Hybrid HMM/ANN Systems for Speech Recognition: Overview and New Research Directions

Hervé Bourlard[1,2] and Nelson Morgan[2,3]

(1) IDIAP, Martigny, Switzerland
(2) Intl. Comp. Science Institute, Berkeley, CA
(3) UC Berkeley, Berkeley, CA

1 Introduction

In recent years there has been a significant body of work, both theoretical and experimental, that has established the viability of *Artificial Neural Networks* (ANNs) as a useful technology for speech recognition. It has been shown that neural networks can be used to augment speech recognizers whose underlying structure is essentially that of *Hidden Markov Models* (HMMs). In particular, we and others have demonstrated that fairly simple ANN structures can be discriminatively trained to estimate emission probabilities for an HMM.

For a number of controlled tests, simple speech recognition systems (using context-independent phone models) based on this approach have been observed to be at least as accurate as HMM-based systems using more common structures for recognition and training. Additionally, they appear to be more efficient than current competitive approaches, in terms of CPU and memory run-time requirements.

In this paper, we first give a brief overview of current state-of-the-art *Automatic Speech Recognition* (ASR), and then describe the use of ANNs as statistical estimators. We then review the basic principles of our hybrid HMM/ANN approach and describe some experiments. We discuss some current research topics, including new theoretical developments in training ANNs to maximize the posterior probabilities of the correct models for speech utterances. Finally, we conclude with the description of a new ASR approach using hybrid HMM/ANN systems to process multiple input streams, with sub-band based ASR as a particular case showing improved robustness to noise.

2 Technology Background

2.1 Automatic Speech Recognition

The basic task of *Automatic Speech Recognition* (ASR) is to derive a sequence of words from a stream of acoustic information. A more general task is automatic speech understanding, which includes the extraction of meaning (for instance, a query to a database) or producing actions in response to speech. For many applications, interaction between system components devoted to semantics, dialog generation, etc., and the speech recognition subsystem can be critical. However, in order to simplify the focus of this article, we will only consider recognition per se.

ASR systems typically consist of several major components that are illustrated in Figure 1.

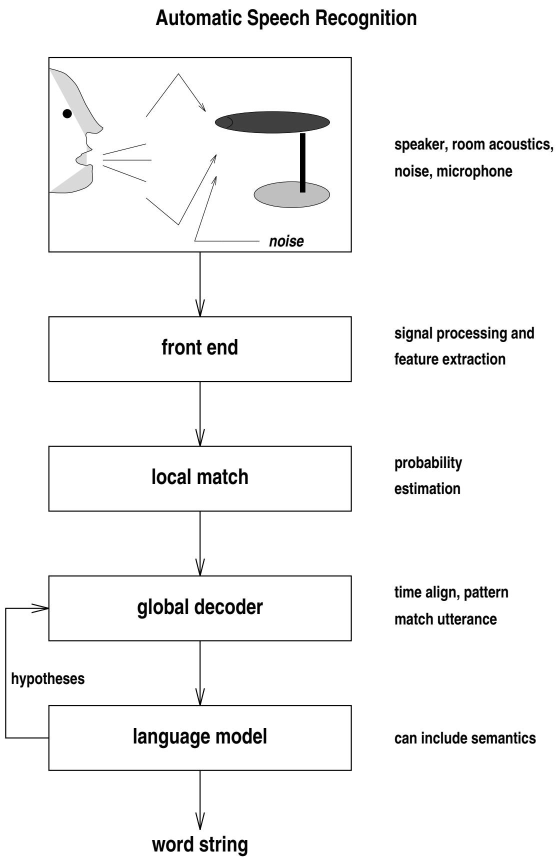

Fig. 1. *Block diagram of continuous speech recognition.*

Note that the first block, which consists of the acoustic environment plus the transduction equipment (microphone, preamplifier, anti-aliasing filter, sample-and-hold, A/D converter) can have a strong effect on the generated speech representations. For instance, additive noise, room reverberation, microphone position and type, dc offsets in the preamplifier or sample-and-hold, and ground loops in the equipment can all be associated with this part of the process. The second block, the feature extraction subsystem (sometimes called the front end) is intended to deal with these problems, as well as deriving acoustic representations that are both good at separating differing classes of speech sounds and effective at suppressing irrelevant sources of variation. These two blocks, though not discussed further in this article, are worthy of significant study, and like the components devoted to understanding cannot ultimately be completely partitioned from the rest of the recognizer.

The next two blocks in Figure 1 illustrate the core acoustic pattern matching operations of speech recognition. In nearly all ASR systems, a representation of speech, such as a spectral or cepstral representation, is computed over successive intervals, e.g., 100 times per second. These representations or speech *frames* are then compared to the spectra or cepstra for speech that were used for training, using some measure of similarity or distance. Each of these comparisons can be viewed as a local match. The global match is a search for the best sequence of words (in the sense of the best match to the data), and is determined by integrating many local matches. The local match does not typically produce a single hard choice of the closest speech class, but rather a group of distances or probabilities corresponding to possible sounds. These are then used as part of a global search or decoding to find an approximation to the closest (or most probable) sequence of speech classes, or ideally to the most likely sequence of words. Another key function of this global decoding block is to compensate for temporal distortions that occur in normal speech. For instance, vowels are typically shortened in rapid speech, while some consonants may remain nearly the same length.

The most common global decoding approach is some form of *dynamic programming* (DP) [26], in which time warping of the input against possible speech representations results in the most likely sequence of sound categories to match the input. There are many variations to this process, but in general the local computation consists of finding the lowest cost path through possible representations by:

1. For each time step, consider possible transitions from the previous time step.
2. For each such possible transition, take the cost of the sound sequence that has been hypothesized so far, and add it to the cost of the transition.
3. Choose the least costly transition according to this number, and add it to the cost of the local match, keeping track of the pointer to the winning prior sequence. The sum is the current global cost of the sequence that can be backtracked at this point from the pointers that have been saved.
4. At the end of the utterance, backtrack from the lowest global cost to generate the corresponding speech sequence.

This description is greatly oversimplified from what is used in most systems; for instance, the local and transition costs are generally implemented as (negative) log probabilities[1], so that the sums can be interpreted as giving the most

[1] Note that for a multivariate Gaussian distribution, the exponent is the negative of the Mahalanobis distance between the data vector and the mean vector.

probable sequences. These sums of logs are equivalent to the log of products of probabilities. Performing these operations in the log domain is preferable for a number of reasons (e.g., numerical stability). Additionally, the decoding procedure is often done using different algorithms, for instance using a tree-based search, or using multiple passes with increasingly detailed models. Nonetheless, the DP (or Viterbi search for statistical systems) described above is at the base of many recognition systems, including the class described here.

This procedure can also be seen as corresponding to an underlying model of speech, namely that of words consisting of sequences of speech units that can have varying length. Implicitly this means that each speech unit has constant spectral properties until one jumps to the next one, an assumption that is clearly wrong for natural human speech. Nonetheless, it is a simplifying assumption that permits the use of powerful statistical techniques that are briefly discussed in the next section.

The last block in Figure 1 consists of the language model, which determines the hypotheses that are considered in the global search. This block can also process the global decoder output further. For instance, if the decoder generates not only the most likely sentence but rather the N most likely, (e.g., N=100), the language model could rescore these sentence according to grammar or semantics. As with the front end, this research topic will not be described further here.

As noted above, most often the local distance computation is implemented probabilistically. Typically, the probability of an observed spectrum or cepstrum is computed for each possible sound. These probabilities are most commonly estimated using a mixtures (weighted sum) of Gaussian distributions, or by vector quantizing the spectra and counting the co-occurrences of spectral prototypes and speech categories in order to derive discrete probability distributions. Additionally, as will be explained in this article, connectionist networks can be used to generate the required probabilities. First, however, we briefly explain the underlying structure of the probabilistic approach.

2.2 Hidden Markov Models (HMMs)

A *hidden Markov model* (HMM) is typically defined (and represented) as a *stochastic finite state automaton* (SFSA) which is assumed to be built up from a finite set of possible states $\mathcal{Q} = \{q_1, \ldots, q_k, \ldots, q_K\}$, each of those states being associated with a specific probability distribution (or probability density function, in the case of likelihoods). A specific HMM M_i will then be represented by a SFSA with L_i states $\mathcal{S}_i = \{s_1, \ldots, s_\ell, \ldots, s_{L_i}\}$, with each $s_\ell \in \mathcal{Q}$, put together according according to a specific (usually predefined, sometimes automatically inferred) topology (usually left-to-right topology when used for speech recognition). Of course, $\mathcal{S}$ may only contain a subset of Q, while also having the same state appearing at different nodes of the SFSA.

According to this formalism, HMMs model the sequence of feature vectors[2] $X = \{x_1, \ldots, x_n, \ldots, x_N\}$ as a piecewise stationary process for which each sta-

[2] Usually, centisecond acoustic vectors resulting from spectral or cepstral analysis.

tionary segment will be associated with a specific HMM state. That is, when using model M, an utterance $X = \{x_1, \ldots, x_n, \ldots, x_N\}$ is modeled as a succession of discrete stationary states $\mathcal{S} = \{s_1, \ldots, s_\ell, \ldots, s_L\}$, $L \leq N$, with instantaneous transitions between these states. As usually defined in our previous papers, notation q_k^n then means that the state of $\mathcal{S}$ hypothesized at time n is associated with the distribution q_k. An example of a simple HMM is given in Figure 2: this could be the model of a short word assumed to be composed of three stationary parts.

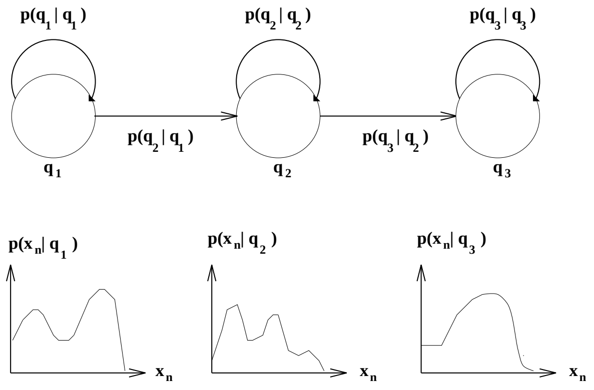

Fig. 2. *A three-state Hidden Markov Model (HMM). A HMM is a stochastic finite state machine, consisting of a set of states and corresponding transitions between states. HMMs are commonly specified by a set of states q_i, an emission probability density $p(x_n|q_i)$ associated with each state, and transition probabilities $P(q_j|q_i)$ for each permissible transition from state q_i to state q_j.*

The approach defines two concurrent stochastic processes: the sequence of HMM states (modeling the temporal structure of speech), and a set of state output processes (modeling the [locally] stationary character of the speech signal). The HMM is called a "hidden" Markov model because the underlying stochastic process (i.e., the sequence of states) is not directly observable, but still affects the observed sequence of acoustic features.

Ideally, there should be a HMM for every possible utterance. However, this is clearly infeasible for all but extremely constrained tasks; generally a hierarchical scheme must be adopted to reduce the number of possible models. First, a sentence is modeled as a sequence of words. To further reduce the number of parameters (and, consequently, the required amount of training material) and to avoid the need of a new training each time a new word is added to the lexicon, word models are often comprised of concatenated sub-word units. Although there are good linguistic arguments for choosing units such as syllables or demi-syllables, the unit most commonly used are speech sounds (phones) that are acoustic realizations of the linguistic categories called phonemes. Phonemes are speech sound categories that are sufficient to differentiate between different words in a language. One or more HMM states are commonly used to model a segment corresponding to a phone. Word models consist of concatenations of phone or phoneme models (constrained by pronunciations from a lexicon), and sentence models consist of concatenations of word models (constrained by a grammar).

Theory and methodology for HMMs are described in many sources, including [29]. Briefly, the fundamental equation relevant for this process is a restatement of Bayes' rule as applied to speech recognition[3]:

$$P(M|X,\Theta) \;=\; \frac{p(X|M,\Theta)P(M|\Theta)}{p(X|\Theta)} \qquad (1)$$

in which Θ is the parameter set and $P(M|X,\Theta)$ is the posterior probability of the hypothesized Markov model M (i.e., associated with a specific sequence of words) given an acoustic vector sequence X. Since it is not known how to compute this probability directly[4], (1) is usually used to split this posterior probability into a likelihood $p(X|M,\Theta)$ that represents the contribution of the *acoustic model*, and a prior probability $P(M|\Theta)$ that represents the contribution of the *language model*. $p(X|M)$ and $P(M)$ are estimated during recognition from subsystems whose parameters are sometimes trained from different training sets (for instance, from an acoustic training set and from a large corpus of written text). For this reason, the two sets of parameters, which we denote as Θ and Θ^* for acoustic and language model parameters respectively, are assumed to be independent. By doing so, and assuming that the estimate of $p(X)$ in (1) is independent of the acoustic parameters Θ (which is actually not true during training), (1) permits the formulation of acoustic model training as a *Maximum Likelihood Estimation* (MLE) problem.

The acoustic likelihood is then computed by expanding it into all possible state paths in M that can generate X:

$$p(X|M,\Theta) \;=\; \sum_{\forall S^j} p(X,S^j|M,\Theta) \qquad (2)$$

where the sum extends over all possible paths S^j of length N in M. This "full" likelihood is sometimes approximated as

$$p^*(X|M,\Theta) \;=\; \max_{\forall S^j} p(X,S^j|M,\Theta) \qquad (3)$$

which is usually referred to as the "Viterbi" approximation, which is often used for recognition without much loss in performance.

[3] In this paper, actual probabilities will be denoted $P(\cdot)$ while probability density functions (likelihoods) will be denoted $p(\cdot)$

[4] However, see [7] for some theoretical work that suggests an approach to training and recognition with a global $P(M|X)$ criterion.

During *recognition* of an unknown utterance X, we have to find the best model M_j from the set of all possible models that maximizes $P(M_j|X,\Theta)$ given a fixed set of parameters Θ, i.e.:

$$j = \underset{\forall i}{\operatorname{argmax}} P(M_i|X,\Theta,\Theta^*) \tag{4}$$

$$\approx \underset{\forall i}{\operatorname{argmax}} p(X|M_i,\Theta)P(M_i|\Theta^*) \tag{5}$$

since Θ is fixed during recognition [consequently turning $p(X|\Theta)$ into a constant factor independent of the model]. This is usually solved by the Viterbi algorithm, a particular case of *dynamic programming* (DP) [26].

During *training*, we want to determine the parameter sets Θ and Θ^* that maximize $P(M_j|X_j,\Theta,\Theta^*)$ for all training utterances X_j, $j = 1,\ldots,J$, associated with M_j (known during training), i.e.,

$$\underset{\Theta,\Theta^*}{\operatorname{argmax}} \prod_{j=1}^{J} P(M_j|X_j,\Theta,\Theta^*) \tag{6}$$

Ideally we thus want to optimize (6) during training. However, as already mentioned above, this problem is usually simplified by using Bayes' rule, yielding

$$P(M_i|X,\Theta,\Theta^*) \simeq \frac{p(X|M_i,\Theta)P(M_i|\Theta^*)}{p(X|\Theta)} \tag{7}$$

$$\simeq \frac{p(X|M_i,\Theta)P(M_i|\Theta^*)}{\sum_j p(X|M_j,\Theta)P(M_j|\Theta^*)} \tag{8}$$

where $\sum_j$ represents the sum over all possible models. It is thus clear that the training of every model should depend on all the other models and on the whole parameter set, yielding proper discrimination between the models. One of the goals of hybrid HMM/ANN systems as discussed here is to actually improve discrimination between the models (see, e.g., [7]); this will be further discussed in Section 4.

However, in standard HMM systems, this optimization is usually simplified by maximizing likelihoods only (maximum likelihood estimation), i.e.,

$$\underset{\Theta}{\operatorname{argmax}} \prod_{j=1}^{J} p(X_j|M_j,\Theta) \tag{9}$$

The parameter set Θ of such a statistical system is trained on acoustic data so that during recognition it produces emission probabilities $p(x_n|q_k,\Theta)$ (see Figure 2) that can be multiplied to produce an approximation to the acoustic probability $p(X|M)$ (assuming statistical independence).

There exist efficient training algorithms to learn the parameters Θ of the probability estimators. The most common form of this procedure is a particular case of Expectation-Maximization (EM) algorithm (often referred to as Baum-Welch or forward-backward algorithm) in which the estimators for the data likelihoods conditioned on each word model ($p(X|M, \Theta)$) are iteratively trained [3]. In the case of Viterbi approximation [equation (3)], the full likelihood is approximated by the likelihood of the most probable path through the states in the models, as given by the DP procedure. The resulting training algorithm will then iteratively improve the DP segmentation and the parameter estimation. This approximation is sometimes more sensitive to poor initializations, but with a good initialization can be particularly straightforward to implement, and ultimately is more convenient for the approaches described in this article. One form of this iteration, then, could be as follows:

1. Given a set of acoustic training data that is phonetically labeled (possibly with errors), train an estimator to generate the data density for any hypothesized state (speech class); i.e., train an estimator of the emission probability density conditioned on the input.
2. Given a probability estimator for the data likelihood of each state, use DP to find the most likely sequence through the states in all the possible model sequences. This step is sometimes called a forced Viterbi alignment (determining the alignment of the sequence of acoustic training vectors with the corresponding phonetic labels).

This procedure, sometimes called embedded Viterbi learning (as used in the segmental k-means algorithm [29], for instance), can be proved to converge to a local optimum; in practice it is repeated until some stopping criterion has been reached. See Sectionřefsec:embedded for the HMM/ANN implementation of this algorithm.

2.3 Artificial Neural Networks (ANNs) as Statistical Estimators

Estimating HMM emission probabilities with an ANN

ANNs can be used to classify speech units such as phonemes or words, typically by mapping temporal representations into spatial ones, or by using recurrences. This is the way ANNs were initially used on simple speech recognition problems. However, ANNs classifying complete temporal sequences have not been successful for continuous speech recognition. In fact, used as such they are not likely to work well for continuous speech, since the number of possible word sequences in an utterance is generally infinite. Also, we presently do not know of any principled way to translate an input sequence of acoustic vectors into an output sequence of speech units with what has commonly been called an ANN. On the other hand, HMMs provide a reasonable structure for representing sequences of speech sounds or words. Assuming such a structure, one good use for ANNs might be to provide the distance measure for the local match block of Figure 1.

For statistical recognition systems, the role of the local estimator must be to approximate probabilities or probability density values. In particular, given the basic HMM equations, we would like to estimate something like the probability

$p(x_n|q_k)$ of Figure 2, that is, the probability of the observed data vector given the hypothesized HMM state (which corresponds to some speech sound). However, HMMs are based on a very strict formalism that is difficult to modify without losing the theoretical foundations or the efficiency of the training and recognition algorithms. Fortunately, ANNs can estimate probabilities that are related to these *emission* probabilities, and so can be fairly easily integrated into an HMM-based approach. In particular, ANNs can be trained to produce the *posterior* probability $P(q_k|x_n)$, that is, the *a posteriori* probability of the HMM state given the acoustic data, if each ANN output is associated with a specific HMM state. This can be converted to emission probabilities using Bayes' rule.

Several authors have shown that the outputs of ANNs used in classification mode can be interpreted as estimates of *a posteriori* probabilities of output classes conditioned on the input [6,12,32]. The proof given in [32], is repeated here. For continuous-valued acoustic input vectors, the *Mean Square Error* (MSE) criterion which is usually minimized during ANN training can be expressed as follows:

$$E = \int p(x) \sum_{k=1}^{K} \sum_{\ell=1}^{K} P(q_k|x) \left[g_\ell(x) - d_\ell(x)\right]^2 \, dx \tag{10}$$

where $g_\ell(x)$ represents the observed output for class q_ℓ given x at the input, and $d_\ell(x)$ represents the associated desired output. Since $p(x) = \sum_{i=1}^{K} p(q_i, x)$, we have:

$$E = \int \sum_{i=1}^{K} \left[\sum_{k=1}^{K} \sum_{\ell=1}^{K} \left[g_\ell(x) - d_\ell(x)\right]^2 P(q_k|x) \right] p(q_i, x) \, dx$$

After a little more algebra, using the assumption that $d_\ell(x) = \delta_{k\ell}$ if $x \in q_k$, and adding and subtracting $p^2(q_\ell|x)$ in the previous equation leads to:

$$\begin{aligned}
E = & \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(g_\ell^2(x) - 2 g_\ell(x) P(q_\ell|x) + P^2(q_\ell|x)\right) \right] p(q_i, x) \, dx \\
& + \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(P(q_\ell|x) - P^2(q_\ell|x)\right) \right] p(q_i, x) \, dx \\
= & \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(g_\ell(x) - P(q_\ell|x)\right)^2 \right] p(q_i, x) \, dx \qquad (11) \\
& + \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(P(q_\ell|x)(1 - P(q_\ell|x))\right) \right] p(q_i, x) \, dx
\end{aligned}$$

Since the second term in this final expression (11) is independent of the network outputs, minimization of the squared-error cost function is achieved by choosing network parameters to minimize the first expectation term. However, the first expectation term is simply the MSE between the network output $g_k(x)$ and the posterior probability $P(q_k|x)$. Minimization of (10) is thus equivalent to minimization of the first term of (11) is independent of the, i.e., estimation of $p(q_k|x)$ at the output of the MLP. This shows that a discriminant function obtained by minimizing the MSE retains the essential property of being the best approximation to the Bayes probabilities *in the sense of mean square error*. A similar proof was given in [32] for the relative entropy cost function.

Since these proofs are only based on the minimized criterion (and not on the architecture of the network), they are valid for any of the ANNs, given two conditions:

1. The system must be sufficiently complex (e.g., contain enough parameters) to be trained to a good approximation of the mapping function between input and the output class, and
2. The system must be trained to a global error minimum (where mean squared error and relative entropy are error criteria that will work for this purpose).

It has been experimentally observed that, for systems trained on a large amount of speech, the outputs of a properly trained ANN do in fact approximate posterior probabilities (see Figure 6.1 in [6]), even for error values that are not precisely the global minimum.

Thus, emission probabilities $p(x_n|q_k)$ for use in (standard) HMMs (see Section 2.2) can be estimated by applying Bayes' rule to the ANN outputs. In practical systems, we actually compute scaled likelihoods

$$\frac{P(q_k|x_n,\Theta)}{P(q_k)} = \frac{p(x_n|q_k,\Theta)}{p(x_n|\Theta)} \tag{12}$$

where Θ represents now the ANN parameter set used as the parameters for all the acoustic models. That is, we divide the posterior estimates from the ANN outputs by estimates of class priors, namely the relative frequencies of each class as determined from the class labels that are produced by a forced Viterbi alignment of the training data. The scaled likelihood of the right hand side can be used as an emission probability for the HMM, since, during recognition, the scaling factor $p(x_n)$ is a constant for all classes and will not change the classification. In Section 4 we will discuss this further and will present a theoretical justification.

Figure 3 shows the basic hybrid scheme, in which the ANN generates posterior estimates that can be transformed into emission probabilities as described above, and then used in DP either for forced alignment (when the word sequence is assumed) or for recognition (when word sequences are hypothesized).

Why this is Good

Since we ultimately derive essentially the same probability with an ANN as we would with a conventional (e.g., Gaussian mixture) estimator, what is the point? There are several potential advantages that we and others have observed:

- Model accuracy: ANN estimation of probabilities does not require detailed assumptions about the form of the statistical distribution to be modeled,

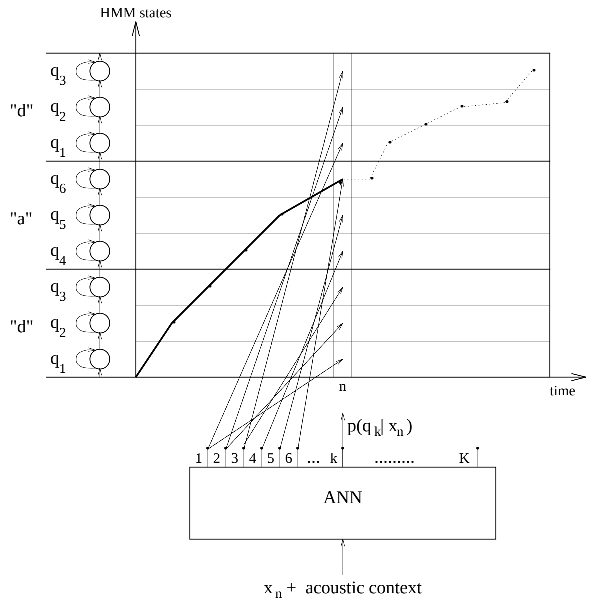

Fig. 3. *At every time step n, the acoustic vector x_n with right and left context is presented to the net (Figure 4). This generates local probabilities that are used, after division by priors, as local scaled likelihoods in a Viterbi DP algorithm. In the figure above, the arrows coming up from each ANN output symbolize the use of these scaled likelihoods (after taking the negative logarithm) as distances from the acoustic input to their corresponding state. The dark solid line shows the best path through the models up to time n (that can be determined by backtracking through pointers), and the dashed path shows its continuation that can be determined once the distances are computed for the last frame in the data.*

resulting in more accurate acoustic models. This is in contrast with more conventional approaches, in which hard choices must often be made, such as between discrete or continuous features, or about the number of significant mixtures used to represent a distribution; within a mixture component, the density is often assumed to be (locally) Gaussian with no correlation between features. For discrete observation variables, the features are assumed to be statistically independent. These types of assumption are not required with an ANN estimator, which will be an advantage particularly when a mixture of feature types are used, e.g., binary and continuous.[5] Specifically, standard HMM approaches require the assumption that successive acoustic vectors are uncorrelated. For the ANN estimator, multiple inputs can be

[5] It is true that one often assumes a certain structure to the neural network, such as a multi-layer perceptron with a single hidden layer composed of some fixed number of hidden units. However, once there are enough hidden units, we have found such structures to be quite insensitive to the precise number of hidden units.

used from a range of time steps, and the network will learn something about the correlation between the acoustic inputs, as discussed below.

- Context sensitivity: In the case of *Recurrent Neural Networks* (RNN) or if several acoustic vectors $X_{n-c}^{n+d} = \{x_{n-c}, \ldots, x_n, \ldots, x_{n+d}\}$ are used at the input of an MLP, local correlation of acoustic vectors can be taken into account in the probability distribution. In the case of an MLP, outputs will be estimates of $P(q_k|X_{n-c}^{n+d})$. This provides a simple mechanism for incorporating acoustic context into the statistical formulation. Of course, ANNs are not the only way to incorporate such context. Many current systems use first and second time derivatives [11,28] computed over a span of a few frames, allowing very limited acoustical context modeling. Some systems transform a context window of a few adjacent frames (typically 3-5 frames in total) with Linear Discriminant Analysis (LDA), which finds a linear transformation that maximizes the between-class variance while minimizing the within-class variance (see, e.g., [13]). The neural network can be seen as a generalization of these approaches that permits arbitrary weights and a nonlinear transformation of the input data.
- Discrimination: ANNs can easily accommodate discriminant training. Of course, as currently done in standard HMM/ANN hybrid discrimination is only local (at the frame level). However, recent theoretical work that allows global discriminant training of hybrid systems will be briefly presented in this paper.
- Parsimonious use of parameters since all probability distributions are represented by the same set of shared parameters. It is also known that it is more "economical" to model boundaries between acoustic classes (i.e., posteriors) than surfaces of density functions (i.e., likelihoods).
- Flexibility: Using a neural network as the acoustic probability estimator permits the easy combination of diverse features, such as a mixture of continuous and categorical (discrete) measures.
- Complementarity: it is sometimes the case that neural networks can supply complementary information to that provided by an existing likelihood-based system. For instance, in one approach, the combination of HMMs with a neural network (referred to as "segmental neural network") provided some improvements over the original system [2]. In that case, an N-best paradigm is used to generate the N-best utterance hypotheses that are then rescored by a neural network taking complete phonetic segments into account.
- Finally, there are two more potential advantages in directly estimating local a posteriori probabilities vs (Gaussian-based) likelihoods:
  1. Recently, it was observed that the availability of posterior probabilities (before division by priors) allowed a more efficient pruning for large vocabulary speech recognition systems [31].
  2. Given the way they are usually computed, the magnitude of the likelihoods depends on the size of the feature space. On the other hand, a posteriori probabilities are independent of the dimension of the input space, allowing for comparisons between different features (e.g., according to the entropy of the posterior distribution).

We and others have performed numerous experiments that have verified these two points. In some of them, a fixed HMM was used and alternate probability estimators were substituted [25,6,30,22]. When these experiments were controlled for the number of parameters, there have been significant improvements using the approaches described here. Some of this quantitative evidence will be briefly summarized in Section 3.5.

3 Hybrid HMM/ANN Recognition System

3.1 The Basic System

As mentioned above, we and others have discriminatively trained large neural networks to estimate HMM emission probabilities for continuous speech recognition. In particular, systems have been developed to perform speech recognition for up to 60,000 word vocabularies, given millions of examples of feature vectors for training. At our laboratories (and those of colleagues throughout the US and Europe), we have focused on using a simple *Multilayer Perceptron* (MLP) that is illustrated in Figure 4, though similar results have been achieved at other labs with structures such as RNNs [16].

It is deceptively simple, consisting of a single large hidden layer, typically with between 500 and 4000 hidden units that receive input from several hundred acoustic variables (e.g., 9 frames of acoustic context consisting of 12th order Perceptual Linear Prediction coefficients (PLP-12) [15] and log energy, along with their derivatives, or 26 features per frame).[6] The output typically corresponds to simple context-independent acoustic classes such as phones defined for the TIMIT phonetic database, using 61 phones. Each word model consists of a succession of phone models, and each phone model uses a single density, with emission probabilities calculated from MLP outputs via Bayes' rule.

Despite this apparent simplicity, there are some significant characteristics of this system that have appeared to be necessary for good performance. The major points are summarized in the following sections; for further explanation, see [6].

3.2 Training

We and others have used on-line training instead of off-line (true gradient) back-propagation. In this approach, the weights are adjusted in the direction of the error gradient with respect to the weight vector, as estimated from a single pattern. With an accurate estimate of the error gradient, one could proceed in the direction of the local training minimum. However, the per-pattern gradient estimate can be viewed as a noisy estimate of the gradient over the entire training set. The size of the learning step can be viewed as the magnitude of the noise; in the limit, very large learning steps move over the error surface randomly, while

[6] Many experiments have been done in our lab that resulted in this choice of input features. Some of these are reported in [25].

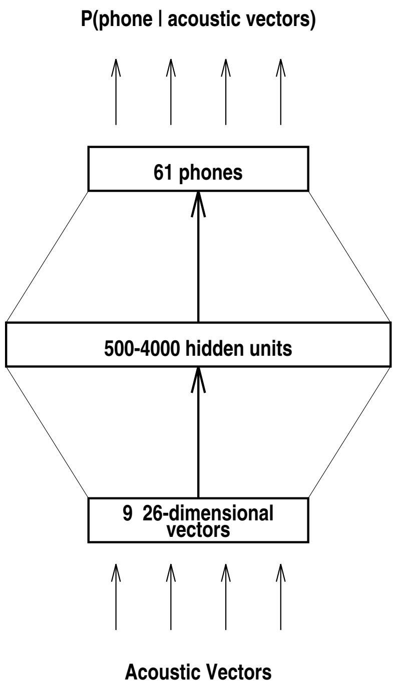

Fig. 4. *Acoustic vectors from the current frame, 4 previous, and 4 following frames are processed by a single large hidden layer. The output corresponds to phonetic categories used for labeling of the TIMIT database.*

very small steps closely correspond to the true gradient. In fact, it can be beneficial to have more noise (larger steps) initially, in order to escape from potentially poor local solutions. Additionally, given realistic training data, which is typically quite redundant, each full pass through the data represents many passes through similar subsets, and thus can be relatively efficient. A compromise approach that we have often found to be quite efficient (particularly for parallel implementations) is to collect gradient information over a moderate number of patterns (e.g., 32) before updating weights. We currently refer to this as "bunch" mode training.

In practice, using on-line gradient search and a relative entropy error criterion, only a small number of passes through the data are required to phonetically train the network (typically 1 to 5). This is generally unchanged for "bunch" mode operation.

In addition to the use of on-line or bunch-mode training, other aspects of the training method include:

– Cross-validation – It is necessary to use a stopping criterion based on an independent portion of the data, i.e., utterances that are not used for training. While this is a good general rule in training pattern classifiers, most of the early published suggestions for neural network stopping criteria were measures based on the training set, e.g., gradient magnitude or slope. The networks that were ultimately successful for continuous speech recognition are quite large, often using hundreds of thousands to millions of parameters. These nets are susceptible to overfitting the training data, resulting in bad probability estimation and very poor generalization performance on the test set. In addition to merely halting the training based on performance for an independent validation set, a training procedure can be used in which the learning rate is also adjusted to improve generalization [23]. Specifically, the learning rate is reduced (typically by a factor of 2) when cross-validation indicates that a given rate is no longer useful. Additionally, we have empirically noted that after the first reduction, only a single epoch at each rate is useful. The heuristic of only permitting a single pass for any learning rate after the initial one cuts down the number of epochs by almost a factor of two, and has little effect on final performance.

- Training criterion – Using relative entropy instead of the MSE criterion speeds convergence. The correction resulting from this criterion is always linear and does not saturate when the output values are at the extremes (tails) of the sigmoid (where the correction for the MSE criterion is negligible).
- Initialization of output biases – Histograms of the output biases of phonetically trained MLPs showed a narrow distribution around a strongly negative value (typically around -4). This is no coincidence, since the input to the sigmoid nonlinearity for and output unit produces the log odds, or $\log \frac{p(q|x)}{1 - p(q|x)}$ when the output produces $p(q|x)$. When the evidence from the data is equivocal, this is roughly equal to $\log \frac{p(q)}{1 - p(q)}$, and since these each $p(q)$ is much less than 1, the sigmoid input is roughly equal to $\log p(q)$. Under the assumption that the data is uninformative, the weighted sum due to the input from the previous layer can be ignored, and the bias should be roughly the log prior for the associated class. This is a rough argument, and for specific distributions (such as a Gaussian) it can be shown to be inaccurate. Nonetheless, the empirical observation (from histograms) is that it is roughly true, at least in the sense that the average output bias of the converged network is close to the average log prior probability. Additionally, it has been confirmed that initializing the biases to the rough range that they will ultimately approach speeds convergence, and slightly improves the results.
- Random pattern presentation – In earlier forms of our analysis we presented the data sequentially according to the speech signal. Sequential presentation of the acoustic vectors to the net (i.e., in the order that they were spoken) can cause slow convergence, requiring a very low learning rate in the case of on-line training. In the current method, the speech vectors are presented at random (preserving the relative frequencies of the classes), which speeds up ANN training, and also slightly improves the results. In a variant on this approach for practical training using speech databases whose size exceeds the physical memory, blocks of sequential sentences (which can be randomized

at the sentence level) are read from disk into physical memory, and frames can be presented randomly from within the block. In both schemes, it does not appear to matter whether random sampling is done with or without replacement (i.e., it does not matter whether each random frame choice is constrained to be a different one than had already been chosen).

We note here that for the case of RNN training, sequential frame presentation is necessary since the structure is one with an infinite impulse response. However, in practice RNN-based hybrid systems have provided equivalent performance to that provided by MLP-based hybrids.

3.3 State Priors and Pronunciation Model

As noted earlier, in the current HMM/ANN paradigm, data likelihoods are estimated by applying Bayes' Rule to the ANN outputs, or, in practice, dividing each posterior probability by the corresponding class priors to get scaled data likelihoods, as shown in (12).

However, it can also be shown that, in theory, HMMs can be trained using local posterior probabilities as emission probabilities [6], resulting in models that are both locally and globally discriminant. See Section 4 for a brief description of some current work in this area, which is described more fully in [7].

For current systems, there are generally mismatches between the prior class probabilities implicit to the training data and the priors that are implicit to the lexical and syntactic models that are used in recognition. For instance, Figure 5 shows the HMM for a pronunciation of "the cat."

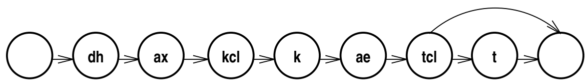

Fig. 5. *Simplified pronunciation model for the phrase "the cat" using Darpa-bet symbols for the phones. The "dh" symbol refers to the voiced form of "th", that is, one in which the vocal cords are vibrating. The initial consonant of the second word is represented by two states, one corresponding to the k-closure and the second corresponding to the k-burst. The final consonant of the second word is represented by two states, one corresponding to the t-closure and the second corresponding to the t-burst, but often the t-burst is omitted; hence the alternate path skipping this sound. The states without labels are non-emitting states representing the start and finish of the phrase.*

The topological definition given in the figure will (in combination with all of the other models) determine the prior probability for each phone during recognition; for instance, if the sound "ax" only occurs after "dh," then the prior probability of the former would be dependent on the prior probability of the latter sound. Depending on both this collection of pronunciation models and the language model, each phone in a sequence like "the cat" will have some prior probability of occurring that may be a poor match to the relative frequencies in the training set, particularly if the pronunciation models come from dictionaries and if the language model is inferred from large text corpora. This can result in significant degradations in recognition performance when the posteriors are used for recognition directly. Thus, it is generally safer to divide the ANN outputs by class priors, taking care to handle the cases of classes that rarely or never occur in the training set, leading to negligible or zero values for the estimates of the class priors.

On the other hand, it would ultimately be desirable to infer statistical word and word sequence models that are consistent with the acoustic training data (or at least to take advantage of the prior information implicit to the acoustic training data). For this case, it would (in principle) be preferable to use posterior probability estimates from the network. Colleagues at ICSI have in fact observed during some experiments with pronunciation inference that division by class priors is not required in this case.

3.4 Embedded Alignment

ANNs trained for classification require supervision (labeled targets for each pattern). An early problem in applying ANN methods to speech recognition was the apparent requirement of hand-labeled frames for ANN training. Since the ANN outputs can be used in a DP procedure for global decoding (after division by the prior probabilities), it is possible to use embedded Viterbi training to iteratively optimize both the segmentation and the ANN parameters. In this procedure, illustrated in Figure 6, each ANN training is done using labels from the previous Viterbi alignment. In turn, an ANN is used to estimate training set state probabilities, and dynamic programming given the training set models is used to determine the new labels for the next ANN training.

Of course, as for standard HMM Viterbi training, one must start this procedure somewhere, and also have a consistent criterion for stopping. Many initializations can be used, including initializing the training set segmentation linearly or in proportion to average phoneme durations. More recently we have achieved better results initializing the procedure by training an ANN on a standard hand-segmented corpus (TIMIT for the case of American English), and using this ANN to align the training set for any new unlabeled corpus.

3.5 Some Results

The major focus in this paper is to describe the ideas that are in common to a family of methods that are being investigated by a large number of laboratories. As such, we have felt that a strong emphasis on numerical results would be a diversion from the major message - it is not difficult to come up with sets of results that show improvements of method X over method Y. However, we do include here a brief litany of results from several laboratories that seem to confirm some of the major points of this paper:

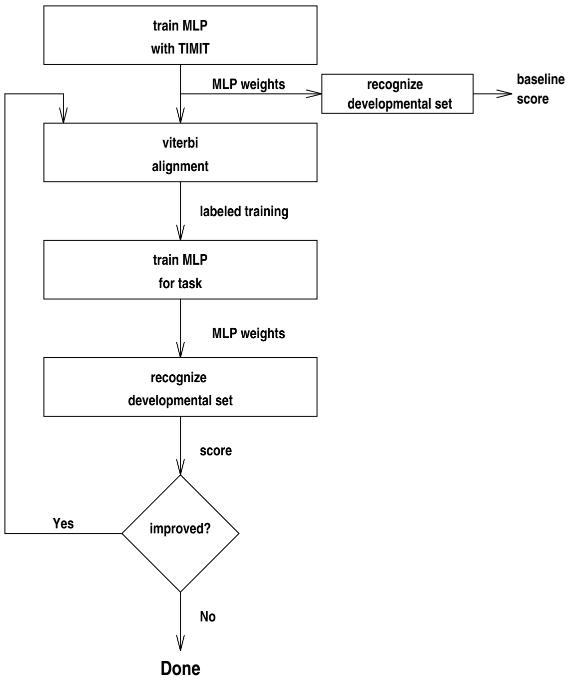

Fig. 6. *Embedded Viterbi learning with MLP.*

1. Many (relatively simple) speech recognition systems based on this hybrid HMM/ANN approach, have been proved, on controlled tests, to be both effective in terms of accuracy (comparable or better than equivalent state-of-the-art systems) and efficient in terms of CPU and memory run-time requirements. More recently, such a system ABBOT from Cambridge University (see, e.g.,[16]) has been evaluated under both the North American ARPA program and the European LRE SQALE project (20,000 word vocabulary, speaker independent continuous speech recognition). In the SQALE evaluation [35] the system was found to perform slightly better than any other leading European system and required an order of magnitude less CPU resources to complete the test. Another striking result is that the acoustic models for this system used several hundred thousand parameters (around 500,000 for ABBOT) while the corresponding models for the competing systems used millions of parameters (around 10^7 as mentioned in the introduction). While the language models for all the systems used a comparable number of parameters (also millions), this still had a significant effect on the practical implementation, and also was an experimental confirmation of the succinctness of neural network probability estimators.
2. It is possible to train networks that are quite large (over a million weights) on millions of speech training patterns with very few epochs; the resulting networks can be used to estimate emission probabilities for HMMs in large

and difficult tasks in continuous speech recognition. This was demonstrated in [24], where we described a 1.6 million-weight network that was trained on 6 million frames of speech from the Wall Street Journal pilot data. This simple estimator was then used to get 16% error on the 5000-word vocabulary evaluation set using a standard bigram grammar. A smaller network (with roughly $\frac{1}{4}$ of the parameters) gave over 20% error on the same test set, showing that at least for our training paradigm and architecture we benefited greatly from the larger number of parameters.

3. For a number of tasks, simple tied-Gaussian mixture systems do not perform as well as hybrid HMM/ANN systems that use a similar number of parameters and the same input features. For equivalent performance, the classical HMM system must be made much more complicated (for instance, typically using context-dependent models and many more parameters). For instance, for the 1000 word vocabulary, perplexity 60 wordpair grammar Resource Management continuous speech recognition task, it was reported in [30] that a context-independent version of SRI's DECIPHER system had 11% word error on a particular Feb 1991 evaluation test set. The same system using MLP outputs as probability estimates achieved 5.8% errors using a similar number of parameters. Even better performance could be achieved by a context-dependent version of DECIPHER (see item 5 below), but this required the use of detailed context and many more parameters. In the same report it was shown that relatively simple forms of context could be incorporated in the network estimators as well and could achieve similar performance as tied-mixture estimators using much more detailed models of context and an order of magnitude more parameters. For further discussion about this with more results, see [33]. In [22], similar conclusions were also drawn for quite a different example, connected digit recognition for a standard TI database. In this case, string error for a moderate-sized MLP (about 11000 parameters) was about 2.5%, while the string error for a 28,000 parameter tied mixture system was 3.8%. In order to get comparable performance, the number of mixtures had to be expanded so that there were an order of magnitude more parameters than in the MLP case.
4. The best current pure-HMM systems currently outperform the best current hybrid systems on some tasks with a large amount of training data. Again, the HMM systems are frequently much more complicated, with many parameters and complex forms of smoothing; frequently the improvement over the simpler hybrid system is small. For instance, in the connected digit case described above [22], by expanding to over 200,000 parameters, Lubensky et al. were able to achieve a 2% string error, which is an error rate reduction of 25% relative to the system incorporating the MLP estimator. For large vocabulary recognition this has also been true as of this writing.
5. Smoothed combinations of hybrid HMM/ANN systems and purely HMM-based systems appear to often give better performance than either one alone. Often this is done by smoothing together probabilities or log probabilities at the frame level. For instance, this was shown for the connected digits example given above. In that case, combining emission probability estimates for

the best Gaussian mixture system with those from the MLP gave roughly a 1.7% string error, which was the best performance reported. Similarly, in [30], the same MLP estimator that yielded a 5.8% word error on the Resource Management task was used to improve a complex Gaussian mixture system from 3.8% error to 3.2% error. In both experiments, the researchers smoothed together log emission probability estimates. Thus, on quite different tasks studied by unrelated laboratories, it was observed that a very good HMM-based system using Gaussian mixture estimators could be further improved by smoothing with estimates from an ANN. Similar results have been observed in other laboratories as well.

6. In work at BBN [39], the subsystems were combined in a different way (by taking a list of the most likely N sentences as estimated by a pure HMM system, and reordering them based on phonetic segment probabilities as estimated by an MLP), but they too reported consistent improvements over the simpler system.

4 Hybrid HMM/ANN Revisited

The hybrid HMM/ANN systems as discussed in the previous section are a modified form of an earlier design we called "discriminant HMMs", which was initially developed to directly estimate and train ANN parameters to optimize global posterior probabilities $P(M|X)$.

This theory has recently been further explored yielding:

1. Better discriminant systems and a new hybrid HMM/ANN approach referred to as REMAP (Recursive Estimation and Maximization of A Posteriori probabilities) [7] based on *conditional transition probabilities* $p(q_\ell|q_k, x_n)$ estimated by a particular form of ANN.
2. Better understanding of the general hybrid HMM/ANN theory and its relationship to what had been done so far.

This work is briefly reviewed in this section.

4.1 Global Posteriors

Similarly to what was done in Section 2.2 (equation 2), it can be shown [6] that the global posterior probability $P(M|X, \Theta)$ can be expressed as:

$$\begin{aligned} P(M|X,\Theta) &= \sum_{\forall S^j} P(M, S^j|X, \Theta) \\ &= \sum_{\forall S^j} P(M, S_1^j, S_2^j, \ldots, S_N^j|X, \Theta) \end{aligned} \tag{13}$$

in which "$\forall S^j$" represents all possible (legal) state sequences in M, S_n^j the specific state of model M visited at time n for path S^j. Of course, as with usual HMMs, S_n^j will be associated with a particular state of the set of possible states $\mathcal{Q}$ and, at time n, will thus take a specific value q_k^n[7], meaning that $S_n^j = q_k$.

If we consider a specific (j-th) state sequence, the posterior probability of the state sequence and the model may be decomposed into the product of an acoustic model and a prior over models ("language model" and state sequences):

$$P(M, S_1^j, \ldots, S_N^j | X, \Theta) \simeq \underbrace{P(S_1^j, \ldots, S_N^j | X, \Theta)}_{\text{ac.model}} \underbrace{P(M | S_1^j, \ldots, S_N^j, \Theta)}_{\text{prior}} \tag{14}$$

The X dependence in the second factor in (14) is dropped since the hidden part (the state sequence) is hypothesized. With the usual assumptions of a first-order Markov process and conditionals on X limited to local context X_{n-c}^{n+c} we can simplify the two factors in (14):

$$P(S_1^j, \ldots, S_N^j | X, \Theta) \simeq \prod_{n=1}^{N} P(S_n^j | S_{n-1}^j, X_{n-c}^{n+d}, \Theta) \tag{15}$$

The "prior" factor in (14) is usually assumed independent of the acoustic model parameters Θ (i.e., parametrized in terms of independent parameters) and can be approximated as:

$$P(M | S_1^j, \ldots, S_N^j) \simeq P(M) \left[\prod_{n=1}^{N} \frac{P(S_n^j | S_{n-1}^j, M)}{P(S_n^j | S_{n-1}^j)} \right] \tag{16}$$

Using these simplifications we can approximate (13):

$$P(M | X, \Theta) \simeq P(M) \sum_{\forall S^j} \left[\prod_{n=1}^{N} P(S_n^j | S_{n-1}^j, X_{n-c}^{n+d}, \Theta) \frac{P(S_n^j | S_{n-1}^j, M)}{P(S_n^j | S_{n-1}^j)} \right] \tag{17}$$

and the Viterbi approximation may be obtained by replacing the sum over state sequences with a maximization.

In hybrid HMM/ANN systems, all emission probabilities are estimated from an ANN. Thus, since all $S_n^j \in \mathcal{Q}$, probabilities $P(S_n^j | S_{n-1}^j, X_{n-c}^{n+d})$ are given by $P(q_\ell^n | q_k^{n-1}, X_{n-c}^{n+d})$ (for all possible k and $\ell \in [1, k]$), q_ℓ^n referring to that specific state distribution q_k of $\mathcal{Q}$ hypothesized at time n in path S^j. All emission probabilities in (17) can thus be estimated to the ANN outputs $P(q_\ell^n | q_k^{n-1}, X_{n-c}^{n+d})$, referred to as conditional transition probabilities.

As with traditional hybrid HMM/ANN systems, these conditional transition probabilities can be estimated by an ANN with K output units and in which the acoustic input X_{n-c}^{n+d} is complemented by a set of additional input units representing the state q^ℓ hypothesized at the previous time step $n-1$. Of course, the network will then have to be estimated for all (q_k, q_ℓ)-pairs allowed by M.

We note here that this formulation (17) has three sets of prior probabilities: $P(M)$ as the usual prior probability of the model (e.g., given by the grammar), $P(q_\ell | q_k)$ representing the training data priors, and $P(q_\ell | q_k, M)$ the Markov

[7] Corresponding to the notation usually used in our previous papers.

model priors. The state transition priors are independent of the HMM topology. The Markov model priors are actually the so-called transition probabilities between states.

In [7], it is shown how the conditional transition probabilities can be obtained at the output of an ANN and how this ANN can be trained to guarantee maximization of $P(M|X, \Theta)$ for the right model M associated with X, yielding global discrimination. This is based on the REMAP algorithm (Recursive Estimation Maximization of A Posterior probabilities), which is a particular kind of GEM (Generalized Expectation Maximization) algorithm. This algorithm is currently under investigation, e.g., to improve current HMM/ANN systems or to yield better estimates of confidence levels.

4.2 Initial HMM/ANN Approach

The above formulation was derived in the context of stochastic finite state acceptor models (also known as discriminative HMMs). However, by removing the dependency on the previous state in (17) we arrive at a hybrid system similar to those previously developed. In this case, (17) becomes:

$$P(M|X, \Theta) \simeq P(M) \sum_{\forall S^j} \left[\prod_{n=1}^{N} P(S_n^j | X_{n-c}^{n+d}) \frac{P(S_n^j | M)}{P(S_n^j)} \right] \tag{18}$$

which gives a clear justification for dividing the local posterior estimate by the training data priors to arrive at the scaled likelihoods that are used in the decoding. This demonstrates that, given the previously stated assumptions, hybrid HMM/ANN systems (as we have formulated them in the past) do produce an estimate of the global posterior $P(M|X)$. Equations (17) and (18) also provide us with a clear way of properly including language model information $[P(M)]$ into the formalism (as part of other local prior information).

4.3 Forward-Backward Training

In the hybrid systems previously developed, a Viterbi training was used in which the summation over state sequences in (17) or (18) is replaced by a maximization over state sequences. However, we can now derive a forward-backward algorithm for hybrid HMM/ANN training without using the Viterbi approximation. This is an application of the Generalized EM algorithm, where the missing data is the state sequence (as usual in HMM estimation), the E-step is the estimation of ANN targets using a forward-backward recurrence and the M-step is the ANN training. This is a generalized EM algorithm since the M-step is not exact. As for standard HMM systems, it is shown in [14], that new forward and backward recurrences can be defined in which the ANN outputs are used to compute and maximize

$$\frac{P(X|M)}{P(X)} = \frac{P(M|X)}{P(M)} \tag{19}$$

also yielding global discrimination.

5 Multi-Stream HMM/ANN Systems

5.1 Motivations

Current automatic speech recognition systems treat any incoming signal as one entity. There are, however, several reasons why we might want to view the speech signal as a multi-stream input in which each stream is processed (up to some temporal level) more or less independently of the others. In this section, we briefly discuss the work which has been done towards multi-stream speech recognition with hybrid HMM/ANN systems. Hybrid HMM/ANN systems could provide a good framework for such problems, where discrimination and the possibility of using temporal context are important features.

In the case of short-term (frame-based) frequency analysis, even when only a single frequency component is corrupted (e.g., by a selective additive noise), the whole feature vector is corrupted, and typically the performance of the recognizer is severely impaired. The work of Fletcher and his colleagues (see the insightful review of his work in [1]) suggests that human decoding of the linguistic message is based on decisions within narrow frequency sub-bands that are processed quite independently of each other. Recombination of decisions from these sub-bands is done at some intermediate level and in such a way that the global error rate is equal to the product of error rates in the sub-bands. Whether or not this is an accurate statement for disparate bands in continuous speech (the relevant Fletcher experiments were done with nonsense syllables using highpass or lowpass filters only), we see some engineering reasons for considering some form of this sub-band approach:

1. The message may be impaired (e.g., by noise) only in some specific frequency bands. When recognition is based on several independent decisions from different frequency sub-bands, the decoding of linguistic message need not be severely impaired, as long as the remaining clean sub-bands supply sufficiently reliable information.
2. Some sub-bands may be inherently better for certain classes of speech sounds than others.
3. Transitions between more stationary segments of speech do not necessarily occur at the same time across the different frequency bands, which makes the piecewise stationary assumption more fragile. The sub-band approach may have the potential of relaxing the synchrony constraint inherent in current HMM systems.
4. Different recognition strategies might ultimately be applied in different sub-bands.

It may also be interesting to define the speech signal in terms of several information streams, each stream resulting from a particular way of analyzing the speech signal. For example, models aimed at capturing the syllable level temporal structure could then be used in parallel with classical phoneme-based models. Another potential application of this approach could be the dynamic merging of asynchronous temporal sequences (possibly with different frame rate), such as visual and acoustic inputs.

Although work has been done on multi-band speech recognition [8] as well as for ASR based on multiple time scales [10], only the multi-band results will be briefly described here.

5.2 Approach

In the following we briefly present the approach presently used to recombine several sources of information represented by different input streams. In this case, an observation sequence X (representing the utterance to be recognized) is assumed to be composed of K input streams X_k (possibly of different lengths and/or different frame rates). A hypothesized model M associated with X will then be built up by concatenating J sub-unit models M_j $(j = 1, \ldots, J)$ associated with the sub-unit level at which we want to perform the recombination of the input streams (e.g., syllables). To allow the processing of each of the input streams independently of each other up to the pre-defined sub-unit boundaries (determined automatically during decoding), each sub-unit model M_j is composed of parallel models M_j^k (possibly with different topologies) that are forced to recombine their respective segmental scores at some temporal anchor points. The resulting statistical model is illustrated in Figure 7.

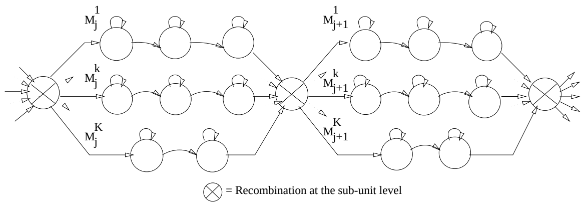

Fig. 7. *General form of a K-stream recognizer with anchor points between speech units (to force synchrony between the different streams). Note that the model topology is not necessarily the same for the different sub-systems.*

In this model we note that:

- The parallel HMMs, associated with each of the input streams, do not necessarily have the same topology.
- The recombination state ($\otimes$ in Fig. 7) is not a regular HMM state since it will be responsible for recombining (according to the possible rules discussed below) probabilities (or likelihoods) accumulated over a same temporal segment for all the streams. This should of course be done for all possible segmentation points. The problem appears to be similar to the continuous speech recognition problem where all of the concurrent word segmentations, as well as all of the phone segmentations, must be hypothesized. However, as recombination concerns sub-unit paths that must begin at the same time, and as the best state path in not the same for all of the sub-stream models, it is necessary to keep track of the dynamic programming paths for all of the

	FB	*No-W*	*Acc-W*	*SNR-W*	*MLP*
clean	3.6%	3.7%	3.7%	3.2%	2.7%
noisy	25.5%	9.2%	6.7%	6.3%	—

Table 1. *Error rates on isolated word recognition (108 German words, telephone speech) and noise was additive white noise in the 1st frequency band, 10dB SNR. Critical band energies were used as features. "FB" refers to regular full-band recognizer; "No-W" refers to sub-band recombination at state level without any weighting; "Acc-W"= state recombination with weights proportional to phonetic sub-band accuracy; "SNR-W"= state recombination with weights proportional to automatically estimated sub-band SNR. The column "MLP" refers to sub-band recombination at word level using an MLP.*

sub-unit starting points. Hence, an approach such as the two-level dynamic programming is required. Alternatively, a particular form of HMM decomposition [38], referred to as HMM recombination, can also be used [8]. Finally, multiple-pass approaches can be used in which lattices are generated by a simpler system and then rescored by one or more multi-stream recognizers.

5.3 Experiments

In one experiment [8], we used 3-state HMM/ANN phone models, 18 critical bands for the full-band system, and three sub-bands (spanning [0-1058], [941-2212], and [1994-4000] Hz) for the three sub-band HMM/ANN recognizers. Note that the overlap is only due to the critical band filter characteristics. Each band roughly encompasses one formant. The database consisted of 108 German isolated command words, telephone speech, with 15 speakers in the test set.

The features used for each recognizer were critical band energies complemented by their first temporal derivatives, and 9 frames of contextual information were used at the input of the ANNs. State level and word level recombinations were tested. In the case of word level merging, an MLP with 108 (words) $\times$ 3 (bands) input units and 108 output units was trained on normalized log-likelihoods from the clean training data.

Resulting error rates are reported in Table 1. Recognition performance of the different recombination strategies are compared with the full-band approach, in case of clean speech and noisy speech (additive white noise in the 1st sub-band, 10dB SNR). For clean speech we have been able to achieve results that were at least as good as the conventional full-band recognizer (though for this size test set the differences are not statistically significant at $p < .05$).

When one of the frequency bands is contaminated by selective noise, the multi-band recognizer yields much more graceful degradation than the broad-band recognizer. The best results have been achieved using weights derived from S/N estimates. However, we have observed that even without any knowledge about the S/N ratio in sub-bands [using equal weighting ("No-W") or sub-band accuracy weighting ("Acc-W")] the sub-band recognizer still yields much better results than the conventional full-band recognizer.

More recently, a similar approach was successfully used to merge acoustic streams with different time scale properties (e.g., respectively capturing phonetic and syllabic dynamics) [10]. In other experiments [37], it was also shown that a similar approach could also be used to better capture the possible asynchrony between frequency bands. These multi-stream results are also reminiscent of earlier experiments in which we showed that the combination of phone models with models trained to emphasize transitions significantly improved robustness to additive noise [5].

6 Other Connectionist Approaches

This paper has focused on the hybrid HMM/ANN approach, in which some kind of network (typically an MLP, RBF, RNN, or TDNN) trained for classification by an MSE or relative entropy criterion is used to estimate probabilities or distances to be used in dynamic programming matching to a HMM. This is currently the most common application of neural networks to continuous speech recognition. However, a range of other approaches and subproblems in speech recognition are being investigated by researchers. Some of the most common are:

- Predictive networks - In this case, ANNs are not trained to perform phonetic (HMM state) classification, but instead are trained (according to a MSE criterion) as an autoregressive (AR) model to predict a feature vector given some previous number of feature vectors and the assumption of a particular HMM state [36][20].
- ANN models of HMMs - Connectionist structures can also be used to represent standard HMM-based algorithms. Examples include the Viterbi network [21], which implements a Viterbi decoder, and the Alpha-Net [9], which simulates the forward recurrence of the forward-backward HMM algorithm.
- Global optimization through nonlinear transformation - Networks can provide a general nonlinear transformation of the observation vectors for an otherwise standard HMM-based system. This permits a global optimization of the input transformation together with a global training of the HMMs [4].
- Minimum classification error optimization - The Generalized Probabilistic Descent/Minimum Classification Error (GPD/MCE) training method [18] is a general framework for classifier optimization. It is based on the incorporation of a smooth classification error function into a gradient search optimization objective. The optimization objective is closely linked to the recognition error rate.
- Preprocessing - Many researchers have used feature map representations, related to one of the formulations from Kohonen and collaborators [19], to generate feature representations for a speech recognizer. In other designs, researchers have experimented with networks to provide mappings from noisy to clean data [34] or from a new speaker to an old speaker [17].

– Postprocessing - As noted earlier, many researchers have used lattice generation (or N-best utterance lists) as an intermediate step in order to test new processing methods without having to embed them in the main system. Neural networks have often been used in such systems. For instance, in the case of the Segmental Neural Network [39], networks are trained on phonetic segments as determined from Viterbi alignments in the training set, and then are run to generate probabilities for sequences of segments in each hypothesized utterance. The resulting scores are blended with the the scores from the primary system, and have been shown to improve overall performance.

Acknowledgements

The work described here has been strongly influenced (and often done) by our collaborators in Europe and the US. In particular, we should thank Steve Renals and Chris Ris for their recent work with us on the use of ANNs as statistical estimators for HMMs. More recently, we have been co-investigating the multiple stream work with Hynek Hermansky at OGI (Portland, OR) and with Stéphane Dupont at the Faculté Polytechnique de Mons (Belgium). Many other researchers at these labs, as well as at our own, continue to contribute to our understanding of these issues. Our joint work in these areas is currently partly funded by the European Union as part of the SPRACH (20077) and THISL (23495) Long Term Research grants.

References

1. Allen, J.B., "How do humans process and recognize speech?," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 4, pp.567-577, 1994.
2. Austin, S., Zavaliagkos, G., Makhoul, J., and Schwartz, J., "Improving state-of-the-art continuous speech recognition systems using the N-best paradigm with neural networks," *Proc. DARPA Speech and Natural Language Workshop* (Harriman, NY), Morgan Kaufmann, pp. 180-184, Feb. 1992.
3. Baum, L., "An inequality and associated maximization techniques in statistical estimation of probabilistic functions of Markov processes," *Inequalities*, no. 3, pp. 1-8, 1972.
4. Bengio, Y., De Mori, R., Flammia, G. and Kompe, R., "Global optimization of a neural network-Hidden Markov Model hybrid," *IEEE Trans, on Neural Networks*, vol. 3, no. 2, pp. 252-259, 1992.
5. Bilmes, J., Morgan, N., Wu, S., and Bourlard, H., "Stochastic perceptual speech models with durational dependence," *Intl. Conference on Spoken Language Processing,* pp. 1301-1304, 1996.
6. Bourlard, H. and Morgan, N., *Connectionist Speech Recognition – A Hybrid Approach*, Kluwer Academic Publishers, 1994.
7. Bourlard, H., Konig, Y. and Morgan, N., "REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities in connectionist speech recognition", *Proc. EUROSPEECH'95* (Madrid, Spain), Sep. 1995.

8. Bourlard, H. and Dupont, S. (1996), "A new ASR approach based on independent processing and recombination of partial frequency bands," *Proc. of Intl. Conf. on Spoken Language Processing (ICSLP)* (Philadelphia), pp. 426-429, Oct. 3-6, 1996.
9. Bridle, J.S., "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, Architectures and Applications*, F. Fogelman Soulié and J. Hérault (Eds.), NATO ASI Series, pp. 227-236, 1990.
10. Dupont, S. and Bourlard, H., "Using multiple time scales in a multi-stream speech recognition system," to be published in *Proc. EUROSPEECH'97* (Rhodes, Greece), Sep. 1997.
11. Furui, S., "Speaker independent isolated word recognizer using dynamic features of speech spectrum," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 52-59, 1986.
12. Gish, H., "A probabilistic approach to the understanding and training of neural network classifiers," in *IEEE Proc. Intl. Conf. on Acoustics, Speech and Signal Processing* (Albuquerque, NM), pp. 1361-1364, 1990.
13. Haeb-Umbach, R., Geller, D., Ney, H., "Improvements in connected digit recognition using linear discriminant analysis and mixture densities," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing* (Adelaide, Australia), pp. II-239-242, 1994.
14. Hennebert, J., Ris, C., Bourlard, H., and Renals, S., "Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems," to be published in *Proc. EUROSPEECH'97* (Rhodes, Greece), Sep. 1997.
15. Hermansky, H., "Perceptual Linear Predictive (PLP) analysis of speech," *Journal of the Acoust. Soc. Am.*, vol. 87, no. 4, 1990.
16. Hochberg, M.M., Renals, S.J., Robinson, A.J., and G.D. Cook., "Recent improvements to the ABBOT large vocabulary CSR system," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing* (Detroit, MI), pp. 69-72, 1995.
17. Huang, X.D., Lee, K.F. and Waibel, A., "Connectionist speaker normalization and its application to speech recognition," *Proc. of IEEE Workshop on Neural Networks for Signal Processing,* pp. 357-366, IEEE Press, 1991.
18. Katagiri, S., Lee, C., and Juang, B., "New Discriminative Training Algorithms Based on the Generalized Probabilistic Descent Method", *Proc. of the 1991 IEEE Workshop on Neural Networks for Signal Processing,* ppp. 299-308, 1991.
19. Kohonen, T., "The 'neural' phonetic typewriter," *IEEE Computer*: 11-22, 1988.
20. Levin, E., "Speech recognition using hidden control neural network architecture," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing* (Albuquerque, NM), pp. 433-436, 1990.
21. Lippmann, R.P., "Review of neural networks for speech recognition," *Neural Computation*, vol. 1, no. 1, pp. 1-38, 1989.
22. Lubensky, D.M., Asadi, A.O. and Naik, J.M., "Connected digit recognition using connectionist probability estimators and mixture-gaussian densities," *IEEE Proc. of the Intl. Conf. on Spoken Language Processing*, pp.295-298, Yokohama, Japan, 1994.
23. Morgan, N. and Bourlard, H., "Generalization and parameter estimation in feedforward nets: some experiments, " in *Advances in Neural Information Processing Systems 2* (D.S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 630-637, 1990.
24. Morgan, N., "Big Dumb Deural Nets (BDNN): a working brute force approach to speech recognition", Proceedings of the ICNN, vol. VII, pp.4462-4465, 1994.

25. Morgan, N. and Bourlard, H., "Neural networks for statistical recognition of continuous speech," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 741-770, 1995.
26. Ney, N., "The use of a one-stage dynamic programming algorithm for connected word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 32:263-271, 1984.
27. Poritz, A., "Linear predictive Hidden Markov Models and the speech signal," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing*, pp. 1291-1294, Paris, 1982.
28. Poritz, A.B. and Richter, A.L., "On hidden Markov models in isolated word recognition", *IEEE Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. 14.3.1-4, Tokyo, Japan, 1986.
29. Rabiner, L.R., "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, 1989.
30. Renals, S., Morgan, N., Bourlard, H., Cohen, M. and Franco, F., "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 1, pp. 161-174, 1994.
31. Renals, S. and Hochberg, M., "Efficient search using posterior phone probability estimates," Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (Detroit, MI), pp. 596-599, 1995.
32. Richard, M.D. and Lippmann, R.P., "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, no. 3, pp. 461-483, 1991.
33. Robinson, T., Almeida, L., Boite, J.M., Bourlard, H., Fallside, F., Hochberg, M., Kershaw, D., Kohn, P., Konig, Y., Morgan, N., Neto, J.P., Renals, S., Saerens, M. and Wooters, C., "A neural network based, speaker independent, large vocabulary, continuous speech recognition system: The WERNICKE Project," *Proc. EUROSPEECH'93* (Berlin, Germany), pp. 1941-1944, 1993.
34. Sorenson, H., "A cepstral noise reduction multi-layer network," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing* Toronto, Canada, pp. 933-936, 1991.
35. Steeneken, J.M. and Van Leeuwen, D.A., "Multi-lingual assessment of speaker independent large vocabulary speech-recognition systems: the SQALE project (speech recognition quality assessment for language engineering)," *Proc. EUROSPEECH'95* (Madrid, Spain), Sep. 1995.
36. Tebelskis, J. and Waibel, A., "Large vocabulary recognition using linked predictive neural networks," in *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing* (Albuquerque, NM), pp. 437-440, 1990.
37. Tomlinson, M.J., Russell, M.J., Moore, R.K., Buckland, A.P., Fawley, M.A., "Modelling asynchrony in speech using elementary single-signal decomposition," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing* (Munich, Germany), pp. 1247-1250, 1997.
38. Varga, A. and Moore, R., "Hidden Markov model decomposition of speech and noise," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. 845-848, 1990.
39. Zavaliagkos, G., Zhao, Y., Schwartz, R. and Makhoul, J., "A hybrid segmental neural net/hidden markov model system for continuous speech recognition" *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 1, pp. 151-160, 1994.