Jean Hennebert, Martin Hasler and Hervé Dedieu
Department of Electrical Engineering
Swiss Federal Institute of Technology
1015 Lausanne, Switzerland
e-mail: martin.hasler@circ.de.epfl.ch

NEURAL NETWORKS IN SPEECH RECOGNITION

Abstract

We review some of the Artificial Neural Network (ANN) approaches used in speech recognition. Some basic principles of neural networks are briefly described as well as their current applications and performances in speech recognition. Strenghtnesses and weaknesses of pure connectionnist networks in the particular context of the speech signal are then evoqued. The emphasis is put on the capabilities of connectionnist methods to improve the performances of the Hidden Markov Model approach (HMM). Some of the principles that govern the so-called hybrid HMM-ANN approach are then briefly explained. Some recent combinations of stochastic models and ANNs known as the Hidden Control Neural Networks are also presented.

1. Introduction

It has been a long standing technical challenge to let machines acquire and expand certain human abilities. Maybe the most difficult tasks, apart from the higher level functions of the brain, are speech and image recognition. Modern computers outperform the human brain by orders of magnitude as far as the execution speed of elementary operations are concerned. On the other hand, they are far behind in the more complex perceptive tasks. It is therefore more than natural that scientists and engineers sought inspiration from biology in order to build better machines. One approach is to imitate the brain by Artificial Neural Networks (ANNs). Of course, Artificial Neural Networks are only very poor imitations, but we shall consider them here simply as a special kind.

6th Microcomputer School, invited paper, Prague, Czech Republic. 1994. pp. 23-40

In this paper we shall be concerned specifically with speech recognition. Ideally, we would like to build a system that is capable of understanding continuous speech, from any speaker, even in a noisy environment such as a cafeteria. This goal is at present completely unrealistic. Therefore, subproblems of different degrees of difficulty are considered, and specific methods are applied to solve them. The subproblems can be classified according to

- isolated words - continuous speech
- vocabulary: small (< 50 words), medium (50 - 500 words), large (> 500 words)
- single speaker - multiple speakers
- noise: noise level, type of noise

Other constraints come from the projected application. Some applications have to be in real time, e.g. automated telephone directory services, and others can be executed without stringent time constraints, e.g. dictaphones with automated speech to text capabilities. Clearly, in the latter case, more complex algorithms can be used for speech recognition.

In order to assess the capabilities of today's speech recognition technology it is interesting to consider the case of telecommunications in which the constraints are tremendously strong. Applications have to be developped in a multi-speaker environment, with a channel that produces distorsion as well as introduces various types of noise such as, additive noise, convolutionnal noise, reverberation, etc. Furthermore real time processing is required.

Despite these technical obstacles, state of the art speech recognitions tools are being introduced into the telecommunication field at increasing pace which both aim to cut the cost of services (a computer can replace thousand of human attendants in the case of directory assistance) or to generate new services such as banking services (Nippon Telephone) or Stock quotation services (Bell Northern Research (BNR). In order to give an idea of the present capabilities of today's Automatic Speech Recognizers through a telephone line we can report the field trials made by the BNR with a new stock quotation service based on an Hidden Markov Model (HMM) recognizer. The BNR has introduced this service since mid-1992. Using the 2000 company names of the New-York stock exchange, entered in phonetic form, callers can obtain the current price of a stock simply by speaking the name of the stock. The experimental system which is freely accessible is called thousands of times a day, and callers are obviously able to obtain the quotation they want (Lennig & al., 1992) (Roe, 1993).

In the case of speaker dependent applications, today's voice dictation machines are examples of a state of the art system. Dragon system (1990) is able to understand 30,000 words; Kurzweil AI's system is a 50,000 word system for medical application; IBM's system (1992) is able to understand 20,000 words. All these systems are HMM based systems, they must be trained by the (mono)-user, either by speaking a prescribed series of sentences, either by letting the system adapt to his voice during an initial period of dictations. Speaking rates of more than

50 words per minute can be achieved and word error-rates are as low as 3 to 5 % (Bahl, 1989) (Roe, 1993).

The principal difficulty for speech recognition lies in the very nature of the speech signal. The speech signal is highly non stationary, and much information is contained in the transient parts. What we as humans identify as the same speech component, e.g. a certain phoneme, has a large variety of different pronunciations. They vary in time and they depend a lot on the context. Therefore, a few simple operations on the speech signal are not sufficient for recognition and a complicated and sophisticated processing chain has to be designed in order to come close to an acceptable recognition rate.

The most widely used recognition tool are HMMs. A very good tutorial on HMMs used in speech recognition can be found in (Rabiner,1989). They are particularly adapted to the time variability of speech. In this respect, the artificial neural networks still remain problematic. On the other hand, they are superior classifiers in situations, where the classes are very heterogeneous, as in the case for classes of speech units (phonemes, words). Finally, some of the algorithms with the highest performance try to combine the advantages of both methods.

In this paper we shall give more specifically an overview of the use of artificial neural networks in speech recognition. There are numerous attempts in this direction reported in the literature. We shall concentrate on some of the main ideas. Other survey articles are (IEEE TSAP, 1994), (Lippmann, 1989).

2. Speech recognition system

In an automatic speech recognition system usually three parts can be distinguished; the preprocessor which essentially give a concise representation of the speech signal and performs data compression, the recognizer and the postprocessor which improve recognition by using additional information and which prepare the desired output. Fig. 1.

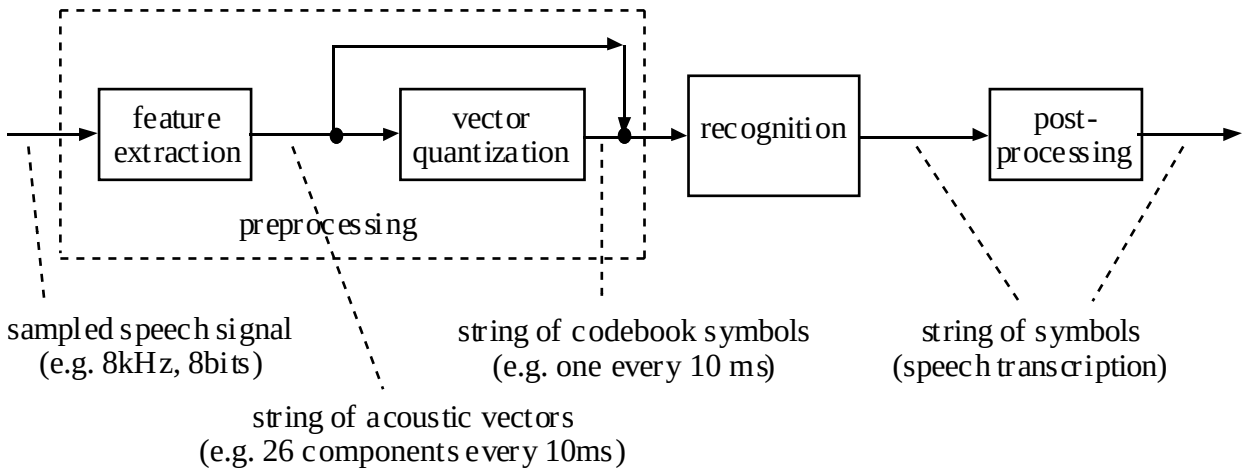

Fig. 1. Speech recognition system (recognition phase)

The feature extraction produces usually a vector every 10 ms, an *acoustic vector*, which represents the salient speech feature of a window of about 30 ms. A popular choice of features are the cepstrum coefficients, the delta-cepstrum coefficients, i.e. the estimation of their temporal derivative, the delta-energy and the delta-delta energy.

The vector quantizer detects clusters in the set of acoustic vectors and determines a representative vector for each cluster. This vector is coded, and the string of codebook vectors is fed to the recognizer. With respect to the incoming speech signal the data flow is considerably reduced. If enough time and computing power is available, the vector quantization step can be eliminated. Vector quantization is usually performed by the classical K-means algorithm, but the **Kohonen neural network** (Kohonen, 1992) has also proved to be at least as efficient, but at a much lower computational cost for the training.

The most widely used recognizers are based on hidden Markov models. It is assumed that the speech utterances are produced by a Markov process, whose states are not directly observable. For a given Markov model of, say, a word, the probability that the pronunciation of the word produces a certain speech signal can be determined. Conversely, for a given speech utterance that represents a word, the probability that the utterance has been produced when pronouncing a certain word, can be calculated in the same way. Therefore, different hypotheses of words can be tested and the most probable can be chosen. This is the recognition method based on hidden Markov models. Choosing a non zero probability that the Markov process remains in the same state at the next time instant, the time variability of speech can efficiently be modelled. The main drawback of hidden Markov models is their relatively modest discriminative power for classification.

Various neural networks have been used for speech recognition. We discuss here the following:

- **Kohonen Self-Organising Maps**
- **Multilayer Perceptron**
- **Time-Delay Neural Network**
- **Hidden Control Neural Network**
- **Combination of hidden Markov model and Connectionist Probability Estimators**

All automatic speech recognition systems, just as the humans, acquire their ability through learning. Speech utterances with known meaning are fed to the system from a database. The system then adapts its parameters such that it reacts similarly to all utterances with the same meaning. The speech recognition system in the learning phase then has the structure of Figure 2.

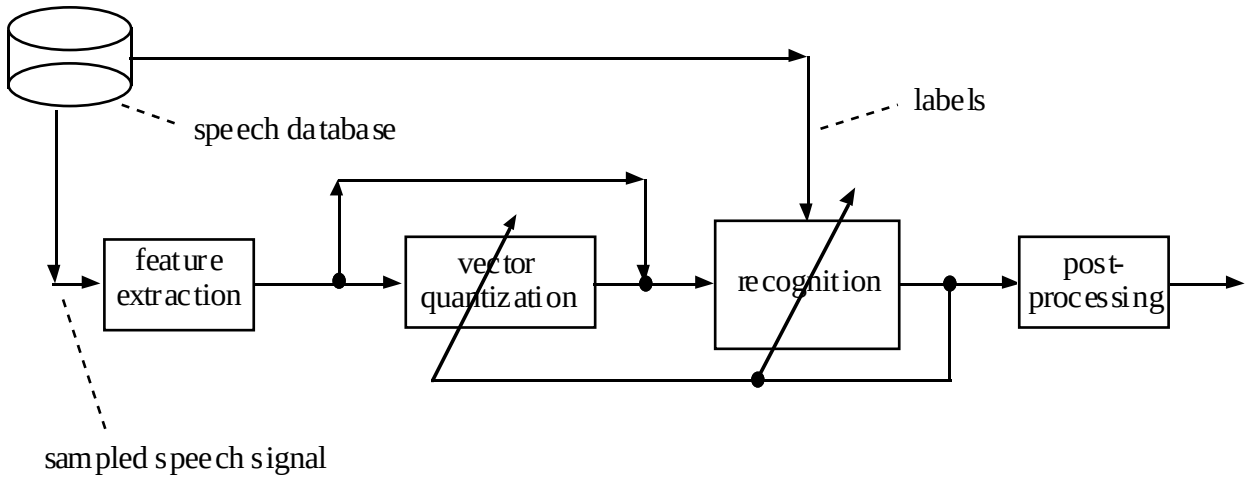

Fig. 2. Speech recognition system (learning phase)

2.Vector quantization with the Kohonen neural network

A vector quantizer can be viewed as a mapping from the input parameter space (a k-dimensional Euclidian space) into a finite set of N vectors often called code vectors, codewords or centroids. The set of code vectors forms the codebook.

The mapping operation from a continuous space into a finite set of N codewords generates a quantization error or distortion measure.

Usually codebook creation is performed using algorithms (LBG, K-means) based on iterative methods which provide an explicit minimization of the average distorsion (Gersho, 1992).

However Kohonen Self Organising Maps (Kohonen, 1992) which were initially introduced with the purpose of producing a special mapping from a high dimensional input-space to a very low dimensional output-space which conserves the topological information of the input-space can serve as Vector Quantizer as explained below.

It has been emphasized recently that Self-Organizing algorithms (Fontaine, 1994) although not explicitly based on a minimization of a distorsion criterion leads to recognition rates always slightly better than the ones provided by K-means family algorithms.

The Kohonen Self Organising Feature Map is a neural network trained by following a non-supervised algorithm. The network is made up of M^n neurons, arranged on a n dimensional lattice. The neurons are characterised by a coordinate $\boldsymbol{i} = (i_1, i_2, \cdots, i_n)$ with $1 \le i_1, i_2, \cdots, i_n \le M$ and by a synaptic weight vector $\mu_{\boldsymbol{i}} = \left(\mu_{\boldsymbol{i}}^1, \mu_{\boldsymbol{i}}^2, \cdots, \mu_{\boldsymbol{i}}^{\boldsymbol{d}}\right)$, where d is the network's input dimension. The output response of each neuron i to a d dimensional input $\xi = (\xi_1, \xi_2, \cdots, \xi_d)$ is given by

$\|\xi(t) - \mu_i(t)\|$. At time t [1], an input $\xi(t)$ is presented to the network. The first phase of the training algorithm is the selection of the winner neuron w following the condition

$$\|\xi(t) - \mu_w(t)\| = min_i \|\xi(t) - \mu_i(t)\| \quad (1)$$

Any norm $\|\cdot\|$ can be used for (1). The weights are then updated according to the equation

$$\mu_i(t+1) = \mu_i(t) + \alpha(t)\Lambda(w-i,t)\big(\xi(t) - \mu_i(t)\big) \quad (2)$$

where $\alpha(t)$ is the *adaptation gain* $\big(0 < \alpha(t) < 1\big)$ generally decreasing with time and $\Lambda(w-i,t)$ is a *neighbourhood function*, whose value is 1 for $i = w$ and generally decreasing with time and distance to the winner neuron. At each iteration, the updating equation (2) attracts the winner neuron and its neighbours towards the input ξ. After the training of the map, the weight vectors form a discrete image of the input space and tend to preserve its probability distribution. These weights can then be used as centroids (code vectors) in order to perform a Vector Quantization. The approach is then quite different to the K-Means where the training criterion is clearly to decrease monotonously the average distortion. Indeed, the distortion produced by a Kohonen Quantizer is known to be greater than the distortion produced by a K-Means' family algorithm.

Surprisingly, despite its poor distortion performance, the Kohonen algorithm produces codebooks which provide recognition rates slightly better than the recognition rates obtained when using K-Means family algorithm. These results have been reported in (Fontaine, 1994) where intensive comparisons have been made between different Vector Quantization approaches used in conjunction with HMM recognizers. The tests were carried out using different families of acoustics features and for a database, medium-vocabulary, speaker independent, american-english recorded over the telephone line. The increase of performances that can be achieved using Self-Organizing maps is in range of 2% to 6% compared to standard approaches. More details can be found in (Fontaine, 1994).

As a by-product of the training, a second piece of information is provided; namely the location of winners on the map in the sense that nearby neurons are excited by nearby inputs. It has been shown in (Zhao, 1992) that this extra piece of information can be usefully taken into account by an HMM recogniser in the case of small training set.

[1] The Kohonen algorithm is expressed in the discrete-time formalism.

3. Multilayer perceptron for recognition

The most obvious way to use multilayer perceptrons for speech recognition is to present all acoustic vectors of a speech unit (phoneme of word) at once at the input and to detect the most probable speech unit at the output by determining the output neuron with the highest activation (Fig.3, for word recognition). The learning algorithm can be the conventional backpropagation, or a more sophisticated variation of it. In the learning phase, the desired output is 1 for the correct speech unit, and 0

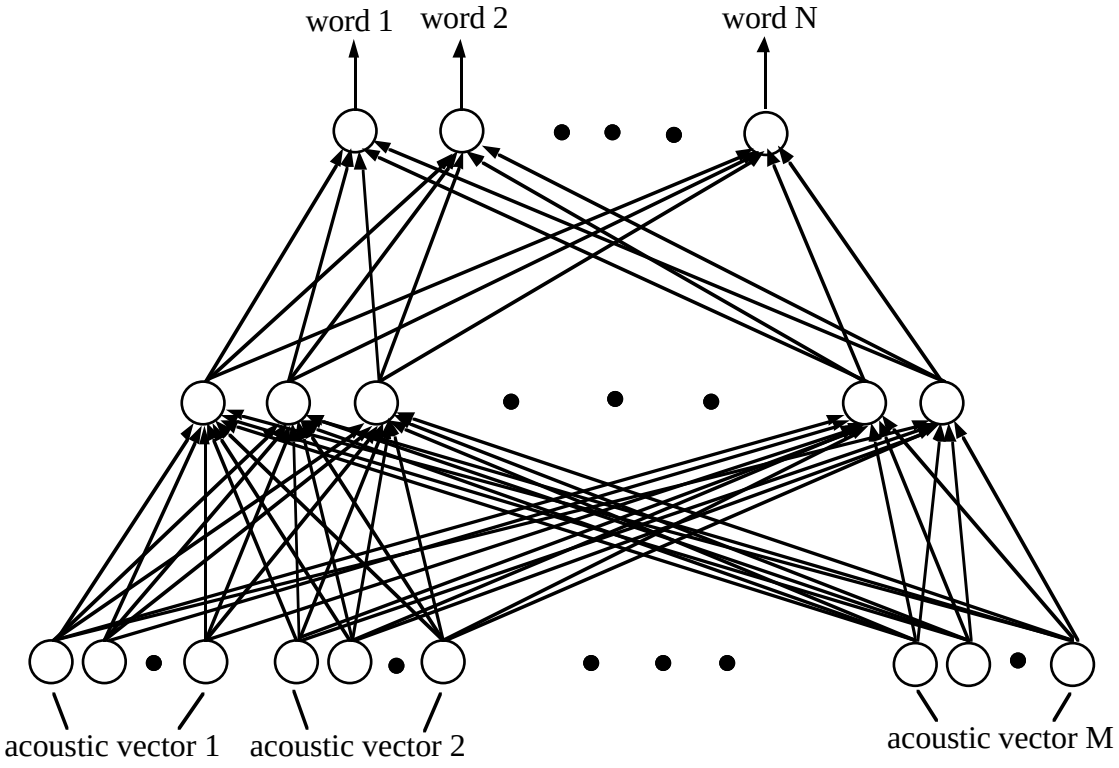

Fig. 3. Word recognition by a multilayer perceptron

for all other speech units. In this way, not only the correct output is reinforced for the corresponding sequence of acoustic vectors, but simultaneously the wrong outputs are weakened. For this reason, the multilayer perceptron has a better discriminating ability than the hidden Markov model.

The problems associated with this approach are multiple. Clearly, for word recognition, a huge number of input units has to be used. This implies an even larger number of parameters to be determined by learning and consequently the necessity to dispose of a large database. The approach is only feasible for a small vocabulary of isolated words. Its use for continuous speech is out of the question.

The method seems to be more appropriate for phoneme recognition. However, in this case, a phoneme segmented database has to be available for learning, which often is not the case. Furthermore, for recognition, in principle the speech signal has to be phoneme segmented which is a nontrivial task. In the corresponding approach with hidden Markov models, the time alignment is performed automatically in the recognition phase by the Viterbi algorithm.

Actually, the time variability of speech in general presents a problem for the multilayer perceptron. The same word pronounced by different speakers or even by the same speaker has quite different durations. Therefore, with a fixed number of inputs to the network, either some acoustic vectors have to be cut if the word pronunciation is too long, or some inputs have to be set to some arbitrary values if the word pronunciation is too short. This implies that a given neuron has not always an input that corresponds to the same part of the word pronunciation. This certainly influences negatively the recognition rate.

In order to improve the situation, various methods to align the speech signal to a fixed length, before feeding it to the multilayer perceptron, have been proposed as in (Huang, 1992).

Some preliminary results we obtained recently in the framework of a current Esprit project in Speech recognition suggests that a powerful approach could be to decompose the speech signal into stationary parts. An efficient way to discover the stationary parts is to use a wavelet decomposition of the speech signal. The decomposition is performed on each 10 ms frame and according to an entropy measure it is decided wether or not each frame can be merged with its neighbour frame (Wesfreid, 1993). Repeating this procedure recursively a very efficient merging of stationary intervals can be performed. Once the stationary parts are discovered, a classical feature extraction is applied to each stationary part. A time warping of the speech signal is then easier to perform and in the case of isolated word recognition it can be possible to constrain the number of intervals to an a priori well suited number. This time-alignement is then used to force the number of inputs to an a priori length before feeding a MLP. Preliminary MLP experiments have been carried out in the context of a small vocabulary (50 words), speaker-independent database recorded through telephone line. The recognition rates we obtained were equivalent to the one obtained with a discrete word-HMM trained for the same task.

4. Time-delay neural network for recognition

Speech recognition experiments using MLPs have been successfully carried out mostly on isolated word recognition for a small vocabulary (e.g. digit recognition task). This obvious limitation in performance of the pure MLP approach is a consequence of the inability of the MLP to deal properly with the dynamic nature of the speech as well as its intrinsic variability.

In order to take into account temporal relationships between acoustic events it has been proposed by Waibel (Waibel, 1989) to modify the architecture of the MLP in such a way that in each layer, delayed inputs are weighted and summed. This modification gives the ability to relate and compare the current inputs to their past history.

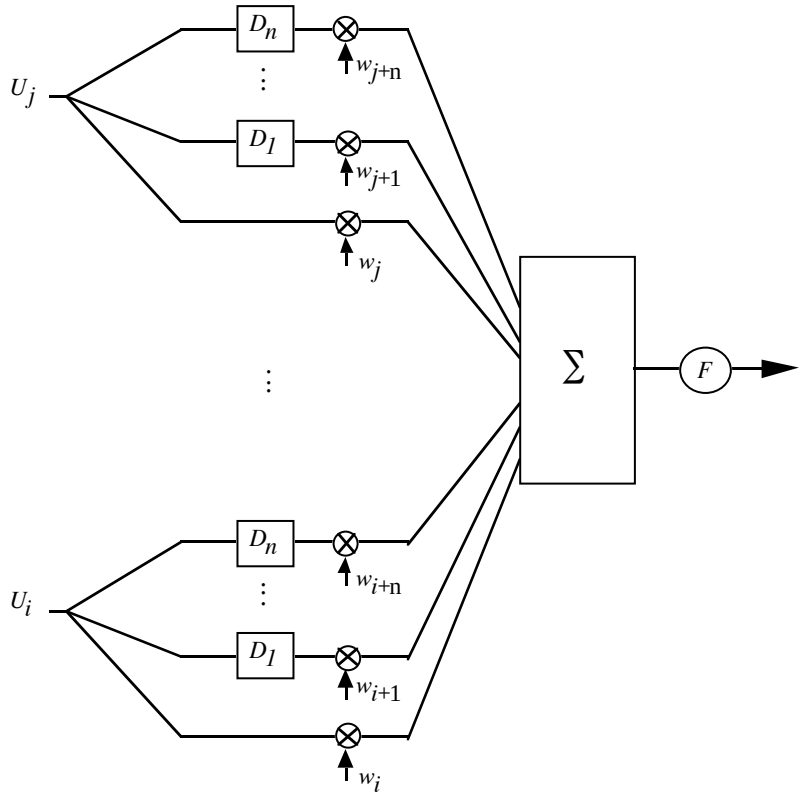

Fig.4: Basic Time Delay Unit

The basic unit of a MLP weights its inputs and sums them through a nonlinear function. In the TDNN version, this basic unit is modified by introducing delays. Figure 4 shows the basic unit of a TDNN in which *n* delayed versions of each input is taken into account.

Since hidden layers are also delayed, the TDNN is not really equivalent to an MLP which takes into account more context, but certainly, as claimed by its authors, allows the discovery of the realtionships between the acoustic-phonetic features.

The training of the TDNN network can be carried out using the backpropagation method traditionnally used for MLP.

Only a few experiments have been reported with TDNN. In (Waibel, 1989 and Lang, 1992) the TDNN has been used for a speaker-dependent recognition of the English phonemes "B", "D" and "G" in varying contexts, furthermore comparisons were made with several HMMs trained to perform the same task. Performance evaluation showed that the TDNN achieved a 98.5% recognition rate while the rate obtained with HMM was 93.7%.

5. Hidden control neural network for recognition

Multilayered neural nets have been mainly proposed as universal approximators for system modelling and nonlinear prediction. However if they are very well suited in the case of time-invariant nonlinear systems, it has been extremely difficult even impossible to apply them directly in the case of complicated nonstationary signals, such as speech signals. The reason for this failing is obvious, it is quite impossible that a network with fixed parameters can take into account and characterise the temporal and spectral variabilities of speech signals. In most of the reported experiments with nonlinear prediction using MLP, additional mechanisms have been implemented in order to enable the network to cope with the time-varying dynamics of the speech signals. In short at some stage a switching control mechanism on the weights of the network has to be implemented (Iso, 1991) or at least as described above a modification (i.e. time alignment) of the input signal has to be performed.

In order to cope with the nonstationary nature and variability of speech signals a control neural architecture has been recently proposed, called the "Hidden Control Neural Network" (HCNN). The nonlinear network (MLP) is used as a predictor model of the speech signal. "A hidden control input signal, allows the network's mapping to change over time, provides the ability to capture the nonstationary properties and to learn the underlying temporal structure of the modelled signal" (Levin, 1993).

The initial idea has been proposed by (Levin, 1993), however HCNN have been mainly investigated by (Iso, 1993) in the context of speech recognition.

In this context the HCNN can be viewed as a source of statistical models.

Suppose that our task is to recognize M words of a vocabulary and that for instance each word (as in the case of the HMM) can be represented by a sequence of left-to-right hidden states the number of which are supposed known for each word. Basically the idea behind HCNN is to train a MLP which is optimized as a predictor of speech features. The originality is that for each word an optimal sequence of control signals has to be discovered. This control signal is merely a constant real input vector which feeds the MLP during the active time of its associated state. It is worth noting that there is only one MLP common for all the M words and not M concurrent MLPs.

The approach is comparable to the HMM approach, the sequence of the hidden control inputs represent in fact the hidden sequence of the HMM states.

This control signal sequence is said to be optimal in the sense that it maximizes the likelihood (average of the logarithmic probabilities to observe each of the M words given their associated control sequence). The main difficulty in the training phase is that various parameters have to be determined (MLP weights, values of each component of the control sequence in a given state) and that a time alignment

algorithm has to be used in order to find the optimal firing instants of the M consecutive control signals.

After the training phase each word is associated with a sequence of control signals and for each word a mapping of the probability of the observation given any of the M control signal sequences is available.

In the recognition phase one has to find the more probable sequence of control signals mainly by a maximisation of the likelihood using a time-alignment algorithm.

We give below a flavour of the method presented in (Iso, 1993).

Consider the figure 5 where the so called input speech is a sequence of feature vectors (*P* components)

$\boldsymbol{a}_i, \cdots, \boldsymbol{a}_t, \cdots, \boldsymbol{a}_T$.

It is assumed that each basic speech unit (phoneme) is associated with a certain number of control signal vectors. For instance each phoneme will be associated with three or four control signals (three or four states in the HMM terminology). A word control sequence is obtained as a concatenation of the phonemes control signals. In short if there are *N* states for a word, there are *N* control signals for a word which have to be determined in the training phase.

$\boldsymbol{c}_i, \cdots, \boldsymbol{c}_t, \cdots, \boldsymbol{c}_N$

In (Iso, 1993), an MLP which acts as a nonlinear predictor is driven by this control sequence. The MLP is fed with τ preceding feature vectors and with the control command $\boldsymbol{c}_n$ (τ=1 in fig. 5). The output is the predicted value

$$\hat{\boldsymbol{a}}_{t,n} = f(\boldsymbol{a}_{t-1}, \cdots, \boldsymbol{a}_{t-\tau}, \boldsymbol{c}_n). \qquad (3)$$

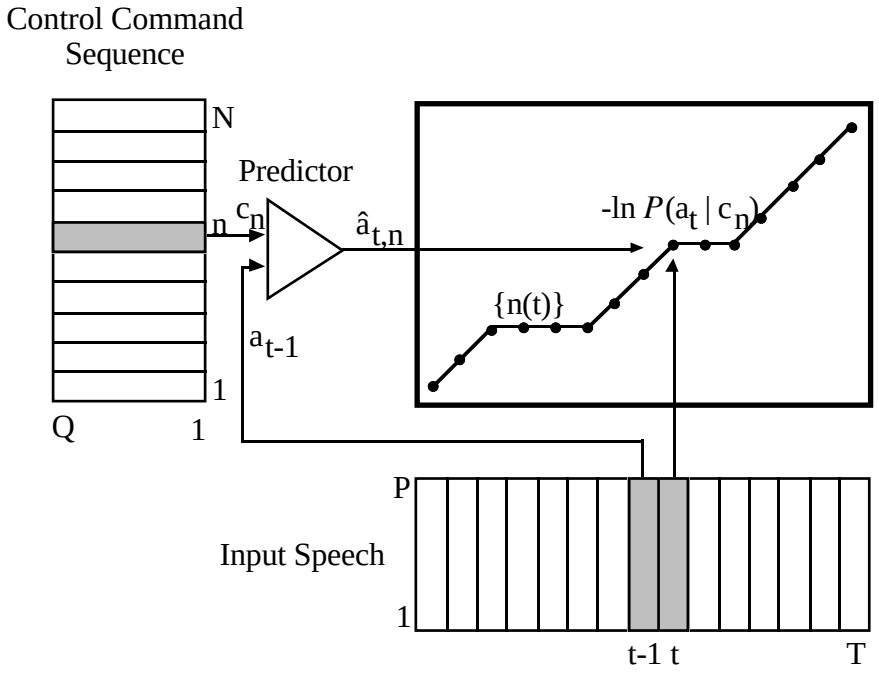

Fig. 5: Hidden Control Neural Network architecture

Figure 5 shows the behaviour of the HCNN in the recognition phase, for each competing word to be recognized a time alignment algorithm computes the best instant firing of the N consecutive control signals associated with each of the words. A score is then computed, in short this score is linked to the prediction power error of the predictive MLP given the control sequence. The "best" score, i.e. the best control sequence traces back to the associate word which is the recognized word.

(Iso, 1993) makes the assumption that the prediction error is distributed following a gaussian, the probability of producing the speech feature vector $\boldsymbol{a}_t$ given the control command $\boldsymbol{c}_n$ is

$$-\ln P\left(\boldsymbol{a}_t \left| \boldsymbol{c}_n \right.\right) = \frac{1}{2}\left(\boldsymbol{a}_t - \hat{\boldsymbol{a}}_{t,n}\right)^T \Sigma_n^{-1}\left(\boldsymbol{a}_t - \hat{\boldsymbol{a}}_{t,n}\right) + \frac{1}{2}\ln(2\pi)^P \left|\Sigma_n\right| \qquad (4)$$

where Σ_n is the $P \times P$ covariance matrix for each control command $\boldsymbol{c}_n$. With this formulation, the probability of an acoustic observation given the control command sequence associated with the m^{th} word of the vocabulary is

$$P\left(\boldsymbol{a}_i, \cdots, \boldsymbol{a}_T \left| \boldsymbol{c}_1^m, \cdots, \boldsymbol{c}_{N(m)}^m \right.\right) = \prod_{t=1}^{T} P\left(\boldsymbol{a}_t \left| \boldsymbol{c}_{n(t)}^m \right.\right). \qquad (5)$$

The control command sequence which maximises the probability of the acoustic observation defines the most probable word (whose index is m^*) given the model parameter:

$$P\left(\boldsymbol{a}_i, \cdots, \boldsymbol{a}_T \left| \boldsymbol{c}_1^{m^*}, \cdots, \boldsymbol{c}_{N(m)}^{m^*} \right.\right) = \max_m \left\{ \max_{\{n(t)\}} \prod_{t=1}^{T} P\left(\boldsymbol{a}_t \left| \boldsymbol{c}_{n(t)}^m \right.\right) \right\} \qquad (6)$$

In the recognition phase, the function $n(t)$ determines the time-alignement between input speech feature vector sequence and control command sequences. The optimal time-alignment $\{n(t)\}$ which gives the maximum probability is determined by dynamic programming.

The MLP weights, the control command and the covariance matrix have to be found during the training phase. Two criteria can be used for this optimization problem. The first one tempts to maximise the average of the logarithmic probabilities (6) for all the training utterances (maximum likelihood criterion). The second one is a discriminative criterion. For more details, the reader is referred to (Iso, 1993), (Levin, 1993) and (Martinelli, 1994).

(Iso, 1993) reports speaker-dependent English spoken letter recognition experiments in which the HCNN configuration exhibited a recognition accuracy rate of 98.8 %. The vocabulary consisted of the 26 letters of the English alphabet. The database contained about 5000 connected letters.

6. Combination of hidden Markov model and Connectionist Probability Estimators

This section refers to a hybrid system combining HMMs and ANNs in which we will give an overview of the capabilities of this mixed approach to improve the performance of pure HMMs methods. Until now, it seems that this mixed approach outperforms both pure HMMs methods and pure ANNs methods. Let us first introduce the underlying mechanism of HMMs and let us review their major weaknesses.

HMMs are widely used for automatic speech recognition. Essentially, a HMM is a stochastic automaton with a stochastic output process attached to each state (Fig. 6). Thus we have two concurrent stochastic processes : an underlying (hidden) Markov process modeling the temporal structure of speech and a set of state output processes modeling the stationary character of the speech signal. For more detailed information about HMMs used in speech recognition, several texts can be recommended such as (Rabiner, 1989) and (Waibel, 1990).

It is possible to use HMMs to represent any unit of speech. For small vocabulary recognition systems, HMMs can be used to directly model words. For large vocabularies, HMMs are defined on subword units. In this case, word and sentence knowledge can be incorporated by representing each word as a network of subword models. A search through all acceptable sentences will spot the pronounced utterance.

The modeling of speech with HMMs assumes that the signal is *piecewise stationary*, that is, HMMs model an utterance as a succession of discrete stationary states, with instantaneous transitions between these states.

HMMs inherently incorporate the sequential and statistical character of the speech signal and they have proved their efficiency in speech recognition. However, standard HMMs still suffer from several weaknesses, namely:

- a priori choice of a model topology, e.g. a number of states is imposed for each subword model
- a priori choice of statistical distributions for the emission probabilities $p(\mathbf{x}|q_i)$ associated with each states: multivariate Gaussian density, mixture of multivariate Gaussian densities,...
- first order Markov assumption, i.e., the probability of being in a given state at time t only depends on the state at time $t-1$

- poor discrimination due to the training algorithm which maximises likelihoods instead of *a posteriori* probabilities [1]

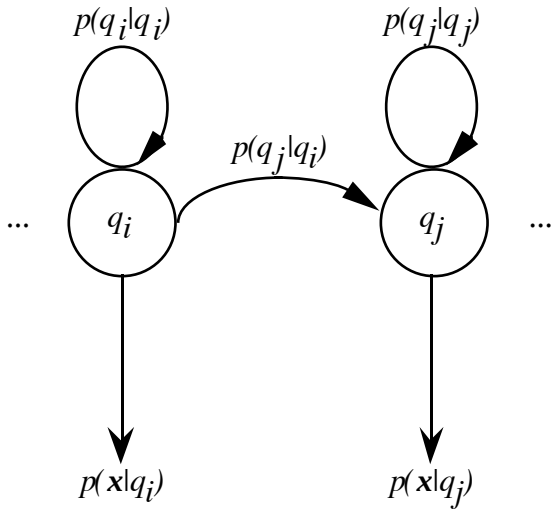

Fig.6: Example of HMM used for modeling speech signal. The i^{th} state is defined by the variable q_i. $p(q_i|q_j)$ refers to the probability of transition from state q_i to state q_j. $p(\mathbf{x}|q_i)$ refers to the emission probability of $\mathbf{x}$ given a state q_i.

As explained in the previous section dealing with MLP, ANNs can be used to solve difficult problems such as vision, pattern and speech recognition. The major strength of ANNs is in the fact that there is no need for any particular assumptions about statistical distributions and independence of input features, and also that ANNs can be trained in such a way that the network exhibits discriminant properties. For speech recognition, the major weakness of ANNs is their inability to deal easily with the time sequential nature of speech.

Recent research (Bourlard, 1990) tried to take advantage of HMMs, i.e. incorporate the time variability and sequential nature of the speech signal, and of MLPs, i.e. classify without making any particular assumptions about the statistical distribution of speech features, leading to a hybrid system HMM/MLP. With this approach MLPs are used to compute the emission probabilities associated with each state of the HMM.

In (Bourlard, 1990) it has been proved that if each output unit of a MLP is associated with a particular state q_k of the set of states $Q=\{q1,q2,...,qK\}$ on which the Markov Models are defined, it is possible to train the MLP to generate *a posteriori* probabilities like $p(q_k|x_n)$ when $\boldsymbol{x}_n$, a particular acoustic vector, is provided to its input. The training data set of the MLP consists of a labelled sequence of N acoustic vectors $\{\boldsymbol{x}1,\boldsymbol{x}2,...,\boldsymbol{x}N\}$. At time n, the input pattern of the MLP is the acoustic vector $\boldsymbol{x}_n$ associated with a particular state. The training of the

[1] General discussion about Maximum Likelihood Estimate criterion (MLE) and Maximum a Posteriori (MAP) criterion can be found in [Bourlard & Wellekens IEEE 1990].

MLP parameters is based on the minimization of the following Mean Square criterion,

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}\left[g_k(\boldsymbol{x}_n) - d_k(\boldsymbol{x}_n)\right]^2 \tag{7}$$

where $g_k(\boldsymbol{x}_n)$ represents the k^{th} output value given $\boldsymbol{x}_n$ at the input and $d_k(\boldsymbol{x}_n)$ is the associated target value and is equal to δ_{kl} (Kronecker Delta) if the input is known to belong to class q_l ("*1-from-K*" training).

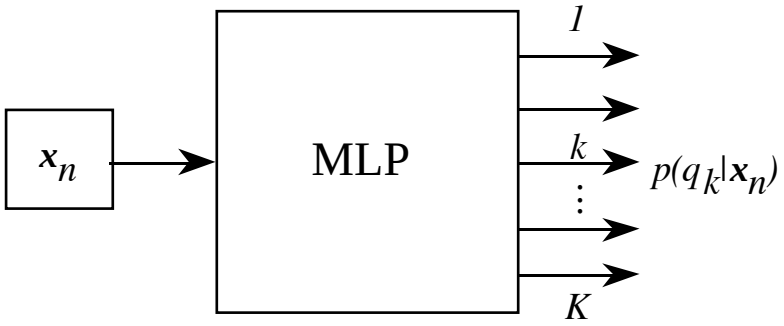

Fig.7: MLP used as a posteriori probability estimators

It has been shown in (Bourlard, 1990) that if the MLP contains enough parameters and if the global minimum of E (7) is reached, the output values of the MLP are the estimates of the *a posteriori* probability density functions (Fig. 7) which are optimal for classification:

$$g_k^{opt}(\boldsymbol{x}_n) = p(q_k|\boldsymbol{x}_n) \tag{8}$$

Emission probabilities for HMMs can be found with Bayes' rule:

$$p(\boldsymbol{x}_n|q_k) = \frac{p(q_k|\boldsymbol{x}_n)P(\boldsymbol{x}_n)}{p(q_k)} \tag{9}$$

where $p(\boldsymbol{x}_n)$ is constant for all the states and where the prior $p(q_k)$ can be easily determined counting the number of times a feature is associated with the state k.

The training procedure of the hybrid system is quite similar to the one of the standard HMMs. Given a parameter set for the HMM (transition probabilities) and the MLP (weights), the best state sequence can be found with a Viterbi algorithm, leading to a labelling of the feature sequence. This labelling allows the MLP to be trained with a standard backpropagation updating the weights and to compute new

values for transition probabilities. This procedure is repeated iteratively until an optimum is reached.

This training procedure has two major advantages in comparison with the maximum likelihood estimation. First, it is based on *a posteriori* probabilities approximation which are discriminant by nature. Second, the MLP gives a model for the emission probability density function without making any assumptions about the distribution of features among the states.

Another advantage of the hybrid system is that these results hold if the MLP is fed with a larger feature window taking more context into account. This procedure allows the time correlation between the successive acoustic vector in the recognition process to be taken into account

Results reported in (Bourlard, 1993) (Renals, 1994) shown that the hybrid HMM/MLP system outperforms standard HMMs system. Other connectionist approaches such as Radial Basis Function Neural Networks, Recurrent Neural Networks, predictive Neural Networks, ... , could be used as well as probability estimators in the framework of hybrid HMM/connectionist systems (Renals, 1994). There is no evidence of the superiority of these alternative approaches in front of the HMM/MLP system. Recently, (Dugast, 1994) have proposed a similar hybrid system combining HMMs and Time Delay Neural Networks (TDNN) used as probability estimators. This approach seems also to outperform standard HMMs.

7. Conclusion

This paper reviews some of the neural techniques used in speech recognition. The strenghness of pure neural techniques is their discriminative properties, however it is now evident that today's neural networks architectures are only able to solve efficiently restricted speech recognition problems such as small vocabulary recognition (e.g. digit recognition). The main problem comes from the fact that pure connectionist methods are not suited to deal with the temporal and spectral variability intra and inter-speakers. In fact ANN methods are really efficient if a segmentation of the speech signal is available. So far the HMM approach appears as the best approach to the segmentation problem, i.e. the best matched method to the problem of speaker variability. The discriminative properties of ANNs make them good candidates to improve the performance of HMMs. It seems that today's efforts in state-of-the art of speech recognizers is mainly directed towards this hybrid approach.

References

1. R. Bahl *et al.* "*Large Vocabulary Natural Language Continuous Speech Recognition*", Proceedings of the IEEE ICASSP-89, pp. 465-468, May 1989.

2. Bourlard and C. J. Wellekens, "Links between Markov models and multi-layer perceptrons", IEEE Trans. Patt. Anal. Machine Intell., Vol. 12, pp. 1167-1178, 1990.
3. Bourlard and N. Morgan, "Continuous Speech Recognition by Connectionist Statistical Methods", IEEE Transactions on Neural Networks, Vol. 4, No. 6, November 1993, pp. 893-909
4. Dugast, L. Devillers and X. Aubert, "Combining TDNN and HMM in a Hybrid System for Improved Continuous-Speech Recognition", IEEE Transactions on Speech and Audio Processing, January 1994, Vol. 2, No. 1, Part II, pp. 217-223
5. Fontaine, J. Hennebert and H. Leich, "Influence of Vector Quantization on Isolated Word Recognition", to be published in Proceedings Eusipco September 1994, Edinburgh
6. Gersho and R. Gray "Vector Quantization and Signal Compression", Kluwer Academic Publishers, 1992
7. Huang,, A. Kuh, "*A combined Self-Organizing Feature Map and Multilayer Perceptron for Isolated Word Recognition*", IEEE Trans. on Signal Processing, Vol. 40, No. 11, Nov. 1992, pp. 2651-2657.
8. IEEE TSAP, "Special Issue On Neural Networks for Speech", IEEE Transactions on Speech and Audio Processing, January 1994, Vol. 2, No. 1
9. Iso and T. Watanabe, "*Speaker-independent word recognition using a neural prediction model*", Proc. ICASSP, Albuquerque, NM, April 1990
10. Iso and T, Watanabe, "*Large Vocabulary Speech Recognition Using Neural Prediction Model*", IEEE ICASSP-91, Toronto, pp. 57-60.
11. Iso and T, Watanabe, "*Speech Recognition Using Demi-Syllabe Neural Prediction Model*", Advances in Neural Information Processing Systems 3, Edited by R. Lippmann, J. E. Moody, D. S. Touretzky, Morgan Kaufmann Publishers, 1991, Inc., pp. 227-233.
12. Ken-ichi Iso, "*Speech Recognition using Dynamical Model of Speech Production*", IEEE ICASSP, Minneapolis, 1993, pp. II-283-II-286.
13. Kohonen, "The Self Organasing Map", Proceedings of the IEEE, September 1992
14. J. Lang, A. Waibel, G. E. Hinton, "*A Time-Delay Neural Network Architecture for Isolated Word Recognition*", Artificial Neural Networks, Paradigms, applications and Hardware Implementations, Edited by E. Sanchez-Sinencio and Clifford Lau, IEEE Press, 1992, pp. 388-408.
15. Lennig, *et al*. "*Automated Bilingual Directory Assistance Trial in Bell Canada*", Proceedings of the first IEEE workshop on Interactive Voice Technology for Telecommunications Applications, Piscataway, N. J., Oct. 1992
16. Lennig, *et al*. "*Flexible Vocabulary Recognition of Speech*", Proc. ICSLp-92, pp. 93-96, Banff, Canada, Oct. 1992
17. Levin, "*Hidden Control Neural Architecture Modeling of Nonlinear Time Varying Systems and its Application*", IEEE Trans. on Neural Networks, Vol. 4, No. 1, January 1993, pp.109-116.

18. Lippmann, "Review of Neural Networks for Speech Recognition", Neural Computation, 1(1):1-38, 1989
19. Martinelli, "*Hidden Control Neural Network*", IEEE Trans. on Circuits and Systems-II: Analog and Signal Processing, Vol 41, No. 3, March 1994, pp. 245-247.
20. R. Rabiner, "A Tutorial on Hidden Markov Models and ted Applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, No. 2, February 1989, pp. 257-286
21. Renals *et al*., "Connectionist Probability Estimators in HMM Speech Recognition", IEEE Transactions on Speech and Audio Processing, January 1994, Vol. 2, No. 1, Part II, pp. 161-174
22. B. Roe and Jay G. Wilpon, "*Whither Speech Recognition: The Next 25 Years*", IEEE Communications Magazine, Nov. 1993, Vol. 31, No. 11, pp. 54-62
23. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang, "*Phoneme Recognition Using Time-Delay Neural Networks*", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 37, No. 3, March 1989, pp. 329-339.
24. Waibel and K-F. Lee, editors. "Readings in Speech Recognition", Vol. 1, Morgan Kaufman Publisher, Inc., San Mateo, California, 1990
25. Weisfreid, M, V. Wickerhauser, "*Signal Processing via Fast Malvar Wavelet Transform Algorithm*", GRETSI, 1993, pp. 379-382.
26. Z.Zhao and C. G. Rowden, "Use of Kohonen Self-Organising Feature Maps for HMM Parameter Smoothing in Speech Recognition", IEE Proceedings-F, Vol. 139, No. 6, December 1992