the training data is limited. We therefore choose parameters so as to maximize the probability of the correct labeling y associated with observation sequence x,

$$P(y|x, \mathcal{M}) = \frac{P(x, y|\mathcal{M})}{P(x|\mathcal{M})} \tag{1}$$

as we have previously proposed in [10]. Maximizing (1) is known as Conditional Maximum Likelihood estimation (CML) and is equivalent to Maximum Mutual Information estimation (MMI) [1, 8] if the language model is fixed during training. For an observation sequence of length L, the labeling y can be either *complete*, *i.e.*, there is one label for each observation ($y = y_1, \ldots, y_L$), or *incomplete*, *i.e.*, the label sequence $y = y_1, \ldots, y_S$ is shorter than the observation sequence ($S < L$). The latter case is more common in speech recognition since we usually only know the spoken words in the training set (and thereby the phonetic transcription), whereas the former is more common in *e.g.* biological sequence analysis. From (1) we see, that in order to compute the probability of the labeling we need to do two forward passes; once in the *free-running* or *recognition* time phase to compute $P(x|\mathcal{M})$ and once in the *clamped phase* to compute $P(y, x|\mathcal{M})$.

Similar to the likelihood for a standard HMM, we define for the HNN

$$\begin{aligned} R(x|\mathcal{M}) &= \sum_{\pi} R(x, \pi|\mathcal{M}) \\ &= \sum_{\pi} \prod_{l=1}^{L} \theta_{\pi_{l-1}\pi_l}(s_{l-1}; u^{\pi_{l-1}}) \phi_{\pi_l}(s_l; w^{\pi_l}) \end{aligned} \quad (2)$$

where $\pi = \pi_1, \ldots, \pi_L$ is a path through the model corresponding to the observation sequence $x = x_1, \ldots, x_L$. Since we do *not* require, that the neural network outputs normalize locally, $R(x|\mathcal{M})$ *will generally not be a probability*. However, if we only allow one label in each state we can define $R(x, y|\mathcal{M})$ in a way similar to equation (2) by only extending the sum over paths that are *consistent* with the observed complete or incomplete labeling. Then it is straightforward to show that,[1]

$$P(y|x, \mathcal{M}) = \frac{R(x, y|\mathcal{M})}{R(x|\mathcal{M})} \quad (3)$$

and the model is normalized at a global level, see [11, 15] for further details. Both $R(x|\mathcal{M})$ and $R(x, y|\mathcal{M})$ can be calculated by a straight-forward extension of the forward algorithm for complete as well as incomplete labeling, see *e.g.* [10, 11]

To maximize (3) we use stochastic online gradient ascent augmented by a momentum term, where the parameter update is performed after each observation sequence. As shown in [11] it turns out that the networks in the HNN are trained by standard backpropagation where the error to backpropagate is computed by running two forward-backward passes on the model; once for the free-running phase, and once for the clamped phase.

In agreement with results reported in [7], we have observed an increased performance for CML estimated models when using full-likelihood based decoders instead of a Viterbi best-path decoder. The reason for this is primarily that several paths have been observed to contribute significantly to the optimal labeling in the CML estimated models, see [11]. In the case of small vocabulary isolated word recognition the likelihood of the model for each word can be computed using the forward algorithm, whereas in the case of continuous speech recognition or large vocabulary isolated word recognition one can use stack decoding [13] or approximative algorithms like the N-best decoder [17]. In this work we use standard full-forward decoding for the PHONEBOOK experiments and N-best decoding for the broad class experiments. The N-best decoder allows 10 active hypothesis during decoding, and only the top-scoring hypothesis is used for recognition at the end of decoding.

3. COMPARISON TO OTHER HYBRIDS

Instead of training the HMM and the NN separately as in the work by *e.g.* Renals *et al.* [14] several authors have recently proposed architectures where all parameters are estimated simultaneously as in the HNN, see *e.g.* [2, 3, 5, 7, 9]. An example of such an approach is to use the neural network as an adaptive input transformation, where the network outputs are used as new observation vectors in a continuous HMM [7]. Our approach is somewhat similar to the idea of adaptive input transformations, but instead of retaining the computationally expensive mixture densities we *replace* these by match networks. This is also done in [3], where a large network with the same number of outputs as there are states in the HMM is optimized using the CML criterion by backpropagating errors calculated by the HMM. Instead of backpropagating errors from the HMM into the neural network some researchers [5] have proposed to let the HMM iteratively reestimate new "soft" targets for the network and then train the network to learn these targets. This method extends the approach by Renals *et al.* [14] to use global estimation where training can be done by a *Generalized* EM (GEM) algorithm.

The HNN is very closely related to the IOHMM [2]. In fact the IOHMM can be considered a special case of the HNN where each state has assigned a label network and a transition network, but no match network. If the transition and label networks all have a softmax output function then, in the free-running phase, $R(x|\mathcal{M}) = \sum_{\pi} R(x, \pi|\mathcal{M}) = 1$ independent of the observations, and thus $P(y|x, \mathcal{M}) = R(x, y|\mathcal{M})$. For this model only one forward pass is needed to compute the probability of the labeling and similarly only one forward-backward pass is needed to find the gradient. Furthermore, this model can be trained by a GEM algorithm. An additional assumption of only one label in each state renders the HNN similar to the purely transition based discriminant HMM/NN hybrid discussed in [9].

4. PHONEBOOK EXPERIMENTS

Often speech recognizers are trained using a fixed vocabulary, *i.e.*, the models are designed only to recognize words also used for estimation. For utterances containing out-of-vocabulary (OOV) words a very poor performance can be expected from such models unless the model incorporates some sort of OOV word detection. Therefore it would be desirable to train the models for task independent recognition, where the vocabulary used for training can be entirely different from that used during recognition. In this section we evaluate the HNN on task independent recognition of isolated words, where there are no identical words in the training and test set. The words are taken from the PHONEBOOK database [12], which is a phonetically-rich isolated word database. The words in PHONEBOOK are uttered by 1300 native American English speakers over a public American telephone line. In PHONEBOOK each of almost 8000 different words are uttered by an average of more than 11 speakers yielding a total of about 92,000 utterances.

We use a training set of 9,000 words randomly selected from the 21 PHONEBOOK wordlists ([a-d][a,h,m,q,t]+ea), see *e.g.* [4, 12], and a crossvalidation set of 1,893 utterances (wordlists ao and ay). The test set is composed of 8 wordlists ([a,b,c,d][d,r]) and results are reported as an unweighted average over the 8 lists. This is identical to the test sets used in [4, 5]. The results are reported for a dictionary size of either about 75 words (one dictionary for each of the 8 wordlists) or a larger dictionary of about 600 words (all 8 wordlists). The 110,000-word CMU 0.4 dictionary was used for phonetically transcribing the words.

For speech corrupted by linear additive channel noise RASTA-PLP cepstral features have been shown to be very robust [6]. We therefore use a RASTA-PLP cepstral preprocessor yielding a feature vector each 10ms based on a 30ms window. The 26 dimensional feature vector is composed of 12 RASTA-PLP cepstral features, the corresponding Δ-features and the Δ- and $\Delta\Delta$-energy.

For each of the 46 phonemes occurring in the transcriptions we use a phoneme submodel with a number of states equal to half the average duration of the phonemes as obtained from an initial forced Viterbi alignment of the training set. No skips are allowed, and the transition probabilities between states in a phoneme submodel are fixed to 1/2. We use a zero-gram grammar between phoneme models to avoid unintended introduction of priors for

[1] If more than one label have non-zero probability in each state equation (3) is still valid provided that the outputs of the label networks normalize locally, $\sum_k \psi_{ik}(s_l; v^i) = 1$ for $\forall i, l$, see [15].

Table 1: *HNN error rates on PHONEBOOK. K is the context-size,* i.e., $s_l = x_{l-K}, \ldots, x_l, \ldots, x_{l+K}$.

75 word dictionary	#Parms	Vit	Forw
Context $K = 0$, 0 hidden	1,242	20.8	15.1
Context $K = 1$, 0 hidden	3,634	16.6	13.3
Context $K = 1$, 10 hidden	36,846	7.3	4.8
600 word dictionary	**#Parms**	**Vit**	**Forw**
Context $K = 1$, 10 hidden	36,846	18.4	14.2

Table 2: *HNN accuracies on TIMIT broad class experiments.*

Model	#Parms	Accuracy
1-state, Match	4,030	80.0
1-state, Transition (sigmoid)	4,225	81.6
1-state, Transition (softmax)	4,225	81.6
3-state, Match	12,055	83.8
3-state, Transition (softmax)	12,290	82.0
3-state, Transition (sigmoid)	12,290	83.7
3-state, Mixed (sigmoid)	12,200	84.4

words in the training vocabulary due to a limited size training set. All states in a phoneme submodel share *one* match network, which replaces the usual emission distribution. All 46 match networks are fully connected MLPs, have the same number of hidden units, share the same input s_l and use sigmoid output functions. The match networks are initially trained to classify the observation vectors into each of the 46 phonemes by a few iterations of standard backpropagation. This speeds up training of the HNN and the model is less prone to getting stuck in local minima. Furthermore the HNN is bootstrapped by a few iterations (usually less than five) of complete label training.

In Table 1 the results obtained by the HNN is shown. First of all it is observed that full-forward decoding gives considerably better results than Viterbi decoding. This can be attributed to two facts: 1) in the CML estimated models several paths contribute to the optimal labeling as discussed above, and 2) the architecture of the phoneme submodels implies a Poisson-like duration distribution when using full-forward decoding, whereas a much weaker exponential duration distribution is implied when using Viterbi decoding. From the table it is also observed that contextual input increases performance considerably. Thus, for a model using a context of one left and right frame and no hidden units an error rate of 13.3% is obtained for the 75 word dictionary. With only 3634 parameters this model is very small and it is interesting that the match networks in this model actually just implement linear weighted sums of the input features (passed through a sigmoid output). No futher improvements in performance was observed for lager contexts. The best model with a context of one frame and 10 hidden units obtains an error rate of 4.8% for the 75 word dictionary. In [4] a continuous density HMM with more than four times as many parameters (162k) is reported to have an error rate of 5% when using a training set of 19,000 words. Similarly, for a Viterbi trained HMM/NN hybrid with 166k parameters an error rate of 1.5% is reported. This is somewhat better than the HNN, but the larger training set contributes significantly to the lower error rate as discussed below.

The effect of using the larger 600 word dictionary is reflected in the higher error rates shown in Table 1. For a model with context $K = 1$ and 10 hidden units an error rate of 14.2% is achieved. For the same training set as used here, Hennebert *et al.* [5] reports an error rate of 13.7% for a Viterbi trained HMM/NN hybrid containing 166k parameters. By iteratively reestimating "soft"-targets for the neural network instead of Viterbi "hard"-target training the error rate drops to 12.2%. For a 19,000 word training set Dupont *et al.* [4] reports an error rate of only 5.3% for the Viterbi trained HMM/NN hybrid. Thus, the size of the training set is indeed very important for the performance of the models.

5. BROAD PHONEME CLASS EXPERIMENTS

In this section we discuss some improvements on results reported at last years ICASSP [16] on the recognition of five broad phoneme classes in continuous speech from the TIMIT database. The broad classes are Vowels (V), Consonants (C), Nasals (N), Liquids (L) and Silence (S) and cover all phonetic variations in American English. We use one sentence from each of the 462 speakers in the TIMIT training set for training, and the results are reported for the recommended TIMIT core test set. The preprocessor is a standard mel cepstral preprocessor, which gives a 26 dimensional feature vector each 10ms (12 mel cepstral coefficients+1 log energy coefficient and the corresponding Δ-coefficients).

In the experiments reported in [16] we used a simple left-to-right three state model for each of the five classes, where the the match distributions were replaced by match networks. However, one major advantage of the HNN is that it allows for using transition probabilities that can depend on the observation context s_l. This can be important in tasks like speech recognition, where the acoustics of one phoneme is indeed highly influenced by the phonetic context in which it is uttered. Here we report results for two different kinds of HNNs with acoustic context dependent transitions: 1) a purely transition based model and 2) a mixed model. In the transition based HNN the transition probabilities in each state are modeled by a transition network and the match distributions are replaced by a constant. The mixed model is based on the 3-state left-to-right submodels also used in [16]. The first two states use standard transitions and match networks and in the last state transitions are modeled by a transition network. The transition based HNN is similar to the discriminant HMM/NN hybrid proposed in [9], except that we do not have to enforce locally normalizing transitions.

For the broad class experiments all match and transition networks have 10 hidden units and use a context of one left and right frame as input ($s_l = x_{l-1}, x_l, x_{l+1}$). The match networks use sigmoid output functions and the transition networks use either a locally normalizing softmax output function or sigmoid outputs. Initialization of the match networks is done in the same way as for the PHONEBOOK experiments, whereas the transition networks are initialized by duplicating the hidden to output weights of a pretrained match network as many times as there are non-zero transitions. For a state with only a transition network assigned, this initially corresponds to a state with a match network and uniform standard HMM transition probabilities.

In table 2 the results for the TIMIT broad class experiments are shown. For a very simple model with only one state per class, where the match distributions are replaced by match networks, an accuracy of 80.0% is obtained. This compares favorably with a result of 69.3% accuracy on the same test set reported for a 3-state ML estimated HMM with six diagonal covariance Gaussians in each state (4,799 parameters) [7]. The purely transition based HNN with one state per class and non-normalizing transitions obtains an accuracy of 81.6%. The same result is obtained if locally normalizing transitions are enforced by softmax output functions. These results indicate that the purely transition based 1-state model is much better capable of modelling the five broad classes than the model with standard transition probabilities and match networks. It is interesting that this very simple model outperforms an HMM/adaptive linear input transformation hybrid with three states per class, which was reported to have an accuracy of 81.3% in [7] for the same training and test set. If we use 3-state submodels with match networks and standard transition probabilities in all states the accuracy increases to 83.8%. Similar to the transition based model above, we tried a 3-state model where the match distributions and standard transition probabilities are replaced by transition networks. If we enforce local normalization of the transitions by using a softmax output function the accuracy drops significantly compared to the 3-state model with match networks and standard transitions, see table 2. This can be explained as follows: assume that in a particular path we have just entered the consonant submodel. Then the next label in this path will also be a consonant with probability one because of the softmax function. That is, after entering a submodel, the path through this submodel will never be terminated due to a very low probability, since no matter what state we make a transition to this will always be with a fairly high probability. This is a fundamental problem, which makes minimum duration modeling very difficult in transition based models. However, the problem can be eliminated in practice by using non-normalizing transition "probabilities", whereby a path can be terminated if all outgoing transition "probabilities" from a state are close to zero. Such a model obtains a performance of 83.7%, which is practically identical to the 3-state model with standard transitions and match networks, see table 2. By replacing the match network and standard transition probabilities in the last state of each 3-state submodel with a transition network, the transitions between submodels become dependent on the acoustic context around the boundaries between the broad phoneme classes. This increases the performance of the HNN to 84.4% accuracy.

These results clearly illustrate the advantage of using transition networks. However, an even larger gain is expected for tasks, where there is a more pronounced context dependency between the phoneme classes like, *e.g.*, the well known TIMIT 39 phoneme recognition task. We are currently investigating this issue.

6. CONCLUSION

The globally normalized HNN has been introduced as a very flexible HMM/NN hybrid that allows for acoustic context dependent transitions. Furthermore, a comparison to other hybrids was given, and the similarity of the HNN to the IOHMM [2] and the discriminant HMM/NN hybrid [9] was discussed. Through a series of experiments it was shown how the HNN in a very parameter efficient way can yield state-of-the-art performance on two different speech recognition benchmarks. The results obtained on PHONEBOOK are slightly inferior to results reported for the HMM/NN hybrid discussed in [4, 5], where a much larger training set is used. On the TIMIT broad class experiments the HNN has been evaluated as a purely transition based model and as a "mixed" model. These experiments clearly illustrated the advantage of using transitions that depend on the acoustic context.

7. ACKNOWLEDGMENTS

The author would like to thank Christoffe Ris for details of his own work and for providing the initial forced Viterbi alignment of the PHONEBOOK database and phonetic transcriptions of words not in the CMU 0.4 dictionary. Also I thank DCS at Sheffield University and in particular Steve Renals for kindly hosting a five months visit in the Spring of 1997 during which much of this work was done. This work was partially supported by ESPRIT LTR project SPRACH (20077).

8. REFERENCES

[1] BAHL, L. R., BROWN, P. F., DE SOUZA, P. V., AND MERCER, R. L. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proceedings of ICASSP'86* (1986), pp. 49–52.

[2] BENGIO, Y., AND FRASCONI, P. Input/output HMMs for sequence processing. *IEEE Transactions on NN 7*, 5 (1996), 1231–1249.

[3] BENGIO, Y., LECUN, Y., NOHL, C., AND BURGES, C. Lerec: A NN/HMM hybrid for on-line handwritting recognition. *Neural Computation 7*, 5 (1995).

[4] DUPONT, S., BOURLARD, H., DEROO, O., FONTAINE, V., AND BOITE, J. Hybrid HMM/ANN systems for training independent tasks: Experiments on phonebook and related improvements. In *Proceedings of ICASSP '97* (1997), pp. 1767–1770.

[5] HENNEBERT, J., RIS, C., BOURLARD, H., RENALS, S., AND MORGAN, N. Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems. In *Proceedings of EUROSPEECH'97* (1997).

[6] HERMANSKY, H., AND MORGAN, N. RASTA processing of speech. *IEEE Transactions on SAP 2*, 4 (1994), 578–589.

[7] JOHANSEN, F. T., AND JOHNSEN, M. H. Non-linear input transformations for discriminative HMMs. In *Proceedings of ICASSP'94* (1994), vol. I, pp. 225–28.

[8] JUANG, B. H., AND RABINER, L. R. Hidden Markov models for speech recognition. *Technometrics 33*, 3 (August 1991), 251–272.

[9] KONIG, Y., BOURLARD, H., AND MORGAN, N. REMAP: Experiments with speech recognition. In *Proceedings of ICASSP'96* (1996), pp. 3350–3353.

[10] KROGH, A. Hidden Markov models for labeled sequences. In *Proceedings of the 12th IAPR ICPR'94* (1994), pp. 140–144.

[11] KROGH, A., AND RIIS, S. Hidden Neural Networks. Submitted.

[12] PITRELLI, J., FONG, C., WONG, S., SPITZ, J., AND LEUNG, H. Phonebook: A Phonetically-Rich Isolated-Word Telephone-Speech Database. In *Proceedings of ICASSP'95* (1995), pp. 101–104.

[13] RENALS, S., AND HOCHBERG, M. Efficient evaluation of the LVCSR search space using the NOWAY decoder. In *Proceedings of ICASSP '96* (1996), pp. 149–152.

[14] RENALS, S., MORGAN, N., BOURLARD, H., COHEN, M., AND FRANCO, H. Connectionist probability estimators in HMM speech recognition. *IEEE Transactions on SAP 2*, 1 (1994), 161–74.

[15] RIIS, S. *Hidden Markov Models and Neural Networks*. PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 1998. To appear May 1998.

[16] RIIS, S. K., AND KROGH, A. Hidden neural networks: A framework for HMM/NN hybrids. In *Proceedings of ICASSP '97* (1997), pp. 3233–3236.

[17] SCHWARZ, R., AND CHOW, Y.-L. The N-best algorithm: An efficient and exact procedure for finding the N most likely hypotheses. In *Proceedings of ICASSP'90* (1990), pp. 81–84.