An Introduction to Hybrid HMM/Connectionist Continuous Speech Recognition

Nelson Morgan and Hervé Bourlard

International Computer Science Institute

Berkeley, CA 94704, USA

Abstract

Since 1988 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 to assist in statistical speech recognition. ANNs are sometimes called *connectionist* due to their representation of information in the connections between units computing simple functions of their inputs. Researchers at a number of laboratories have shown that these nets can be used to estimate probabilities that are useful in statistical pattern recognition, and in particular for speech recognition. Simple systems based on this hybrid approach have performed very well on large vocabulary continuous speech recognition. Research is continuing on extending these results to more complex systems. In this tutorial paper, we will briefly introduce the theoretical and practical underpinnings of this approach.

1 Introduction: Advantages and Drawbacks of the Dominant Paradigm

If speech recognition by machine were a solved problem, then work in the area would consist of the application of proven techniques to new tasks, and basic research would not be necessary. By some metrics, this is almost the current situation. Statistical representations called *Hidden Markov Models* (HMMs) have been developed to the point where impressive recognition performance can be achieved in the laboratory on very large vocabulary speaker-independent continuous speech recognition tasks. Related methods have been transferred to commercial applications on a limited scale, and the current prospect is that many such useful deployments can be done with the existing technology. Further, this technology is now reasonably mature, having been built up over the last twenty years, based on the pioneering work of Baker [3] and Jelinek [36]. Additionally, many of the engineering issues were addressed by speech researchers in the late 1980's and early 1990's under the auspices of the ARPA program [16, 18, 42]. It is now feasible to develop new applications by following mathematical prescriptions that have been developed under these previous programs. Many people have been able to use HMM-based approaches as the core of impressive recognition systems. Recently, one such system, the HMM Tool Kit (HTK) from Cambridge [84] has been widely distributed so that students and developers can build up speech recognition systems based on some of the best known statistical approaches.

However, speech recognition is *not* a solved problem in any fundamental sense. While the existing technology may be sufficient for some specific commercial applications, human beings

Pros	Cons
Rich mathematical framework	Poor discrimination
Powerful learning and decoding methods	Practical requirements for distributional assumptions (e.g., uncorrelated features within an acoustic vector)
Good abstractions for sequences, temporal aspects	1st order Markov model assumption for phone or subphone states
Flexible topology for statistical phonology and syntax	Typically ignore correlation between acoustic vectors

Table 1: Advantages and disadvantages of classical HMM-based approaches to continuous speech recognition.

expect speech input systems to behave much as people would. Human beings must often contend with unfamiliar accents, background noise and reflective room acoustics, improper English grammar, and unfamiliar words. Nonetheless, we can generally understand what is being said. Human recognition performance under many realistic conditions is still much better than that of any machine, even for the limited tasks we have undertaken so far.

Since there is great room for improvement, research continues in this area. However, it may be the case that some of the characteristics of mainstream approaches to speech recognition are limiting factors in the long term road to improvement. For this reason, it is important to examine alternate approaches, such as the neural-network-based method described in this article. This seems particularly justified if there are some theoretical reasons why the current approaches are suboptimal. However, given the strong base of mathematical tools for statistical speech recognition, it has been useful to modify only a few aspects of the existing approaches at a time, rather than starting from scratch.

Table 1 shows some of the pros and cons of the dominant speech recognition paradigm, in which HMMs are used to represent the production of speech and are compared with new speech to determine the most likely utterance that could have been said.

HMM-based structures and algorithms provide a rich and flexible mathematical framework for building recognition systems. They also feature powerful learning and decoding methods for temporal sequences without requiring any explicit (hand) segmentation in terms of the speech units (typically phones or phonemes) used as a basis for continuous speech training and recognition. Also, they easily accommodate different levels of constraints (e.g., phonological and syntactical constraints), as long as these are expressed in terms of the same statistical formalism. However, to take advantage of this representational power, algorithms must explicitly or implicitly make numerous assumptions about speech, although some of them are obviously unrealistic. For instance, it is often necessary to assume that the features extracted within a phonetic segment are uncorrelated with one another. This is a poor match to most kinds of speech segments and features. Also, HMM training algorithms are based on likelihood maximization, which assumes correctness of the models (which is known not to be true; see, e.g., the correlation problem mentioned above) and implies poor discrimination (since, ideally, training for minimization of the error rate should be based on a posteriori probabilities). In a sense, we have settled for a poor model because we know how to use it more effectively than we know how to use more realistic ones.

Connectionist or *Artificial Neural Network* (ANN) methods provide one way to reduce system dependence on unrealistic assumptions about speech. Any system can be improved in a variety of ways, and so there is no conclusive evidence that connectionist speech recognition algorithms are better than other approaches. Nonetheless, in a number of controlled experiments, the use of connectionist elements in a larger statistical system has significantly improved performance [2, 47, 65]. Further, connectionist speech research can provide the opportunity to develop a new paradigm that also has a firm theoretical footing, and which may lead to newer innovations that will improve the state of the art. Currently, though, connectionist approaches for continuous speech recognition are embedded within a framework that is essentially a standard HMM, in which only a few of the usual assumptions have been dropped. As research continues, hopefully the models can represent speech more faithfully. However, most laboratories that have published reports of work in continuous speech recognition using ANNs are using them to generate probabilities (or distances) that are used with an underlying model that is explicitly or implicitly an HMM [2, 7, 19, 24, 33, 45, 47].[1] For more on this class of hybrid systems, see [9].

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. 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 [71] is computed over successive intervals, e.g., 100 times per second. These representations or speech *frames* are then compared (in some sense) 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

[1] Unavoidably, we will tend to give a description that is close to our own implementation (for instance, to use a multi-layered perceptron, as described in [56]). The reader should keep in mind that most of the basic ideas should still hold for other hybrid implementations, e.g., using a recurrent net for the probability estimates as done in [33].

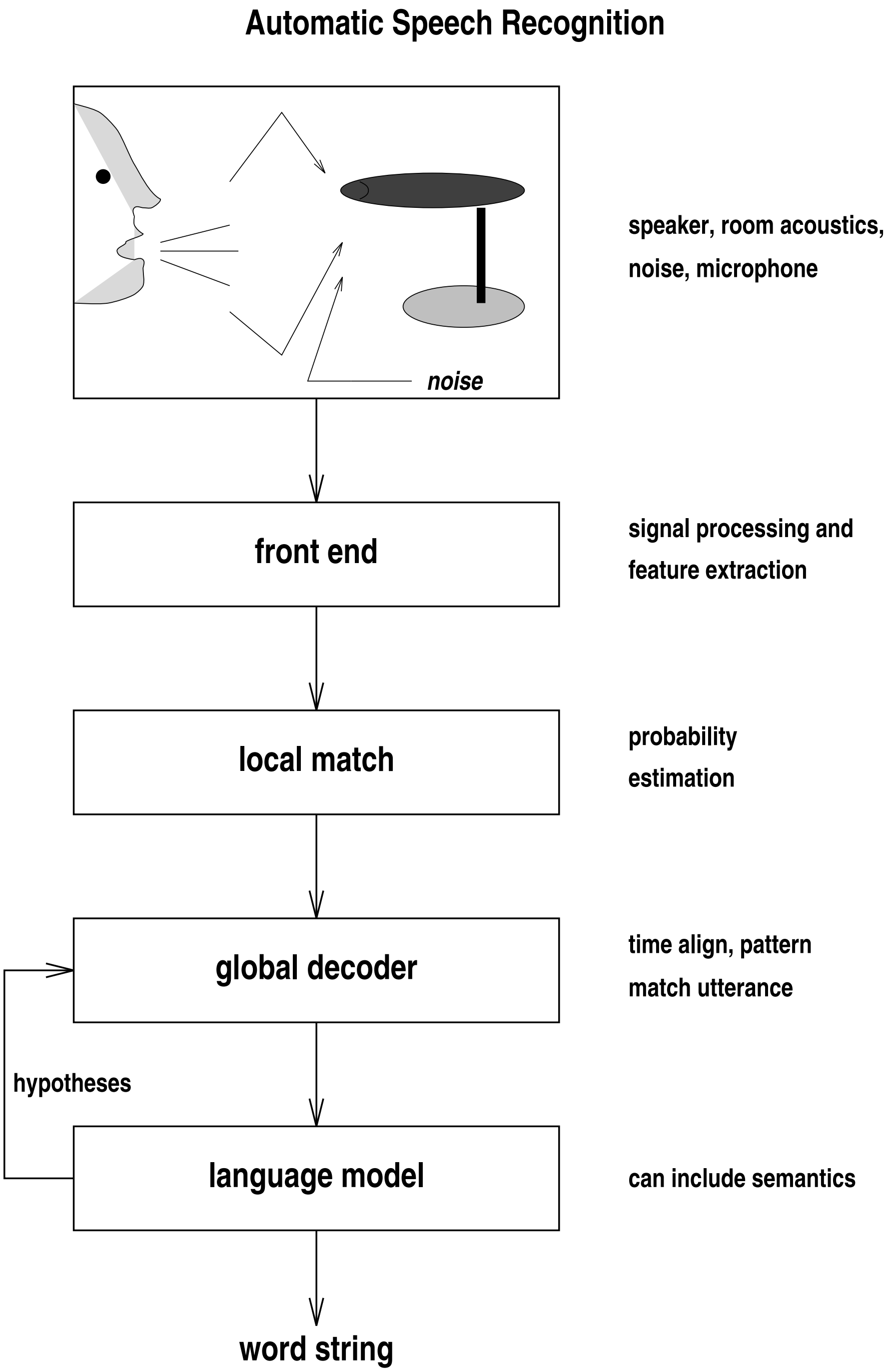

Figure 1: Block diagram of continuous speech recognition.

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 can be 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 [57], 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[2], so that the sums can be interpreted as giving the most probable sequences. These sums then can be viewed as computing products of probabilities in the log domain, which 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. Nonetheless, the dynamic programming (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

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

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)

HMMs model the sequence of feature vectors as a piecewise stationary process. That is, an utterance $X = \{x_1, \ldots, x_n, \ldots, x_N\}$ is modeled as a succession of discrete stationary states $Q = \{q_1, \ldots, q_k, \ldots, q_K\}$, $K < N$, with instantaneous transitions between these states. A HMM is typically defined (and represented) as a stochastic finite state automaton (usually with a left-to-right topology when used for speech). 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. 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.

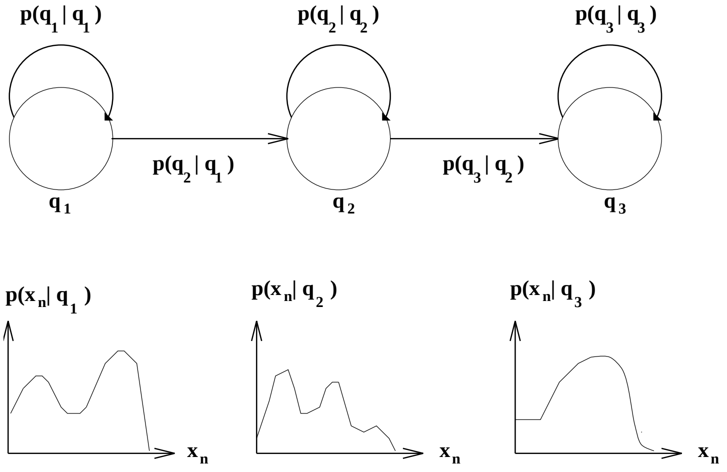

Figure 2: A three-state Hidden Markov Model (HMM). A HMM is a stochastic finite state machine, consisting of a set of states and corresponding transitions between states. HMMs are commonly specified by a set of states q_i, an emission probability density $p(x_n|q_i)$ associated with each state, and transition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

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 [63]. Briefly, the fundamental equation relevant for this process is a restatement of Bayes' Rule as applied to speech recognition:

$$P(M|X) \;=\; \frac{P(X|M)P(M)}{P(X)} \tag{1}$$

in which $P(M|X)$ 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,[3] equation (1) is usually used to split this posterior probability into a likelihood $P(X|M)$ that represents the contribution of the *acoustic model*, and a prior probability $P(M)$ that represents the contribution of the *language model.* Usually $P(X|M)$ and $P(M)$ are trained separately on independent training corpora. By doing so, and assuming that $P(X)$ in (1) is independent of the parameters of the models (which is actually not true during training), (1) turns the acoustic model training into a Maximum Likelihood Estimation (MLE) problem.[4]

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

$$P(X|M) \;=\; \sum_{\forall Q} P(X,Q|M) \tag{2}$$

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

$$P^*(X|M) \;=\; \max_{\forall Q} P(X,Q|M) \tag{3}$$

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

Such a statistical system is trained on acoustic data so that during recognition it produces emission probabilities $p(x_n|q_k)$ (see Figure 2) that can be multiplied to produce an approximation to the acoustic probability $P(X|Q)$ (assuming statistical independence). There exist efficient training algorithms to learn the parameters of the probability estimators. The most common form of this procedure is often called the forward-backward algorithm, in which the estimators for the data likelihoods conditioned on each word model ($P(X|M)$) are iteratively trained [4]. For the Viterbi approximation, the full likelihood is approximated by the likelihood of the most probable path through the states in the models, as given by the dynamic programming procedure. 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:

[3] However, see [6] for some theoretical work that suggests an approach to training and recognition with a global $P(M|X)$ criterion.

[4] This actually explains the lack of discrimination mentioned in Table 1, as systems trained to maximize $P(X|M)$ for the correct model are not trained to minimize that quantity for the incorrect models.

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 dynamic programming 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 [63], for instance), can be proved to converge to a local optimum; in practice it is repeated until some stopping criterion has been reached.

2.3 Artificial Neural Networks (ANNs)

2.3.1 Multilayer Perceptrons (MLPs)

Our discussion of neural networks for speech will be focused on *Multilayer Perceptrons* (MLPs), which are the most common ANN architecture used for speech recognition. However, all of the basic conclusions about the utility of these structures for estimating probabilities or local costs for an HMM will also hold for other structures such as a recurrent network, as used in [33], or a Time-Delay-Neural-Network (TDNN), as used in [24]. These alternative structures will be briefly discussed in Section 2.3.2.

Typically, MLPs have a layered feedforward architecture with an input layer (consisting of the input variables), zero or more *hidden* (intermediate) layers, and an output layer, as shown in Figure 3. Each layer computes a set of linear discriminant functions [22] (via a weight matrix) followed by a nonlinear function, which is often a sigmoid function

$$f(x) = \frac{1}{1+\exp(-x)} \tag{4}$$

As discussed in [9], this nonlinear function performs a different role for the hidden and the output units. On the hidden units, it serves to generate high order moments of the input; this can be done effectively by many nonlinear functions, not only by sigmoids. On the output units, the nonlinearity can be viewed as a differentiable approximation to the decision threshold of a threshold logic unit or perceptron [69], i.e., essentially to count errors. For this purpose, the output nonlinearity should be a sigmoid or sigmoid-like function. Alternatively, a function called the *softmax* can be used, as it approximates a statistical sigmoid function. For an output layer of K units, this function would be defined as

$$f(x_i) \;=\; \frac{\exp(x_i)}{\sum_{n=1}^{K}\exp(x_n)} \tag{5}$$

MLPs with enough hidden units can (in principle) provide arbitrary mappings $g(x)$ between input and output. MLP parameters (the elements of the weight matrices) are trained to associate a “desired” output vector with an input vector. This is achieved via the *Error Back-Propagation* (EBP) algorithm (see [58], [59], [70] and [81] for multilayer networks; [1], [69] and [83] for single layer networks; [15] for a control theory version) that uses a steepest descent procedure to iteratively minimize a cost function.

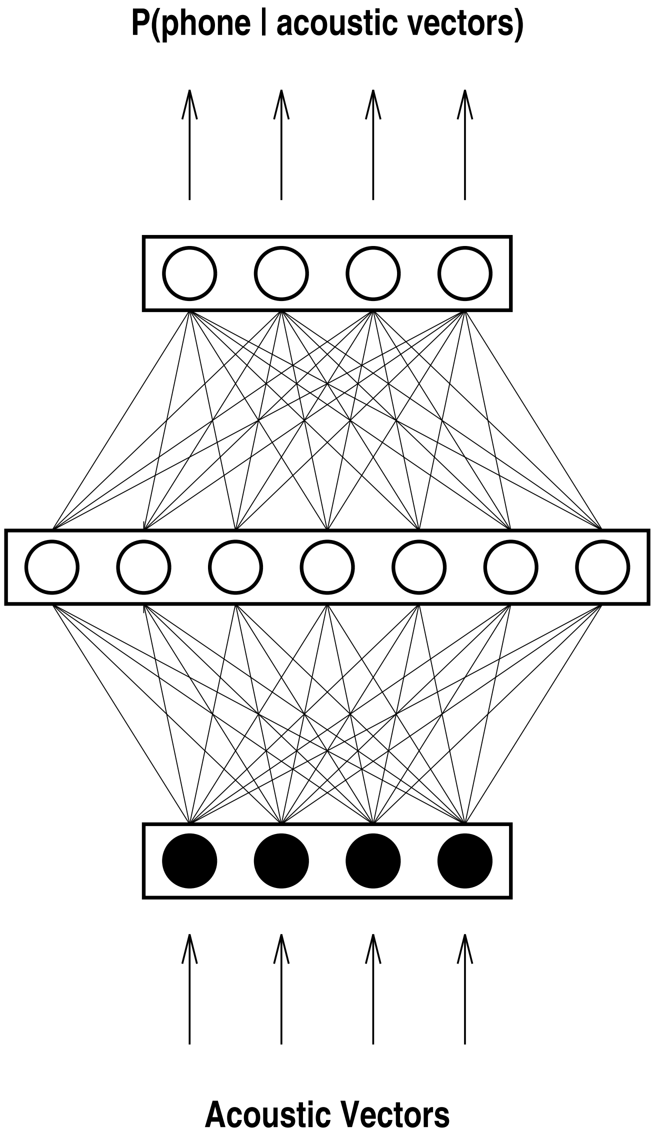

Figure 3: A generic multi-layer perceptron (MLP). Each horizontal box is a layer of the MLP, and consists of a series of units represented by circles. Each unit represented by an open circle computes a simple function of its inputs, such as a sigmoidal nonlinearity of the weighted sum of its inputs. The darker circles represent the inputs to the network. The intermediate layer is referred to as a hidden layer. The output layer computes phonetic probabilities if the net has been trained under certain conditions, as described in the text.

Popular cost functions are, among others, the *Mean Square Error* (MSE) criterion:

$$E = \sum_{n=1}^{N} \parallel g(x_n) - d(x_n) \parallel^2 \tag{6}$$

or the relative entropy criterion[5] (also referred to as Kullback-Leibler distance):

$$E_e = \sum_{n=1}^{N} \sum_{k=1}^{K} d_k(x_n) \ln \frac{d_k(x_n)}{g_k(x_n)} \tag{7}$$

where x_n is the pattern to be classified, $d(x_n) = (d_1(x_n), \ldots, d_k(x_n), \ldots, d_K(x_n))^t$ represents the desired output vector (for classes q_k, $k = 1, \ldots, K$), $g(x_n) = (g_1(x_n), \ldots, g_k(x_n), \ldots, g_K(x_n))^t$ the observed output vector, K the total number of classes, and N the total number of training patterns.

MLPs, as well as other neurally-inspired architectures, have been used in a variety of ways for speech-related tasks. For instance, for some tasks the entire input sequence (e.g., isolated word) is processed at once by the MLP, which views the temporal acoustical sequence as a spatial pattern. For the case of isolated words, for instance, each word can be associated with an output of the network. However, this kind of structure has not been useful for continuous speech recognition and will not be discussed further here.

2.3.2 Other Connectionist Architectures

As we shall see in the next sections, the hybrid approach is based on a statistical perspective that is valid for any connectionist architecture, given some assumptions about the training procedure. Consequently we are not giving a significant treatment here of the possible neural networks that could be used for this purpose (see [51] or [32] for more about the possible architectures). However, we briefly describe here some of the other kinds of network that can be used in an ANN/HMM hybrid besides the MLP described above.

- Radial Basis Functions (RBFs)- many researchers use a variant of a layered feedforward network in which a hidden layer with a fixed-variance Gaussian is used instead of one with sigmoids [13]. Further, these networks commonly use a single Gaussian layer and an output layer that is strictly linear. Once reasonable positions are found for the hidden Gaussians, the optimal output layer weights can be determined analytically with a matrix inversion. These systems can be trained with much less computation than sigmoid or softmax-based MLPs. The outputs can also be used as probabilities and integrated into a hybrid system [64, 76]. Many researchers have reported, however, that the final performance is somewhat poorer than with the far more computationally expensive MLP.

- Recurrent Neural Network - the group at Cambridge University Engineering Dept (CUED) has developed an approach that is very similar to the one described here, except that the emission probability is generated by a recurrent neural network (RNN) [67]. The network uses a set of state units that takes the acoustic input and has a recurrent connection from the output of the state layer back to its input. The state layer output along with the acoustic input is connected to the output layer. The network is trained using back

[5] This criterion can be derived from viewing the network outputs as a posterior distribution over the values of a random variable that is the pattern class (conditioned on the acoustic data), and evaluating a discrete form of the classical definition of relative entropy between the target distribution and this output distribution.

propagation through time. Recent experiments show this method to work roughly as well as the MLP approach, but with even fewer parameters. The principal drawback appears to be a somewhat tricky training schedule, due to potential instabilities.

- Time-Delay-Neural Network - recurrent networks can be approximated over a finite time period by a feedforward network in which the loops are replaced by the explicit use of several preceding activation values. This feedforward approach was shown to be useful with a single convolutional layer for consonant recognition in 1983 by Makino [48], and later by Waibel et al for a similar task using multiple convolutional layers [79] and finally for hybrid systems such as those described in this article [24]. It is a flexible structure, and in practice many researchers (including us) have used a compromise between the simple MLP and a full TDNN in which delays are only used at the input layer.

Note that in the case of the TDNN and RNN, the hidden and output units can be of the same form as for an MLP (sigmoidal or softmax), and that in all cases the error criteria can be MSE or relative entropy (among others).

3 ANNs as Statistical Estimators

3.1 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 (see, e.g., [60, 80, 79]). 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 an ANN only. 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. 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 [9], [10], [27], [66]. Before we repeat this proof, the general principle can be geometrically motivated (see Figure 4).

The figure shows that an equilibrium that is reached in the ideal case does in fact correspond to the network output g being equal to the posterior probability p. Consider an area in feature

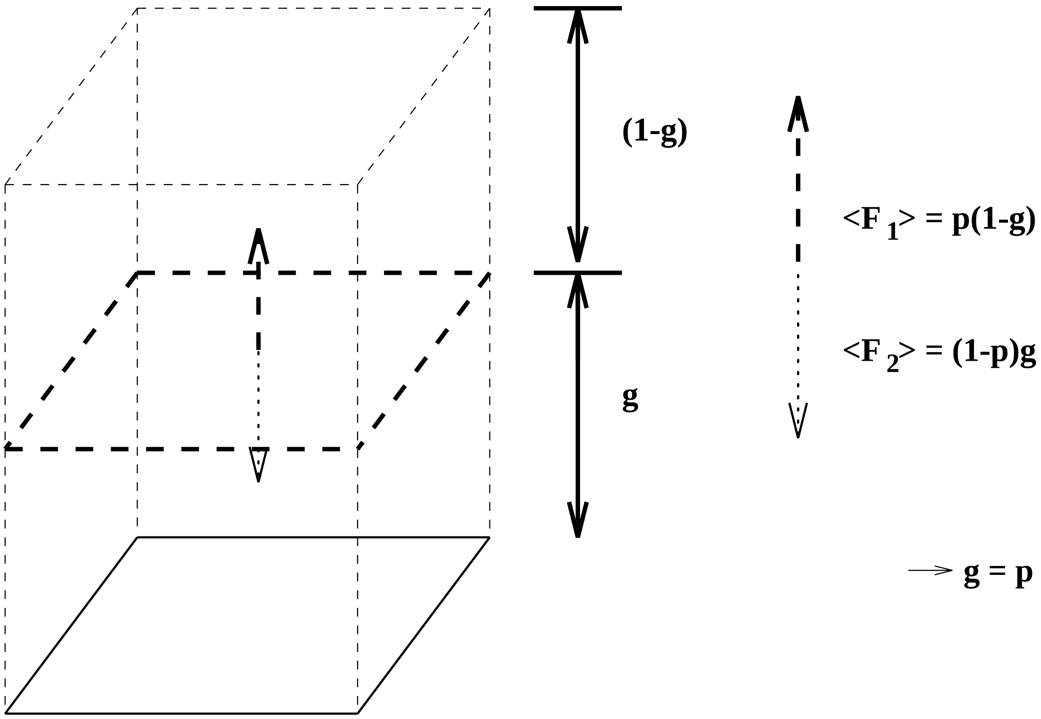

Figure 4: Geometric illustration of proof that ANN will (ideally) produce posterior probabilities. Equilibrium is reached when the upward force, (due to the average deviation of the ANN output from 1 for a target that should be high), is equal to the downward force (due to the average deviation from 0 for a target that should be low).

space around a training pattern x_n, and assume that we consider only two classes: the target class for this pattern, and the class of all patterns that do not belong to the first class. Vectors in the selected area will undergo an upward "force" corresponding to the gradient of the error term for all patterns with a target of 1; the quadratic error term in this case is $\frac{1}{2}(1-g)^2$, with a derivative of $(1-g)$, and this case will occur a fraction of the time given by probability p. Therefore, the upward force (average gradient term) applied to the region is equal to $p(1-g)$. The downward force is similarly defined, and will balance for the equilibrated case. This will only occur when the network output is equal to the posterior probability[6].

A more formal proof, originally given in [66], is repeated here. For continuous-valued acoustic input vectors, the MSE in (6) can be expressed as follows:

$$E = \int p(x) \sum_{k=1}^{K} \sum_{\ell=1}^{K} p(q_k|x) \left[g_\ell(x) - d_\ell(x)\right]^2 \, dx \qquad (8)$$

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:

$$\begin{aligned} E &= \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(g_\ell^2(x) - 2 g_\ell(x) p(q_\ell|x) + p^2(q_\ell|x) \right) \right] p(q_i, x) \, dx \\ &\quad + \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(p(q_\ell|x) - p^2(q_\ell|x) \right) \right] p(q_i, x) \, dx \\ &= \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(g_\ell(x) - p(q_\ell|x) \right)^2 \right] p(q_i, x) \, dx \qquad (9) \\ &\quad + \int \sum_{i=1}^{K} \left[\sum_{\ell=1}^{K} \left(p(q_\ell|x)(1 - p(q_\ell|x)) \right) \right] p(q_i, x) \, dx \end{aligned}$$

Since the second term in this final expression (9) 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 mean squared-error between the network output $g_\ell(x)$ and the posterior probability $p(q_\ell|x)$. Minimization of (8) is thus equivalent to minimization of the first term of (9), i.e., estimation of $p(q_\ell|x)$ 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 [66] 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 discussed in Section 2.3.2, 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

[6] Thanks to Raul Rojas of Freie Universitaet, Berlin for the figure and for this perspective on the proof.

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 MLP do in fact approximate posterior probabilities (see Figure 6.1 in [9]), even for error values that are not precisely the global minimum.[7]

Thus, emission probabilities can be estimated by applying Bayes' rule to the ANN outputs. In practical systems, we actually compute

$$\frac{p(x_n|q_k)}{p(x_n)} = \frac{p(q_k|x_n)}{p(q_k)} \tag{10}$$

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 left 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.

Figure 5 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 dynamic programming either for forced alignment (when the word sequence is assumed) or for recognition (when word sequences are hypothesized).

3.2 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 at least two potential advantages that we and others have observed:

1. As noted in the introduction, standard statistical recognizers require strong assumptions about the statistical character of the input, such as parameterizing the input densities as mixtures of Gaussian densities with no correlation between features, or as the product of discrete densities for different features that are assumed to be statistically independent. This type of assumption is 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. Specifically, standard HMM approaches require the assumption that successive acoustic vectors are uncorrelated. For the ANN estimator, multiple inputs can be used from a range of time steps, and the network will learn something about the correlation between the acoustic inputs. Note that the use of such a network will lead to a more general emission probability that may also be used in the global decoding. That is, if $c + d + 1$ frames of acoustic vectors $X_{n-c}^{n+d} = \{x_{n-c}, \ldots, x_n, \ldots, x_{n+d}\}$ are used as input to provide contextual information to the network, the output values of the ANN will estimate $p(q_k|X_{n-c}^{n+d}), \forall k = 1, \ldots, K$. This provides a simple mechanism for incorporating acoustic context into the statistical formulation.[8]

[7] Recently, Lippmann [46] has noted that "local minima usually represent alternative solutions which change decision regions and posterior probabilities only far from the training data. They thus have little effect on performance when there is sufficient training data."

[8] Of course, ANNs are not the only way to incorporate such context. Many current systems use first and second time derivatives [25, 62] computed over a span of a few frames, allowing very limited acoustical context modelling. 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., [28]). The neural network can be seen as a generalization of these approaches that permits arbitrary weights and a nonlinear transformation of the input data.

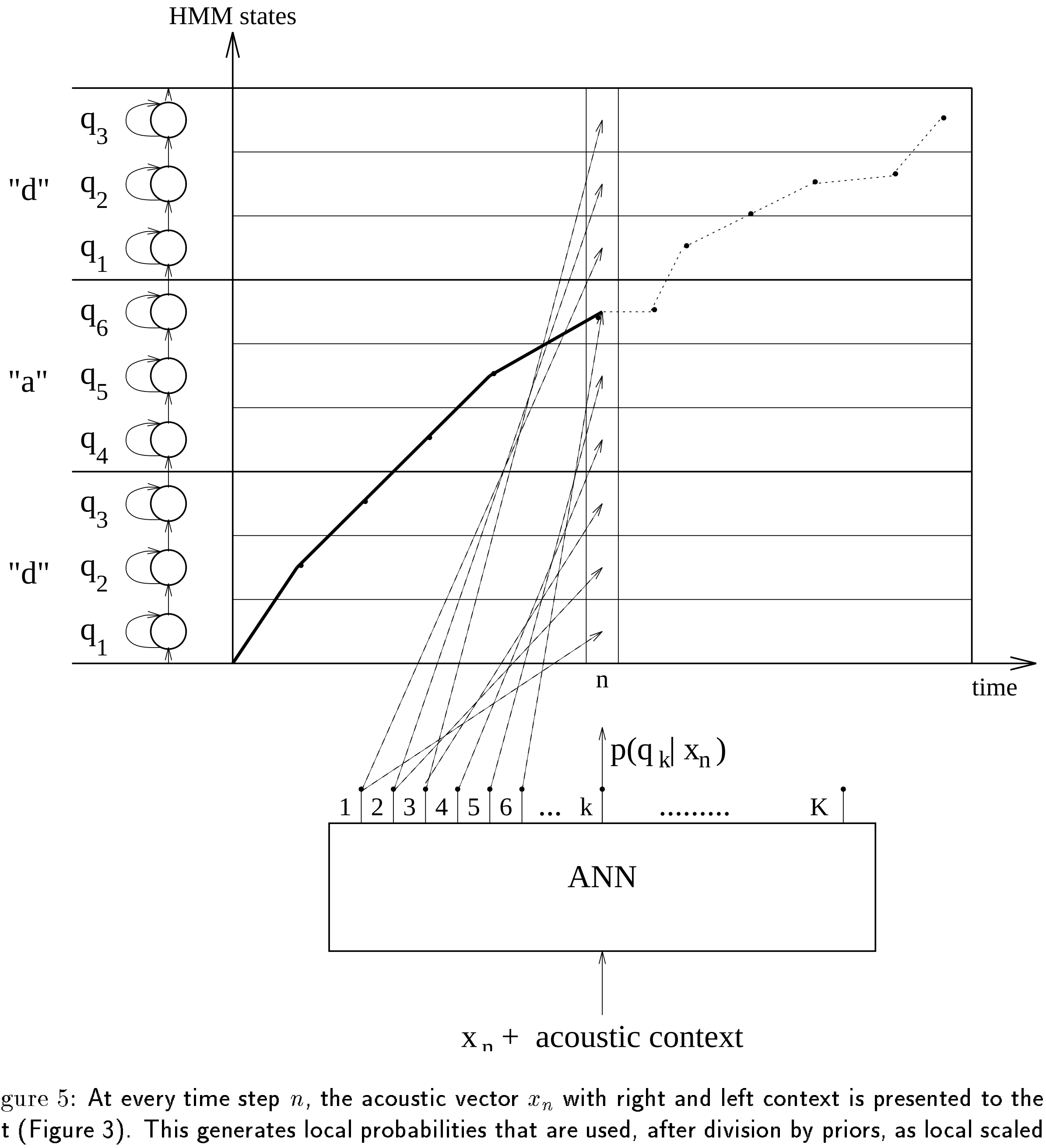

Figure 5: At every time step n, the acoustic vector x_n with right and left context is presented to the net (Figure 3). This generates local probabilities that are used, after division by priors, as local scaled likelihoods in a Viterbi dynamic programming algorithm. In the figure above, the arrows coming 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.

2. MLPs are a good match to discriminative objective functions (e.g., mean squared error), and so the probabilities will be optimized to maximize discrimination between sound classes, rather than to most closely match the distributions within each class. It can be argued that such a training is conservative of parameters, since the parameters of the estimate are trained to split the space between classes, rather than to represent the volumes that compose the division of space constituting each class. Some of the results cited in Section 5 appear to support the notion that fewer parameters are required for these systems than in non-discriminative systems with similar performance (see, in particular, [47]).

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 [7, 56, 9, 65, 67, 47]. 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 5.

4 Hybrid HMM/MLP Recognition System

4.1 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 1000 and 5000 word vocabularies, given millions of examples of feature vectors for training. At our laboratory, we have focused on using a simple MLP that is illustrated in Figure 6, though similar results have been achieved at other labs with structures such as RNNs. 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) [30] and log energy, along with their derivatives, or 26 features per frame).[9] 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 [9].

4.2 Training

We and others have used on-line training instead of off-line (true gradient) backpropagation. 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 very small steps closely correspond to the true gradient. In fact, it can be beneficial to have more noise

[9] Many experiments have been done in our lab that resulted in this choice of input features. Some of these are reported in [56].

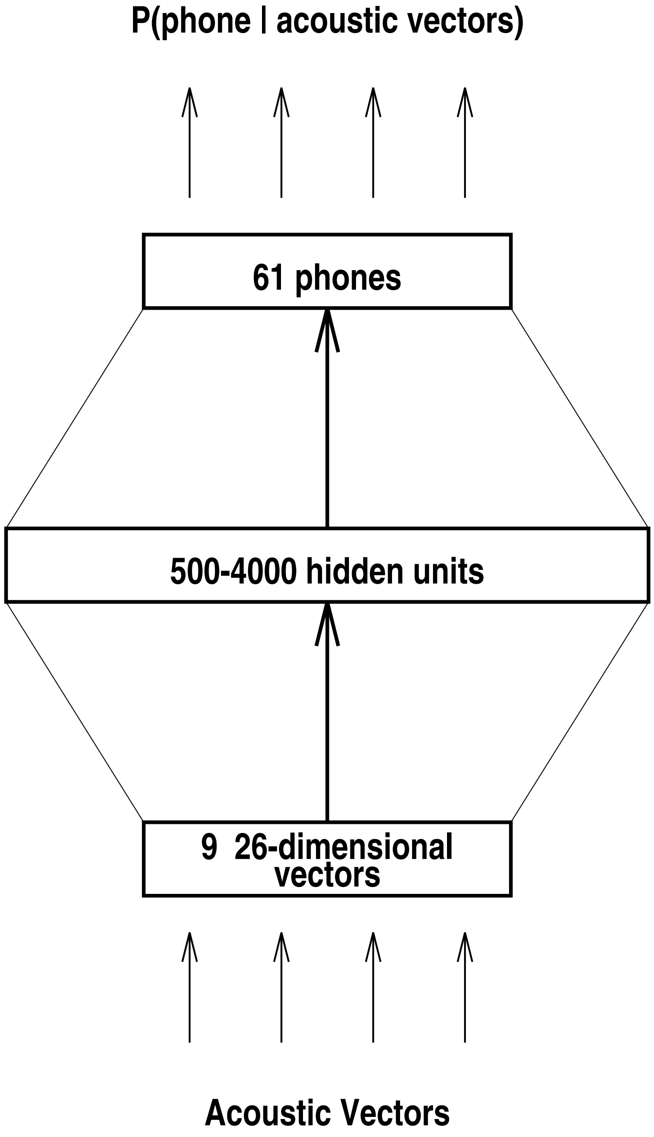

Figure 6: Acoustic vectors from the current frame, 4 previous, and 4 following frames are processed by a single large hidden layer. The output corresponds to phonetic categories used for labeling of the TIMIT database.

(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.

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).

In addition to the use of on-line 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 [54]. 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 (7) 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.

4.3 Division by Priors

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 equation 10). However, it can also be shown that, in theory, HMMs can be trained using local posterior probabilities as emission probabilities [9], resulting in models that are both locally and globally discriminant. (See Section 6.3 for a brief description of some current work in this area, which is described more fully in [6]). 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 7 shows the HMM for a pronunciation of "the cat." 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. We have recently conducted some experiments with the inference of pronunciation from data (see Section 5.1), and the preliminary observation is that division by class priors is not required in this case.

4.4 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-

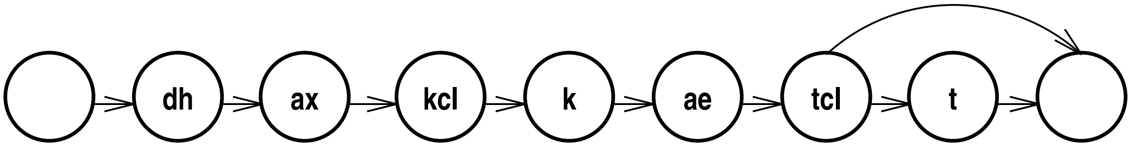

Figure 7: Simplified pronunciation model for the phrase "the cat" using Darpa-bet symbols for the phones. The "dh" symbol refers to the voiced form of "th", that is, one in which the vocal cords are vibrating. The initial consonant of the second word is represented by two states, one corresponding to 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.

labeled frames for ANN training. Since the ANN outputs can be used in the dynamic programming 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 8, 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.

4.5 Incorporating Constraints

The statistical interpretation of ANNs permits the flexible incorporation of model constraints. For instance, gender or phonetic context can be viewed as an additional random variable whose probability is estimated jointly with that of the phonetic unit. This joint probability can in turn be estimated, without simplifying assumptions, as the product of two simpler probabilities that are each estimated by ANNs.

For a constrained model, we may wish to estimate the joint probability of a current HMM state with a constraint class (such as phonetic context). Using $\mathcal{C}$ to represent the set of possible constraints, we wish to estimate $p(q_k, c_j|x_n)$, where $c_j \in \mathcal{C} = \{c_1, \ldots, c_J\}$. If there are J constraint classes, this will require $K \times J$ output units for an ANN estimator. However, using the definition of conditional probability, the desired expression can be broken down in two ways as follows [8, 19]:

$$p(q_k, c_j|x_n) = p(q_k|x_n)p(c_j|q_k, x_n) = p(c_j|x_n)p(q_k|c_j, x_n) \tag{11}$$

Thus, the desired probability is the product of an unconstrained posterior probability and a new conditional. The former can be realized with a network like that of figure 6 (though for the second case the output targets are context classes). Viewing an ANN as an estimator of the probability of the left side of a conditional given the right side as input, the $K \times J$ output probabilities can be estimated by two MLPs with K and J output units respectively.

This approach can be applied to contextual constraints to estimate biphone probabilities with two networks, or triphone probabilities with three. In each case, the use of extra conditionals removes any assumption of independence.

The conditionals can be implemented in a number of ways. For instance, a net can be constructed with sparsely coded binary inputs representing the classes (where all inputs are zero except for the class that we are conditioning on). These inputs can either be connected to a hidden layer, or, for greater computational simplicity, directly to an output layer. This latter architecture, as applied to phonetic context, is illustrated in Figure 9.

In collaborative work with SRI International [19] this class of approach was tested for left and right generalized biphone context. In this work, an architecture was used in which the output layer was replicated for each of 8 possible broad phonetic contexts (8 for the left and 8 for the right, each associated with one state of a 3-state model). This worked somewhat better than the architecture of Figure 9, but at the cost of a larger number of parameters. On the other hand, most of them were not modified for any given training frame, so that the computation was not significantly larger. In both cases, the weights to the context-dependent output layers were initialized to values learned from context-independent training, so that cross-validation (early

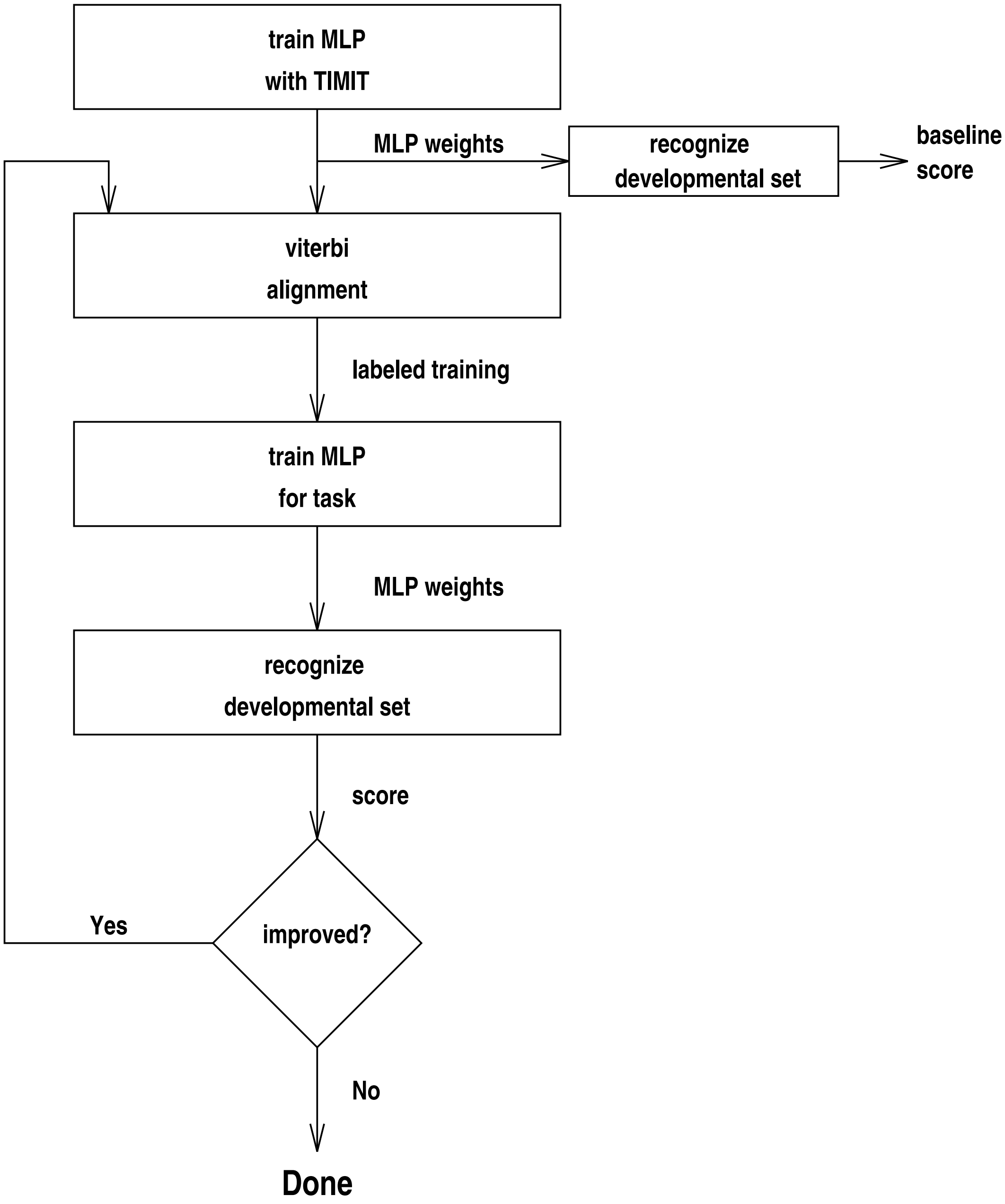

Figure 8: Embedded Viterbi learning with MLP.

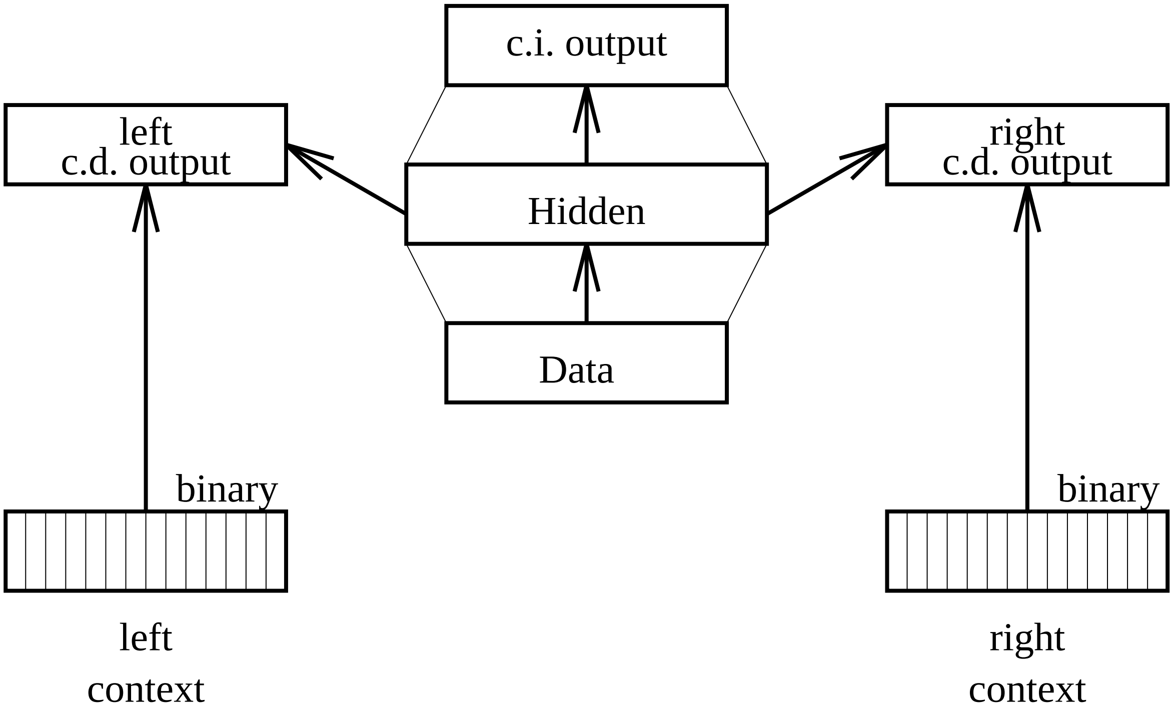

Figure 9: Context dependent neural network. Each output layer corresponds to a state of a 3-state HMM. The central output layer is trained in a context-independent manner. The left context-dependent output layer is only trained for frames that are labeled as the first state, and receives an input from a binary layer that codes the broad category of left context, i.e., only one input is on for any particular context, as indicated by the transcription during training or by the current hypothesis during recognition.. The right context-dependent output layer is trained similarly.

stopping) determined a smooth compromise between fully relying on the context-dependent data (which is sparser for each class) and the context-independent training [19]. These experiments showed significant improvements over the simpler context-independent approach, eliminating roughly one-fifth of the errors in a speaker-independent Resource Management test set [23]. The resulting system was comparable in accuracy to an HMM system that used many more parameters and which modeled much more detailed context, including word-specific models for the most common words [19]. Researchers at the Oregon Graduate Institute have also recently reported very large improvements for one of their tasks, using an architecture and method similar to that used by SRI [38].

Thus, statistical factorization of ANN probabilistic estimators appears to have practical significance. The approach has also been applied in other ways, for instance for gender and combinations of speaker-specific models [41]. See Figure 10 for an illustration of a network trained with gender dependency. This is also an example of the incorporation of long-term consistency constraints, as the input in this case is constrained to either be male or female for the entire utterance.

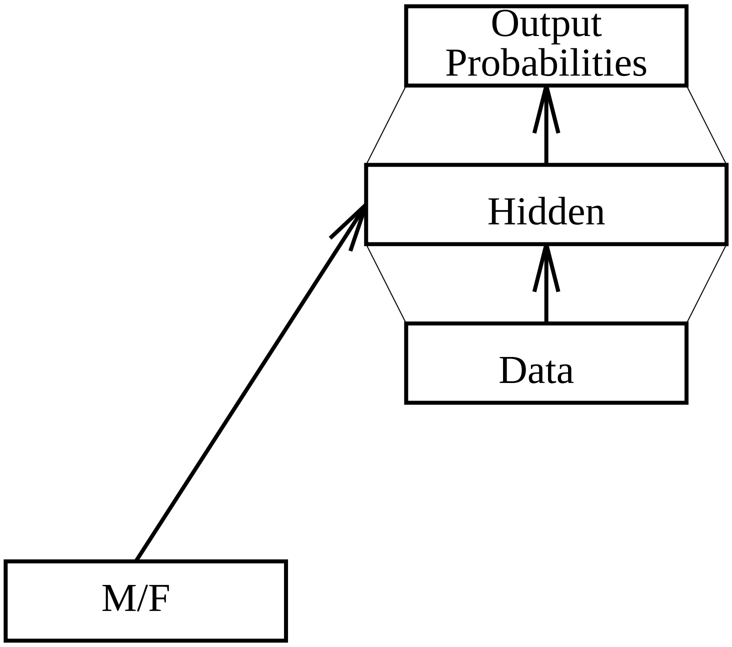

Figure 10: A gender-dependent MLP. Gender is represented by 1 or 2 bits that are additional variables presented to the network. For instance, for the 1-bit case, the binary input will be turned on for one gender and off for the other. During recognition, one gender or the other is assumed throughout an utterance, and the gender corresponding to the best overall score is chosen. Alternatively, nets can be trained separately for each gender. A gender classification net is used to weight the two choices, analogous to context-dependent networks.

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:

1. 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 [52], 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.

2. 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 [65] 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 [68].

   In [47], 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.

3. The best current pure-HMM systems currently outperform the best current hybrid systems on many tasks. 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 [47], 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.

4. 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 [65], 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.

5. In work at BBN [2], 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.

6 Current Related Research Areas

The following are a few areas of current research interest in our own laboratory, but similar work is being done at a number of labs around the world. We note briefly that related work is also being done to apply hybrid HMM/ANN methods to non-speech applications such as handwriting recognition (see, e.g., [72]).

6.1 Learning Pronunciations with Nets

The quality of phonological models have sometimes been observed to have a significant effect on ASR performance [17]. It is also our hypothesis that discriminant models such as a phonetically trained ANN can have severe problems when the phonological model is very inaccurate (since the ANN may strongly discriminate against the correct phone) and when class priors observed on the training data are not sufficiently representative of priors used in the recognition models.

In recent work at ICSI on a restaurant query speech understanding system called the Berkeley Restaurant Project (BeRP) [39], Chuck Wooters developed an approach for generating models of multiple pronunciations using a modification of our iterated training. Briefly, this procedure consisted of the following steps (sketched in Figure 11):

1. Initialize the neural network architecture by training on TIMIT, which (as noted previously) is a phonetically hand-labeled database.

2. Initialize the word models to be several equally probable possibilities, as determined by dictionary look-up, phonological rule generation, and/or examples from hand-labeled speech (e.g., TIMIT).

3. Run the training data forward through the network, generating posterior probabilities. Use either these probabilities[10] or scaled likelihoods (after division by priors) in a dynamic

[10] As suggested in [9], our results in these experiments confirmed that division by priors was not actually necessary

programming procedure (forced Viterbi) to determine the alignment between word models and speech. This will result in the recognition of pronunciations in the data.

4. Using the framewise phone labels determined by the previous step, train the neural network. Training techniques are the same as those described earlier in this article.

5. Generate new word models by including all examples of pronunciations as labeled in the training data. Merge them using the Stolcke-Omohundro HMM merging technique [77], which uses a Bayesian criterion to create a HMM from examples.

6. Run steps 3 through 5 until performance on an independent cross-validation set does not improve appreciably.

In our internal BeRP task, inclusion of multiple pronunciations reduced word recognition errors by 20% (i.e., one-fifth of the errors went away), and semantic recognition errors (i.e., errors in filling out the database query) by 30%. We are currently applying this approach to some other tasks of interest to us.

6.2 Segmental Approaches

One of the principal limitations of standard HMM-based approaches is the required assumption of independent and identically distributed (i.i.d.) acoustic vectors within the speech segment corresponding to a state. In fact, not only are there significant dependencies between these vectors, but the nature of this dependence may be well represented by a dynamic relationship that succinctly describes the speech sound in a way that will be helpful for discrimination from other sounds. For these reasons, a number of researchers have recently focused on statistical representations of complete segments, as opposed to i.i.d. statistical estimates for sub-segmental frames [20, 21, 26].

In general, these approaches are used to estimate segment likelihoods. An alternate approach would be to estimate segment posteriors, and to incorporate some representation of the segment dynamics into the model. We have done some experimentation in this direction in which we have explicitly incorporated the time index within a segment as part of the network input. In another class of approaches, segments can be modeled as being produced by an explicit articulatory model. Finally, attention can be focused on the transitions rather than the relatively stationary portions of segments in an attempt to model the perceptual sensitivity to such regions. This has been done for diphone likelihoods [26], and a posterior model focusing on transitions is described in [55].

In all of these cases, there is an attempt to better model the nonstationary nature of speech. All of this work is at an early stage, both for connectionist and non-connectionist approaches. However, it is possible that connectionist algorithms may provide a useful paradigm for the incorporation of more faithful representations of speech.

6.3 Transition-based Approaches

In the previous section, we briefly mentioned that it might be desirable to model transitions in speech. One approach to this is to use time derivative features [25]. In general, though, the

when the pronunciations were learned from the data. This result was probably due to the fact that the phone priors which are implicit to the induced models are fairly similar to those implied by the relative frequency of phone classes in the training data.

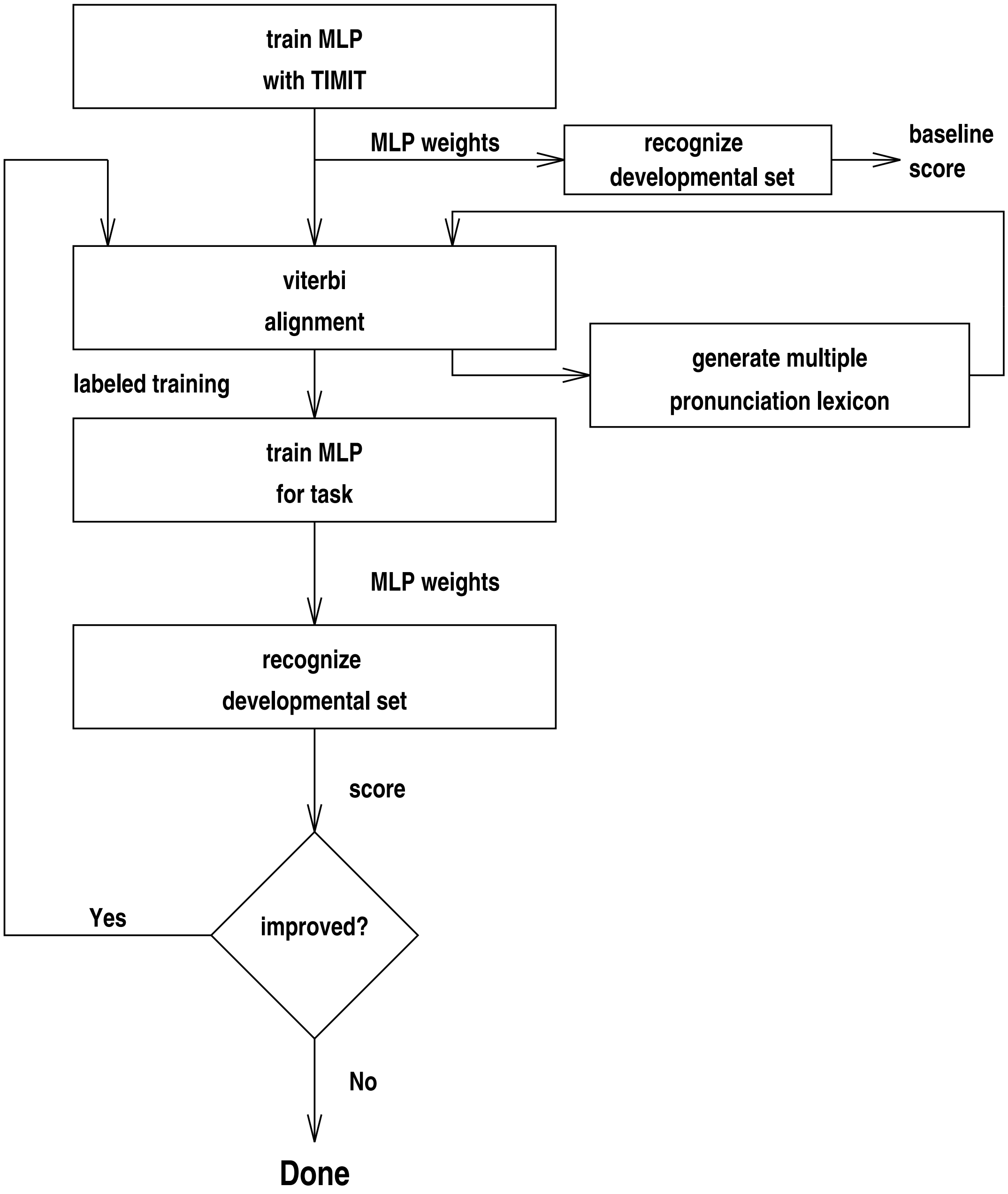

Figure 11: Embedded multi-pronunciation Viterbi learning with MLP.

problem of modeling (non-stationary) transitions is still an open one. Another step in this direction is to use highpass or bandpass filtering of critical band trajectories (RASTA processing) to emphasize transitions [31]. While this is sometimes helpful in reducing errors due to mismatches between training and testing conditions, the resulting observation sequence is a representation that has emphasized the regions of strong change and de-emphasized temporal regions without significant spectral change. This is a mismatch to the underlying speech model in standard HMMs, which has been designed to represent piecewise stationary signals. Therefore, it is likely that transition-based systems will require a fundamentally different kind of underlying statistical model. The segment-based approaches mentioned in the previous section offer one possibility. Another is to model speech as a succession of auditory events or *avents*, separated by relatively stationary periods (ca. 50-150 ms). Avents correspond to times when the spectrum and amplitude are rapidly changing, which are believed to be the most important regions for phonetic discrimination [25]. The stationary periods are mapped to a single tied state, and so modeling power is focused on regions of significant change. This is described further in [55].

It is implicit in transition classification that the transition occurs at some unique time (e.g., for a single frame). This is difficult for a classifier to learn, as the neighboring acoustic vectors are often extracted from time regions with very similar acoustics. We are currently investigating an approach in which an iterative procedure reminiscent of the Baum-Welch algorithm (which is used to iteratively converge to Maximum Likelihood estimates) is used to generate "soft" labels for each frame that correspond to estimates of the posterior probability of each possible transition. These estimates are generated in order to optimize the global posterior probability (known to minimize the actual classification error rate) for the sequence of word models, given a sequence of acoustic vectors. A neural network can then be trained to approximate the mapping to this probability. Algorithms to do this and a proof of their convergence can be found in [6].

6.4 Merging Neural Networks as Statistical Experts

Multiple networks can, in principle, be used together in order to improve performance over a single "expert." Additionally, this kind of approach can be used to speed training; this is the application that is discussed here.

As we have gradually increased the size of our speech corpora and networks, the computation required for training has greatly increased. Surprisingly, the increase has been sub-quadratic, as the number of required passes through the data has decreased. For instance, for training with six million frames of Wall Street Journal data, a net with 1.6 million free parameters trained with a single pass through the data (though the initial weights were loaded from a TIMIT-trained network, which required several passes through a significantly smaller database). However, as the corpora get even larger, and the nets correspondingly get larger to take advantage of the increased data, training times could become prohibitive, even with fast hardware (see Section 6). There are a number of possible solutions to this problem, some of which we are currently investigating:

- Modularization with broad classes - while a number of early promising experiments were done with broad class classification, we had found that a single large network with many fine classes appeared to work quite well given sufficient training data. However, it is possible that breaking up the probability estimation into several pieces can ultimately be more computationally efficient. In some preliminary experiments with small networks that discriminate between each class and all others, we have observed comparable performance to what we achieve with a monolithic network [74]. While performance was not improved, and even the number of parameters seemed roughly comparable (for comparable performance), communication between elements was greatly improved, which could be helpful for future parallel implementations.

- Merging random data splits - the complete fine category network can also be split into two or more networks that are each trained by random splits of the data. The probability estimates from each network are averaged, as illustrated in Figure 12. If the corpus is large enough that each split is representative of the complete space, this provides a way to both decrease communication and computation, as each of the nets can be made proportionately

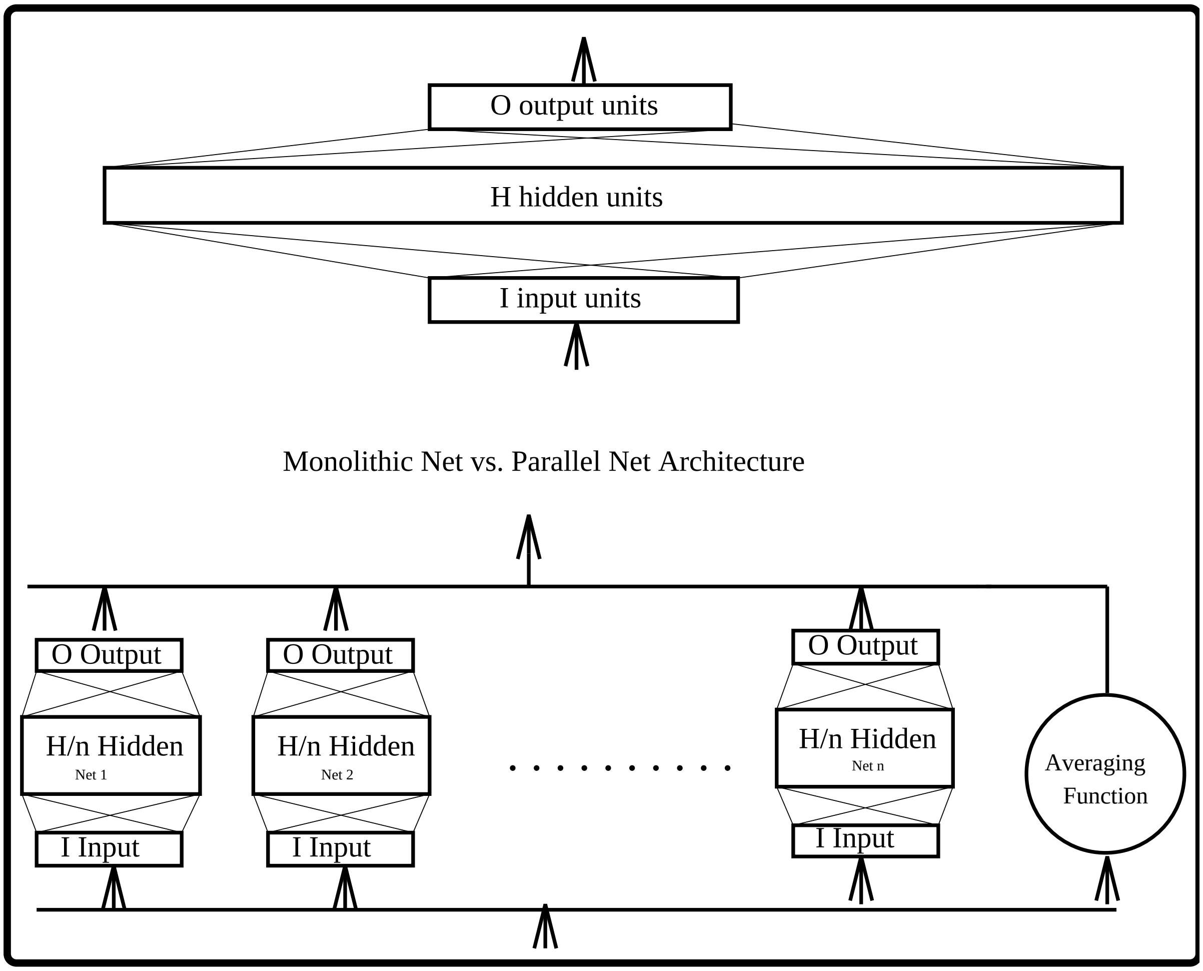

Figure 12: Training multiple smaller probability estimation MLPs instead of a single monolithic network.

smaller as well.[11] Preliminary tests seem to show that this works.[50]

- Merging known categories - as noted earlier, gender-based splits of the data have been successfully used to train separate networks (or to modify monolithic networks with a gender input variable). Other categories, such as rate of speech, dialect region, or unsupervised data clusters could be used to define a partition of the data for the training of multiple networks. An approach of this kind was proposed in [29]. Speaker-dependent estimators for voiced stop consonant probabilities were weighted and summed with gating elements trained with error back-propagation. More generally, a number of authors have recently discussed the mixture of information from separately-trained experts [35]. During recognition, two approaches can be taken: either weight each network with probabilities from the gating network on every frame, or perform multiple recognitions based on the assumption of one partition being the "correct" one for the duration of the utterance. The latter approach probably makes sense for classes such as gender that don't ordinarily change in mid-sentence (speaker consistency).

- Selective sampling - for extremely large corpora, a fraction of the data is sufficient to train the parameters to values that are quite close to the optimum values. For further refinement by the rest of the corpus, it is likely that only examples near the boundaries are necessary. It is possible that training time could be reduced significantly by some combination of random and selective choice of training examples. During training, the boundaries are unknown, but the network that has been trained in such a way can nonetheless provide class probabilities for the statistical search. However, the class priors must be compensated for as with the other methods described earlier.

It is often the goal to apply some of these approaches to improve classification performance, in addition to providing modularity as described above. However, for the large vocabulary continuous speech recognition tasks (with a large training database) that we have explored, we have not observed such improvements in comparison to very large monolithic neural networks.

7 System Issues

We have justified our interest in connectionist approaches by the known limitations of current systems, and potential relief from these limitations with newer methods. However, connectionist algorithms have at least one apparent disadvantage: training requires orders of magnitude more computation than the more common HMM training paradigm. This is a necessary consequence of phonetically discriminant training, in which all parameters are potentially modified by patterns belonging to any class. As noted earlier, large vocabulary HMM systems frequently have many more parameters than are used in ANNs; however, since they are typically not trained discriminatively, most parameters are not modified for any particular pattern (those parameters associated with classes other than the class of the pattern).

As a result, the amount of computation required for ANN training can be orders of magnitude more than is required for traditional HMM training. Because of this, connectionist speech researchers have found it necessary to use fast computational hardware. In our own laboratory, we have worked on developing computational systems (both hardware and software) to facilitate the training [53]. These computers need to be fully programmable machines, since general programs must run on them as we alter our training paradigms in the course of research. On the other

[11] Of course, the smaller network must still be large enough to perform the required mapping function.

hand, having some architectural specialization to make common training operations efficient can save significant amounts of time.

In the recognition process, neural network evaluation is only moderately more computationally intensive than the evaluation of large Gaussian mixtures. Although this cost must be considered, most of the time required for large searches and grammars is typically due to memory access rather than raw computation. Therefore, what is generally required for large vocabulary recognition is a large memory that is connected with high bandwidth to a processor that can be easily programmed, regardless of whether ANNs or Gaussian mixtures are used for the emission probability estimation.

As workstations continue their inevitable climb to higher performance over the rest of this decade, many laboratories will be able to experiment with neural network estimators, as the increased costs (even for training) will no longer be a significant limitation for many problems. However, experimental work with the largest models will still require enough computation for training so that the increased cost (relative to pure HMM systems) may still justify some specialized systems.

8 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. We briefly list here the most common of these.

- 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. This can be shown to be a variant of the maximum likelihood estimate in which the dynamic of the system (acoustic correlation) is taken into account. This could either be done using multiple networks (one per state) [78], or with one network that uses the state identity as a control input [43]. Such systems have been successfully trained to do connected digit recognition. They also have sometimes been used for larger tasks, but apparently have not been as successful as classification-based hybrids. In any event, the predictive systems are also used as hybrids, in that a search such as the Viterbi is used to integrate the information from the network, which in this case is a prediction error signal. These networks are the nonlinear equivalent of autoregressive HMMs [61]. Potentially, predictive networks can represent segment dynamics, once we learn how to use them effectively; however, they do not alleviate the piecewise stationarity assumption that is endemic to classical HMM-based approaches.

- ANN models of HMMs - In this paper we have discussed ways in which connectionist algorithms can replace others in generating probabilities used in an HMM. In some other work, researchers have shown that connectionist structures can be used to represent standard HMM-based algorithms. Once such subsystem was the Viterbi network [44]. In this case, each HMM is associated with a neural network in which each output corresponds to a HMM state. Each output unit is complemented by time-delayed connections between the different output units to represent the topology of the HMM, followed by a comparator sub-network to compute the minimum of the activation values of output nodes at the previous time step. This network does implement the core of the dynamic time warp algorithm, but it does not replace the pointer bookkeeping that is required for a practical Viterbi implementation.

In another connectionist formulation of what had previously been considered an entirely non-connectionist algorithm, the Alpha-Net [11] was introduced to simulate the forward recurrence of the forward-backward HMM algorithm. The Alpha Net is recurrent, the recurrent units are linear, and the acoustic vectors enter the loop via a multiplication to simulate the operation of a standard HMM state using a full likelihood criterion (as opposed to the Viterbi criterion).

- Global optimization through nonlinear transformation - Instead of using the network outputs as probabilities, the network can be viewed as providing a general nonlinear transformation of the observation vectors that are then used for an otherwise standard HMM-based system. This permits a global optimization of the input transformation together with a global training of the HMMs [5].

- Preprocessing - Many researchers have used feature map representations, related to one of the formulations from Kohonen and collaborators [40], to generate feature representations for a speech recognizer. In other designs, researchers have experimented with networks to provide mappings from noisy to clean data [75] or from a new speaker to an old speaker [34].

- Postprocessing - Because of interest in more complex models of speech and the difficulty in training such models (such as the segmental models mentioned above), some researchers have used more computationally demanding approaches as a postprocessing step in speech recognition. In this method, the primary recognizer generates either a list of candidate utterances (commonly referred to as an *N-best list*), or a lattice of potential words and associated acoustic information (such as emission probabilities from the primary system). The secondary system then rescores the candidate list or reprocesses the lattice hypotheses with some new criterion. For instance, in the case of the Segmental Neural Network [2], 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.

  Although this approach generates segment probabilities as opposed to the frame probabilities of the approach focused on in this paper, it is still close in spirit, as it generates posterior phone probabilities and then uses them in an HMM-based system as part of a global decoding strategy.

9 Summary and Challenges for the Future

In this paper, we have focused on a tutorial description of the hybrid HMM/ANN method. The approach has been applied to large vocabulary continuous speech recognition, and variants are in use by many researchers. The method provides a mechanism for incorporating a range of sources of evidence without strong assumptions about their joint statistics, and may have applicability to much more complex systems that can incorporate deep acoustic and linguistic context. The method is inherently discriminant and conservative of parameters.

Despite these potential advantages, the hybrid method thus far has been focused on implementing fairly simple systems, which do surprisingly well on large continuous speech recognition tasks. We and others are only beginning to explore the use of more complex structures with this paradigm. In particular, we are just beginning to look at the connectionist inference of language models (including phonology) from data, which may be required in order to take advantage of locally discriminant probabilities rather than simply translating to likelihoods.

Finally, our current intuition is that more advanced versions of the hybrid method can greatly benefit from a perceptual perspective. In such an approach, we may be able to use some simple properties of the auditory system to greatly constrain the space of important dependencies between each state and the prior states and acoustic vectors [12]. This could permit a much more comprehensive model than we are currently using in any of our HMM-based systems, including those with connectionist probability estimators. Such models may be necessary in order to determine the ultimate utility of neural networks for speech recognition.

[12] Some initial theory for this extension is described in [55], and the mathematical basis for training of such systems with a global maximum a posteriori (MAP) criterion as described in [6].

References

[1] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. on Elec. Com.*, vol. EC16, pp. 279-307, 1967.

[2] G. Zavaliagkos, Y. Zhao, R. Schwartz, and J. Makhoul, "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.

[3] J. Baker, "The DRAGON System - An overview," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 24-29, 1975.

[4] L. Baum, "An inequality and associated maximization techniques in statistical estimation of probabilistic functions of Markov processes," *Inequalities*, no. 3, pp. 1-8, 1972.

[5] Y. Bengio, R. De Mori, G. Flammia, & R. Kompe, "Global optimization of a neural network-Hidden Markov Model hybrid," *IEEE Trans, on Neural Networks*, vol. 3, no. 2, pp. 252-259, 1992.

[6] H. Bourlard, Y. Konig and N. Morgan, "REMAP: recursive estimation and maximization of a posterior probabilities – Application to transition-based connectionist speech recognition," *ICSI Technical Report TR-94-064*, 1994.

[7] H. Bourlard and N. Morgan, "A continuous speech recognition system embedding MLP into HMM," in *Advances in Neural Information Processing Systems 2* (D. S. Touretzky, Ed.), pp. 413–416. Morgan Kaufmann, San Mateo CA, 1990.

[8] H. Bourlard and N. Morgan, N. "CDNN: A context dependent neural network for nontinuous speech recognition," *IEEE Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing* (San Francisco, CA), pp. II:349-352, 1992.

[9] H. Bourlard and N. Morgan, *Connectionist Speech Recognition - A Hybrid Approach*, Kluwer Academic Press, 1994.

[10] H. Bourlard and C. J. Wellekens, "Links between Markov models and multilayer perceptrons," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1167–1178, 1990.

[11] J. S. Bridle, "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.

[12] J. S. Bridle, "Alpha-Nets: a recurrent neural network architecture with a hidden Markov model interpretation," *Speech Communication*, vol. 9, pp. 83-92, 1990.

[13] D. Broomhead, & D. Lowe, "Multi-variable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321-355, 1988.

[14] P. F. Brown, "The Acoustic-Modelling Problem in Automatic Speech Recognition," *PhD Thesis*, School of Computer Science, Carnegie Mellon University, 1987.

[15] A. Bryson and Yu Chi Ho, *Applied Optimal Control*, Blaisdel Publishing Company, 1969.

[16] Y. Chow, M. Dunham, O. Kimball, M. Krasner, G. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwarz, "BYBLOS: The BBN continuous speech recognition system," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing* Dallas, Texas, pp. 89-92, 1987.

[17] M. Cohen, "Phonological Structures for Speech Recognition," *PhD Thesis*, University of California at Berkeley, 1989.

[18] M. Cohen, H. Murveit, J. Bernstein, P. Price, and M. Weintraub, "The DECIPHER speech recognition system, " in *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing* (Albuquerque, NM), pp. 77-80, 1990.

[19] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash, "Context-dependent multiple distribution phonetic modeling," in *Advances in Neural Information Processing Systems 5* (S.J. Hanson, J.D. Cowan, and C.L. Giles, Eds.), pp. 649-657, 1993.

[20] L. Deng, "A generalized hidden markov model with state-conditioned trend functions of time for the speech signal." *Signal Processing*, 27:65–78, 1992.

[21] V.V. Digalakis, J.R. Rohlicek, and M. Ostendorf. "Segment-based stochastic models of spectral dynamics for continuous speech recognition." *IEEE trans. on Speech and Audio Processing*, 1(4):431–442, October 1993.

[22] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley Interscience, New York, 1973.

[23] H. Franco, M. Cohen, N. Morgan, D. Rumelhart, and V. Abrash, "Context-dependent connectionist probability estimation in a hybrid hidden Markov model-neural net speech recognition system," *Computer Speech and Language,* vol. 8, no. 3, pp. 211-222, July 1994.

[24] M. Franzini, K.F. Lee, and A. Waibel, "Connectionist Viterbi training: a new hybrid method for continuous speech recognition," *IEEE Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. 425-428, Albuquerque, NM, 1990.

[25] S. Furui, "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.

[26] O. Ghitza and M.M. Sondhi. "Hidden markov models with templates as non-stationary states: an application to speech recognition." *Computer Speech and Language*, 2:101–119, 1993.

[27] H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing* (Albuquerque, NM), pp. 1361-1364, 1990.

[28] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," *IEEE Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. I-13-16, San Francisco, CA, 1992.

[29] J. Hampshire and A. Waibel, "Connectionist architectures for multi-speaker phoneme recognition," in *Advances in Neural Information Processing Systems 2* (D. S. Touretzky, Ed.), Morgan Kaufmann, CA, 1990.

[30] H. Hermansky, "Perceptual Linear Predictive (PLP) analysis of speech," *Journal of the Acoust. Soc. Am.*, vol. 87, no. 4, 1990.

[31] H. Hermansky and N. Morgan, "RASTA processing of speech", *IEEE Transactions on Speech and Audio Processing*, special issue on Robust Speech Recognition, vol.2 no. 4, pp. 578-589, Oct., 1994

[32] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Networks,* Addison Wesley, 1991.

[33] M. Hochberg, S. Renals, and A. Robinson, "ABBOT: The CUED hybrid connectionist-HMM large-vocabulary recognition system," in *Proc. ARPA Spoken Language Technology Workshop*, 1994.

[34] X.D. Huang, K.F. Lee, and A. Waibel, "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.

[35] R. Jacobs and M. Jordan M. "Linear piecewize control strategies in a modular neural network architecture", *IEEE Trans. on Systems, Man, and Cybernetics,* March/April 1993, vol. 23, nr. 2, pp. 337-345, 1993

[36] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532-555, 1976.

[37] F. Jelinek "Self-organized modelling for speech recognition," in *Readings in Speech Recognition*, A. Waibel and K. Lee (eds.), pp. 450-503, Morgan Kaufmann, 1990.

[38] L. Jiang and E. Barnard, "Choosing contexts for neural networks," *Oregon Graduate Institute Technical Report*, 1994.

[39] D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, and N. Morgan, "The Berkeley restaurant project," in *Proc. Intl. Conf. on Spoken Language Processing* (Yokohama, Japan), In Press, 1994.

[40] T. Kohonen, "The 'neural' phonetic typewriter," *IEEE Computer*: 11-22, 1988.

[41] Y. Konig, N. Morgan, C. Wooters, V. Abrash, M. Cohen, and H. Franco, "Modeling consistency in a speaker independent continuous speech recognition system," in *Advances in Neural Information Processing Systems 5* (S.J. Hanson, J.D. Cowan, and C.L. Giles, Eds.), pp. 682-687, 1993.

[42] K. F. Lee, *Large vocabulary speaker-independent continuous speech recognition: The SPHINX system*, Kluwer Academic Publishers, 1988.

[43] E. Levin, "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.

[44] R. P. Lippmann, "Review of neural networks for speech recognition," *Neural Computation*, vol. 1, no. 1, pp. 1-38, 1989.

[45] R. P. Lippmann and E. Singer, "Hybrid neural-network/HMM approaches to wordspotting," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing* Minneapolis, Minn., pp. I-565-568, 1993.

[46] R. Lippmann, personal communication, 1994.

[47] D.M. Lubensky, A.O. Asadi, and J.M. Naik, "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.

[48] S. Makino, T. Kawabata, and K. Kido, "Recognition of consonant based on the Perceptron model," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing,* Boston, Mass., pp. 738-741, 1983.

[49] M. Minsky, & S. Papert, *Perceptrons*, Cambridge, MA: MIT Press, 1969.

[50] Mirghafori, N., Morgan, N., and Bourlard, H., "Parallel training of MLP probability estimators for speech recognition: a gender-based approach" *IEEE Workshop on Neural Networks for Signal Processing*, Greece, pp.289-298, 1994.

[51] D. P. Morgan and C. L. Scofield, *Neural Networks and Speech Processing,* Kluwer Academic Publishers, 1991.

[52] N. Morgan, "Big Dumb Deural Nets (BDNN): a working brute force approach to speech recognition", Proceedings of the ICNN, vol. VII, pp.4462-4465, 1994.

[53] N. Morgan, J. Beck, P. Kohn, J. Bilmes, E. Allman, and J. Beer, "The Ring Array Processor (RAP): a multiprocessing peripheral for connectionist applications," *Journal of Parallel and Distributed Computing,* Special Issue on Neural Networks, vol. 14, pp.248-259, 1992.

[54] N. Morgan and H. Bourlard, "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.

[55] N. Morgan, H. Bourlard, S. Greenberg, and H. Hermansky, "Stochastic Perceptual Auditory-Event-Based Models (SPAM) for speech recognition", Intl. Conference on Spoken Language Processing, pp. 1943-1946, 1994.

[56] N. Morgan, H. Hermansky, H. Bourlard, P. Kohn, and C. Wooters, "Continuous speech recognition using PLP analysis with multilayer perceptrons," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing*, pp. 49-52, Toronto, Canada, 1991.

[57] H. Ney, "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.

[58] D. Parker, *Invention Report S81-64*, File 1, Office of Technology Licensing, Stanford University, 1982.

[59] D. Parker, "Learning logic," *Technical Report TR-47*, Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, 1985.

[60] S.M. Peeling & R.K. Moore, "Isolated digit recognition experiments using the multi-layer perceptron," *Speech Communication*, vol. 7, pp. 403-409, 1988.

[61] A. Poritz, "Linear predictive Hidden Markov Models and the speech signal," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing*, pp. 1291-1294, Paris, 1982.

[62] A. B. Poritz and A.L. Richter, "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.

[63] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, 1989.

[64] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist optimization of tied mixture Hidden Markov Models," in *Advances in Neural Information Processing Systems 4* (J. Moody, S. Hanson, and R. Lippmann, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 167-174, 1992.

[65] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 1, pp. 161-174, 1994.

[66] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities." *Neural Computation*, no. 3, pp. 461-483, 1991.

[67] T. Robinson and F. Fallside, "A recurrent error propagation network speech recognition system," *Computer Speech and Language*, no. 5, pp. 259–274, 1991.

[68] T. Robinson, L. Almeida, J.M. Boite, H. Bourlard, F. Fallside, M. Hochberg, D. Kershaw, P. Kohn, Y. Konig, N. Morgan, J.P. Neto, S. Renals, M. Saerens, & C. Wooters, "A neural network based, speaker independent, large vocabulary, continuous speech recognition system: The WERNICKE Project," *Proc. EUROSPEECH'93* (Berlin, Germany), pp. 1941-1944, 1993.

[69] F. Rosenblatt, *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, 1962.

[70] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Procressing* (D. E. Rumelhart and J.L. McClelland, Eds.), vol. 1, pp. 318–362. MIT Press, Cambridge MA, 1986.

[71] R. Schaefer and L. Rabiner, "Digital representations of speech signals," in *Proceedings of the IEEE*, vol. 63, no. 4, pp. 662-667, 1975.

[72] M. Schenkel, I. Guyon, I., & D. Henderson, "On-line cursive script recognition using neural networks and hidden Markov models," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. II.637-640, Adelaide, Australia, 1994.

[73] R. Schwartz, Oral presentation, *Speech Research Symposium XIII*, Johns Hopkins, 1993.

[74] R. Salomon, Oral presentation, ICSI, July, 1994.

[75] H. Sorenson, "A cepstral noise reduction multi-layer network," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing* Toronto, Canada, pp. 933-936, 1991.

[76] E. Singer and R. Lippmann, "A speech recognizer using radial basis function neural networks in an HMM framework," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing* (San Francisco, CA), pp. 629-632, 1992.

[77] A. Stolcke and S. Omohundro, "Hidden Markov model induction by Bayesian model merging," in *Advances in Neural Information Processing Systems 5* (S.J. Hanson, J.D. Cowan, and C.L. Giles, Eds.), San Mateo, CA, Morgan Kaufmann, 1993.

[78] J. Tebelskis and A. Waibel, "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.

[79] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition: neural networks vs. hidden Markov models," in *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing* (NY, NY), pp. 107-110, 1988.

[80] R. Watrous and L. Shastri, "Learning phonetic features using connectionist networks: an experiment in speech recognition," in *Proc. First Intl. Conf. on Neural Networks*, (San Diego, CA), vol. 2, pp. 619-627, 1987.

[81] P. Werbos, "Beyond regression: new tools for prediction and analysis in the behavioral sciences," *PhD Thesis*, Harvard University, Cambridge, MA, 1974.

[82] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings IEEE*, vol. 78, pp. 1150–1160, 1990.

[83] B. Widrow and M. Hoff, "Adaptive Switching Circuits," *Technical Reports 1553-1*, Stanford University, Electron. Labs., Stanford, CA, 1960.

[84] P. Woodland, and S. Young, "The HTK tied-state continuous speech recognizer," Eurospeech '93, pp. 2207-2210, 1993.

[85] C. Wooters, "Lexical modeling in a speaker-independent speech understanding system," ICSI Technical Report TR-93-068, also a UC Berkeley PhD Thesis.