speech recognition system. *Proceedings of the 1990 International Conference on Spoken Language Processing*, Kobe, Japan, pp. 1033-1036.

Rumelhart D.E., Hinton G.E. and Williams R.J. 1986. Learning internal representation by error propagation. *Parallel Distributed Processing* vol. 1, MIT Press, pp. 318-362.

Seneff S. and Zue V. 1988. Transcription and alignment of the TIMIT database, in J.S. Garofolo ed. *Getting started with the DARPA TIMIT CD-ROM : An acoustic phonetic continous speech database.* National Inst. of Stand. and Tech. (NIST), Gaithersburgh, MD.

Stevens K.N. (1975) The potential role of properties detectors in the perception of consonants, in *Auditory analysis and perception of speech* G. Fant and M.A. Tatham ed., Academic Press, London, pp. 303-330.
Stewart G.W. 1973. Introduction to matrix computations. *Academic Press.*

Waibel A., Hanazawa T. and Shikano K. 1988. Phoneme recognition: neural networks *vs* hidden Markov models. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, (ICASSP88), New-York NY, April 88, pp.107-110.

Zue V., Seneff S. and Glass J. 1990. Speech database development: TIMIT and beyond. *Speech Communication*, Vol. 9. No. 4. august 1990, pp. 351-356.

multilayer perceptrons with hidden Markov models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, NM, April 90, pp. 413-416.

Nadas, A., Nahamoo, D. and Picheny, M.A. 1988. On a model-robust training method for speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-36, no. 9., pp. 1432-1436.

Pearlmutter B.A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation* vol. 1, no. 2, pp. 263-269.
Poggio T. and Girosi F. 1989. A theory of networks for approximation and learning. MIT AI Memo No. 1140.

Rabiner L.R. and Levinson S.E. 1985. A speaker-independent, syntax-directed, connected word recognition system based on hidden Markov models and level building. *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-33, no. 3, pp.561-573.

Rabiner L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, vol. 77, no. 2, Feb. 89., pp. 257-285.

Robinson, T. and Fallside, F. 1990. Phoneme recognition from the TIMIT database using recurrent error propagation networks. Engineering Dept., Cambridge University, CUED/F-INFENG/TR 42.

Robinson T., Holdsworth J., Patterson R. and Fallside F. (1990). A comparison of preprocessors for the Cambridge recurrent error propagation network

Huang, X.D. and Jack, M.A. (1989). Semi-continuous hidden Markov models for speech signals. *Readings in Speech Recognition*, Waibel & Lee eds., Academic Press.

Irino T. and Kawahara H. (1990). A method for designing neural network multivariate analysis: application to speaker-independent vowel recognition. *Neural Computation 2*, 3, pp. 386-397.

Kolen, J.F. and Pollack, J.B. 1990. Backpropagation is sensitive to initial conditions. *Complex Systems*, vol. 4, no. 3, pp. 269-280.

Lee K.F., and Hon H.-W. 1989. Speaker-Independent Phone Recognition Using Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. ASSP-37, pp. 1641-1648.

Levin, E. 1990. Word recognition using hidden control neural architecture. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, NM, April 90, pp. 433-436.

Lippman R.P. 1989. Review of neural networks for speech recognition. *Neural Computation*, vol. 1, no. 1, pp. 1-38.

Makhoul, J. and El-Jaroudi, A. 1990. Posterior Probability Estimation with Neural Nets. Presented at the *Neural Networks for Computing* conference at Snowbird, April 1990.

Morgan, N. and Bourlard, H. 1990. Continuous speech recognition using

Bourlard, H. and Wellekens, C.J. 1988. Links between Markov models and multi-layer perceptrons. *Advances in Neural Information Processing Systems 1*, (ed. D.S. Touretzky) Morgan Kauffman Publ., pp. 502-510.

Bridle, J.S. 1990. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in Neural Information Processing Systems 2*, (ed. D.S. Touretzky) Morgan Kauffman Publ., pp. 211-217.

Bridle J.S. and Cox S.J. 1991. RECNORM: simultaneous normalisation and classification applied to speech recognition. To appear in *Advances in Neural Information Processing Systems 3*, (ed. D.S. Touretzky) Morgan Kauffman Publ.

Brown P.F. 1987. The acoustic-modeling problem in automatic speech recognition. PhD Thesis, Dept. of Computer Science, Carnegie-Mellon University.

Cohen J.R. 1989. Application of an auditory model to speech recognition. Journal of the Acoustical Society of America, 85 (6), pp. 2623-2629.

Cosi P., Bengio Y. and De Mori R. 1990. Phonetically-based multi-layered networks for acoustic property extraction and automatic speech recognition. *Speech Communication* special issue on neurospeech, vol. 9, no. 1, pp. 15-30.

Franzini, M., Lee K-F. and Waibel A. 1990. Connectionist Viterbi training: a new hybrid method for continuous speech recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, NM, April 90, pp. 425-428.

explore further the possibilities of such a hybrid system, and extend it to the recognition of all American-English phonemes.

References

Bahl L.R., Jelinek F., and Mercer R.L. 1983. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. ASSP-5, no. 2, pp. 179-190.

Bellagarda J.R., and Nahamoo D. 1989. Tied Mixture Continuous Parameter Models for Large Vocabulary Isolated Speech Recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Glasgow, Scotland, May 89, pp. 13-16.

Bengio Y., Cardin R., De Mori R. and Normandin Y. 1990. A hybrid coder for hidden Markov models using a recurrent neural network. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, NM, April 90, pp. 537-540.

Bengio Y., Cardin R., De Mori R. and Merlo E. 1989. Programmable execution of multi-layered networks for automatic speech recognition. *Communications of the Association for Computing Machinery*, vol. 32, no. 2, Feb. 89, pp. 195-199.

Bottou L., Fogelman Soulie F., Blanchet P. and Lienard J.S. 1990. Speaker-independent isolated digit recognition: multilayer perceptrons *vs* dynamic time warping. *Neural Networks* 3 (4) : 453-465.

rate (100% - %deletions - %substitutions) for the 8 units with the hybrid system after two training iterations is 90% on a total of 7214 phonemes, and its accuracy (100% - %deletions - %substitutions - %insertions) is 86%. Note that this is a significative improvement over the performance obtained with a HMM trained without global optimization (86% recognition and 75% accuracy). The ANNs alone yielded 85% recognition but only 53% accuracy, because of the high number of insertions (32%), mostly due to short plosive segments. The ANNs perform a good classification but have a noisy output with many insertions. The HMM eliminates most of these insertions because of its better duration and temporal modeling.

In addition to providing a good temporal model the HMM provides more appropriate target values for the outputs of the ANN. With these target outputs for the ANN, the hybrid system significantly improves its performance. It is interesting to note that the effect of equation 16 is to generate a gradient that tends to bring the output of the ANN closer to the means of the normal densities which are close to the ANN output as well as consistent with the the training string.

8 Conclusion

A system has been proposed to combine the advantages of ANNs and HMMs for speech recognition. The parameters of the ANN and HMM subsystems can influence each other. We showed how to perform a global optimization of such a system by driving the network gradient descent with parameters computed in the HMM. We have seen how such a hybrid system could be extended to integrate multiple ANN modules, which may be recurrent.
Encouraged by the results of the above-described initial experiments, we will

short intervals of frequency and time[10], 7 slope coefficients describing the frequency derivative of the spectrum, the total energy and the voicing energy (in the 60-500 Hz band) and their time derivatives, and a measure of distance (dot product) between neighbouring spectral frames. New input parameters are fed to the networks every 5 msec. ANN2 has time-delay links, while ANN3 has time-delay links between the input nodes and the hidden layer, and recurrent links between some of the hidden nodes and the output nodes. ANN1 computes 8 features for the continuous densities HMM. There is one HMM per unit. Each of the 11 unit models [11] had 14 states, 28 transitions, 3 self loops, without explicitly modeling the state duration. Each HMM has tied distributions with 3 basic different distributions characterizing the beginning, middle and final part of a segment modeled by the unit. Each of these distributions is modeled by a gaussian mixture with 5 densities. The covariance matrix is assumed to be diagonal because the parameters are principal components.

In order to assess the value of the proposed approach as well as the improvement brought by the HMM as a post-processor for time alignment, the performance of the hybrid system was evaluated and compared with that of a simple post-processor applied to the outputs of the ANNs. The simple post-processor assigns a symbol to each output frame of the ANNs by comparing the target output vectors with actual output vectors. It then smooths the resulting string to remove very short segments and merges consecutive segments that have the same symbol.

The comparative results are summarized in Table I. The overall recognition

[10]This parameter is inspired by studies in Acoustic-Phonetics (see Stevens 1975).

[11]In order to improve its modeling, the rejection class was composed out of four models: nasals, fricatives, non-nasal sonorants, and silence. The recognition results are obtained by merging these four subclasses, such that the total number of classes to recognize is eight.

Table 1: Comparative Results

	% rec	% ins	% del	% subs	% acc
ANNs	85	32	0.04	15	53
ANNs+HMM	86	11	0.7	13	75
ANNs+HMM+global opt.	90	3.8	1.4	9	86

sonants or may be included into other unit segments by a recognition system.

The experimental system is based on the scheme shown in figure 2 with two ANNs initially trained to perform plosive recognition (ANN3) and broad classification (ANN2) respectively. Another network (ANN1) computed the principal components of the concatenated output vectors of ANN2 and ANN3. The combined network (ANN1+ANN2+ANN3) had 23578 weights.
The broad classification net (ANN2) has five outputs corresponding to five broad categories[8]. The twelve input nodes to ANN2 are the energies of five band-pass filters in the time domain covering the range up to 7 kHz, the signal total energy, and their six time derivatives. The plosive recognition net (ANN3) has sixteen outputs corresponding to place and manner and degree of voicing, with different firing output nodes depending on the right context[9]. The 74 inputs to ANN3 are the outputs of 32 Bark-scaled triangular filters computed from the short-time Fast Fourier Transform of the windowed signal, 30 property detectors approximating a second order derivative over

[8]non-nasal sonorant, nasal, plosive, fricative, and silence.

[9]Each of the four different places of articulation (labial, alveolar, velar, and flapped alveolar) corresponds to two different firing nodes, depending whether the following phoneme has a front or non-front place of articulation. The remaining eight nodes are labeled: unvoiced plosive, voiced plosive, vocalic front,vocalic non-front, liquid, fricative, nasal, silence.

unchanged. The gradient of the optimization criteria is still passed from the HMM to the ANN but HMM parameters remain unchanged. The ANN is trained in such a way that different speakers tend to produce similar output parameters for the same speech unit (See (Bridle & Cox 91) for a similar speaker adaptation mechanism).

7 Experimental Results

A preliminary experiment has been performed using a prototype system based on the integration of ANNs with HMMs. The task is the recognition of plosive sounds in every context and pronounced by a large speaker population. The TIMIT continuous speech database (Seneff, Zue & Glass 90; Seneff & Zue 88) has been used for this purpose. SI and SX sentences from regions 2, 3 and 6 were used, with 1080 training sentences and 224 test sentences, 135 training speakers and 28 test speakers [6].
The goal of these experiments was to evaluate the potential performance of the system by simply allowing the HMM to be a sophisticated post-processor for the ANNs, thus providing a benchmark for further and more complex experiments. The following units have been considered:

/p/,/t/,/k/,/b/,/d/,/g/,/dx/[7], /all other phonemes/

Speaker-independent recognition of plosive sounds in continuous speech is a particularly difficult task because these sounds are made of short and non-stationary events that are often confused with other acoustically similar con-

[6] The training speakers were those with initial between "a" and "r" inclusively; the remaining speakers were used for test.

[7] The flapped alveolar plosive */dx/* is considered as a distinct phoneme in the TIMIT database.

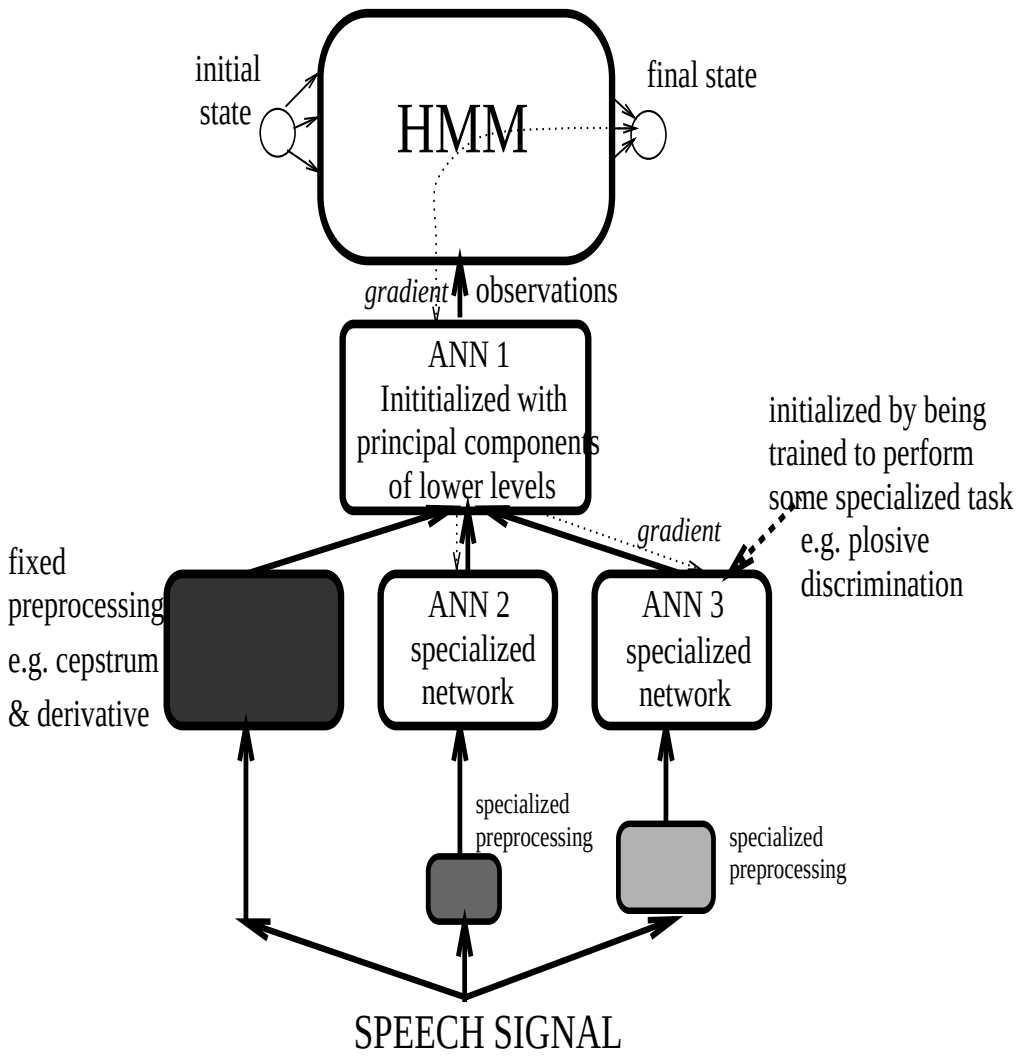

Figure 2: *Extension of the ANN/HMM hybrid to a hierarchy of modules, with three levels.*

fusion between voiced and unvoiced plosives) and suitable time and frequency representations can be conceived for the input of these networks (see for example Bengio *et al* 89). The outlined hierarchical system can be trained as described in Section 5. The gradient of the optimization criterion can be back-propagated from level 2 to level 1.

Networks at level 1 can be initially trained separately as classifiers of certain phoneme sets. After training, the output layer can be removed so that the outputs of the last hidden layer produce acoustic parameters for the input of the level 2 network. An alternative would be to train networks explicitly as detetectors of phonetic properties such as place and manner of articulation. After level 1 networks are trained, the level 2 network can be initialized so that it computes the principal components of the output of the level 1 modules, using the method described in Section 5.

The just introduced algorithms can be used even if ANNs are recurrent. Networks of this type can capture short-time temporal regularities with less parameters than a feedforward network with time delays (see for example (Bengio, De Mori & Gori 89; Pearlmutter 89)). In general, the output of a recurrent network is smoother than the output of a feedforward network scanning the input data. In that case, the HMM temporal resolution could be reduced, thus reducing the number of necessary parameters.

ANNs trained as proposed can perform speaker adaptation. For such a purpose, the network has to map speaker-dependent observations into a speaker-independent representation. This objective could be obtained by first training a hybrid system as previously described for multiple speakers, and in a second step, adapting *only the ANN parameters* with known sentences from the new speaker. In such a system, the ANN adaptation represents a tuning of the feature space to the new speaker, whereas the temporal model remains

tral frames of the input signal.

Initially the network settled as described performs a decorrelation of the input data over the spectra and a few frames of speech. Indeed, we know that there is a degree of redundancy in each spectrum as well as in a set of adjacent spectra (the spectra generally change smoothly over time). The network reduces the dimensionality of the input data so that the process described by the network output can be better modeled by an HMM trained with limited sample data. The advantage of the proposed scheme over a simple fixed linear transformation that computes the principal components or the linear discriminants in a preprocessing phase is that the function computed by the network can evolve with further training in order to optimize the learning criterion, based on the reduction of the rate of recognition errors.

6 Extensions

Rather than having a single ANN that computes the vector Y of parameters, we can have a hierarchy of networks, as shown in figure 2. Such an architecture is built on three levels. Level 3 contains the HMMs. Level 2 is made of a single ANN that acts as an integrator of parameters generated by more specialized ANNs or computed by other algorithms, like the ones of an ear model (see (Cohen 89) or (Cosi, Bengio & De Mori 89)). Networks at level 1 are specialized to compute parameters that are particularly useful for characterizing certain acoustic situations. As an example, a set of parameters may represent the acoustic cues useful for distinguishing between sonorant and non-sonorant sounds. A network that computes these parameters may have some special acoustic features at the input that are different from those feeding other networks. Networks at level 1 can be added when systematic recognition errors have been noticed (for example a frequent con-

in a local minimum (see (Irino & Kawahara 90)) [5]. Another advantage of a deterministic initialization is that parameter estimation does not depend on random initial conditions. Different initial conditions may give rise to different networks, with very different performances (Kolen & Pollack 90). A useful preprocessing step that the ANN can perform consists in the computation of the principal components of the input data. An interesting alternative is the computation of linear discriminants (Brown 87). In both cases, network outputs represent statistically independent variables, such that it is reasonable to model their distributions with a mixture of gaussians with diagonal covariance matrices.

The computation of the principal components can be split into two parts by decomposing the principal components or the linear discriminants matrix A into the product of two matrices, *e.g.*, with an LU decomposition (Stewart 73):

$$A = LU \tag{17}$$

This decomposition leads to a 2-layer network with inputs connected to the hidden layer and the hidden layer connected to the output layer. The hidden layer weight matrix is initialized with ϵL where ϵ is a small positive number, and the output layer weight matrix is initialized with U. By multiplying L by a small scalar we make sure that initially the weights of the connections to hidden nodes are very small so that the input-output transformation of each hidden node is almost linear. If the hidden units compute symmetric sigmoids, with output ranging in the interval [-1,1] rather than [0,1], then the ANN output Y can be approximately expressed as:

$$Y \approx \epsilon\, L\, U\, X \; = \epsilon\, A\, X \tag{18}$$

where X is the network input, which could be fed by a small number of spec-

[5] A similar situation occurs for Radial Basis Function networks, which can be trained very fast if the hidden layer is initialized based on a cluster analysis of the input data.

simultaneously the HMM parameters a_{ij}, μ_{kl}, Σ_k, Z_k and the ANN weights w_{mn}. This minimum is of course constrained by the number of free parameters and is not guaranteed to be a global minimum. A similar situation occurs when training ordinary HMMs or multilayer networks.

A practical problem encountered when an HMM is simultaneously trained with an ANN, is that algorithms for training the HMM (*e.g.*, the forward-backward or the Viterbi algorithms (Rabiner 89)) converge in fewer iterations than the algorithms for training the multilayer network (*e.g.*, backpropagation (Rumelhart *et al* 86)).

In view of this problem, we propose the following procedure:

1. Initialize the ANN in such as way that it performs a probably useful transformation, *e.g.*, principal component analysis of the training data.

2. Train the HMM with MLE.

3. Train the hybrid system with MMIE, to produce a globally optimized discriminant system [4].

5 Initialization of the ANN Weights

There are several ways for initializing the weights of the ANN. If initialization is with random weights, training is very slow because ANN training is in general slower than HMM training, and the HMM parameters depend on the transformation performed by the ANN. Another possibility consists in a deterministic initialization that would decrease the chance of getting stuck

[4] To speed up training, input patterns that produce errors can be presented more frequently (Normandin & Cardin 90). A noisy approximation of the gradient can be used to adapt the network parameters after each pattern or group of pattern (on-line learning) rather than after the presentation of the whole training set (batch learning). It is well known (Bottou *et al* 90) that on-line learning, which uses a noisy gradient, yields much faster convergence of multilayer networks than batch learning.

In the case of isolated units modeling, for states i that are in a unit model ω:

$$\frac{\partial H_{isolated}}{\partial b_{i,t}} = \frac{(\delta_{c\omega} - H_c)}{\sum_\omega L_\omega} \tag{14}$$

For continuous speech, we have the following derivative:

$$\frac{\partial H_{continuous}}{\partial b_{i,t}} = \alpha_{F_\rho,T} \frac{\partial \alpha_{F_\tau,T}}{\partial b_{i,t}} - \alpha_{F_\tau,T} \frac{\partial \alpha_{F_\rho,T}}{\partial b_{i,t}} \tag{15}$$

In general, for every optimization criterion C that can be expressed as a differentiable function of the likelihood L, it is possible to compute $\frac{\partial C}{\partial L}$.

By differentiating equation (11), $\frac{\partial b_{i,t}}{\partial Y_{jt}}$ can be expressed as follows:

$$\frac{\partial b_{i,t}}{\partial Y_{jt}} = \sum_k \frac{Z_k}{((2\pi)^n \mid \Sigma_k \mid)^{1/2}} (\sum_l d_{k,lj}(\mu_{kl} - Y_{lt})) \exp(-\frac{1}{2}(Y_t - \mu_k)\Sigma_k^{-1}(Y_t - \mu_k)^T) \tag{16}$$

where $d_{k,lj}$ is the element (l,j) of the inverse of the covariance matrix (Σ^{-1}) for the k^{th} gaussian distribution and μ_{kl} is the l^{th} element of the k^{th} gaussian mean vector μ_k.

Then, following Bridle (1990), it is possible to compute using (2)

$$\begin{aligned}
\frac{\partial \alpha_{F_{model},T}}{\partial b_{i,t}} &= \frac{\partial \alpha_{F_{model},T}}{\partial \alpha_{i,t}} \frac{\partial \alpha_{i,t}}{\partial b_{i,t}} \\
&= (\sum_j \frac{\partial \alpha_{j,t+1}}{\partial \alpha_{i,t}} \frac{\partial L_{model}}{\partial \alpha_{j,t+1}})(\sum_j a_{ji}\alpha_{j,t-1}) \\
&= (\sum_j b_{j,t+1} a_{ji} \frac{\partial \alpha_{F_{model},T}}{\partial \alpha_{j,t+1}})(\sum_j a_{ji}\alpha_{j,t-1}) \\
&= \beta_{i,t} \frac{\alpha_{i,t}}{b_{i,t}}
\end{aligned}$$

for any hidden Markov *model*, where *model* is ω for isolated units modeling, or ρ (recognition model) or τ (training model) for continous speech modeling. A consequence of the derivation of the gradient of the optimization criterion with respect to the network outputs ($\frac{\partial C}{\partial Y_{jt}}$) is that a minimum of the criterion function can be attained with the proposed hybrid model, by estimating

Jack 89). A derivative of the cost function with respect to $b_{i,t}$ can be computed and used for estimating the parameters of the ANN as it will be shown in the next section.

4 Estimation of ANN parameters

As the optimization criterion C depends on the parameters Y_1^T computed by the ANN, it is possible to express C as a function of them and derive the following equation, using the chain rule:

$$\frac{\partial C}{\partial Y_{jt}} = \sum_i \frac{\partial C}{\partial b_{i,t}} \frac{\partial b_{i,t}}{\partial Y_{jt}} \tag{11}$$

for all the ANN output units j (Y_{jt} is the j^{th} element of the network output vector Y_t). The negative of this gradient can be used with backpropagation [2] to estimate the ANN weights w_{mn} [3].

In the case of MLE, the derivative of C_{MLE} with respect to $b_{i,t}$ is simply

$$\frac{\partial C_{MLE}}{\partial b_{i,t}} = \frac{\partial L_{model}}{\partial b_{i,t}} = \frac{\partial \alpha_{F_{model},T}}{\partial b_{i,t}} \tag{12}$$

where *model* is the training model (the correct word model, in the case of isolated units modeling).

In the case of MMIE, the gradient of the optimization criterion C_{MMIE} with respect to the observation probabilities $b_{i,t}$ can be expressed as follows:

$$\frac{\partial C}{\partial b_{i,t}} = \frac{1}{H} \frac{\partial H}{\partial b_{i,t}} \tag{13}$$

where H is defined as in equations 7 and 9 for isolated and continuous speech modeling, respectively.

[2] It replaces the usual $\partial E_p / \partial Y_{jt} = (Y_{jt} - target_{jt})$ for output units, for a particular pattern p (here a sentence), as used in (Rumelhart *et al* 86), where $target_{jt}$ would be the desired output at time t for unit j.

[3] w_{mn} is the weight of the connection from unit n to unit m

Assuming equal prior probabilities for each model, maximizing C_{MMIE} as in equation 6 also maximizes the mutual information I. For convenience, let us define

$$H_{isolated} = \frac{L_c}{\sum_\omega L_\omega} \tag{7}$$

For continuous speech, we assume that there is a single HMM built by concatenating unit models. During *training*, we consider a constrained model τ that is made of the concatenation of the units that form the training sentence. On the other hand, during *recognition* (almost) all the transitions from one unit to another one are possible and we use an unconstrained model ρ, for example a loop model (see Lee & Hon 89). Hence, for continuous speech, C_{MMIE} can be expressed as

$$C_{MMIE} = \log(H_{continuous}) = \log(\frac{L_\tau}{L_\rho}) \tag{8}$$

where $H_{continuous}$ is defined as follows:

$$H_{continuous} = \frac{L_\tau}{L_\rho} \tag{9}$$

$L_\tau = \alpha_{F_\tau, T}$ denotes the likelihood of the training model and $L_\rho = \alpha_{F_\rho, T}$ denotes the likelihood of the recognition model.

Assume $b_{i,t}$ can be represented by shared gaussian mixtures for SCHMM (Bellagarda & Nahamoo 89):

$$b_{i,t} = \sum_k \frac{Z_k}{((2\pi)^n \mid \Sigma_k \mid)^{1/2}} \exp(-\frac{1}{2}(Y_t - \mu_k)\Sigma_k^{-1}(Y_t - \mu_k)^T) \tag{10}$$

where n is the number of observation features of the HMM. It is common practice to introduce approximations that consider only the largest contributions to $b_{i,t}$. The transition probabilities a_{ij}, normal distribution mean vectors μ_k, covariance matrices Σ_k, and gains Z_k can be estimated, with algorithms used for training SCHMMs (Bellagarda & Nahamoo 89; Huang &

probabilities for partial sequences (up to time t, from time $t+1$ on)

$$\begin{aligned} \alpha_{i,t} &= P(Y_1^t \text{ and } S_t = i \mid model) &= b_{i,t} \sum_j a_{ji} \alpha_{j,t-1} \\ \beta_{i,t} &= P(Y_{t+1}^T \mid S_t = i \text{ and } model) &= \sum_j a_{ij} b_{j,t+1} \beta_{j,t+1} \end{aligned} \tag{2}$$

with appropriate boundary conditions presumed. If the task is to model isolated units (*e.g.*, isolated words), there will be multiple models ω, one for each unit. For continuous speech recognition, unit models (for example phonemes) are concatenated to make word and sentence models. The likelihood that a HMM has generated the observation corresponding to the pronounciation of the unit ω is

$$L_\omega = \alpha_{F_\omega, T} \tag{3}$$

where F_ω is the final state for model ω. HMM parameters can be estimated with different criteria. Two popular criteria are Maximum Likelihood (ML) and Maximum Mutual Information (MMI). Modeling with these two criteria is discussed in (Nadas, Nahamoo & Picheny 89). Maximum Likelihood Estimation (MLE) is based on the maximization of the criterion C expressed as follows:

$$C_{MLE} = L_c \tag{4}$$

where, for isolated unit modeling, c represents the pronounced unit. In the case of Maximum Mutual Information Estimation (MMIE) for isolated unit modeling, the following criterion can be used:

$$C_{MMIE} = \log(H_{isolated}) = \log(\frac{L_c}{\sum_\omega L_\omega}) \tag{5}$$

The mutual information between the correct model c and the observation Y_1^T is

$$I = \log(\frac{P(Y_1^T, model_c)}{P(Y_1^T)P(model_c)}) = \log(\frac{P(Y_1^T \mid model_c)}{\sum_\omega P(Y_1^T \mid model_\omega)P(model_\omega)}) \tag{6}$$

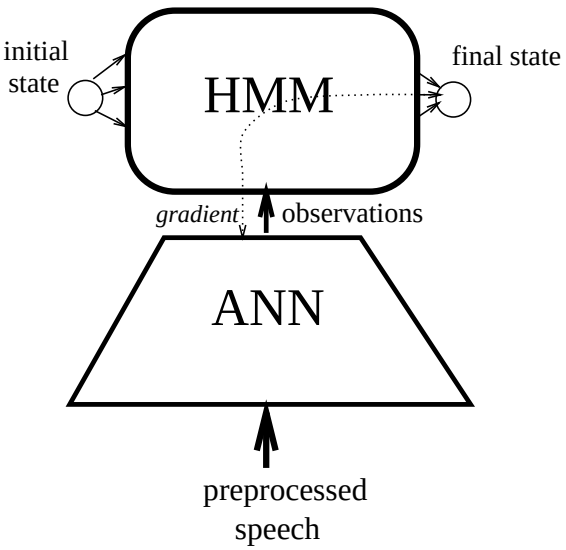

Figure 1: *Proposed ANN/HMM hybrid model: the outputs of the ANN constitute the observation sequence for the HMM. The parameters of both the ANN and the HMM can be estimated in order to perform a global optimization of a given criterion.*

3 Gradient Computation in the hybrid ANN/HMM system

For this paper, only left-to-right models with a single final state are assumed. Let Y_t be the vector of ANN outputs at time t. These outputs are considered as observations of a continuous or a semi-continuous HMM used in the scheme shown in Figure 1. Let us briefly review how observations are dealt with in HMMs (see (Rabiner 89) for more details). Let Y_1^T be the whole observation sequence for the HMM, T is the length of the observation sequence, and Y_t a particular observation, made when the HMM is in the state S_t at time t. Let a_{ij} be the transition probability from state i to state j. The probability that the HMM generates Y_t in state S_t at time t is represented as follows:

$$b_{i,t} = P(Y_t \mid S_t = i) \tag{1}$$

Algorithms (Rabiner 89) allow one to efficiently compute the following joint

bining ANNs with HMMs. In some of the proposed approaches (*e.g.*, Franzini, Lee & Waibel 90; Bridle 90) the activation value of each output node of the network corresponds to $P(\textit{observation} \mid \textit{state})$, the observation probability conditional to the state of the HMM (that will be indicated later as $b_{i,t}$) in equation (1)). The ANN is trained to compute these observation probabilities for the best sequence of states produced by the alignment. In (Franzini, Lee & Waibel 90) the input data are aligned with the model of the spoken utterance with the Viterbi algorithm. In this case, the observation probabilities are approximated by the network outputs. Network parameters are determined with the backpropagation algorithm and the Least Mean Square criterion. Another approach was proposed by Bridle (1990) and consists in computing the gradient of an optimization criterion with respect to all the observation probabilities and to use gradient descent to estimate network parameters (including the parameters of the HMM, which is viewed as a recurrent ANN). Other hybrid systems combining ANNs with HMMs (*e.g.*, Bourlard & Wellekens 88; Morgan & Bourlard 90) theoretically require that the ANN parameter estimation has converged to the global minimum in order to express the posterior probability $P(\textit{state} \mid \textit{observation})$. Our previous work on hybrid models (Bengio *et al* 90) used ANNs merely to compute an additional set of symbols considered as observations for a discrete HMM. A vector-quantized codebook was generated for these parameters and added to codebooks obtained for other popular parameter sets. This did not require any assumption on the network outputs but had the disadvantage that the ANN and the HMM were optimized separately. The method described in the present paper allows to perform global parameter optimization by transmitting to the ANN a gradient computed for the HMM.

constraints [1]. Along a similar line, researchers have attempted to combine the classification power of ANNs with the time-domain modeling capability of HMMs (Bengio *et al* 90; Bourlard & Wellekens 88; Franzini, Lee & Waibel 90; Morgan & Bourlard 90) or to formalize HMMs in the framework of ANN theory (Bridle 90; Levin 90).

In this paper, semi-continuous HMMs (SCHMMs) (Bellagarda & Nahamoo 89; Huang & Jack 89) and continuous densities HMMs (CDHMMs) will be considered in conjunction with networks trained with the generalized delta rule (Rumelhart *et al* 86). It will be shown how to perform a joint *global optimization* of both the ANN and the HMM parameter estimation. In the proposed algorithm, the gradient of the optimization criterion with respect to the transformed observations is computed for the HMM system. The HMM can be trained with traditional methods (Rabiner 89) with which the gradient of an optimization criterion is computed. This gradient is sent to the ANN for the estimation of the weight associated to each connection of the network. No assumption need to be made or constraints imposed on the network outputs, except that the network output distribution should be modeled by a mixture of multivariate gaussians. Since training of HMMs is usually much faster than ANN training, we consider how to initialize the ANN in order to start from parameter values that are not too far from those obtained after training. Multiple ANNs are combined and an incremental design method is described in which specialized networks are integrated to the recognition system in order to improve its performance.

2 Related Work

Interesting papers have been published recently, describing attempts at com-

[1] Probability constraints on the duration and the sequences of phoneme pairs.

1 Introduction

In spite of the fact that speech exhibits features that cannot be represented by a first-order Markov model, Hidden Markov Models (HMMs) of speech units (*e.g.*, phonemes) have been used with a good degree of success in Automatic Speech Recognition (ASR) (*e.g.*, Bahl *et al* 83; Rabiner & Levinson 85; Lee & Hon 89). The effect on ASR performances of first order HMM inadequacies in modeling the speech signal can be reduced if HMM observations are based on carefully chosen parameters. Parameter choice can be inspired by speech knowledge (consider for example samples produced by a mathematical ear model (Cohen 89)). It is often desirable to add new dimensions to the set of parameters used as observations for HMMs. Addition of parameters may be motivated by systematic recognition errors (like poor performances on nasal sounds) for which suitable acoustic cues are worth being considered (like cues for nasality). If the size of the parameter set becomes too large, it is useful to reduce its dimensionality with suitable functions derived from a-priori knowledge and analysis of the system, in order to make HMM parameter estimation effective with the available training data.

Architectures for performing function approximation can be conceived based on Artificial Neural Networks (ANNs) (Poggio & Girosi 89). ANN parameters are estimated from samples of the input/output function that is to be approximated. ANNs have proven to be useful for classifying speech properties and phonemes based on the analysis of a speech segment of limited duration (*e.g.*, Waibel *et al* 88; Bengio *et al* 89; see Lippman 89 for review). Various attempts have been made to interpret the time evolution of ANN outputs. Worth mentionning is the post-processor proposed by Robinson and Fallside (1990) which uses dynamic programming with duration and bigram