CDNN: A CONTEXT DEPENDENT NEURAL NETWORK FOR CONTINUOUS SPEECH RECOGNITION

Hervé Bourlard [†,‡], *Nelson Morgan* [‡], *Chuck Wooters* [‡], *and Steve Renals* [‡]

[†] L&H SpeechProducts, Ieper, B-8900 BELGIUM
[‡] International Computer Science Institute, 1947 Center St., Berkeley CA 94704, USA

ABSTRACT

Over the period of 1987-1991, a series of theoretical and experimental results have suggested that MultiLayer Perceptrons (MLP) are an effective family of algorithms for the smooth estimate of highly-dimensioned probability density functions that are useful in continuous speech recognition [1]-[4]. All of these systems have exclusively used context-independent phonetic models, in the sense that the probabilities or costs are estimated for simple speech units such as phonemes or words, rather than biphones or triphones. Numerous conventional systems based on Hidden Markov Models (HMM) have been reported that use triphone or triphone-like context-dependent models [6]-[9]. In one recently reported case [10], the outputs of many context-dependent MLPs (one per context class) were used to help choose the best sentence from the N best sentences as determined by a context-dependent HMM system. In this paper, we show how, without any simplifying assumptions, we can estimate likelihoods for context-dependent phonetic models with nets that are not substantially larger than our context-independent MLPs.

1. INTRODUCTION

Earlier work has shown the ability of Multilayer Perceptrons (MLPs) to estimate emission probabilities for Hidden Markov Models (HMM) [1]-[4]. In these reports, we have shown that these estimates led to improved performance over standard estimation techniques when a fairly simple HMM was used. Current results on the speaker-independent DARPA Resource Management database for MLP monophone estimators continue to support this contention [5], and are roughly at the 90% accuracy level for a wordpair grammar. However, current state-of-the-art continuous speech recognizers (that have roughly half this error rate for the same task) require HMMs with greater complexity, e.g. multiple densities per phone and/or context-dependent phone models. Will the consistent improvement we have seen in these tests be washed out in systems with more detailed models?

One difficulty with more complex models is that many more parameters must be estimated with the same limited amount of data. Brute-force application of our earlier techniques would result in an output layer with many thousands of units, and a network with many millions of connections. This network would be impractical to train, both in terms of computation and learnability, using current-sized public data bases. In each of our earlier studies, a simple context-independent trained network used a single output unit for each phone. For our most recent Resource Management tests, we use 69 of these units. Were one to consider the coarticulatory effects from the right only, this number would expand out to 69^2, or over 4000. Considering both right and left context, we would require 69^3 units, or about 328,000. With a typical hidden layer of 500 units, we would have over 10^8 connections, which is far too many for a practical system.

Of course, HMM researchers have had a similar consideration in reducing the number of parameters in their VQ-based or tied-mixture-based systems. The solution has been, in one form or another, to use a reduced number of context-dependent models (typically a few thousand). However, this is still a large number. For instance, with 4000 outputs and 500 hidden units, a network would still have over 2 million connections, which makes good generalization difficult for training sets of a few hundred thousand frames. Even if enough training data were available, networks with millions of parameters can be expected to take impractical amounts of time to train using back-propagation approaches, even with fast special-purpose machines such as our Ring Array Processor (RAP) [11].

In the approach reported here, we are able to estimate, without any simplifying assumptions, likelihoods for context-dependent phonetic models with nets that are not substantially larger than our context-independent MLPs, and that require only a small increase in computation.

2. CDNN: A CONTEXT-DEPENDENT NEURAL NETWORK

As described in [1]-[4], with a few assumptions an MLP may be viewed as estimating the probability $p(q_k|x_n)$ where q_k is a speech class or an HMM state $\in \mathcal{Q} = \{q_1, \ldots, q_K\}$, the set of all possible HMM states from which phoneme models are built up, and x_n is the input data (speech features) for frame n. If there are K such classes, then K outputs are required in the MLP. This probability may be considered "context-independent" in the sense that the left-hand side of the conditional probability contains no term involving the neighboring phones.

For a context-dependent model, we may wish to estimate

0-7803-0532-9/92 $3.00

the joint probability of a current HMM state with a particular neighboring phonetic category. Using $\mathcal{C}$ to represent the set of possible contexts, we wish to estimate $p(q_k, c_j|x_n)$, where $c_j \in \mathcal{C} = \{c_1, \ldots, c_L\}$. If there are L context classes, this will require $K \times L$ output units for an MLP estimator. However, if we use the definition of conditional probability, the desired expression can be broken down as follows:

$$p(q_k, c_j|x_n) = p(q_k|x_n) \times p(c_j|q_k, x_n) \quad (1)$$

Thus, the desired probability is the product of the monophone posterior probability and a new conditional. The former can be realized with the usual monophone network. Viewing an MLP as an estimator of the left side of a conditional given the right side as input, the second term can be estimated by an MLP trained to generate the correct context class given inputs of the current class and the speech input frame. The latter network only has as many outputs as there are context classes.

This procedure reduces the training of a single network with $K \times C$ outputs to the training of two smaller networks with K and C outputs respectively, and represents a generic way of splitting large MLPs used in classification mode into several smaller ones. It has the potential, however, of requiring much greater computation during the recognition phase. Indeed, if one implements this method naively, the second network must be computed K times for each frame during recognition, since the output probabilities depend on an assumption of the current class (corresponding to a monophone model in a hypothesized word sequence at that point in the dynamic programming). The next section will describe how this expense can largely be circumvented.

As a particular application of this general rule, the problem of hybrid HMM/MLP approaches for triphone modelling will be considered. In this case, the main problem lies in the estimation of probabilities like $p(x_t|q_k, c_j^\ell, c_\ell^r)$ where c_j^ℓ and c_ℓ^r respectively represent the left and right phonemic contexts of state q_k. If one wants to model triphones with neural networks, a straightforward approach could consist in having $K \times L \times L$ output units to model the $K \times L \times L$ possible contextual state probabilities. This would require an excessive number of parameters.

Expanding the joint triphone likelihood as in (1), we get:

$$p(q_k, c_j^\ell, c_\ell^r|x_t) = p(c_j^\ell|q_k, c_\ell^r, x_t).p(c_\ell^r|q_k, x_t).p(q_k|x_t) \ , \quad (2)$$

and

$$p(q_k, c_j^\ell, c_\ell^r) = p(c_j^\ell|q_k, c_\ell^r).p(c_\ell^r|q_k).p(q_k) \ . \quad (3)$$

During recognition and (Viterbi) training, the context-dependent likelihoods $p(x_t|c_j^\ell, q_k, c_\ell^r)$ can then be estimated from (2) and (3) as:

$$p(x_t|c_j^\ell, q_k, c_\ell^r) = \frac{p(c_j^\ell, q_k, c_\ell^r|x_t).p(x_t)}{p(c_j^\ell, q_k, c_\ell^r)}, \quad (4)$$

in which $p(x)$ can be ignored during dynamic time warping.

In fact, relations (1), (2) and (3) are examples of a general approach for splitting a huge MLP used in classification mode into smaller ones without requiring any simplifying assumptions. Now, exploiting the conclusions we derived from the theory of our hybrid HMM/MLP approach for phoneme models (i.e., in classification mode, the output values of the MLP are estimates of the a posteriori probabilities of the output classes conditioned on the input), it can be shown that all the right hand factors appearing in (2) and (3) can be estimated separately by different MLPs:

- $p(c_j^\ell|q_k, c_\ell^r, x_t)$ can be estimated by an MLP in which the output units are associated with the left phonemes of the triphones and in which the input field consists of the current acoustic vector x_t (possibly extended to its left and right contexts), the current state and the right phonetic contexts in the triphones (which are known during training).
- $p(c_\ell^r|q_k, x_t)$ can be estimated by a neural network in which the output units are associated with the right phonemes and in which the input field is constituted by the current acoustic vector x_t (possibly extended to its left and right contexts) and the current state (associated with x_t).
- $p(q_k|x_t)$ is estimated by the same neural network as the one used for modeling phonemes where the input field contains the current acoustic vector only and the output units are associated with the current labels.
- $p(c_j^\ell|q_k, c_\ell^r)$ can be estimated by a neural network in which the output units are associated with the left phonemes of the triphones and where the input field represents the current state and the right phonemes. This provides us with the a priori probability of observing a particular phoneme in the left part of a triphone given particular current state and right phonetic context.
- $p(c_\ell^r|q_k)$ can be estimated by a neural network in which the output units are associated with the right phonemes of the triphones and where the input field represents the current state. This provides us with the a priori probability of observing a particular phoneme on the right side of a particular state. Given the limited number of parameters in this model (i.e., $K \times L$), this probability can also be estimated by counting (i.e., this does not require a network).
- $p(q_k)$ is the a priori probability of a phoneme as also used in the standard hybrid HMM/MLP phonetic approach, and is simply estimated by counting on the training set (i.e., this also does not require a network).

By training these different MLPs and using their output activation values in (4), it is possible to estimate, without any particular assumptions, the probability $p(x_t|q_k, c_j^\ell, c_\ell^r)$ that is required for modelling triphone probabilities to be used in HMMs. This generalizes to triphones the approach developed in [1]-[4] and which was restricted to phoneme models. Of course, for limited training sets, as done with standard HMMs, smoothing of context-dependent and context-independent probabilities [9] still may be important even

with MLPs. Also, training improvements presented in [2], [3] (such as the use of cross-validation technique to improve generalization performance) remain valid in this new approach. Additionally, if c^ℓ and c^r represent broad phonetic classes or clusters rather than phonemes, the above results apply to the estimation of "generalized triphones", such as are defined in [9]. As done previously [1]-[4], the input field containing the acoustic data (e.g., x_t) may also be supplied with contextual information. In this case, the x_t appearing in all the above (and subsequent) probabilities have to be replaced by $X_{t-c}^{t+c} = \{x_{t-c}, \ldots, x_t, \ldots, x_{t+c}\}$, in which c represents the width of the contextual window. This leads to the estimation of triphone probabilities given acoustic contextual information, which is even more important in the case of triphone models.

3. IMPLEMENTATION ISSUE

While the previous section appears to have solved the problem of triphone modelling by neural network, an important implementation issue still has to be taken into account.

We have shown theoretically how to transform, without any particular assumptions or simplifications, the huge neural network which would result from the brute force application of our earlier hybrid HMM/MLP approach for phoneme modelling to triphones. Indeed, instead of having a single MLP estimating $p(q_k|x_t)$, we need to estimate

$$p(q_k, c_j^\ell, c_\ell^r|x_t) \quad , \tag{5}$$

which requires an MLP with $K \times L \times L$ output units. In the previous Section, we have shown that it was possible to estimate the same probability with three smaller MLPs respectively estimating

$$p(c_j^\ell|q_k, c_\ell^r, x_t) \quad , \tag{6}$$

$$p(c_\ell^r|q_k, x_t) \tag{7}$$

and

$$p(q_k|x_t) \quad . \tag{8}$$

However, while this strongly reduces the memory requirement and the number of parameters, a naive implementation of these smaller networks would require much more computation.

In the case of phonetic modelling, a single MLP provided with the current acoustic vector x_t as input can estimate $p(q_k|x_t)$ for all possible classes q_k on the associated output units. This remains valid for triphone modelling if we use the huge network with x_t at its input and $K \times L \times L$ output units, each output unit being associated with a particular triphone. However, when this huge network is decomposed into smaller networks computing (6), (7) and (8), the first two networks must have input values depending on the phonetic contexts constituting the triphones. For example, the input field of the neural network estimating (8) on its output units is constituted by the concatenation of the current acoustic vector x_t and the middle and right phonetic contexts in the triphones. Since the MLP training is supervised, i.e. we know exactly which triphone is associated with a particular acoustic vector, this is not a problem during training. However, this is no longer the case during recognition where we do not know in advance which triphone is associated with x_t.

Therefore, in principle one would have to compute network activations at each frame for each possible phonetic context. This would amount to $L \times L$ times the monophone network computation, and would generally be prohibitive. Fortunately, a simple restriction on the network topology permits the pre-calculation of contextual phonetic contributions to the output; this computation can be done at the end of the training phase, prior to the recognition of any speech. By simply partitioning the net so that no hidden unit receives input from both phonetic context units and data input units, we can pre-calculate the contribution to the output units (prior to the output nonlinearity) for all possible combinations of left and right contexts, and form a table of these contributions. During recognition, the pre-sigmoid output values resulting from data vectors can be computed by a forward pass on the net for each frame. For each hypothetical triphone model, these contributions from the data inputs can be added to the corresponding context contributions from the table. The major new computation (in comparison with the monophone case) then is simply the cost of some lookups, both for the contextual contributions, and for the final sigmoidal nonlinearity, which must now be re-computed for each hypothesized triphone (as opposed to once per frame, as in the monophone case). In practice this only doubles or triples the computation time, a reasonable cost for triphone models.

As an example, let us consider the case of $p(c_j^\ell|q_k, c_\ell^r, x_t)$. More formally, letting $Y_j(q_k, c_\ell^r)$ be the contribution to the pre-sigmoid output for state q_j for the phonetic context-dependent partition of the net, and letting $Z_j(x_t)$ be the contribution to the pre-sigmoid output for state q_j for the data vector input. Then

$$p(c_j^\ell|q_k, c_\ell^r, x_t) = f(Y_j + Z_j) \quad , \tag{9}$$

where f is the standard sigmoid function. A $(K \times L \times L)$-dimensional table Y is computed after network training by running the phonetic-context-dependent partition of the network (which has no inputs from the data vector) $K \times L \times L$ times, i.e. for all possible output units and for all possible combinations of phonetic contexts, with no output sigmoid computation. This table loading is a negligible amount of computation compared to the training of the network. During recognition, for each acoustic vector x_t, it is then enough to run each MLP only once to get the contribution Z_j of the data inputs for each output unit q_j. For each hypothetical triphone model, this contribution Z_j just has to be added to the corresponding context contribution Y_j obtained by a simple lookup in table Y. In fact, this is equivalent to considering Y_j as an added bias (of output unit q_j) that depends on the phonetic context. Of course, the same method can be applied to the MLP computing $p(c_\ell^r|q_k, x_t)$. Also, for $p(c_j^\ell|q_k, c_\ell^r)$ and $p(c_\ell^r|q_k)$ it is sufficient to compute look-up tables at the end of the training phase for use in (3).

4. DISCUSSION AND RESULTS

4.1. The unrestricted split net

In equation (1), when splitting the original MLP with $K \times L$ output units into two smaller networks with K and L outputs respectively, the number of parameters is drastically reduced, which could affect the quality of the conditional distributions' estimation. However, parameter reduction is exactly the aim of the proposed approach, both to reduce computation and to improve generalization. As it was done for $p(q_k|x_n)$ [1]-[4], it will be necessary to find (e.g. by using cross-validation techniques) the number of hidden units (and hence the number of parameters) leading to the best estimate of $p(c_j|q_k, x_n)$. The desired probabilities can in principle be estimated without any statistical assumptions (e.g., independence). Of course, this is only guaranteed if the training does not get stuck in a local minimum and if there are enough parameters.

4.2. The topologically restricted net

As shown above, while reducing the number of parameters, the splitting of the network into two smaller networks results in much greater computation in the contextual network. To avoid this problem it is proposed to restrict the topology of the second network so that no hidden unit shares input from both q_k and x_n. Consequently, the q_k input only changes the output thresholds. However, a recent experiment with frame classification for continuous speech (trained using 160,000 patterns from 500 sentences uttered by a speaker in the Resource Management continuous speech recognition corpus) suggested that this did not affect the correct estimation of $p(c_j|q_k, x_n)$. In this example, the network with a split hidden layer predicted (for a test set of 32,000 patterns from 100 sentences) the correct right context 63.6% of the time, while a network with a unified hidden layer predicted the context 63.5% of the time, an equivalent figure.

4.3. Preliminary results and conclusion

Prior to experimenting with the CDNN for continuous speech recognition using biphone and triphone models (to be reported at a later date), we wanted to check experimentally that the split MLP was equivalent to the original one. We compared biphone probabilities generated by the original and split MLP for the speaker independent Resource Management database. The number of hidden units in each MLP was chosen such that the number of parameters was approximately the same in both cases. After having trained both cases on 4,000 sentences, biphone probabilities were computed on a test set of 100 sentences pronounced by 4 different speakers, yielding a total of 17,012,088 probabilities. To compare both sets of probabilities we computed the correlation coefficient to be 0.65, and the mean absolute difference that was equal to 0.0017. Thus, the two sets of probabilities are significantly correlated. This suggests that CDNN may be a good way to compute context-dependent probabilities with nets that have a limited number of parameters and that require an acceptably small increase in computation over the context-independent case.

REFERENCES

[1] H. Bourlard & C.J. Wellekens, "Links Between Markov Models and Multilayer Perceptrons", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 12, pp. 1167-1178, December 1990.

[2] H. Bourlard, & N. Morgan, "Merging Multilayer Perceptrons & Hidden Markov Models: Some Experiments in Continuous Speech Recognition" *Artificial Neural Networks: Advances and Applications*, North Holland Press, 1990, E. Gelenbe editor, 1991.

[3] N. Morgan & H. Bourlard, "Continuous Speech Recognition Using Multilayer Perceptrons with Hidden Markov Models", *Proc. IEEE Intl. Conf. on Acoustics, Speech, & Signal Processing*, pp. 413-416, Albuquerque, New Mexico, 1990.

[4] N. Morgan, C. Wooters, H. Bourlard, & M. Cohen, "Continuous Speech Recognition on the Resource Management Database using Connectionist Probability Estimation", *Proceedings of ICSLP-90*, 1337-1340, Kobe, Japan, 1990.

[5] S. Renals, N. Morgan, M. Cohen, H. Franco, & H. Bourlard, "Connectionist probability estimation in the DECIPHER speech recognition system", *ICASSP92.*

[6] H. Murveit, M. Cohen, P. Price, G. Baldwin, M. Weintraub, "SRI's DECIPHER System", *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1989, pp.238-242.

[7] D.B. Paul, "The Lincoln Robust Continuous Speech Recognizer", *IEEE Proc. of the 1989 Intl. Conf. on ASSP*, Glasgow, Scotland, May 1989, pp. 449-451.

[8] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, & J. Makhoul, "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech", *IEEE Proc. of the 1985 Intl. Conf. on ASSP*, Tampa, FL, April 1985, pp.1205-1208.

[9] K.F. Lee, "Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition", *IEEE Trans. on ASSP*, Vol. 38, No. 4, pp. 599-609, 1990.

[10] S. Austin, J. Makhoul, R. Schwartz, & G. Zavaliagos, "Continuous Speech Recognition Using Segmental Neural Networks", *Proc. of the Fourth DARPA Workshop on Speech and Natural Language*, in press, February 1991.

[11] N. Morgan, J. Beck, P. Kohn, J. Bilmes, E. Allman, & J. Beer, "The RAP: a Ring Array Processor for Layered Network Calculations," *Proc. of Intl. Conf. on Application Specific Array Processors*, pp. 296-308, IEEE Computer Society Press, Princeton, N.J., 1990.