A Connectionist Expert Approach for Speech Recognition

Halima Bahi and Mokhtar Sellami
Department of Computer Science, University of Annaba, Algeria

Abstract: *Artificial Neural Networks (ANNs) are widely and successfully used in speech recognition, but still many limitations are inherited to their topologies and learning style. In an attempt to overcome these limitations, we combine in a speech recognition hybrid system the pattern processing of ANNs and the logical inferencing of symbolic approaches. In particular, we are interested in the Connectionist Expert System (CES) introduced by Gallant [10], it consists of an expert system implemented throughout a Multi Layer Perceptron (MLP). In such network, each neuron has a symbolic significance. This will overcome one of the difficulties encountered when we built an MLP, which is how to find the appropriate network configuration and will provide it with explanation capabilities. In this paper, we present a CES dedicated to Arabic speech recognition. So, we implemented a neural network where the input neurons represent the acoustical level, they are defined using the vector quantization techniques. The hidden layer represents the phonetic level and according to the Arabic particularities, the used phonetic unit is the syllable. Finally, the output neurons stand for the lexical level, since they are the vocabulary words.*

Keywords: *Artificial intelligence, speech recognition, hybrid system, neuro-symbolic integration, expert system, neural networks.*

Received February 23, 2004; accepted July 8, 2004

1. Introduction

The Artificial Intelligence (AI) approach tries to reproduce the natural human reasoning which incorporate several approaches of reasoning in particularly in perception problems. This allows us to recognize and to react instantly to sensory cues. This kind of hybrid intelligence has inspired AI researchers to combine multiple artificial methods and several information sources to deal with knowledge in an attempt to simulate human thought.

Some researches in this area deal with the integration of expert systems and neural networks [7, 10, 16, 17, 20, 23]. In particular, we are interested in the connectionist expert system introduced by Gallant [10], which is a multi layer perceptron with symbolic aspect related to domain knowledge.

Our system is dedicated to Arabic speech recognition; it is an MLP which recognizes isolated spoken words in Arabic. We attached to its architecture a symbolic meaning. So, the input layer represents the acoustical level, the hidden layer the phonetic level, and the output layer, stands for the lexical one.

In this paper, we describe our investigations throughout the expert system-neural network integration, and we propose an integration approach which we applied to Arabic speech recognition. The remainder of the paper is structured as follows. In the second section 2, we give a brief introduction of expert neural networks. In section 3, we present the connectionist expert system approach. In section 4, we describe the conceptual elements of our recognizer. In section 5, we give implementation issues. The obtained results are presented in section 6. Finally, conclusion is drawn and perspectives are presented.

2. Expert Neural Networks

2.1. Expert Systems

An expert system consists of programs that contain knowledge bases and a set of rules that infer new facts from knowledge and from incoming data. The rules are used in the inference process to derive new facts from given ones. The strength of expert systems is the high abstraction level. Knowledge can be declared in a very comprehensive manner, making possible to easily verify the knowledge base with the domain experts. The system also gives explanations for the given answers in the form of inference traces. Typical weakness is dealing with incomplete, incorrect and uncertain knowledge. Also, the system does not learn anything by itself.

2.2. Artificial Neural Networks

An Artificial Neural Network (ANN) is basically a dense interconnection of simple, non-linear computation elements called “neurons”. It is assumed that a neuron has N inputs, labeled x_1, x_2, .., x_N, which are summed with weights w_1, w_2, …, thresholded, and

linearly compressed to give the output y, defined as: $y = f(\sum wixi - \Phi)$, where Φ is an internal threshold, and the function *f* is a non-linearity, usually *f* is a sigmoid function [14, 18].

There are several issues to consider in the design of ANNs, in terms of the neurons organization. In particular, the multi layer perceptron is a category of ANN that is usually used in classification problems. It consists of a network composed of more than one layer of neurons, with some or all of the outputs of each layer connected to one or more of the inputs of another layer. The first layer is called the input layer, the last one is the output layer, and in between there may be one or more hidden layers.

There are two phases in neural information processing: The learning phase or training and the retrieving phase. In the training phase a training data set is used to determine the weight parameters that define the neural model. This trained neural model will be used later in the retrieving phase to process real test patterns and yield classification results.

ANNs are good pattern recognizers. In particular, MLP is known to be a universal classifier. They are able to recognize patterns even when the data is noisy, ambiguous, distorted, or has a lot of variation. Although, big problem in neural networks is the choice of architecture: The only way to decide on a certain architecture is by trial-and-error. Another weakness of neural net is the lack of explanation.

2.3. Expert-Neural Systems

Both expert systems and neural networks have strong and weak points. Researchers have tried to overcome expert systems and neural networks limitations by creating hybrid systems. Various classification schemes of hybrid systems have been proposed [13, 17, 21, 24] as a brief introduction, we present a simplified taxonomy, where such systems are grouped into two categories: Transformational and coupled models.

1. In the transformational models (translational as [13]), the expert system could be transformed to a neural network or the neural network could be transformed to an expert system.
2. In the coupled models, the application is constituted of separated two components, that can exchange knowledge. A neural network can be used like component of pre-treatment for the expert system. The expert system can prepare data for neural network and can contribute to the determination of the network configuration. The most used of tightly coupled systems are connectionist expert systems [22].

3. Connectionist Expert Systems

Gallant S. was the first to describe a system combining the domain expert knowledge with neural training [10]. It consists of an expert system implemented throughout a multi layer perceptron. In this approach, the knowledge is incorporated into a neural network in different ways:

- By setting the topology (hidden layers, nodes and connections between nodes).
- By setting weights and bias values.
- By pre-wiring or pruning connections.
- By choosing the adequate learning procedure.

3.1. From Domain Knowledge to Network

The system starts with dependency information from which it builds a structured neural network with only feedforward connections. All specified connections initially have weight of 0. CES have commonly the following properties [10]:

- Each neuron has a meaning.
- Positive weight mean reinforcement of the links while negative ones mean inhibition.
- Input values are discrete, usually {-1, 0, 1} or {0, 1} standing for true, false or not known.
- Cell activation are discrete, taking on values +1, -1, or 0. Cell ui computes its new activation ui' as a linear discriminant function: $S_i = \sum_{j \geq 0} w_{ij} u_j$.

The typical example presented by Gallant, is a CES for diagnostic and treatment of Acute Sarcophagal disease [10]. The input neurons represent symptoms, they will be present, absent or not known yet (like for some analysis of blood). The hidden neurons stand for disease and diagnosis and output neurons represent the treatments.

3.2. Learning

The final information supplied to the system is the set of training examples. It is important that the training examples specify the desired activations for intermediate and output cells in the network: Easy learning. This allows us to decompose the problem and consider each cell independently in terms of training generated.

To train the network, Gallant suggests the use of a relevant algorithm called: Pocket algorithm [7, 10]. The Pocket algorithm is a modification of the perceptron learning rule [6, 14]. It repeatedly executes the perceptron algorithm and maintains (in a Pocket) the weight vector which is remained unchanged for the highest number of iterations. In the following, we describe the principles of the learning algorithm:

Let *w**, be the weights vector of a cell *u*. Set *w** = 0 for every connection from a cell that is not in the *u*'s

dependency list. So, we ignore these weights for the remainder of the computations. To compute w_j for the other cells, we use the following procedure:

For cell u let $\{E^k\}$ be the set of training example activation, and $\{C^k\}$ the corresponding correct activation for u. C^k takes on values {+1, -1}, and E^k takes on values {+1, -1, 0}. P is the perceptron weight vectors, which occasionally replace pocket weight vectors w*.

1. Set P to the 0 vector.
2. Let P the current perceptron weights. Randomly pick a training example E^k (with corresponding classification C^k).
3. If P classifies correctly E^k, that is
   {P. $E^k > 0$ and $C^k = +1$} or {P. $E^k < 0$ and $C^k = -1$}
   then
   - 3.1. If the current run of correct classification with P is longer than the run of correct classification with w* in your pocket.
   - 3.2. replace pocket weights w* by P, and remember the length of its correct run.
4. Otherwise, form a new set of weights $P' = P + C^kE^k$
5. Go to 2.

3.3. Explanation

The user can ask the system, why a particularly concluded cell was true or false. The system will answer with if-then rule application to the current case [10].

4. CES as a Speech Recognizer

Neural networks are widely and successfully used in pattern recognition [6]. In speech recognition many improvements were made to increase their recognition rate [12]. It is commonly agreed, that the difficulty in using neural networks is related to how to configure the neural network, and what are initial weights of links between neurons. So, since there is no deterministic way to configure the network and to set the initial values of connections, some researches deal with the integration of domain knowledge to assist the network conception. In particular, the purpose in the CES approach is to assist the conception of the network so the neurons reflect the semantic rules.

To generate the connectionist knowledge base, we must specify the name of each cell corresponding to variable of interest (acoustic characteristics, phonetic units, …). In parallel, the considered network topology is a MLP. So neurons are regrouped into layers which correspond to the levels of our application which consists of the isolated word recognition, so that the input layer represents the acoustical level, the hidden layer the phonetic level, and the output layer, stands for the lexical one (Figure1).

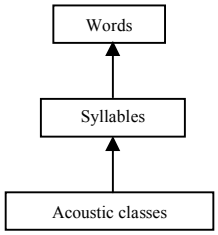

Figure 1. The MLP configuration.

Once, our variables are named and structured into layers, we should determine the dependency between them. This is done as follows:

The output neurons which represent the lexical level are the vocabulary words: The ten Arabic digits. The ten Arabic digits are: sifr (0), waaHid (1), ʔiθnaani (2), θalaaθa (3), ʔarbaʕa (4), xamsa (5), sita (6), sabʕa (7), θamaania (8), tisʕa (9).

In parallel, while considering the Arabic structural characteristics, it appears that syllables are the most suitable phonetic units to consider in a segmentation task [1]. Thus, the hidden units stand for the syllables related to the various pronunciations of our vocabulary words. Each output neuron is linked to the correspondent syllables. For example the word /sifr/ (0) is formed by only one syllable, while the word/ waaHid/ (1) is formed by the syllables: /waa/ and / Hid/.

The input neurons should represent the acoustic signal characteristics. The features extraction stage produces a collection of acoustic vectors each of them represents a temporal frame of the original signal, but in CES, as already said, only discrete values are authorized. Then, we transformed these vectors into discrete symbols using the vector quantization techniques.

Vector quantization allows us to cluster vectors into classes, each of them represents one characteristic of the signal. Each of the input neurons corresponds to an acoustic class and then represents an acoustic characteristic of the spoken word. When a characteristic is present in the signal the corresponding neuron is set to 1 else it will be 0.

Each hidden neuron is linked to the corresponding input neurons, because the acoustical classes are related to syllables.

Overall, our approach could be decomposed in two stages. The first one consists of collecting the acoustic and phonetic knowledge implied by the network conception. Then, when the network is built, it must be trained and the recognition tests will be done.

5. Implementation Issues

Let us consider a set of word pronunciations, we call training base. This set will serve to determine the information dependency and in the second phase to train the MLP.

5.1. Features Extraction

The signal of the spoken word is sampled at a rate of 11025 Hz. Then, all background before and after the word is eliminated. After that, in the first stage, each word is segmented into syllables and for every obtained wave file, we proceed to features extraction. In the second stage (after the MLP conception) the signal of the whole word is analysed similarly. The steps we followed are [1, 2, 19]:

- Preemphasis: The sampled signal is processed by a first-order digital filter in order to spectrally flatten the signal. š (n) = s (n) – a * s (n), a = 0,97.
- Blocking into frames: Sections of N consecutives samples are blocked into a single frame (N = 512 samples of signal). Frames are spaced M samples (M = 256). $x_t(n) = \check{s}(M * t + n)$, $0 \le n \le N - 1$.
- Frame windowing: Each frame is multiplied by a N-sample Hamming window. W (n) = 0,54 - 0,46 * cos (2πn/ N), i. e., $\xi_t(n) = x_t(n) * W(n)$, $0 \le n \le N - 1$.
- Autocorrelation analysis: Each windowed set is autocorrelated to give a set of (p + 1) coefficients, where p is the order of the LPC analysis.

$$R_t(m) = \sum_{N=1}^{N-M} \xi_t(n) \times \xi_t(n+m),\ 0 \le m \le p,\ p = 8$$

- LPC/ Cepstral analysis: For each frame, a vector of LPC coefficients is computed from the autocorrelation vector using the Levinson method [8]. An LPC derived cepstral vector is computed with q coefficients, with q > p, we use q = 12.
- Cepstral weighting: The cepstral vector of q component c_t (m) at time frame t is weighted by a window w_c (m) of the form:

$$W_c(m) = 1 - q/2 * \sin(\pi * m/q)$$

$$\check{C}_t(m) = c_t(m) * W_c(m),\ 1 \le m \le q$$

- Delta cepstrum: The time derivative of the sequence of weighted cepstral vectors is approximated by a first-order orthogonal polynomial over a finite length window of (2K + 1) frames, centred around the current vector (we use k = 2, hence a 5 frame window is used). The cepstral derivative (or the delta cepstrum) is computed as:

$$\Delta\check{c}_t(m) = [\sum_{k=-K}^{K} k\, \check{c}_{t-k}(m)] * G,\ G = 0.375;\ 1 \le m \le q$$

The acoustic vector is the concatenation of the weighted cepstral vector, and the corresponding weighted delta cepstrum vector, i. e., $V_t = \{\check{c}_t(m), \Delta\check{c}_t(m)\}$; $1 \le m \le q$. Each window of the signal will correspond a numerical vector of 24 coefficients.

5.2. Vector Quantization

Given a training set of continuous observation vectors, the Vector Quantization (VQ) partitions the training vectors into M disjoint regions (M is the size of the codebook), and represents each set by a single vector v_m, which is generally the centre of the training set assigned to the m^{th} region.

We consider all acoustical vectors we obtain during the training stage, we regroup them into disjoint classes using the LBG algorithm, a variant of the k-means [12]. The 32 prototypes we obtain represent acoustical frames, and are the entries of the codebook.

At the recognition phase, the vector quantizer compares each acoustical vector v_j of the signal to stored vectors c_i, that represent the code-words, and v_j is coded by the vector c_b that best represents v_j according to some distortion measure d. $d(v_j, c_b) = \min(d(v_j, c_i))$.

5.3. Syllables as Decision Units

Arabic speech has the particularity to present few vowels, few consonants and a regular structure of syllables [1, 9, 11]. Syllables could also be easily processed and have well defined linguistic statute, especially in the phonetic level where they represent suitable unit for the lexical access. These elements have motivated our choice to consider the syllable for modelling the phonetic level. Another element sustained this choice, which is, given a set of Arabic syllables in the nearly totality of cases only a single word could be formed.

In the following, we present the five possible patterns of Arabic syllable presented in [13]. In their representation C, stands for all consonants, V for short vowels and VV for long vowels. The first four patterns occur initially, medially and finally. The fifth pattern, occurs only finally or in isolation:

1. *CV* e. g., /bi/ (with).
2. *CVC* e. g., /sin/ (tooth).
3. *CVV* e. g., /maa/ (not).
4. *CVVC* e. g., /baab/ (door).
5. *CVCC* e. g., /sifr/ (zero).

From our vocabulary, we have extracted 29 syllables that represent the majority of phonetic variants of the ten Arabic digits. The prototypes, we define in the earlier stage VQ, will characterize a syllable if they exist or not in a given signal.

5.4. Dependency Information

The network architecture reproduces two categories of dependency information, the first one represents the relationship between the input and the hidden layer and the second between the hidden and the output layer.

We choose to represent the first category of dependency by using simple rules having the following form:

If conjunction (classes) *then* syllable.

They explain connections between the input and the hidden nodes. The input nodes are classes issued form the VQ stage. We decided to keep 32 classes. Every one of these classes represent an acoustical characteristic of the signal, since we are interested with the ten Arabic digits an example of these rules is:

If C_7 *and* C_8 *and* C_9 *and* C_{11} *and* C_{15} *and* C_{20} *and* C_{22} *and* C_{23} and C_{25} *and* C_{27} *and* C_{28} *and* C_{30} *then* waa

If C_7 *and* C_9 *and* C_{10} *and* C_{11} *and* C_{12} *and* C_{20} *and* C_{24} *and* C_{30} *then* naan

If C_7 *and* C_9 *and* C_{11} *and* C_{12} *and* C_{13} *and* C_{14} *and* C_{15} *then* ?ar

If C_1 *and* C_2 *and* C_7 *and* C_8 *and* C_{19} *and* C_{20} *and* C_{28} *and* C_{30} *then* γam

If C_8 *and* C_{10} *and* C_{11} *and* C_{12} *and* C_{16} *and* C_{24} *and* C_{30} *then* tis

Where /waa/, /naan/, and /?ar/, … are Arabic syllables which represent nodes of the hidden layer.

The second category of dependencies is easier to determine. It is concerned with relationship between hidden and output nodes, which represent the vocabulary words. Each word (output neuron) depends on the corresponding syllables

5.5. The CES Topology

The neural network is a multi layer perceptron (Figure 2). It has the following characteristics: the input layer, contains thirty two neurons representing all the acoustic classes. A signal in entry of the system is analyzed, then transformed to a symbolic chain by the vector quantizer. Each symbol is the index corresponding to the prototype of the vector in the codebook. Every entry of the network is going to receive a binary value (1 or 0) according to the existence of the corresponding characteristic in the signal.

The output layer, contains ten neurons representing the words of the vocabulary, here the ten Arabic digits. Hidden layer contains twenty nine neurons corresponding to the syllables of our vocabulary.

6. Results

We performed some tests to evaluate the CES performances comparatively to other approaches in Arabic speech recognition. Experimentations were performed on a Dell-PC, 1.2 Ghz with sound Blaster 16 card.

We consider a training corpus constituted by three speakers, each of them uttered three times the ten digits. The test corpus, comprise four speakers each of them uttered twice the ten digits. In the Table 1, we present the results obtained with the implementations below, considering the same conditions as for our proposition (same corpus of data and the same steps for the signal analysis):

1. With a classical MLP trained with backpropagation algorithm.
2. With Hidden Markov Models [2].
3. With CES, 17 hidden neurons [3, 4].
4. With CES, 29 hidden neurons [5].

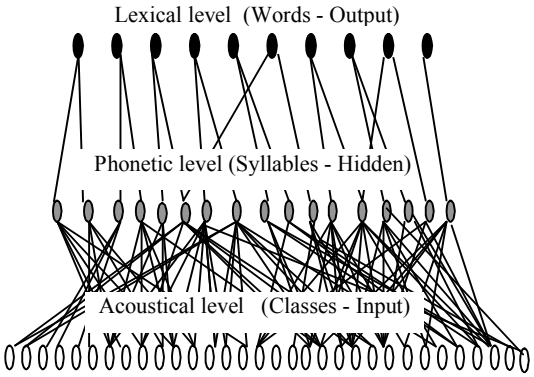

Figure 2. The general structure of the network.

Table 1. Results.

(a)	(b)	(c)	(d)
94	95	95.71	97.86

7. Conclusion and Perspectives

The model proposed by Gallant is one of the simplest possible connectionist models, since there is no feedback and all computations can be performed using integer arithmetic.

We also notice that, when using domain knowledge, the network trains faster and generalizes better than the classical ones.

Another aspect appears when we assume a knowledge-based approach relying on the recognition phase. In the recognition phase, a word could be recognized and be well categorized or recognized and badly categorized; this addresses the question of the system reliability, then the explanation aspect becomes to much important. This aspect is absent in the connectionist models but is effectively present in rule-based systems.

In parallel, some improvements can be brought to our suggestion, so as to include additional layers standing for other phonetic or acoustic aspects, such as: Phonemes, phones, voiced or voiceless sounds characteristics, etc. This permits to consider other languages, in particular those where syllables do not have the same importance as in Arabic.

Overall, the connectionist expert models are a promising trend in resolution of perception problems, since this category of problems involves both neuronal models and symbolic reasoning.

References

[1] Bahi H. and Sellami M., "An Acoustical Based Approach for Arabic Syllables Recognition," *in Proceedings of the Workshop on Software for the Arabic Language*, Beirut, Lebanon, June 2001.

[2] Bahi H. and Sellami M., "Combination of Vector Quantization and Hidden Markov Models for Arabic Speech Recognition," *in Proceedings of the ACS/IEEE International on Computer Systems and Applications (AICCSA'01)*, Beirut, Lebanon, pp. 96-100, June 2001.

[3] Bahi H. and Sellami M., "Hybrid Approach for Arabic Speech Recognition," in *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'03)*, Tunis, Tunisia, July 2003.

[4] Bahi H. and Sellami M., "Hybrid Approach for Speech Recognition," *in Proceedings of the IAPR International Conference on Image and Signal Processing (ICISP'03*), Agadir, Marocco, pp. 362-367, June 2003.

[5] Bahi H. and Sellami M., "Système Expert Connexioniste Pour la Reconnaissance de la Parole," *in Proceedings of the RFIA*, Toulouse, France, pp. 659-665, January 2004.

[6] Bishop C. M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[7] Boz O., *Knowledge Integration and Rule Extraction in Neural Networks*, University of Lehigh, 1995.

[8] Calliope, *La Parole et Son Traitement Automatique*, in Masson (Ed), 1989.

[9] El-Ani S. H., *Arabic Phonology*: *An Acoustical and Physiological Investigation*, Indiana University, 1989.

[10] Gallant S. I., "Connectionist Expert Systems," *Communications of the ACM*, vol. 31, no. 2, pp. 152-169, 1988.

[11] Harkat M., *Les Sons et la Phonologie*, in Dar el Afaq (Ed), Algeries, Algeria, 1993.

[12] Haton J. P., "Les Modèles Neuronaux et Hybrides en Reconnaissance Automatique de la Parole: état des Recherches," http://www.biblio theque.refer.org/html/parole/haton/haton.htm.

[13] Hilario M., *An Overview of Strategies for Neurosymbolic Integration*, in Sun R. (Ed), Kluwer Academic Publishers, 1996.

[14] Jodouin J. F., *Les Réseaux Neuromémitiques: Modèles et Applications*, in Hermès (Ed), Paris, 1994.

[15] Linde Y., Buzo A., and Gray R. M., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Computers*, no. 36, pp. 84-95, 1980.

[16] Louie R., "Hybrid Intelligent Systems Integration into Complex Multi-Source Information Systems," *Master Thesis*, MIT, 1999.

[17] Medsker L. R., *Hybrid Neural Network and Expert Systems*, Kluwer Academic Publishers, 1995.

[18] Morgan N. and Bourlard H. A., "Neural Networks for Statistical Recognition of Continuous Speech," *Proceeding of the IEEE*, vol. 83, no. 5, pp. 742-770, 1995.

[19] Rabiner L. and Juang B. H., *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[20] Sima J., "Neural Expert System," *Journal of Neural Networks*, vol. 8, no. 2, pp. 261-271, 1995.

[21] Sima J., "Review of Integration Strategies in Neural Hybrid Systems," *citesser.nec.nj.com.*

[22] Towell G. G., "Symbolic Knowledge and Neural Networks: Insertion, Refinement and Extraction," *PhD Thesis*, University of Wisconsin, Madison, 1991.

[23] Towell G. G. and Shavlik J. W., "Knowledge-Based Artificial Neural Networks," *Artificial Intelligence*, vol. 70, pp. 119-165, 1994.

[24] Wermter S. and Sun R., *Hybrid Neural Systems*, Springer, New York, 2000.

Halima Bahi is an assistant professor at the Department of Computer Science, University of Annaba, Algeria, and a researcher at the LRI Laboratory. She received her MSc in computer science from the University of Annaba, Algeria, in 1996. Currently, she is preparing for her PhD. Her research interests include speech recognition and its applications, hidden Markov models, and neural networks.

Mokhtar Sellami received a PhD in computer science from the University of Grenoble, France, in the specialty of artificial intelligence and logic programming. He participates in many international research projects in Europe and Algeria. He is currently director of Computer Research Laboratory at Annaba University and a senior lecturer in artificial intelligence and expert systems. His professional interests include pattern recognition applied to Arabic image processing, hybrid systems and knowledge engineering.