Institut EURECOM
2229, route des Crêtes
B.P. 193
06904 Sophia Antipolis
FRANCE

Research Report N° 94-0~~15~~ 16

Improved Hidden Markov Models for Speech Recognition through Neural Network Learning

Chris J. Wellekens

September 1st, 1993

Telephone: +33 93 00 26 26 E-mail:
Chris J. Wellekens: +33 93 00 26 28 welleken@eurecom.fr
Fax: +33 93 00 26 27

Improved Hidden Markov Models for Speech Recognition through Neural Network Learning

Chris J. Wellekens
Institut EURECOM - 06904 Sophia Antipolis FRANCE

Abstract. Multilayer perceptrons generate a posteriori probabilities related to emission probabilities of Hidden Markov Models through Bayes rule. This property is used to improve the discrimination of HMM. Moreover, it gives rise to many statistical interpretations which can be cast in neural architectures for nonlinear prediction and triphone probability estimation.

Keywords. Multilayer perceptrons, Hidden Markov Models, Speech Recognition, Discrimination.

1 Introduction

Mathematical formalization is definitely the intellectually most satisfactory approach to solve technical issues. It offers a continuous and clear insight into the problem since any intermediate step in the computing process can be observed and has a mastered physical meaning. Unfortunately, some problems get so intricate that the number of equations or their nature prevent any reasonable mathematical treatment e.g. the model of sophisticated robot arm with multiple degrees of freedom and nonlinearities which should be real time operated. Other problems involve unexplained behavior of industrial plants or human beings e.g. the auditory or visual human apparatus. In both cases, the unability to formalize forces to alternative methodologies. In microphysics, the huge number of particles prevents any traditional use of the laws of mechanics and a combined use of statistics and mechanics gave rise to one of the most impressive discipline. In that case, hypotheses on the hidden phenomena help the scientist formalize the statistical behavior. If no reasonable hypotheses can be made, the sole approach relies on observations which are used for model training.

This is the case in speech recognition where the phonatory process is quite versatile depending on speaker's identity, stress status, cultural history and even inconsistency. Best results in speech recognition have been obtained by using statistical models the parameters of which are estimated in a training phase on speech databases. Hypotheses were made on statistics which restrict to some extent the power of this approach. Weaker and weaker hypotheses are accepted leading to improved models. However, the number of parameters grows up with the generality of the models and any significant training requires huger and huger data bases. Neural networks provide an elegant method to estimate statistics with few hypotheses and few parameters. The joint use of neural networks and statistical models lead to speech recognizers the performances of which compare favorably with the best known techniques and require less a priori knowledge on the speech production process. Also less parameters are necessary to describe the underlaying statistics and as a consequence, less training material. However, the training process of neural networks is known to be so CPU time greedy that research work is impaired by the unacceptable delays between research tests and the expected results! Dedicated machines have been built which partly circumvent the problem. While training is part of the design phase and consumes such expensive computing resources, recognition can be achieved very fast using custom devices.

For all the reasons above, speech recognition has been identified as a particularly *representative research field for neural networks and statistics.*

Speech recognition relies on comparisons between an utterance and a cascade of speech unit models. Since the pertinent information for recognition is hidden and mixed with other features (like phase lags and prosody) in the waveform, a preprocessing is required. Section 2 gives a quick summary of this representation problem.

In the early years of automatic speech recognition (ASR), word models were plain utterances of words. Speaker's variability was not easily taken into account: the only manner was to use several models per word at the price of a prohibitive computation time in the recognition phase and of storage needs. Recognition was based on the time alignment between reference and test signals: this alignment results in a distorted path in the matrix of local distances between acoustic vectors of speech frame pairs from the reference and the test signal. This distortion is known as *time warping* [17]. The resulting distorsion score is used as a classification criterion.

Statistical models (Hidden Markov Models (HMM)) have been introduced by Jelinek [1] and proved to be very efficient to create robust and even speaker independent models. From that time, the idea of learning was included as a major issue in ASR. Very soon, embedded training was proposed: the speech data base consisting in sentences of connected words is "matched" by Viterbi training on a global model formed by concatenating word models in the same order as they appear in the sentence. The most commonly used training criterion is the likelihood that an utterance be associated with a given model but it does not yield the best recognition rates due to an inherent lack of discrimination. Indeed, a word model is tuned to be highly probable for the corresponding word but no care is brought to make it as less probable as possible for the other words. Section 3 will describe in more detail this statistical approach which will be combined later with a neural network approach in a hybrid recognizer.

Neural networks are known for their learning capabilities. Interesting results in speech recognition were obtained with Time-Delay-Neural Networks ($TDNN$) i.e. multilayer perceptrons endowed with memory at each layer [2]. Memory adds on contextual representation to the static signal but $TDNN$'s hardly take time warping into account. Most interesting results were obtained for phoneme and isolated word recognition without any use of HMM.

Section 4 will be devoted to the use of Multilayer Perceptrons (MLP) in ASR. It will be showed that outputs of MLP's used for classification can be interpreted as a posteriori probabilities [3-5] of observing a given class when a signal is observed. These probabilities are related by Bayes rule to the emission (local) probabilities of HMM's. These MLP-derived probabilities are more robust than Gaussian parametric densities traditionally used in HMM because they are obtained from discriminant training with neither assumption on the shape of probability density functions (pdf's) nor quantization errors as observed in a quantized representation space. MLP architecture eases incorporation of contextual information by extension of the input field to right and left informations [4]. Nonlinearities of neural nets extend the capabilities of class partitioning [6].

Hybrid models using MLP-derived probabilities in conjunction with HMM's combine the advantages of both techniques: time warping is taken into account and discrimination is significantly improved by MLP classifier training. Recent results show that scores obtained with hybrid HMM/MLP's compare favorably with those obtained with carefully designed traditional triphone HMM's [7].

Thanks to the statistical meaning of MLP classifier outputs, most of standard laws of statistics can be cast into connectionist architectures. For instance, the difficult problem of triphone (phoneme in left and right contexts) model training can be simplified [8-9].

The emission probabilities in HMM can be extended to the probability of observing an acoustic vector on a state given the previous vector: the prediction error can then be used as a local distance between a state and an acoustic vector [10]. To alleviate the restriction of linear prediction, neural architectures have been proposed [11] to train nonlinear predictors.

2 Speech representation

2.1 Waveform preprocessing

Since speech waveforms contain information under different forms not all essential for the recognition, a preprocessing is required to provide an efficient representation in term of what is called *acoustic vectors*. Speech is by nature a nonstationary process. However due to the mechanical inertia of the phonatory apparatus, it may be considered stationary on short time intervals of a centisecond duration called *time frames*. Spectral analysis can be performed on overlapping intervals defined by a window spanning two or more centisecond time frames. The results of this analysis are stored in a vector called *acoustic vector*. Its content could be either some linearly or nonlinearly (MEL Scale) frequency spaced values of the smoothed spectrum or LPC (linear prediction coefficients) or cepstral coefficients or any other description of the speech signal over the window (PLP or Rasta-PLP coefficients)[20-21]. Analysis windows are shifted by one centisecond so that there is one acoustic vector ($\in R^d$) associated with each time frame. Some representation parameters suit better for some applications (speaker dependent or not, recognition over the phone, robust to noise recognizers). A multiresolution analysis could probably improve recognition rates.

Spectral analysis based on an FFT roughly applied to the waveform samples in an analysis window, results in a spectrum exhibiting formant peaks but also, in case of voiced speech, a fine structure due to the fundmental frequency of the glottal excitation (*pitch* is in the range of 100 to 200 Hz according to speaker's sex). Since pitch information is commonly admitted not to be pertinent for speech recognition (mainly because it is related to high level analysis like semantics and hard to deal with at the front end analysis layer!), only spectral envelope will be kept by smoothing and decimation (data compression).

Another way to get rid of pitch information is to apply homomorphic filtering [18-19]. Since the speech spectrum is the product of the vocal tract transfer function modulus and the Fourier transform of the pitch, application of a log operator transforms it into a sum. Taking the FFT of that sum generates the cepstrum which is a function of a new variable denoted quefrency that has dimension of time. By removing the high quefrency components of the cepstrum (quefrency has the dimension of time and forms with frequency the Fourier variable pair in the log domain), pitch information is liftered (i.e. filtering in the quefrency domain) out. This operation amounts to spectral smoothing. Moreover cepstral coefficients are independent of the acquisition channel gain except c_0 which is thus frequently discarded. Both spectral and cepstral approaches obviously overlook phase information since only moduli are considered.

Linear prediction is a parametric representation since it assumes an underlaying speech production model (i.e. source coding). Only information on the vocal tract model is retained i.e. the LP coefficients or any other equivalent representation set such as Parcor's or log-area ratios [17-19]. Cepstral coefficients can also be derived from the prediction coefficients but they differ from those obtained through homomorphic analysis since contrary to the latter ones, they assume an underlaying source model.

Speech representation is denoted *continuous* in opposition with the *discrete* one obtained when acoustic vectors are replaced the nearest vector taken out of a codebook. Codebook clusters are computed with vector quantization algorithms such as K-Means or LBG [22] applied on a large amount of the acoustic vectors of the preprocessed speech data base.

2.2 Lexicon representation

Speech recognition relies on comparisons between a preprocessed utterance and a cascade of speech unit models. If the number of words in the lexicon is small (less than 50), word models are used. For larger lexicons, sub-units like phonemes or triphones (phonemes in right an left contexts) are mandatory: indeed any lexicon regardless to its size can be represented in terms of a limited number of sub-units (about 40 phonemes or 2500 triphones). The addition of a new word in a lexicon amounts then to give its transcription in terms of sub-units. A transcription may be read in a dictionary but also could be automatically generated by a grapheme to phoneme algorithm as used in the front end of a text-to-speech system. Mostly, word models are merged into a lexical tree resulting in a saving of computation and storage.

Grammars used to constraint word sequences and so to reduce the search space may be either a priori described or stochastically trained. Discussions of these aspects fall beyond the scope of this paper.

3 Hidden Markov Models

Speech variability causes most of the difficulties in recognition: indeed, pronunciation of words depends of course on the speaker but even the same speaker articulates words unconsistently. To take the variability of speech into account, it is useful to create a statistical model for each speech unit: *Hidden Markov Models* [1] are perfectly suited. An utterance of a speech unit (a time sequence of acoustic vectors) is viewed as being generated by a state machine. Each time a state is visited, an acoustic vector is emitted according to a probability density function (pdf) associated with the current state. Usually due to the time progressive nature of speech, only left to right models are considered i.e. states are ordered and transitions are allowed from left to right only. No other loops but self-loops on states are accepted (fig. 1).Self loops and jumps over states allow duration control of the utterance. Two kinds of probabilities appear in an HMM: emission pdf's and transition probabilities. Emission probabilities are either computed from a parametric pdf's (Gaussian, Gaussian mixture or Laplace densities) or read from a look-up table if the representation space is vector quantized. In the latter case, no assumption on the pdf is made but a distance definition between acoustic vectors and clusters is assumed. Transition probabilities are estimated by plain frequencies.

The parameters of the models are determined by training.

3.1 Training

Two main criteria are used for training. The first one is the maximum likelihood criterion where the probability that several utterances of the word

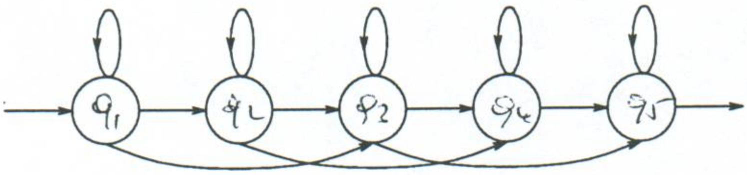

Figure 1: A left-to-right Hidden Markov Model

corresponding to the model is maximized by iterative reestimation of the parameters (Estimate-Maximize algorithm). This probability is the sum of the probabilities associated with all possible paths through the model from the input state to the output one. Probabilities of all paths are considered in the *forward backward* algorithm proposed by Baum and Welch [19].

In the sequel, we will focus on a second criterion known as *Viterbi's criterion*, which is a simplified version of the maximum likelihood where only the best path is considered. So, HMM training amounts in aligning a large amount (N) of utterances of the corresponding sub-unit W on the model. The best match is measured by the accumulated probability of the optimal (most probable) path computed with the Viterbi algorithm (Bellman's dynamic programming) in the matrix of the local emission probabilities:

$$\max \prod_{j=1}^{N} P(X_j|W) \qquad (1)$$

where $X_j = \{x_{j,1}, x_{j,2}, \ldots, x_{j,m_j}\}$ stands for an utterance of word W consisting in a sequence of m_j acoustic vectors ($\in R^d$).

Bellman's principle consists in solving a sequence of subproblems. In our case, the best scores computed from the first acoustic vector of the test utterance to any subsequent one. They represents the probability that this part of the utterance can be matched with the model. Frame by frame, it is then possible to recurrently compute the score up to the last frame. During this forward process, one keeps track of the optimal local decisions and eventually the optimal path is obtained by backtracking. It should be pointed out that the path is globally optimal: no hypotheses have been discarded during the forward process.

In practice, probabilities are commonly replaced by the opposite of their logarithms and can be viewed as local distances. Pdf's parameters are estimated by collecting all vectors emitted on a given state and transition probabilities by plain counting during the backtracking process. For instance, for Gaussian pdf's associated with a given state, the mean vector μ is estimated by the classical formula:

$$\mu = \frac{1}{n}\sum_i x_i \tag{2}$$

where n denotes the number of vectors x_i emitted on that state along the optimal path.

In case of discrete models, pdf's are the observation probabilities of cluster prototypes on the current state (look-up table). The probability of emitting cluster ξ on a given state q is estimated by:

$$\hat{p}(\xi|q) = \frac{n_\xi}{n} \tag{3}$$

where n_ξ and n are respectively the number of emissions of cluster ξ and the number of vectors emitted on state q.

Isolated utterances of speech units (as words for instance) are not mandatory for training. *Embedded training* makes use of word labeled but unsegmented (detection of boundaries between words) sentences. Optimal alignment results in a path in the matrix of local emission probabilities. This path generates the segmentation of the training material as a by-product. It is interesting to notice that this process can be extended to the training of subunits such as phonemes if the training sentence can be labeled in subunits: the tedious and unaccurate segmentation process by experts is avoided.

Using short subunits such as phonemes leads to less accurate results but requires less models and less training material. An additional advantage is the possibility to add a new word to the lexicon by just giving its phonetic spelling. A word model is then built by concatenating phoneme models. The recognition degradation comes from the *coarticulation effects*: the utterance of a phoneme is indeed strongly affected by its left (due to the continuity of the motion of the articulatory apparatus) and right (due to anticipation in the speech production process) neighbors. Occurrences of a speech-unit in various contexts are merged into a single signal model and reduces coarticulation to some extent but a definite improvement is obtained by replacing phonemes by triphones i.e. phonemes in right and left contexts. Triphone training sets up new problems: a larger training data base should be available and the frequencies of triphones span over a wide range. Significance of the estimators is questionable if no special interpolation techniques are used. However, using short subunits is the only way to tackle the large vocabulary recognition task.

An important improvement was observed when taking spectral (or cepstral) contextual effects into account. The easiest way was to increase the size of the representation space by concatenating two neighboring acoustic vectors or one vector and the difference with its neighbor [12]. Even second order differences are commonly used. The increased space dimension slows down the recognition process since the local probability computation consumes more CPU time. Another technique is to define local probability given a previous vector: it has been shown that the logarithm of this local probability is closely related to a linear prediction of acoustic vectors [10]. Both techniques require larger training data bases.

Equation (1) shows the criterion used for training word W model. So, no correlation between word models is taken into account during the training process which does not enforce discrimination; indeed, the objective is to maximize the global probability produced by an HMM when it matches a corresponding utterance but a very close probability might be observed when it is matched a wrong word. Several attempts to modify the criterion have been proposed such as Minimum Mutual Information Estimator ($MMIE$) or corrective training. Unfortunately, no convergence proofs can be given if such approaches are used: these techniques should be considered as heuristics.

Training capability is not restricted to HMM. In particular, studies in discriminant classification put into evidence the role of the perceptrons [13]. Unfortunately, perceptrons can be trained only on linearly separable data sets for two reasons: first, the criterion used is looking for an exact solution and does not use an LMS approach and second, the monolayer structure is not rich enough to describe non convex or non connected partitions. Multilayer perceptrons trained along a LMS criterion remove these severe restrictions[6].

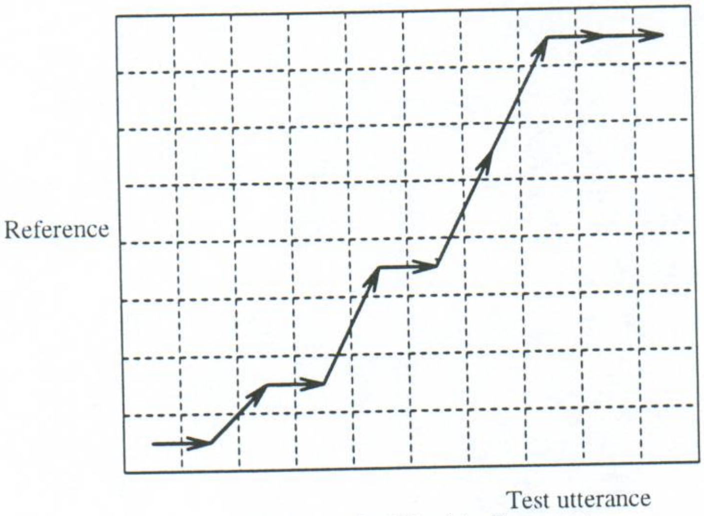

Figure 2: Viterbi alignment.

3.2 Recognition

Isolated word recognition is based on the comparison of the test utterance with all word models. The highest score identifies the recognized word. Scores are nothing but the probabilities computed using either a Baum Welch algorithm or a Viterbi alignment. So recognition uses the same algorithm as for training except backtracking. This is no longer the case for connected word recognition.

Concatenated word recognition amounts to find an optimal path aligning the test utterance with a sequence of word models. The best alignment is obtained by a generalization of Viterbi's algorithm which allows discontinuities (cfr figure 3) of the optimal path; they correspond to jumps between the end of a word and the beginning of the next one. Possible starting points are the grey-colored entries while ending points are in black-colored entries. Here the absolute value of the score does not matter but the optimal path is crucial since it yields the word sequence and boundaries simultaneously. Since determination of the best path is crucial to segment and label the test sentence, Viterbi algorithm is traditionally used [23-24].

The process can also be applied to *phonemic segmentation* but results in low recognition scores due to the intrinsic fragility of phonemic models. Word models can be built as phoneme concatenations: this word description can be viewed as a syntactical constraint on the phonemes and allows description of a full lexicon from the small phonemic set: it is the mostly used lexicon description.

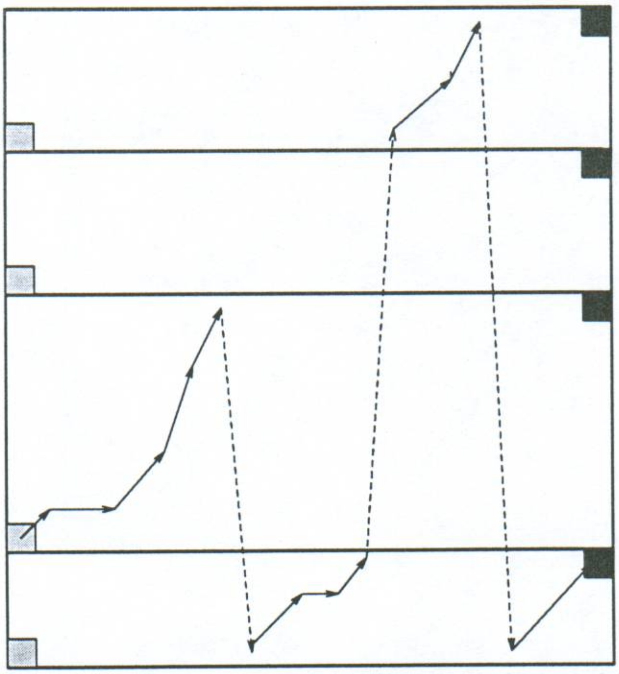

Figure 3: One-level method for connected word recognition.

4 Multilayer Perceptrons

The perceptron has been extensively studied as a classification device having an abrupt decision law [13]. A major advantage of such a decision rule is in its squashing property: how much a data is distant from the *linear* separation surface does not care. If data are linearly separable, a convergent training algorithm assigning the perceptron parameters exists. However, by trivial non linearly separable counterexamples such as the XOR, algorithm failure is easily demonstrated. Hidden layers should be inserted between the output (decision) field and the input one. No convergent training algorithm has been proposed for such structures but if the abrupt nonlinearities of nodes are smoothed out to form sigmoids, a gradient algorithm can be used (Error Back Propagation EBP) [14] to compute the optimal values of the connections. EBP is nothing else but a clever use of Newton's formula for derivating functions of functions to get gradient corrections.

Supervised training offers a powerful manner to learn mappings between input/output pairs of data. Unfortunately EBP is extremely time consuming. When used in the classification mode i.e. with all outputs equal to zero but one which corresponds to the class the input data belong, they produce a posteriori probabilities $p(\mathcal{C}_j|x)$ after training. Here $\mathcal{C}_j$ denotes classes and x is an input vector. Application of this result to speech recognition has been proposed in [4] with vector quantized inputs. While justification for continuous inputs has been published by Lippmann [5], we focus in this paper on the discrete approach for the sake of simplicity [25].

Let us thus assume the acoustic vectors of the training database have been replaced by their nearest clusters and I denotes the number of different clusters. The sequence of vectors becomes a sequence of indices i_n where $n \in [1, N]$ and $i_n \in [1, I]$. Each acoustic vector is also known to belong to a given class $\mathcal{C}_k$ where $k \in [1, K]$. An MLP with I inputs and K output is trained to map a sequence of N input index vectors y_n (all entries equal to zero but the one corresponding to the input cluster index) to a sequence of output index vectors (all entries equal to zero but the one corresponding the current class). The same cluster vector can show up in different classes depending on its time of occurrence in the sequence (class overlapping). The objective LMS function is thus:

$$E = \sum_{n=1}^{N} \sum_{k=1}^{K} [g_k(i_n) - d_k(i_n)]^2$$

where $g_k(i_n)$ stands for the output value of the k-th output given the i_n-th input is turned on and $d_k(i_n)$ is a target equal to one or zero according to the current input is associated with class $\mathcal{C}_k$. If we partition the terms of () to form groups corresponding to the same input cluster, sum over n is split into a double sum over the clusters and over the classes:

$$E = \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{\ell=1}^{K} n_{ik}[g_\ell(i) - d_\ell(i)]^2$$

where n_{ik} denotes the number of occurrences of cluster i in class $\mathcal{C}_k$. Thus, whatever the MLP topology may be i.e. number of layers and number of units per layer, the optimal output values $g_\ell^{opt}(i)$ are obtained by canceling the partial derivatives of E with respect to the $g_\ell(i)$:

$$\frac{\partial E}{\partial g_\ell(i)} = 2\sum_{k=1}^{K} n_{ik}[g_\ell(i) - d_\ell(i)] = 0$$

So, the optimal values are:

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

or by using the definition of $d_\ell(i)$:

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

This expression can be interpreted as the *a posteriori* probability:

$$g_\ell^{opt}(i) = P(\mathcal{C}_\ell|y_i).$$

This is the key result leading to hybrid HMM's: indeed, MLP's generate at their outputs a posteriori probabilities which are related to the emission probabilities of HMM by Bayes rule. The emission probabilities obtained from a MLP's generated a posteriori probabilities are more discriminant since discrimination between classes was taken into account by forcing outputs to be index vectors during the training phase.

4.1 Hybrid HMM/MLP's

By assuming for the sake of simplification that a state in a HMM corresponds to a class, we observe the local emission probabilities defined in the previous section $P(x|q)$ do not correspond to the a posteriori probabilities $P(q|x)$ but are related to them through Bayes rule:

$$P(x|q) = \frac{P(q|x)P(x)}{P(q)}. \tag{4}$$

Since in the alignment process, $P(x)$ is constant in each time frame, this factor can be discarded and $P(x|q)$ can be replaced by $P(q|x)/P(q)$. In place of using parametric pdf's or non discriminant probability look-up tables, MLP outputs divided by the a priori probability of state visits can be used in HMM : this approach is usually known as hybrid HMM/MLP. A major advantage lays in the discriminant training of $P(q|x)$ as explained above: this probability will be high or low according to the fact x belongs to state q or not.

Taking account of spectral or cepstral contexts is quite easy with hybrid HMM/MLP's. MLP's input is split into several fields: the central one contains the current vector while left anf right fields contain neighboring vectors (fig. 2). Inputs are fully connected to all hidden units, regardless to the field they belong.

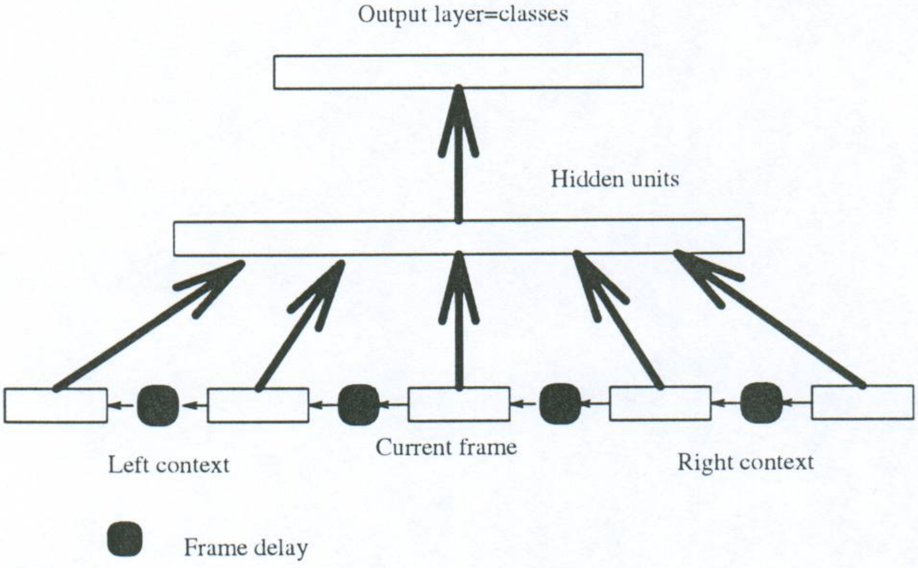

Figure 4: Context-dependent multilayer perceptron classifier.

An obvious objection could be: "How do you segment your data in state intervals as it is required in a supervised training?". Actually, embedded training also applies for hybrid HMM/MLP [15]: an initial guess segmentation is defined and EBP training is applied. Then a Viterbi alignment is performed using the probabilities generated by the trained MLP resulting in a new segmentation in state intervals. The process is iterated. Since EBP is also an iterative process, the training program contains a double iteration loop: but using crossvalidation data subset to control both loops, the training process duration can be kept within reasonable limits. However for ambitious ASR tasks like speaker independent 1000 word vocabulary continuous speech recognition (or more), dedicated boards are required [16]. The restriction of having one state per phoneme is removed since no segmentation is required.

Segmentation even in states if necessary may result from embedded training.

4.2 Triphone emission probabilities

In the previous section, triphones have been shown an efficient but expensive way to cope with coarticulation effects. The local probability of a state taking left and right contexts into account can be written $P(x|q, c^\ell, c^r)$ where c^ℓ and c^r respectively denote left and right phonemes. where c^ℓ and c^r respectively denote left and right phonemes.It factorizes according standard probability rules:

$$P(x|q, c^\ell, c^r) \;=\; \frac{P(q, c^\ell, c^r|x).P(x)}{P(q, c^\ell, c^r)}$$

Let us denote by L the number of possible contexts (regardless they appear on the right or on the left).

Using standard probability factorization properties, numerator of (4) can be written:

$$P(q|x).P(c^r|q, x).P(c^\ell|q, x, c^r). \qquad (5)$$

where the factor $P(x)$ has been overlooked since it plays no role in the alignment process as already pointed out in section 4.1. This expression suggests an original method to compute triphone probabilities [8-9]. The first factor is the standard a posteriori emission probability and may be computed as described in section 4.1.

The middle factor can be viewed as the output of an MLP with two input fields containing respectively the acoustic vector and an index vector showing the current state. This MLP has one output per possible right context (L). The last factor has also L outputs but three input fields: one contains the acoustic vector, the second the current state and the third is an index vector showing the right context.

Thus three MLP's can be used to compute the factors leading to the numerator of the triphone emission probability. The last two ones are shown in figure 5.

Some restrictions on the MLP architecture simplifies the amount of computations f.i. of $P(c^\ell|q, x, c^r)$ [8-9]. Slight assumptions allow to merge these three MLP's resulting in a substantial weight saving when compared with the single MLP which could produce the triphone probability without going to the factorization formula. Weight saving implies training data saving.

The denominator is also factorized in a similar way:

$$P(q).P(c^r|q).P(c^\ell|q, c^r). \qquad (6)$$

These factors depend on the lexicon only, are easily estimated by counting once for ever and stored in a look-up table.

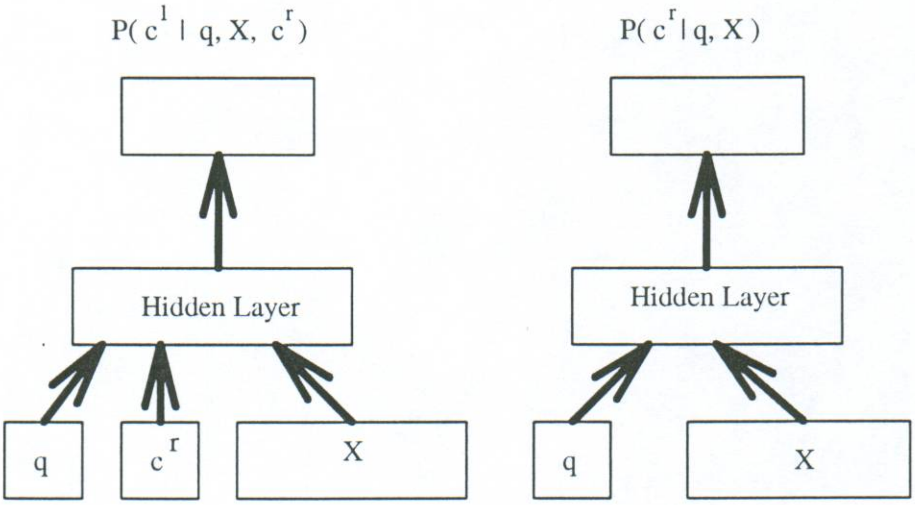

Figure 5: Triphone emission probabilities estimation by MLP's.

4.3 Nonlinear predictors

In [10], Wellekens showed that taking context into account in a Gaussian emission probability amounts to use the error of a acoustic vector prediction based on this context as a representation of the signal. This vectorial prediction error is then equivalent to a distance in log-presentation ($-logP$ equivalent to a distance) of the algorithm. Levin [11] got rid of the Gaussian assumption and the linear predictor and suggested to extend the concept to nonlinear prediction by using an MLP to generate the predictor of any acoustic vector. The principle is illustrated in figure 6. Theoretically, ther should be one such predictor per state, but in practice an additional input field (not shown on the figure) controls the state dependency. The prediction error $x(t) - \hat{x}(t)$ is then used as a local distance. The corresponding emission probability is then:

$$P(x_t|q, X_{t-p}^{t-1}) \tag{7}$$

where X_{t-p}^{t-1} stands for $\{x_{t-1}, x_{t-2}, \ldots, x_{t-p}\}$. Training is easy since the target is to minimize the LMS prediction error on a large database. In that case, the predicted value is generated as an output. No interpretation in terms of probability is possible. Prediction error is directly used as a local distance. Unfortunately, results with nonlinear predictors have never reached the recognition scores obtained with the MLP used as classifiers. This is probably because the discriminant properties are no longer enforced in the training process.

It is interesting to observe that application of Bayes rule leads another method to compute context dependent probabilities [8-9]. Indeed,

$$P(x_t|q, X_{t-p}^{t-1}) = \frac{P(x_t|X_{t-p}^{t-1}).P(q|X_{t-p}^{t})}{P(q|X_{t-p}^{t-1})}$$

The first factor in the denominator is irrelevant for time alignment since it depends only on the the signal itself and not on the states and the context dependent probability is estimated as the ratio of two MLP classifier outputs. In this case, discriminative constraints can be enforced on the MLP's which are both used as classifiers. No comparative results have been reported by the authors [8-9].

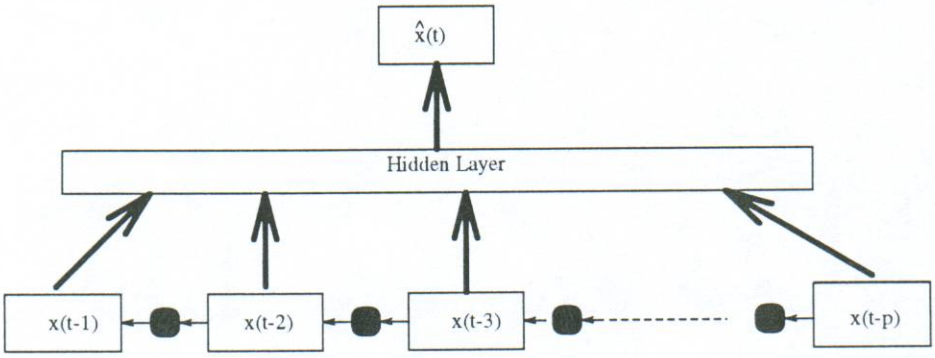

Figure 6: Nonlinear prediction using MLP's.

5 Conclusions

In the long history of ASR, a major step was the introduction of statistical techniques. But since the underlaying statistical rules completely escape to our knowledge and also for the sake of simplification, standard parametric pdf's have been hypothesized for the description of the speech production process. Even when discrete pdf's were considered, the clustering process was constraining the generality of the statistical behavior through distance definition.

The relationship between Least Mean Square minimization and statistics has been known for several decades. The idea to interpret outputs of MLP classifiers as probabilities is thus not new. However, relationship through the Bayes rule with the emission probabilities and discriminant properties inherited from the training process lead very soon to significant scores comparing favorably with very sophisticated recognizers affected by the above constraints. The price to pay was a CPU time greedy training algorithm but the regular structure of the MLP architectures allows design of modular dedicated boards and integrated chips for fast training.

Statistical interpretation of MLP outputs shows the way to cast standard results of statistics into new architectures. Building on statistics and neural networks, researchers may still expect significant progresses in ASR and more generally in most of applications where complexity forbids any explicit formulation and forces to accept training on examples as the only reasonable approach.

References

[1] F.Jelinek, Continuous Recognition by Statistical Methods, Proceedings IEEE, vol. 64 no 4, pp.532-555, 1976.

[2] A.Waibel, T.Hanazawa, G.Hinton, K.Shikano, K.Lang, Phoneme Recognition : Neural Networks vs. Hidden Markov Models, Proc. Intl. Conf. on Acoustics, Speech and Signal Processing, New York, vol.1, pp107-110, 1988.

[3] H.Bourlard & C.J.Wellekens, Multilayer Perceptrons and Speech Recognition, IEEE Proc. of the First International Conf. on Neural Networks, vol IV, pp.407-416, San Diego, CA, 1987.

[4] H.Bourlard & C.J.Wellekens, Links between Markov Models and Multilayer Perceptrons, Computer, Speech and Language, vol.3,pp.1-19, 1989

[5] R.P.Lippmann, Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities, Neural Computation, vol.3, no 4, pp. 461-484, 1991.

[6] R.P.Lippmann, An Introduction to Computing with Neural Nets, IEEE Acoustic, Speech, and Signal Processing Magazine, vol. 4, no 2, pp.4-22, 1987.

[7] M.Cohen, H.Franco, N.Morgan, D.E.Rumelhart & V.Abrash, Hybrid Neural Network/Hidden Markov Model Continuous Speech Recognition, Proc. Intl. Conf. on Spoken Language Processing, vol2, pp. 915-918, Banff, Canada, 1992.

[8] H. Bourlard, Continuous Speech Recognition: From Hidden Markov Models to Artificial Neural Networks, Doctoral Dissertation, Faculté Polytechnique de Mons, Belgium, Feb.1992.

[9] H.Bourlard, N.Morgan, Chuck Wooters & Steve Renals, CDNN: A Context Dependent Neural Network for Continuous Speech Recognition,

IEEE Proc. Intl. Conf. on Acoustics, Speech and Signal Processing San Francisco, CA, vol.2 pp. 349-352, 1992.

[10] C.J.Wellekens, Explicit Time Correlation in Hidden Markov Models in Speech Recognition, Proc. IEEE Conference on Audio, Speech and Signal Processing, pp.384-387, April 1987.

[11] E.Levin, Speech Recognition Using Hidden Control Neural Network Architecture, IEEE Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing, pp. 433-436, Albuquerque, NM, 1990.

[12] S.Furui, Speaker Independent Isolated Word Recognizer Using Dynamic Features of Speech Spectrum, IEEE Trans. on Acoustic, Speech, ans Signal Processing, vol.34, no 1, pp 52-59, 1986.

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

[14] D.E. Rumelhart, G.E.Hinton & R.J.Williams, Parallel Distributed Processing. Exploration of the Microstructure of Cognition. vol.1: Foundations, Ed. D.E.Rumelhart & J.L.McClelland, MIT Press, 1986.

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

[16] N.Morgan, J.Beck, P.Kohn, J.Bilmes, E.Allman & J. Beer, The Ring Array Processor (RAP): A multiprocessing peripheral for connectionist applications, Journal of Parallel and Distributed Computing, 1992.

[17] L.R.Rabiner, B.H.Juang, "Fundamentals of Speech Recognition", Prentice Hall, 1993.

[18] L.R.Rabiner, R.W.Schafer, "Digital Processing of Speech Signals", Prentice Hall, 1978.

[19] J.R.Deller, J.G.Proakis, J.H.L.Hansen,"Discrete-Time Processing of Speech Signals", Mc Millan, 1993.

[20] H.Hermansky, Perceptual Linear Predictive (PLP) Analysis of Speech, Jl of Acoustical Soc. of America, vol 87, no 4.

[21] H.Hermansky, J-C Junqua, Optimization of Perceptually-Based ASR Front-End, ICASSP 88, pp. 219-222.

[22] Y.Linde, A.Buzo, R.M.Gray, An Algorithm for Vector Quantizer Design, IEEE Trans. on Communications, vol 28, no 1, pp. 84-95, 1980.

[23] H.Ney, The Use of One-Stage Dynamic Programming Algorithm for Connected Word Recognition, IEEE Trans. on Acoustic, Speech and Signal Processing, pp 833-836. (1984)

[24] J.S.Bridle, M.D.Brown, R.M.Chamberlain, An Algorithm for Connected Word Recognition, ICASSP 1982, pp.899-902.

[25] H.Bourlard, C.J.Wellekens, Links between Markov Models and Multilayer Perceptrons, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 12, no 12, pp. 1167-1178.