Merging Multilayer Perceptrons and Hidden Markov Models: Some Experiments in Continuous Speech Recognition

H. Bourlard[1], N. Morgan[2]

TR-89-033

July 1989

Abstract

The statistical and sequential nature of the human speech production system makes automatic speech recognition difficult. Hidden Markov Models (HMM) have provided a good representation of these characteristics of speech, and were a breakthrough in speech recognition research. However, the a priori choice of a model topology and weak discriminative power limit HMM capabilities. Recently, connectionist models have been recognized as an alternative tool. Their main useful properties are their discriminative power and their ability to capture input-output relationships. They have also proved useful in dealing with statistical data. However, the sequential character of speech is difficult to handle with connectionist models.

We have used a classic form of a connectionist system, the Multilayer Perceptron (MLP), for the recognition of continuous speech as part of an HMM system. We show theoretically and experimentally that the outputs of the MLP approximate the probability distribution over output classes conditioned on the input (i.e., the Maximum a Posteriori (MAP) probabilities). We also report the results of a series of speech recognition experiments. By using contextual information at the input of the MLP, frame classification performance can be achieved which is significantly improved over the corresponding performance for simple Maximum Likelihood probabilities, or even MAP probabilities whthout the benefit of context.

However, it was not so easy to improve the recognition of words in continuous speech by the use of an MLP, although it was clear that the classification at the frame and phoneme levels was better than we achieved with our HMM system. We present several modifications of the original methods that were required to achieve acceptable performance at the word level. Preliminary results are reported for a 1000 word vocabulary, phoneme based, speaker-dependent continuous speech recognition system embedding MLP into HMM. These results show equivalent recognition performance using either the Maximum Likelihood or the outputs of an MLP to estimate emission probabilities of an HMM.

---

1. Philips Research Laboratory Brussels, Belgium.

2. International Computer Science Institute, Berkeley, California.

Merging Multilayer Perceptrons & Hidden Markov Models: Some Experiments in Continuous Speech Recognition

H. Bourlard [†,‡] & N. Morgan [‡]

[‡] International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704, USA.

[†] Philips Research Laboratory Brussels
Av. Van Becelaere 2, Box 8,
B-1170 Brussels, Belgium.

1 Introduction

Hidden Markov Models (HMM), which are widely used for automatic speech recognition, inherently incorporate the sequential and statistical character of the speech signal. However, their discriminant properties are weak if they are trained along the *Maximum Likelihood Estimate (MLE)* [Brown, 1987]. An algorithm based on another criterion, *Maximum Mutual Information (MMI)* [Brown, 1987] provides more discrimination, but the mathematics are more complex, and many constraining assumptions must be made. Finally, the incorporation of acoustic or phonetic contextual information requires a complex *HMM* and a large (possibly prohibitive) storage capacity.

On the other hand, connectionist architectures, and more particularly *Multilayer Perceptrons (MLP)*, have been recognized as an alternative tool for pattern recognition problems such as speech recognition. Their main useful properties are their discriminative power and their capcaity to learn and represent implicit knowledge. Also, contextual information can easily be incorporated. Good results for phonetic decoding have already been reported [Bourlard & Wellekens, 1989a], but are restricted to local decisions, as *MLPs* are feedforward machines that are generally used for classification of static inputs with no sequential processing. If the connections are supplied with delays, feedback loops can be added providing dynamic and implicit memory. Several authors [Jordan, 1986; Watrous, 1987; Elman, 1988] have proposed original architectures of this type.

In this report, we discuss the link between stochastic models used in speech recognition and connectionist devices used as classifiers [Bourlard & Wellekens, 1989b]. The hypotheses made when using Markov models are compared with the potential solution offered by *MLPs*. It is shown theoretically and empirically that the outputs of the *MLP* approximate the probability distribution over classes conditioned on the input (i.e., the *Maximum a Posteriori (MAP)* probabilities, also referred to here as the Bayes probabilities). It is also shown that these estimates, made using contextual information at the input of the *MLP*, lead to frame classification performance which is significantly improved over the corresponding performance for *MLE* or *MAP* probabilities, where the latter are estimated without the benefit of context.

Although it was clear that the classification at the frame and phoneme levels was better, the recognition of words in continuous speech was not so simply improved by the use of an *MLP*. Several modifications and improvements of the initial ideas were necessary for getting acceptable performance at the word level. These modifications are presented here and preliminary results are reported for a 1000 word vocabulary, phoneme based, speaker-dependent continuous speech recognition system embedding *MLP* into *HMM*.

2 Hidden Markov Models

In the generic discrete *HMM*, the acoustic vector (e.g., cepstra calculated for each 10 ms speech frame) is quantized in a front-end processor. Each vector is replaced by the closest (in the Euclidean sense) prototype vector y_i, selected in a predetermined finite set $\mathcal{Y}$ of cardinality I. Let $\mathcal{Q}$ be a set of K different states $q(k)$, with $k = 1, \ldots, K$. Markov models [Bahl & Jelinek, 1975] are constituted by the association of some of these states according to a predefined topology. If *HMM* are trained using the *MLE* criterion, the parameters of the models (defined below) are optimized for maximizing $P(X|W)$, where X is a training sequence of quantized acoustic vectors $x_n \in \mathcal{Y}$, with $n = 1, \ldots, N$ and W is its associated Markov model made up of L states $q_\ell \in \mathcal{Q}$ with $\ell = 1, \ldots, L$. Of course, $L \neq K \neq N$since the same state may occur several times with different indices ℓ, since all states do not appear in the model, and since loops on states are allowed. Dropping the parenthesized index for a particular model, we denote by q_ℓ^n the presence of state q_ℓ at a given time $n \in [1, N]$. Events q_ℓ^n are mutually exclusive so that probability $P(X|W)$ can be written for any arbitrary n:

$$P(X|W) = \sum_{\ell=1}^{L} P(q_\ell^n, X|W) \,, \tag{1}$$

where $P(q_\ell^n, X|W)$ denotes the probability that X is produced by W while associating x_n with state q_ℓ. Maximization of (1) can be calculated from the forward-backward recurrences of the *Baum-Welch* algorithm [Brown, 1987].

Maximization of $P(X|W)$ is usually approximated using the *Viterbi* method. It uses a simplified version of the *MLE* criterion, in which only the most probable state sequence in W capable of producing X is determined. To explicitly show all possible paths, (1) can also be rewritten as

$$P(X|W) \;=\; \sum_{\ell_1=1}^{L} \cdots \sum_{\ell_N=1}^{L} P(q_{\ell_1}^1, \ldots, q_{\ell_N}^N, X|W) \,.$$

The *Viterbi* criterion can be obtained by replacing all summations by a "max " operator. Probability (1) is then approximated by:

$$\overline{P}(X|W) \;=\; \max_{\ell_1, \ldots, \ell_N} P(q_{\ell_1}^1, \ldots, q_{\ell_N}^N, X|W) \,, \tag{2}$$

and can be calculated by the classical *Dynamic Time Warping (DTW)* algorithm [Bourlard et al., 1985; Ney, 1984; Sakoe, 1979]. For this case, each training vector is uniquely associated with only one particular transition (at time n) $\{q(k) \rightarrow q(\ell)\}$ between two states $\in Q$.

In both cases (*MLE* and *Viterbi*), it can be shown that probabilities $P(X|W)$ and $\overline{P}(X|W)$ can be recursively computed from "local" contributions $p[q_\ell^n, x_n|Q_1^{n-1}, X, W]$,

where Q_1^{n-1} stands for the state sequence associated with the previously observed vector sequence $x_1, \ldots, x_{n-1}$. For simplicity, it is generally assumed that the model is a first order Markov model (i.e., each state is conditioned on the previous state only) and that the acoustic vectors are not correlated (i.e, X may be overlooked in the conditional). These "local" contributions are then estimated from the set of local parameters $p[q(\ell), y_i | q^-(k), W]$, for $i = 1, \ldots, I$ and $k, \ell = 1, \ldots, K$. Notations $q^-(k)$ and $q(\ell)$ denote states $\in \mathcal{Q}$ observed at two consecutive instants. In the particular case of the Viterbi criterion, these parameters are estimated by:

$$\hat{p}[q(\ell), y_i | q^-(k)] = \frac{n_{ik\ell}}{\sum_{j=1}^{I} \sum_{m=1}^{K} n_{jkm}}, \quad \forall i \in [1, I], \quad \forall k, \ell \in [1, K], \tag{3}$$

where $n_{ik\ell}$ denotes the number of times each prototype vector y_i has been associated with a particular transition from $q(k)$ to $q(\ell)$ during the training. To reduce the number of parameters, (3) may be split into an *emission probability* $p(y_i | q(\ell))$ and a *transition probability* $p(q(\ell) | q^-(k))$. However, criterion (1) and estimates (3) do not minimize the error rate, either at the word level or at the acoustic vector level. Discrimination is not a criterion for training the models in a *MLE*. Consequently, the local probability (3) is not a good measure for the labeling of a prototype vector y_i, i.e., to find the most probable state given a current input vector and a specified previous state. Indeed, the optimal decision should be based on the *Maximum a Posteriori Probability (MAP)*, (also referred to here as the Bayes probability). In that case, the most probable state $q(\ell_{opt})$ is defined by

$$\ell_{opt} = \underset{\ell}{\operatorname{argmax}} \; p[q(\ell) | y_i, q^-(k)], \tag{4}$$

and not on the basis of (1).
It is easy to prove that the estimates of the Bayes probabilities in (4) are:

$$\hat{p}[q(\ell) | y_i, q^-(k)] = \frac{n_{ik\ell}}{\sum_{m=1}^{K} n_{ikm}}. \tag{5}$$

Thus, the optimal criterion, minimizing the decoding error rate at the word or at the frame level, should be based on the *MAP*. This assertion and the role of other probabilities used in stochastic speech recognition were clearly explained in [Nadas et al., 1988]. It was also shown that the *MAP* estimate appears safer for the training of speech recognizers when the language model is poor, e.g., when the a priori word probabilities are poorly estimated. However, the conclusions of this paper were only valid for isolated word recognition, and did not apply to local probabilities used in the lower level decoding process (i.e., frame probabilities used in an *HMM* for continuous speech recognition).

In the next section it is shown that the output values of an *MLP* are estimates of *MAP*-like probabilities, i.e., (5) (or something more general if there is feedback from higher layers to the input field). Using an *MLP* with this feedback and extending the input field to include context, output probabilities can be generated which depend on a fixed temporal window on both states and observations. These requirements were explained in [Poritz, 1988].

3 Statistical Inference in Multilayer Perceptrons

Let $q(k)$, with $k = 1, \ldots, K$, be the output units of an *MLP* associated with different classes (each of them corresponding to a particular state of $\mathcal{Q}$). Assume that the training

is performed on a sequence of N vector quantized inputs $\{y_{i_1}, \ldots, y_{i_N}\}$ where $y_{i_n} \in \mathcal{Y}$. The training of the *MLP* parameters is usually based on the minimization of the following *mean square criterion (LMSE)*:

$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} [g(i_n, k) - d(i_n, k)]^2 \ , \tag{6}$$

where $g(i_n, k)$ represents the output value of unit k given y_{i_n} at the input and $d(i_n, k)$ is the associated target value and is equal to $\delta_{k\ell}$ (Kronecker delta) if the input is known to belong to class $q(\ell)$. Multiple presentations of the same current prototype are not necessarily associated with the same class (inconsistent training). Expanding summations to collect all terms depending on the same y_i, (6) can be rewritten as:

$$E = \frac{1}{2} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{\ell=1}^{K} n_{ik} \, . \, [g(i, \ell) - d(i, \ell)]^2 \ , \tag{7}$$

where n_{ik} represents the number of times y_i has been classified as having been generated from $q(k)$. Thus, whatever the *MLP* topology is, e.g., the number of its hidden layers and of units per layer, the optimal output values $g_{opt}(i, k)$ are obtained by canceling the partial derivative of E versus $g(i, k)$. It can easily be proved that, doing so, the optimal values for the outputs are then

$$g_{opt}(i, k) \ = \ \frac{n_{ik}}{\sum_{\ell=1}^{K} n_{i\ell}} \ , \tag{8}$$

which are the estimates of the Bayes probabilities (5) (not including transition probabilities). However, these optimal values can only be reached if the *MLP* contains enough parameters, does not get stuck into a local minimum during the training, and is trained long enough to reach the minimum.

These results follow directly from the minimized criterion, not from the topology of the model. In fact, the same optimal values (8) may also result from other criteria, such as the entropy or relative entropy of the targets with respect to the output [Bourlard & Wellekens, 1989b]. This solution, obtained by cancelling the partial derivative of the error criterion (*LMSE* or entropy) versus the output vector g, is the optimal set of values that could be reached by the algorithm actually used for the training of the *MLP*, the error *back-propagation algorithm (EBP)*. In this procedure, a gradient estimate is used to cancel the partial derivatives of the error versus the weight parameters W.
Indeed:

$$\frac{\partial E}{\partial w_{ij}} \ = \ \nabla_g^t E \, . \frac{\partial g}{\partial w_{ij}} \quad , \forall i, j \ ,$$

where t signifies the transpose operation. Thus, a minimum in the output space ($\nabla_g E = 0$) is also a minimum in the parameter space ($\partial E / \partial w_{ij} = 0, \ \forall i, j$). However, it is also clear that $\partial E / \partial w_{ij} = 0, \ \forall i, j$, does not necessarily lead to $\nabla_g E = 0$ which then implies that the network has converged to a *local minimum* of the error function. In this case, the outputs will not be the *MAP* probabilities. In fact, it is no longer guaranteed that the output values will look like probabilities, e.g., that they sum up to unity. An elegant way to circumvent that problem is to replace the classical sigmoidal function applied at the output units by a "softmax" function [Bridle, 1989] defined, for any i, as:

$$g(i, k) = \frac{e^{x(i,k)}}{\sum_{\ell=1}^{K} e^{x(i,\ell)}} \ , \tag{9}$$

where $x(i,k)$ is the output value of unit k before the nonlinearity for an input y_i. This function generalizes the sigmoid and has a nice relationship with the Gibbs distribution [Bridle, 1989].

For these experiments we have used a discrete-input *MLP* for classification with "one-from-K coding", i.e., one output for each class, with all targets zero except for the correct class where it is unity. If there are enough parameters in the system and if the training does not get stuck in a local minimum, the output values of the *MLP* will approximate the a posteriori probabilities (Bayes probabilities). Section 4 will show some empirical evidence for this assertion.

This conclusion can also be generalized to continuous inputs. It is known by regression theory that an *LMSE* criterion (as well as other criteria , including the entropy) converges, if there is enough training data, to the conditional expectation of the output given the input. That is, the estimate will converge to $E[d(t)|v(t)]$, where $v(t)$ stands for the input vector at time t and $d(t)$ the associated desired output. In classification mode, as $d(t)$ is a "one-from-K coding", we have then $E[d(t)|v(t)] = P[d(t)|v(t)]$, i.e., the probability distribution over classes conditioned on the input.

Since these results are independent of the topology of the models, they remain valid for linear discriminant functions. In practice, performance is limited by the number of parameters, so it is not guaranteed that the optimal values (equation 8) can be reached (even if we can escape from local minima). However, it can be shown [Devijver & Kittler, 1982, see pages 171-172] that the discriminant functions obtained by minimizing a *LMSE* criterion retains the essential property of being the best approximation, in the sense of mean-square-error, to the Bayes probabilities.

4 Classification at the frame level

As shown in Section 3, the *MLP* can at best approximate Bayes *(MAP)* probabilities. The only potential advantage of using an *MLP* instead of counting as in (5) is for interpolated estimates when there is insufficient training data for the input space, e.g., when the input is highly-dimensioned through the use of multiple frames as contextual input. This fact is clearly illustrated in the following experiments.

Two databases have been considered. The first one is a German database and is speaker dependent. It will be referred as the SPICOS data base. The second is a DARPA database and is speaker independent. It will be referred as the TIMIT data base.

4.1 SPICOS data base

Two independent sets of vocabularies for training and test are used. The characteristics of the acoustic analysis are:

- 16kHz sampling rate,
- 512-point FFT in a 10-ms frame rate with 25-ms windows,
- 'mel-scale'-dependent cepstral smoothing of spectra,
- 30 sample points of smoothed, 'mel-scaled' logarithmic spectra and the intensity value.

	training set 26767 patterns	test set 27702 patterns
Full Gaussian	65.1	64.9
MLE	45.9	44.8
MAP	53.8	53.0

Table I: Classification rates at the frame level on SPICOS by standard approaches

The training data-set consists of two sessions of 100 German sentences per speaker. These sentences are representative of the phoneme distribution in the German language and include 2430 phonemes in each session. These 2 sessions of 100 sentences are phonetically segmented on the basis of 50 phonemes. However, as the segmentation of the test set is not available, only the first session of the training set was used for training the *MLP* while the other one was used for testing the generalization capabilities and was also used as the stopping criterion (cross validation). The lexicon words of that database were not available.

The test set consists of one session of 200 sentences per speaker. The recognition vocabulary contains 918 words, including the 'silence' word. The overlap between training and recognition is 51 words, which are mostly articles, prepositions and other structural words.

The acoustic vectors were coded using an alphabet of 132 prototype-vector labels. These prototype vectors were calculated from the training data by using a standard cluster-analysis technique (K-means).

Vector-quantized mel cepstra were used as binary input to a hidden layer. Multiple input frames provided context to the network. While the size of the output layer was kept fixed at 50 units, corresponding to the 50 phonemes to be recognized, the width of the contextual input and the number of hidden units were varied. The acoustic vectors were coded as one of 132 prototype vectors by a simple binary vector with only one bit 'on', so the input field contained $132 \times a$ bits where a represents the number of frames in the input field. In that case, the total number of possible inputs is equal to 132^a. There were 26767 training patterns and 26702 independent test patterns. Of course, in the case of contextual inputs, this represented only a small fraction of the possible inputs, so that generalization was potentially difficult.

Training was done by an "error-back-propagation" algorithm, first minimizing an entropy criterion [Hinton, 1987; Solla, 1988] and then the standard least-mean-square error [Rumelhart et al, 1986]. In each iteration, the complete training set was presented, and the parameters were updated after each training pattern. To avoid overtraining of the *MLP*, improvement on the test set was checked after each iteration [Morgan & Bourlard, 1989]. If the classification rate on the test set was decreasing, the adaptation parameter of the gradient was also decreased; otherwise, it was kept constant. This test set stopping criterion was also used to determine when to switch the error measure from entropy to least-mean-square.

For comparison with classical approaches, results obtained with a Gaussian classifier described by a full covariance matrix for each class are given in Table I ("Full Gaussian"). In this case the results are very good, perhaps because the continuous mel-cepstra are

	training set 26767 patterns	test set 27702 patterns
MLE	45.9	44.8
MLP5×132-20-50	65.5	59.0
outputs/priors	60.2	51.7
MLP9×132-5-50	62.8	54.2
outputs/priors	61.5	51.9
MLP9×132-20-50	75.7	62.7
outputs/priors	72.1	57.5
MLP9×132-50-50	86.4	61.4
MLP9×132-200-50	86.9	59.4
MLP9×132-50	76.9	65.0
outputs/priors	67.7	54.5
MLP15×132-50-50	83.6	64.2
outputs/priors	86.8	64.9
MLP21×132-20-50	93.0	64.0
outputs/priors	89.7	59.1
MLP21×132-50-50	95.0	67.7
outputs/priors	95.4	66.1
MLP21×132-50	92.6	68.6
outputs/priors	87.8	62.7
MLP25×132-20-50	92.8	62.7

Table II: Classification rates at the frame level on SPICOS for different MLPs

classified directly without losing any information through the vector quantization process. In Table I, results obtained with *Maximum Likelihood Estimates (MLE)* and *Maximum a Posteriori (MAP)* probabilities are also given. In those cases, the parameters have been obtained by standard methods for estimating discrete probabilities (i.e., simply by counting). Since we know the phonemic transcription and segmentation of the training set, we can count the frequencies $F(i,j)$ of observation of label i, $i = 1,\ldots,132$ within a phoneme j, $j = 1,\ldots,50$.
The *MLE* of phoneme j is then given by:

$$p(i|j) = \frac{F(i,j)}{N_j}$$

where N_j is the overall frequency of phoneme j, and the *MAP* is:

$$p(j|i) = \frac{F(i,j)}{N_i}$$

where N_i is the overall frequency of prototype i in the training set.

Results obtained from different architectures of *MLP* are given in Table II. In that Table, "MLPa×b-c-d" stands for an *MLP* with a blocs of b (binary) input units, c hidden units and d output units. For Spicos, the size of the output layer is kept fixed at 50 units,

# hidden units	# parameters/ # training patterns	training	test
MLP9×132-5-50	.23	62.8 (1.010)	54.2 (1.012)
MLP9×132-20-50	.93	75.7 (1.030)	62.7 (1.035)
MLP9×132-50-50	2.31	86.4 (1.018)	61.4 (1.000)
MLP9×132-200-50	9.3	86.9 (1.053)	59.4 (0.995)
MLE	.25	45.9	44.8
MAP	.25	53.8	53.0
50 NCI	.34	53.5 (1.011)	52.7 (1.012)

Table III: Classification rates and *MAP* approximation at the frame level on SPICOS

corresponding to the 50 phonemes to be recognized. For the binary input case, b is the number of prototype vectors (132 for Spicos). If c is missing, there are no hidden units. Results reported in Table II clearly show that it is possible to improve the classification rates (at the frame level) obtained by classical approaches (e.g. *MLE*) by providing context to the network, which seems to be one of the potential advantages of the *MLP*. For simple relative frequency (counting) methods, it is not possible to use contextual information, because the number of parameters to be learned would be too large. Therefore, in Table I and *MLE* in Table II, the input field was restricted to a single frame. This restriction explains why the Bayes classifier (*MAP*, in Table I), which is inherently optimal for a given pattern classification problem, is shown in Table II yielding a lower performance than the potentially suboptimal *MLPs*. Frame performance is also shown for the cases where the *MLP* outputs were divided by the respective a priori class probabilities. While this generally degraded classification performance, we believed that it might lead to improved word recognition. This was later verified, as described in a later section.

An interesting empirical question was whether the *MLP* was indeed able to approximate Bayes probabilities at the output (proved theoretically in [Bourlard & Wellekens, 1989b]). We have compared the results obtained with a fixed contextual input window (9 frames) for a hidden layer which varied from 5 to 200 units (Table III). The line denoted "50 NCI" stands for the results obtained from the training of a *MLP* with 50 hidden units and *no contextual input* (only the current acoustic vector coded by 132 input units). The mean of the error between the " 50 NCI" output values and the actual *MAPs* (which can be obtained by counting) are, for the training and the test sets, equal to 2.78×10^{-4} and 2.93×10^{-4} respectively. The standard deviation was 1.15×10^{-2} in both cases, which leads to the confidence interval:

$$P(|g(i,k) - p(q_k|y_i)| > 0.04) < 0.001 \quad ,$$

using the standard assumption of normality.

In Table III, the numbers in parentheses give the average sum, over all the training or test patterns, of the *MLP* output values. Since these outputs approximate *MAP*, their sum is approximately unity. It can also be observed that the *MLP* solution converges to the optimal *MAP* performance (53.5 and 53.8 for the training set and 52.7 and 53.0 for the test set). Again, the average sums of the output values are very close to unity. All these results clearly suggest that the training did not get stuck in a very suboptimal local minimum (since the optimal global minimum can be proven to correspond to Bayes

probabilities at the output of the *MLP*). Therefore, an *MLP* can be useful in estimating Bayes probabilities associated with acoustic vectors in a temporal context which is too large for the training of a classical *HMM*.

It is also interesting to notice that large values for the parameterization ratio (# parameters / # training measurements) only corresponded to a slight degradation of generalization performance (3.3% over a factor of 10 in number of parameters). The iterative estimation process was stopped when generalization degraded for an independent data set (cross-validation) [Morgan & Bourlard, 1989], which explains the insensitivity of test set classification scores to the net size.

It can be observed in Table II that the best results are sometimes obtained with no hidden layer. We also wished to learn if the sigmoid function at the output was useful or not. Without this nonlinearity the *MLP* would reduce to simple linear discriminant functions. We wanted to observe the effect of reducing the strong discrimination due to the sigmoid function which approximates a logical decision. Accordingly, we trained one of the best *MLPs* (with 9 contextual input frames) with a linear function at the output. Two results are reported in Table IV: "LMLP9×132-50" stands for the *MLP* with no hidden units and linear outputs, "LCMLP9×132-20-50" stands for the *MLP* with 20 hidden units with linear outputs and the desired outputs that correspond to the confusion between classes (e.g,. 0.9 for the correct class, 0.6 for the classes which are close to the good one and 0.1 for the others). For comparison, Table IV also shows some results obtained with standard *MLPs* of Table II: MLP9×132-50 and MLP9×132-20-50.

It can be observed in Table IV that, for these preliminary experiments, the classification results at the frame level are worse than for the nonlinear case. This is probably because we no longer approximate the perceptron when the sigmoidal function is removed from the output; i.e., we no longer minimize the number of errors but simply a standard least square criterion. This can be seen by comparing "MLP9×132-50" and "LMLP9×132-50", where the only difference is the presence or absence of the output sigmoid. It is also important to notice that the result reported in "LMLP9×132-50" is probably a good approximation to the optimal linear discriminant since we are minimizing a standard quadratic function that has no local minima. The training is also faster.

Finally, we did a few preliminary experiments using continuous-valued inputs from the same Spicos data base. As we knew that the training of a standard *MLP* by error back-propagation was comparatively slow with continuous-valued cepstral input vectors, we only tried this for single-frame (no context) input. Results are reported in Table V:

	training set 26767 patterns	test set 27702 patterns
LMLP9×132-50	57.2	52.3
MLP9×132-50	76.9	65.0
LCMLP9×132-20-50	54.2	50.5
MLP9×132-20-50	75.7	62.7

Table IV: Classification rates at the frame level on SPICOS with linear and nonlinear outputs

	training set 26767 patterns	test set 27702 patterns
MLP1×31-50	34.6	33.8
MLP1×31-132-50	58.8	56.6
MAP	53.8	53.0
MLP9×132-5-50	62.8	54.2

Table V: Classification rates at the frame level on SPICOS with continuous inputs

"MLP1×31-50" stands for an *MLP* with one 31-dimensional cepstral vector at the input, no hidden units and the classical sigmoid function at the output. The results are very bad, probably because we are limited to 50 linear functions in a small input space.

The results denoted "MLP1×31-132-50" are more promising and were produced in an experiment inspired by radial basis function theory [Broomhead & Lowe, 1988; Niranjan & Fallside, 1988]. The function computed by the hidden units was replaced by a well-defined continuous function (e.g., a Gaussian function), and the error back-propagation was modified accordingly. Such an *MLP* can generate any kind of dichotomies if there are enough hidden units. To test the potential advantage of continuous inputs versus discrete ones, we needed to compare two systems with the same number of parameters. The *MLP* incorporated 31 continuous inputs corresponding to one 10-ms cepstral vector, 132 hidden units and 50 classical output units. If c_{ik} represents the k-th component of the i-th prototype vector (as used for quantization in the previous experiments), the weights w_{ik} between input units k and hidden unit i was fixed to c_{ik}. If $x = (x_1, \ldots, x_K)$, $K = 31$, is the input cepstral vector, the activation value of hidden unit i was defined as:

$$h_i = exp(-\sum_{k=1}^{K}(w_{ik} - x_k)^2) .$$

From these activations, the hidden-output weights were trained using the same EBP algorithm. However, as the function on the hidden unit is continuous, it is clear that the input-hidden weights could also be trained by a simple modification of the learning algorithm. However, in this preliminary test, the input-hidden weights were kept fixed to the prototype vectors, so that the second layer optimized the classification using the distances between the input vector and the 132 prototypes. It is thus a kind of "generalized" quantization where we keep all the distances instead of considering only the closest prototype. In spite of the fact that the system has no more parameters than a classical *MAP* classifier (132 prototype vectors and 132 × 50 real values, e.g., discrete emission probabilities), it can be seen in Table V that it yields better classification rates. Those results obtained with a single, but continuous, input frame are even quite equivalent to the results obtained for an *MLP* with similar parameters but with discrete and contextual inputs (MLP9×132-5-50). Of course, the continuous input case could be improved by adding contextual information and training the first layer too (which corresponds to the training of the prototype vectors in order to optimize the performance of the last layer).

	training	test	validation
MLP1×562-60	41.4 (1.176)	36.1 (1.116)	37.3 (1.129)
MLP7×562-60	69.8 (1.043)	40.3 (0.998)	40.9 (0.960)
MLE	36.0	29.2	30.2
MAP	42.2	36.4	38.2

Table VI: Classification rates at the frame level on TIMIT data base

4.2 TIMIT data base

In a further set of experiments, we attempted speaker-independent continuous speech recognition using the SRI discrete features extracted at SRI from the TIMIT data base [Murveit & Weintraub, 1988]. Each acoustic vector was described by 4 features, the mel-cepstrum (f_1), the delta mel-cepstrum (f_2), the energy (f_3) and the delta energy (f_4). These features are independently described by 256, 256, 25 and 25 prototypes, respectively. Even without contextual information from the input field, it is impossible to directly estimate the probability of observing a set of 4 features given a class (or a state) q_k without any independence assumption (as there are $256 \times 256 \times 25 \times 25$ or 4×10^7 possible inputs). Therefore, assuming independence, the joint probability estimate is

$$p(f_1, f_2, f_3, f_4|q_k) = \prod_{i=1}^{4} p(f_i|q_k) \quad . \tag{10}$$

Using Bayes' rule, the *MAP* estimate can then be calculated:

$$\hat{p}((q_k|f_1, f_2, f_3, f_4) = \frac{\prod_{i=1}^{4} p(f_i|q_k).p(q_k)}{p(f_1, f_2, f_3, f_4)} \quad . \tag{11}$$

If we now consider an *MLP* with four input groups, each of them coding a particular feature ($256 + 256 + 25 + 25 = 562$ input units), the $k - th$ output will approximate, in theory, the *MAP* probability $p(q_k|f_1, f_2, f_3, f_4)$ without any independence assumption.

Preliminary results are reported in Table VI where classification rates obtained from *MLE* (equation 10), *MAP* estimates (equation 11, i.e. under the hypothesis of independence of the 4 features) and *MLPs* (no independence assumption) are compared. The TIMIT data base was described using 60 phonemes. "MLP1×562-60" and "MLP7×562-60" stand for *MLPs* with one input frame (no contextual information) and seven input frames, respectively. For the training of this data base, the crossvalidation technique already used on SPICOS was extended by splitting the data in three parts: one for the training, one for the test and a third one absolutely independent of the training procedure for validation. As reported in Table VI, no significant difference was observed between classification rates for the test and validation data.

The classification rates obtained from the *MAP* estimated by (11) are similar to those obtained with the *MLP* without contextual input (1 x 562 - 60). This suggests that there is not enough class-dependent correlation of the input features to make a difference in the recognition scores. However, it is also true that the training procedure is not guaranteed to reach the optimal solution. If some contextual information (7 x 562 - 60) is added , the network performance is somewhat better. As in Table I, the numbers in parentheses

give the average sum, over the training set, test set or validation set, of the output values. Again, these values are very close to unity.

5 Integrating an MLP into an HMM

In the experiments described in the previous Sections, each acoustic vector was classified independently of the preceding classifications; the sequential character of the speech signal was not modeled, except in the sense of the time-space mapping which was done to use the context of multiple time frames in the classification of each frame. The system has no short-term memory from one classification to the next one, and successive classifications may be contradictory. As shown by both the theoretical and experimental results above, *MLP* output values may be considered to be *MAP* probabilities for pattern classification. Either these, or some other related quantity (such as the output normalized by the prior probability of the corresponding class) may be used in the Viterbi search *(DTW)* to determine the best time-warped succession of states (speech sounds) to explain the observed speech measurements. This hybrid approach (*MLP* to estimate probabilities, *HMM* to incorporate them to segment continuous speech into a succession of words) has the potential of exploiting the interpolating capabilities of *MLPs* while using the *DTW* procedure to capture the dynamics of speech. We know of no one else who has been successful at using this or any other method to recognize large-vocabulary continuous speech (at least 1000 words) with an *MLP* at an accuracy as good as that of conventional statistical systems. Nonetheless, it is likely that any successful effort of this sort must make use of some method (such as the *DTW*) to recognize the succession of states which comprises an entire continuous utterance.

Thus, a feedforward *MLP* can be used to estimate the emission probabilities of an *HMM*. When recurrent connections are added to the *MLP*, state transition probabilities may also be incorporated. Several recurrent networks have already been proposed for speech recognition. In [Prager et al., 1986], a particular Boltzmann machine dealing with sequential inputs was defined where some of the hidden units, called "carry units", were supplied as extra inputs with the purpose of generating a time dynamic. In [Watrous & Shastri, 1987], sequential processing is obtained with the "temporal flow model". Delayed self-loops are added to the hidden and output units of the *MLP*. The system described in [Elman, 1988] is also an alternative implementation of this network in which the output self-loops are eliminated, and where the delayed hidden unit values are fed back as supplementary input units. Feedback is easily implemented by extending the input field with an additional vector containing the hidden unit values generated by the preceding input frame. All these machines can be referred as "hidden-to-input feedback" models. Approximation of these recurrent networks over a finite time period has been presented in [Waibel et al., 1988]. In that case, the loops at each layer are replaced by the explicit use of several preceding activation values. The activations in each non-input layer are computed from the current and multiple delayed values of the preceding layer.

In the *HMM* formalism, the speech signal is modeled as being produced by a (first order) Markov source for which the probability of reaching a particular state depends entirely on the previous state and on the observed acoustic vectors associated with the speech time slots. In this framework, it can be shown [Bourlard & Wellekens, 1989b] that local contributions as (5) can be generated by an *MLP* feeding back to the input field, the

output values associated with the previous input frame. Canceling the partial derivative of the associated *LMSE* criterion with respect to the outputs leads to the estimates given in (5) which take the transition probabilities into account. The basic architecture of the corresponding *MLP* (using contextual information for the input acoustic vectors) is similar in design to the net developed in [Jordan, 1986] to produce output pattern sequences. It can be referred as an "output-to-input-feedback" model. Again, the generated local contributions can be used in a *DTW* process to achieve a global discriminant recognition. A straightforward generalization can be made to more general or high order Markov models by replicating outputs corresponding to several previous frames in the input field. Another way to take account of previous decisions is to represent, in the same extra input vector, a weighted sum (e.g. exponentially decreasing with time) of the preceding outputs [Jordan, 1986]. This can be achieved by adding a self-loop on the fed-back units with a scalar weight μ, with $0 < \mu < 1$. The $k-th$ output unit will then estimate the *a posteriori probability* of state $q(k)$ given the current input vector and given the probabilities that previous vectors were classified into $q(\ell)$, $\ell = 1, \ldots, K$.

The main advantage of the "output-to-input-feedback" topology, when compared with other recurrent models proposed for sequential processing [Elman, 1988; Watrous, 1987], besides the possible interpretation in terms of *HMM*, is the control of the information fed back during the training. Indeed, since the training data consists of consecutive labeled speech frames, the correct sequence of output states is known and the training is supervised by providing the correct information. All of these recurrent nets can be interpreted in terms of state space equations of "nonlinear" control theory [Robinson & Fallside, 1987].

6 Recognition at the word level

6.1 Algorithm

After training the *MLP*, each output unit is associated with a particular phoneme. As described above and in [Bourlard et al, 1989], the activation value of an output unit k for a given input y will be the estimate of $p(q(k)|y)$. These outputs (or their logarithms) can be used in a classical one-stage *DTW* [Ney, 1984] for connected speech recognition in the same way as the local contributions (5) of a discrete discriminant *HMM*. If the *MLP* has been trained for phonemic labeling (each output of the *MLP* is associated with a phoneme or a state of a phonemic *HMM*), "word models" can be build along the vertical axis of a *DTW* table by concatenating the outputs of their constituting phonemes. The horizontal axis then corresponds to the time ordering of the acoustic vectors. The *DTW* table then defines a set of grid points (i, ℓ, w) associated with time slot i of the speech signal ($i = 1, \ldots, N$) and state $q(j_\ell(w))$ with $\ell = 1, \ldots, L_w$ and where $j_\ell(w)$ is the index of the ℓ-th state in word w.

The local DTW contributions (the emission probabilities of the *HMMs*) are computed for all states by a single calculation of the *MLP* outputs per time slot (e.g., 10 msec) of the speech signal. Then, using the technique of dynamic programming [Ney, 1984], we seek, among the paths starting from grid points $(1, 1, w)$ and arriving at (N, L_w, w), $\forall w = 1, \ldots, S$ (S = size of the lexicon), that path which provides the smallest accumulated "distance" $G(N, L_w, w)$. This "distance", alternatively referred to as a matching score, is associated with the best path and defines the optimal word sequence. These accumulated

distances are obtained by the following recurrences:

$$G(i,\ell,w) \;=\; \min_{\{k\}} \{G(i-1,k,w) \;-\; \ln[g(i,k,j_\ell(w))]\}$$

within a word model, and:

$$G(i,1,w) \;=\; \min_{\{w'\}} \left\{G(i-1,L_{w'},w') \;-\; \ln p_t \;-\; \ln\left[g(i,j_{L_{w'}}(w'),j_1(w))\right]\right\}$$

between word models, where $\{k\}$ and $\{w'\}$ respectively stand for the set of possible predecessor states of $q_{j_\ell(w)}$ and possible predecessor words of w. The term $-\ln p_t$ in the second equation stands for the word transition penalty chosen to minimize the number of errors. The best path is recovered by backtracking through the *DTW* table.

If a recurrent *MLP* is used to generate local contributions as in (5), taking the transition probabilities into account, some modifications of the dynamic programming procedure are necessary and have been explained in [Bourlard & Wellekens, 1988].

6.2 Results

Table VII shows the word level performance using a simple *HMM* with a single state type per phoneme, duration being modeled by concatenating such states into a chain half as long as the number of frames in the average length of the phoneme, with a self-loop and forward transition of a fixed arbitrary probability (.4 for the self-loop and .6 for the forward transition). The performance for the best of the cases is roughly comparable with the standard *MLE*. "MLPa×b-c-d/MLE" stands for the outputs divided by the priors (which essentially estimates the *MLE*). "MLPa×b-c-d/MAP " means there has been no division by the priors. The word transition penalties have been optimized empirically on the test set (first 100 sentences) and appear to generalize well to the second set of 100 sentences. The number in the parentheses is the value of the optimal word transition probability. It appears (from our experimental results) that the severity of this penalty must be increased for larger input context. Note also that the validation set performance, shown here for the classical *MLE* case and for one of the best *MLPs*, was not substantially different from the test set performance.

Table VIII shows the results of an additional experiment in which we did not optimize the word transition penalty but instead scaled -ln(output values) between [0,1000]. In other words, an output $x = -ln(voutput[o])$ is transformed into $1000\frac{(x-x_{min})}{(x_{max}-x_{min})}$. In that way, the values in the DTW are bounded and it is easier to fix a good word transition penalty (in our case, 1000). As we had learned that *MLE* appears to give better word recognition performance than *MAP*, we only tried the *MLE* for 1, 3 and 9 input frames.

	test	validation
MLP9×132-20-50/MLE	51.9 (10^{-14})	52.2
MLP9×132-20-50/MAP	40.9 (10^{-14})	
MLP9×132-50/MLE	53.3 (10^{-14})	
MLP1×132-50-50/MLE	49.7 (10^{-7})	
MLP1×132-50-50/MAP	27.3 (10^{-7})	
regular MLE	52.6 (10^{-8})	52.5

Table VII: Word recognition rate with the optimized word transition penalties

	test
MLP1×132-50-50/MLE	47.8
MLP3×132-50-50/MLE	47.1
MLP9×132-20-50/MLE	45.2

Table VIII: Word recognition rate with output scaling

6.3 Discussion

Although it was clear that classification at the frame and phoneme levels was much better using an *MLP* than it was for simple relative-frequency Maximum Likelihood methods, the recognition of words in continuous speech was not so simply improved. Several modifications and improvements of the initial ideas were necessary for getting comparable word recognition performance. The most important modifications were:

- Better training procedure using crossvalidation techniques, in order to avoid overtraining. That has been explained in [Morgan & Bourlard, 1989]
- Division of the output values of the *MLP* by the a priori probabilities of the classes as observed on the set training. As shown in the previous Tables, this leads to an improvement of 10-20 % at the word level.
- Optimization of the transition probabilities of the *HMMs* depending on the size of the contextual window at the input to the *MLP*.

These modifications to our basic methods led to word performance comparable to those obtained with standard *HMM*. This progress, as well as the good classification rates at the frame level, seems to justify our further investigations.

7 Conclusion

It is now clear, both from a theoretical perspective and from empirical measurements, that the outputs of an *MLP* (when trained for pattern classification using the mean square criterion) approximate the *MAP* probabilities, which are the class probabilities conditioned on the input. It has also been shown that these estimates, given the use of contextual inputs to the *MLP*, lead to frame classification performance which is significantly improved over the corresponding performance for simple *MLE*, or even *MAP* without the benefit of context (at least for the speech data examined here).

The recognition of words in continuous speech is not yet significantly improved by the use of an *MLP*. Dynamic time warping *(DTW)* for decoding *HMM* is not only useful in handling the variability of speech pronunciation, but is also an efficient tool for connected speech segmentation. This property is difficult to achieve using a "neural" architecture. Indeed, even assuming a perfect dynamical system taking the complete history into account, the output of these machines is just the "instantaneous" activation value of each

output class (e.g. associated with words or phonemes) and does not determine the underlying segmentation. For example, with the *HMM* approach, even if we were able to build a very high order Markov model, the time warping would still be useful. Thus, avoiding explicit *DTW* for a real task continuous speech recognition remains a challenging problem. On the other hand, it is also questionable whether it is even desirable to replace the *DTW* algorithm, as it is well known that this process is very effective, and also has efficient hardware implementations [Murveit & Brodersen, 1986]. Lippmann & Gold (1987) have defined a neural net architecture, called the "Viterbi Net", that can implement a *DTW* decoder, but there is no obvious advantage to its use. In short, we have yet to learn how to incorporate "neural" networks to any advantage in the recognition of words in continuous speech. However, the progress that has been made for the levels which we appear to understand (e.g., frame classification) seems to justify our further investigation. Furthermore, the good levels of word recognition performance reported here suggest the viability of *MLP* methods when used in the context of an *HMM* speech recognition system.

References

[1] Bahl, L.R. & Jelinek, F. (1975). Decoding for Channels with Insertions, Deletions, and Substitutions with Applications to Speech Recognition, *IEEE Trans. on IT,* vol. 21, no. 4, pp. 404-411.

[2] Bourlard, H., Kamp, Y., Ney, H., Wellekens, C.J. (1985). Speaker-Dependent Connected Speech Recognition via Dynamic Programming and Statistical Methods, *Speech and Speaker Recognition*, Ed. M.R. Schroeder, Karger, Basel.

[3] Bourlard H. & Wellekens, C.J. (1988). Links between Markov Models and Multilayer Perceptrons, *Philips Manuscript M.263.*

[4] Bourlard, H. & Wellekens, C.J. (1989a). Speech Pattern Discrimination and Multilayer Perceptrons, *Computer, Speech and Language*, vol. 3, pp. 1-19.

[5] Bourlard, H. & Wellekens, C.J. (1989b). Links between Markov Models and Multilayer Perceptrons, *Advances in Neural Information Processing Systems 1*, Morgan Kaufmann, pp. 502-510,

[6] Bourlard H., Morgan N. & Wellekens C.J. (1989). Statistical Inference in Multilayer Perceptrons and Hidden Markov Models with Applications in Continuous Speech Recognition, to appear in *Neuro Computing, Algorithms, Architectures and Applications*, NATO ASI Series.

[7] Bridle, J.S. (1989). Probabilistic Scoring for Back-Propagation Networks, with Relationships to Statistical Pattern Recognition, *Neural Network for Computing*, Snowbird, UT.

[8] Broomhead, D.S. & Lowe, D. (1988). Radial Basis Functions, multi-variable functional interpolation and adaptive network, *Technical Report RSRE Memorandum No. 4148*, Royal Speech and Radar Establishment, Malvern, Worcester, UK.

[9] Brown, P. (1987). The Acoustic-Modeling Problem in Automatic Speech Recognition, *Ph.D. thesis*, Comp. Sc. Dep., Carnegie-Mellon University.

[10] Devijver, P.A. & Kittler, J. (1982). *Pattern Recognition: A Statistical Approach*, Prentice Hall International.

[11] Elman, J.L. (1988). Finding Structure in Time, *CRL Tech, Report 8801*, University of California, San Diego.

[12] Hinton G.E. (1987). Connectionist Learning Procedures, Technical Report CMU-CS-87-115, Carnegie Mellon University, 1987

[13] Jordan M.L. (1986). Serial Order: A Parallel Distributed Processing Approach, UCSD, Tech. Report 8604.

[14] Lippmann, R.P. & Gold, B. (1987). Neural Classifiers Useful for Speech Recognition, *1st Int. Conf. on Neural Networks*, pp. IV-417, San Diego, CA.

[15] Morgan, N. & Bourlard, H. (1989). Generalization and Parameter Estimation in Feedforward Nets: Some Experiments, *Neural Networks for Computing*, Snowbird, UT, also in *ICSI Technical Report TR-089-017.*

[16] Murveit, H. & Brodersen, R.W. (1986). An Integrated-Circuit-Based Speech Recognition System, *IEEE Trans. ASSP*, vol. 34, no. 6, pp. 1465-1472.

[17] Murveit, H. & Weintraub M. (1988). 1000-Word Speaker-Independent Continuous - Speech Recognition Using Hidden Markov Models, *Proc. Int. Conf. on ASSP-88*, pp. 115-118, New York.

[18] Nadas, A., Nahamoo, D. & Picheny M.A. (1988). On a Model-Robust Training Method for Speech Recognition, *IEEE Trans. on ASSP*, vol. 35, no.9, pp. 1432-1436.

[19] Ney, H. (1984). "The use of a one-stage dynamic programming algorithm for connected word recognition", *IEEE Trans. ASSP* vol. 32, pp.263-271.

[20] Niranjan, M. & Fallside, F. (1988). Neural Networks and Radial Basis Functions in Classifying Static Speech Patterns. *Technical Report CUED/F-INFENG/TR 22*, Cambridge University Engineering Department, UK.

[21] Poritz, A.B. (1988). Hidden Markov Models: A Guided Tour, *Proc. Int. Conf. on ASSP-88*, pp. 7-13, New York.

[22] Prager R.W., Harrison T.D. & Fallside F. (1986). Boltzmann Machines for Speech Recognition, *Computer, Speech and Language*, vol. 1, pp. 3-27.

[23] Robinson, A.J. & Fallside, F. (1987). The Utility Driven Dynamic Error Propagation Network, *Tech. Report* CUED/F-INFENG/TR.1, Cambridge Univ., UK.

[24] Rumelhart D.E., Hinton G.E. & Williams R.J. (1986). Learning Internal Representations by Error Propagation , *Parallel Distributed Processing. Exploration of the Microstructure of Cognition. vol. 1: Foundations*, Ed. D.E.Rumelhart & J.L.McClelland, MIT Press,

[25] Sakoe, H. (1979). Two-level DP-matching - A dynamic programming-based pattern matching algorithm for connected word recognition, *IEEE Trans. on ASSP,* 27, Dec. 1979, pp. 588-595.

[26] Solla S.A., Levin E. & Fleisher M. (1988). Accelerated Learning in Layered Neural Networks, AT&T Bell Labs. Manuscript,

[27] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K. (1988). Phoneme Recognition: Neural Networks vs. Hidden Markov Models, *Proc. Int. Conf. on ASSP-88*, pp. 107- 110, New York.

[28] Watrous, R.L. & Shastri, L. (1987). Learning Phonetic Features Using Connectionist Networks: an Experiment in Speech Recognition, *1st Int. Conf. on Neural Networks*, pp. IV-381-388, San Diego, CA.