A NEW NEURON MODEL FOR AN ALPHANET-SEMICONTINUOUS HMM

J.E. Díaz-Verdejo *J.C. Segura-Luna* *A.J. Rubio-Ayuso* *A.M. Peinado-Herreros* *J.L. Pérez-Córdoba*

Dpto. de Electrónica y Tecnología de Computadores
Facultad de Ciencias, Universidad de Granada
Campus Universitario Fuentenueva
18071 - Granada, SPAIN

ABSTRACT

The main goal of this work is the automatic speech recognition using Artificial Neural Networks. We define a generalized type of neuron that, grouped in a Recurrent Neural Network (an Alphanet), implements a Semicontinuous Hidden Markov Model (SCHMM). The neurons are grouped in a single layer that generates the Alphanet in such a way that some of its inputs come from the outputs. The so-built network allows an interpretation according to SCHMM models, evaluating symbol sequences that constitute the second type inputs. The network is trained using backpropagation algorithm, and has been applied to an isolated word recognition task. Our experimental results show recognition rates reaching multi-speaker recognition rates of 97.81%.

1. INTRODUCTION

In the last years, some neural network approaches have been proposed for speech recognition, nevertheless in most cases the nets present a fixed number of inputs. This brings out problems when dealing with the speech sequential nature and durational variability of elements to be classified. On the other hand, the standard neural networks are independent of HMM-based recognizers, which are at present the best systems for speech recognition.

A new approach has recently been proposed by Bridle [1] and others [2,3] that solves the mentioned problem incorporating an appropriate system to manage the sequential nature of speech. Therefore, this approach includes the knowledge about the well-known and widely used HMM-systems.

The proposed net is the so-called *Alphanet* that will be described in brief in the following sections. This net uses a generalized neuron model [3] that allows us to implement a Discrete HMM (DHMM). In the present work we make a higher level generalization of the neuron in order to implement a Semicontinuous HMM (SCHMM) [4]. SCHMM yields better results than DHMM when applied to speech recognition.

*This work has been supported by Spanish CICYT under project TIC92-0662.

2. HMM: DISCRETE AND SEMICONTINUOUS.

A HMM is a finite state automaton that can model the production of temporary vector sequences. In the HMM formalism, the temporary signal is supposed to be produced by one of this automatons built up from a set of states $Q = \{q_1, q_2, ..., q_N\}$. The evolution of the states is governed by statistical laws and the observed signal is obtained from symbols that the automaton emits in the transitions between states according to a probability that depends on the actual state.

An HMM-based recognizer is composed by a set of HMM-s each one standing for a recognition unit. Given an input vector sequence $X = \{x_1, x_2, ..., x_T\}$ the recognizer will evaluate the probabilities of this sequence being emmited by the model λ, $Pr(X|\lambda)$. The sequence will be classified as belonging to the class whose model yields the higher probability.

Given a model, the probability of a sequence can be obtained from the so-called *forward probability* α_i[5]

$$Pr(X|\lambda) = \sum_{i=1}^{N} \alpha_i(T) \tag{1}$$

The forward probability $\alpha_i(t)$ is the probability that the model were in state i at time t given the input sequence. This probability can be obtained in a recurrent way [5]

$$\left.\begin{aligned} \alpha_i(1) &= \pi_i \cdot B_i(x_t) \\ \alpha_i(t) &= \left[\sum_{j=1}^{N} \alpha_j(t-1) \cdot w_{ij}\right] B_i(x_t) \end{aligned}\right\} \quad \begin{aligned} 2 \le t \le T \\ 1 \le i \le N \end{aligned} \tag{2}$$

where π_i is the probability that the initial state is state i, $B_i(x_t)$ is the probability that vector x_t was emmited while the model is at state i (observation probability) and w_{ij} is the probability of a transition from state j to state i.

The temporary signal that is to be classified (e.g. speech signal) is divided into fixed length segments that are characterized by a feature vector. The only difference between the DHMM and SCHMM approaches consists in the way the observation probabilities of those feature vectors are estimated.

In discrete HMM, the vectors are quantized and the sequence of vectors X is replaced by a sequence of symbols $O = \{O_1, O_2, ..., O_T\}$. A observation probability $b_i(v_l)$ is

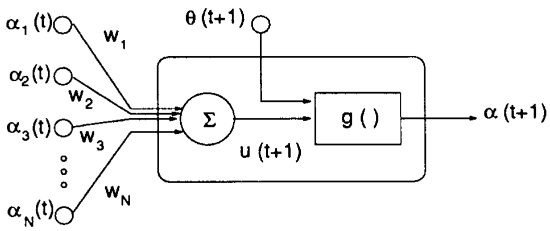

Figure 1. Neuron model for Discrete Alphanet

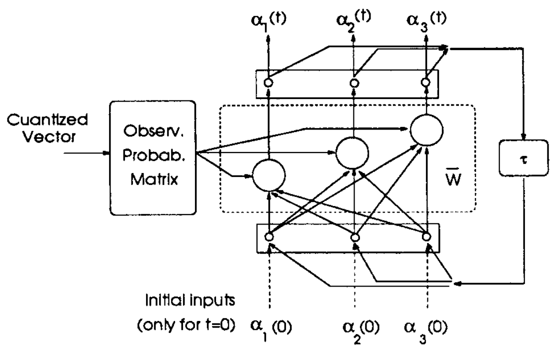

Figure 2. Net topology used for implementing a DHMM

estimated for every symbol v_l during the training process. Therefore the observation probability of a vector $B_i(x_t)$ is obtained from a table lookup in which the vector is replaced by one of the centroids of the codebook (symbols).

$$B_i(x_t) \equiv b_i(O_t) \tag{3}$$

In SCHMM the sequence of feature vectors is preprocessed by a quantizer and each of them is replaced by a vector composed by quantization density probabilities $f(x_t|v_m)$ according to the given codebook centroids v_m. The observation probability is obtained by the linear combination of these probabilities whose weights can be viewed as the observation probabilities of the codebook vectors $b_i(v_m)$.

$$B_i(x_t) \equiv \sum_{m=1}^{M} f(x_t|v_m) \cdot b_i(v_m) \tag{4}$$

3. DISCRETE ALPHANET.

A Discrete Alphanet is a Recurrent Neural Network (RNN) with a topology suitable to evaluate the forward probabilities defined in the HMM context. In this way, an equivalence between HMM and RNN can be established using the Alphanet topology.

The generalized neuron model used for the Alphanet and net topology are shown in figures 1 and 2.

For the *ith* neuron, the $\alpha_j(t)$ signals correspond to the *forward probabilities* associated to the state (neuron) j at time t, and the output signal $\alpha_i(t+1)$ is the probability of state i at time $t+1$. In order to evaluate such probabilities, the *external stimulus* $\theta_i(t+1)$ is also involved, as depicted in figure 1. This external stimulus corresponds

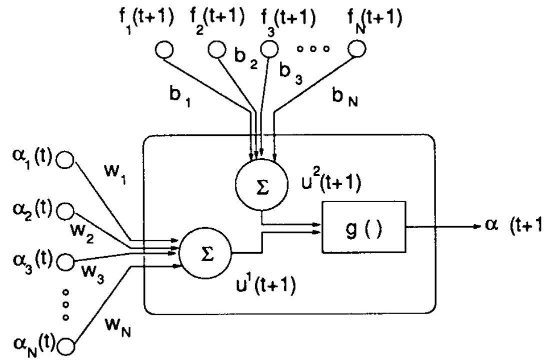

Figure 3. New neuron model proposed for implementing a SCHMM

to the observation probability of symbol O_{t+1} in state i ($b_i(O_{t+1})$ in HMM notation). The neuron equations are:

$$u_i(t+1) = \sum_{j=1}^{N} w_{ij} \cdot \alpha_j(t) \tag{5}$$

$$\alpha_i(t+1) = g(\theta_i(t+1),\ u_i(t+1)) \tag{6}$$

where N is the number of neurons (states). For the Alphanet case, the function $g()$, called *activation function*, corresponds to the product of *net input*, $u_i(t+1)$ by the external stimulus.

In this approach, the observation probabilities of the equivalent DHMM must be provided to the neuron as an external stimulus. Therefore it is independent of the network evaluation and must be obtained previously. These observation probabilities are obtained from a matrix generated from the quantized feature vectors (figure 2).

4. SEMICONTINUOUS ALPHANET.

For Semicontinuous HMM (SCHMM), the observation probabilities are obtained from a mixture of probability density functions (eq. 4). Traditional neuron models can not evaluate this mixture because they only accept a single external stimulus. In order to solve this problem, we need a neuron that accepts several external stimuli, each one corresponding to one pdf.

In the present work, we propose a generalized model of neuron, depicted in figure 3, that solves the above mentioned problem. This new neuron model obtains the observation probabilities internally from the quantization probability densities. These pdf's are generated from a quantization codebook which centers are shaped with Gaussian probability densities. The neuron processing equations for the i-th neuron in the $t+1$ layer are:

$$\left.\begin{aligned} u_i^{(1)}(t+1) &= \sum_{j=1}^{N} w_{ij} \cdot \alpha_j(t) \\ u_i^{(2)}(t+1) &= \sum_{j=1}^{M} b_{ij}(t+1) \cdot f_j(t+1) \\ \alpha_i(t+1) &= g(u_i^{(1)}(t+1), u_i^{(2)}(t+1)) \end{aligned}\right\} \tag{7}$$

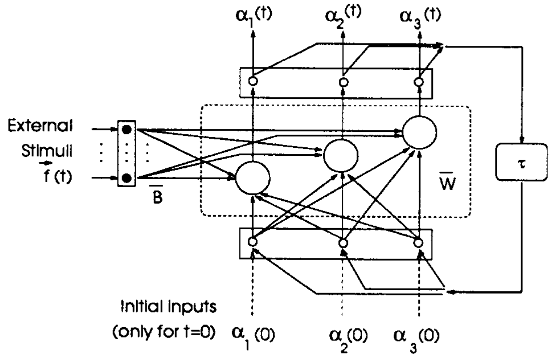

Figure 4. Alphanet scheme using proposed neuron

where N is the number of neurons and M the number of external inputs $f_j(t+1)$ obtained from the observed vector, and b_{ij} is the weight of the j-th component of the mixture.

For the Semicontinuous Alphanet the selected *activation function* $g()$ is the product of the two *net inputs* $u_i^{(1)}(t+1)$ and $u_i^{(2)}(t+1)$.

Figure 4 shows the interconnection scheme of the neurons making up a RNN (Alphanet) in which outputs are connected to the inputs through a delay block.

5. NETWORK TRAINING.

The RNN training is performed (both Discrete and Semicontinuous) by applying the *backpropagation algorithm.* The backpropagation corrective signal is defined from the measure used as [1]

$$\delta_i(t) = \frac{\partial E}{\partial \alpha_i(t)} \tag{8}$$

that can be recurrently computed[1]

$$\delta_i(t) = \sum_{j=1}^{N} \delta_j(t+1) \cdot u_i^{(2)}(t+1) \cdot w_{ji} \tag{9}$$

The selected error measure is the likelihood of the sequence

$$E = \sum_{i=1}^{N} \alpha_i(T) \tag{10}$$

where T is the number of elements of the sequence. This way the net will be trained to implement the Maximum Likelihood Criterion (MLE) although it is possible to train it according to other criterions, such as MMI or MDI [6].

The application of the backpropagation algorithm is coarried out by using the multiplicative updating rules

$$w'_{ij} = w_{ij} \cdot \eta \cdot \frac{\partial E}{\partial w_{ij}} \tag{11}$$

$$b'_{ij} = b_{ij} \cdot \eta' \frac{\partial E}{\partial b_{ij}} \tag{12}$$

[1] In the Discrete Alphanet, $u_i^{(2)}(t+1)$ corresponds to the external stimulus $\theta_i(t+1)$, while b_{ij} is the probability $b_i(v_j)$.

where $'$ stands for the reestimated values of net parameters.

If as usual we express the derivatives of the error measure with respect to the parameter to be updated in function of the derivative with respect to the outputs of the neurons we will obtain the updating equations

Discrete Alphanet

$$w'_{ij} = w_{ij} \cdot \eta_j \sum_{t=1}^{T-1} \delta_i(t+1) b_i(O_{t+1}) \alpha_j(t) \tag{13}$$

$$b'_i(v_k) = \eta'_j \sum_{t=1}^{T} \delta_i(t) \alpha_i(t) \cdot \delta_{kr}(O_t, v_k) \tag{14}$$

Semicontinuous Alphanet

$$w'_{ij} = w_{ij} \cdot \eta_j \sum_{t=1}^{T-1} \delta_i(t+1) u_i^{(2)}(t+1) \alpha_j(t) \tag{15}$$

$$b'_{ij} = b_{ij} \cdot \eta'_j \sum_{t=1}^{T} \frac{\delta_i(t)\alpha_i(t)}{u_i^{(2)}(t)} f_j(t) \tag{16}$$

The HMM interpretation of the Alphanets imposes a normalization constrain to the weights

$$\sum_{i=1}^{N} w_{ij} = 1; \qquad \sum_{j=1}^{M} b_{ij} = 1 \tag{17}$$

This constrains can be introduced in the net reestimation equations by fixing the learning rates, η and η'

$$\eta_j = \frac{1}{\sum_{t=1}^{T-1} \alpha_j(t) \cdot \delta_j(t)} \quad ; \quad \eta'_j = \frac{1}{\sum_{t=1}^{T} \alpha_j(t) \cdot \delta_j(t)} \tag{18}$$

The reestimation algorithm described yields the same expressions for updating as the Baum-Welch for a maximum likelihood estimation [4,7].

6. EXPERIMENTAL RESULTS.

The experimental results of the above explained Alphanet, in an isolated word recognition task, are shown in table 1; the results from DHMM and SCHMM implemented with the proposed neurons are compared in this table. The database has a 16-word vocabulary, 40 speakers and 3 utterances per speaker (a total of 1920 utterances). All the experiments have been carried out in multi-speaker mode, the test set has 640 utterances (one per word and speaker) and the training set 1280. Each frame was caracterized by a vector composed by Cepstrum, ΔCepstrum, Energy and ΔEnergy, these vectors were quantized by using a weighted Euclidean distance [7] and a codebook with 64 centroids.

The vocabulary used is composed by the Spanish digits and the words (/cuerpo/, /hombro/, /codo/, /muñeca/, /mano/, /dedos/) thought to control the motors of a robot.

The experimental results show an increase in the error rate for the training set when comparing the Semicontinuous Alphanet with the Discrete one. Nevertheless the error rate for the test set decreases about 30% (from 3.12 to 2.19) when using the Semicontinuous Alphanet. This is an important qualitative improvement because this implies

N	DHMM		SCHMM	
	Train	Test	Train	Test
2	4.84	7.97	6.48	6.25
3	3.91	6.09	6.02	5.00
4	3.67	6.09	4.45	4.38
5	3.12	5.62	4.14	4.22
6	2.81	5.78	3.75	3.28
7	2.03	5.16	3.05	2.97
8	2.03	4.06	3.28	3.59
9	1.80	4.38	2.89	2.66
10	1.48	3.12	2.66	2.50
11	1.25	3.44	2.27	2.81
12	0.94	3.44	2.42	2.19

Table 1. Error rates for training and evaluation sets

better generalization properties for the Semicontinuous Alphanet than for the Discrete one.

The number of iterations needed to train the nets is small (50-100) in both cases because of the multiplicative update rule selected. In any case it has been observed that the number of iterations needed in the Semicontinuous Alphanet is almost twice that of Discrete Alphanet. This is due to the increase in complexity introduced by the lineal combination of eq (4).

7. CONCLUDING REMARKS.

We have proposed a new neuron model in order to implement a Semicontinuous Alphanet. This new model has the advantage of integrating the observation probabilities as part of the RNN evaluation, so the parameter reestimation can be obtained applying directly backpropagation techniques, as all the parameters correspond to connection weights between neurons or between neurons and input nodes.

In addition, with the proposed neuron, the pdf's can be obtained by using the outputs of a neural based VQ. In this way, the two processes involved (quantization and recognition) can be simultaneously optimized applying backpropagation algorithm to the whole network composed by these two blocks. This approach is currently under study and promising results are expected.

REFERENCES.

[1] Bridle, J.S. "Alpha-nets: A Recurrent Neural Network Architecture with a Hidden Markov Model interpretation", Speech Communication, vol. 9, 1990.

[2] Hwang, J.N; Vlontzos, J.; Kung, S. "A Systolic Neural Network Architecture for Hidden Markov Models", IEEE Trans. Acoust., Speech, Signal Processing, vol. 37, pp. 1967-1979, Dec. 1989.

[3] Kung, S; Hwang, J. "A unifying Algorithm/Architecture for Artificial Neural Networks", Proc. ICASSP-89, pp. 2505-2508.

[4] Huang, X.D.; Jack, M.A. "On Semi-continuous Hidden Markov Modeling". Proc. ICASSP-90, vol. 2, pp. 689-692.

[5] Ephraim, Y; Rabiner, L. "On the Relations Between Modeling Approaches for Speech Recognition". IEEE Trans. Information Theory, vol. 36, n. 2, March 1990

[6] Rabiner, L. "A Tutorial on Hidden Markov Model Parameters for Speech Recognition". Proc. IEEE, vol. 77, n. 2, Feb. 1989

[7] Segura Luna, J.C. "Modelos de Markov con Cuantización Dependiente para Reconocimiento de Voz", Thesis Dissertation. Universidad de Granada, 1990.