# Voice Recognition Using Modular Neural Networks and Genetic Algorithms

**Patricia Melin, Felma Gonzalez and Gabriela Martinez**
Department of Computer Science
Tijuana Institute of Technology
Tijuana, Mexico
pmelin@tectijuana.mx

**Abstract**
We describe in this paper the evolution of modular neural networks using hierarchical genetic algorithms. Modular Neural Networks (MNN) have shown significant learning improvement over single Neural Networks (NN). For this reason, the use of MNN for pattern recognition is well justified. However, network topology design of MNN is at least an order of magnitude more difficult than for classical NNs. We describe in this paper the use of a Hierarchical Genetic Algorithm (HGA) for optimizing the topology of each of the neural network modules of the MNN. Simulation results shown in this paper prove the feasibility and advantages of the proposed approach.

## 1 Introduction

The results of the different applications involving MNN lead to the general evidence that the use of modular neural networks implies a significant learning improvement comparatively to a single NN and especially to the backpropagation NN [2]. Indeed, to constrain the network topology on connectivity, increases the learning capacity of NN and allow us to apply them to large-scale problems [6]. This is highly confirmed by the experience carried out by [1], which shows that a random pruning of connections before any learning improves significantly the network's performance. Also, we can note that [4] argue that a complex behavior requires bringing together several different kinds of knowledge and processing, which is, of course, not possible without structure (modularity).

Usually the MNN implementations are based on the "divide and conquer" principle, which is well known in computer science. This principle consists first in breaking down a task into smaller and less complex subtasks, to make learn each task by different experts (i.e. NN modules) and then, to reuse the learning of each subtask to solve the whole problem. For example, [8] has shown that NN training can be greatly simplified by identifying subtasks in the problem and embedding them into the network structure.

This method can only be applied if the a-priori knowledge concerning the task is sufficiently precise to enable a split up of the task into subtasks. If one can separate a task in distinct subtasks, each task can be trained off-line and later integrate in the global architecture [12]. This enables an acceleration of the learning [5].

To make an efficient use of modular neural networks, however, we need to optimize their topology for a specific problem. Here is where the evolutionary approach to modular neural network design comes into place. We propose the use of a hierarchical genetic algorithm for evolving the topology design of the complete modular network, which means optimizing each and all of the modules of the MNN. An HGA is needed because the MNN topology information is hierarchical, i.e. we need to manage information about the number of modules, number of layers and nodes of the network. As a consequence the chromosome of an HGA contains genetic information about the different levels of the MNN topology.

## 2 Modular Neural Networks

This section describes a particular class of "modular neural networks", which have a hierarchical organization comprising multiple neural networks; the architecture basically consists of two principal components: local experts and an integration unit, as illustrated in Figure 1. A variety of modular connectionist architectures have been discussed, and thus such diverse names as "committees of networks", "adaptive mixtures", and "hierarchical mixtures of experts" have all been mentioned.

In general, the basic concept resides in the idea that combined (or averaged) estimators may be able to exceed the limitation of a single estimator. The idea also shares conceptual links with the "divide and conquer" methodology. Divide and conquer algorithms attack a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the complex problem. In other words, the central idea is task decomposition. When using a modular network, a given task is split up among several local experts NNs. The average load on each NN is reduced in comparison with a single NN that must learn the entire original task, and thus the combined model may be able to surpass the limitation of a single NN. The outputs of a certain number of local experts ($O_i$) are mediated by an integration unit. The integrating unit puts those outputs together using estimated combination weights ($g_i$). The overall output Y of the modular network is given by

$$Y_i = \Sigma g_i O_I \qquad (1)$$

Nowlan, Jacobs, Hinton, and Jordan [7] described modular networks from a competitive mixture perspective. That is, in the gating network, they used the "softmax" activation function, which was introduced by McCullagh and Nelder [9]. More precisely, the gating network uses a softmax activation $g_i$ of the ith output unit given by

$$G_i = \exp(ku_i) / \Sigma_j \exp(ku_j) \quad (2)$$

Where $u_i$ is the weighted sum of the inputs flowing to the ith output neuron of the gating network. Use of the softmax activation function in modular networks provides a sort of "competitive" mixing perspective because the ith local expert's output $O_i$ with a minor activation $u_i$ does not have a great impact on the overall output $Y_i$.

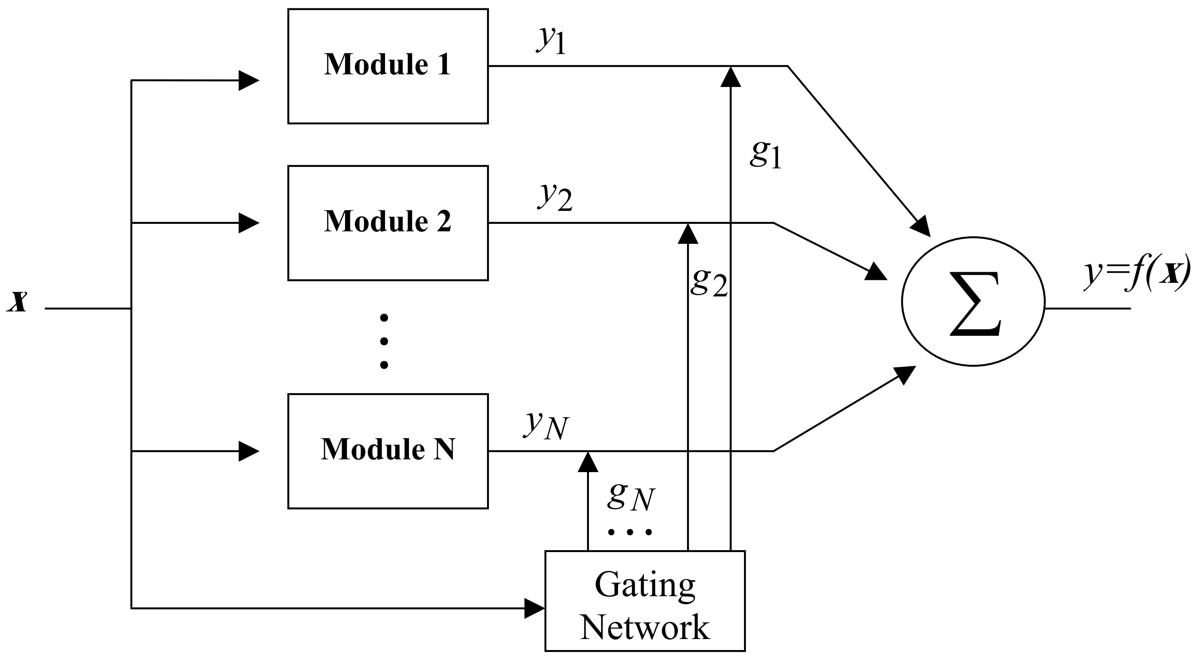



Figure 1 Architecture of a modular neural network.

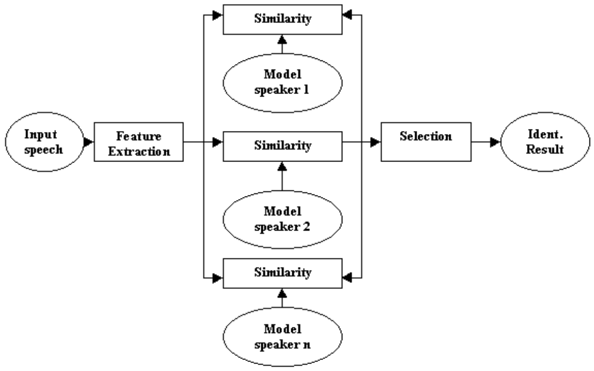
## 3 Genetic Algorithms for Neural Networks

The bottleneck problem for NN application lies within the optimization procedures that are used to obtain an optimal NN topology. Hence, the formulation of the Hierarchical Genetic Algorithm (HGA) is applied for this purpose [10]. The HGA differs from the standard GA with a hierarchy structure in that each chromosome consists of multilevel genes. Each chromosome consists of two types of genes, i.e. control genes and connection genes. The control genes in the form of bits, are the genes for layers and neurons for activation. The connection genes, a real value representation, are the genes for connection weightings and neuron bias. With such a specific treatment, a structural chromosome incorporates both active and inactive genes. It should be noted that the inactive genes remain in the chromosome structure and can be carried forward for further generations. Such an inherent genetic variation in the chromosome avoids any trapping at local optima, which has the potential to cause premature convergence. Thus it maintains a balance between exploiting its accumulated knowledge and exploring the new areas of the search space. This structure also allows larger genetic variations in chromosome while maintaining high viability by permitting multiple simultaneous genetic changes.

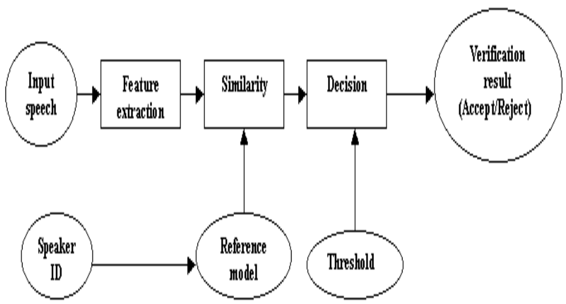
## 4 The Problem of Speech Recognition

Speaker recognition, which can be classified into identification and verification, is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers.

Figure 2 shows the basic components of speaker identification and verification systems. Speaker identification is the process of determining which registered speaker provides a given utterance. Speaker verification, on the other hand, is the process of accepting or rejecting the identity claim of a speaker.




(a) Speaker identification




(b) Speaker Verification

Figure 2 Basic structures of speaker recognition systems

Both text-dependent and independent methods share a problem however. These systems can be easily deceived because someone who plays back the recorded voice of a registered speaker saying the key words or sentences can be accepted as the registered speaker. To cope with this problem, there are methods in which a small set of words, such as digits, are used as key words and each user is prompted to utter a given sequence of key words that is randomly chosen every time the system is used. Yet even this method is not completely reliable, since it can be deceived with advanced electronic recording equipment that can reproduce key words in a requested order. Therefore, a text-prompted speaker recognition method has recently been proposed by [11, 12].

## 5 Simulation Results

We describe below some simulation results of our approach for speaker recognition. First, in Figure 3 we have the signal of the word "example" in Spanish with noise. Next, in Figure 4 we have the identification of the word "example" without noise. We also show in Figure 5 the word "layer" in Spanish with noise. In Figure 6, we show the identification of the correct word "layer" without noise.

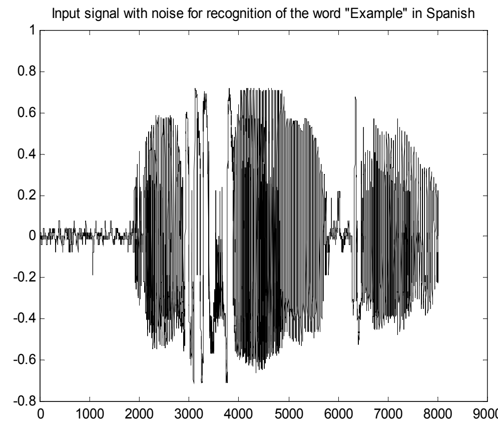



Figure 3 Input signal of the word "example" in Spanish with noise.

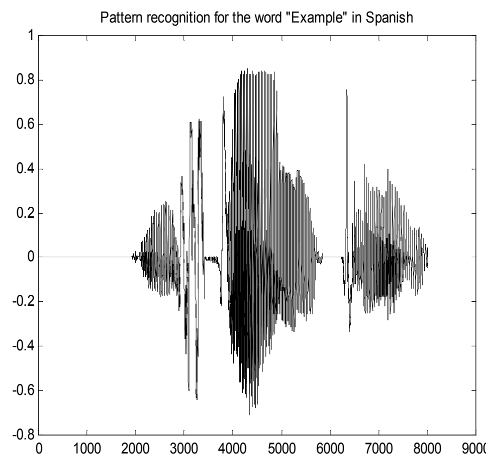



Figure 4 Indentification of the word "example".

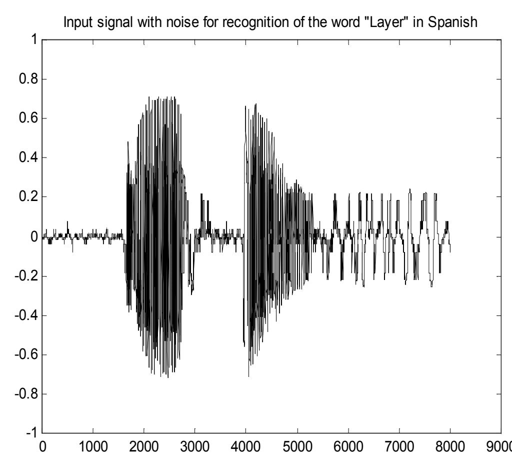



Figure 5 Input signal of the word "layer" with noise added.

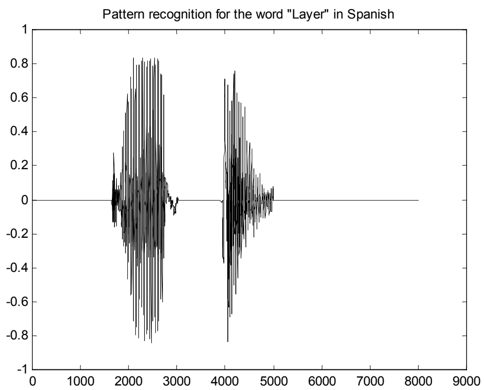



Figure 6 Identification of the word "layer".

The idea behind this data partition is that learning will be easier in each of the modules, i.e. a simple NN can learn more easily the behavior of the function in one of the regions. We used three-layer feed-forward NNs for each of the modules with the Levenberg-Marquardt training algorithm. Regarding the genetic algorithm for MNN evolution, we used a hierarchical chromosome for representing the relevant information of the network. First, we have the bits for representing the number of layers of each of the three modules and then we have the bits for representing the number of nodes of each layer. This scheme gives us the chromosome shown in Figure 8.

The fitness function used in this work combines the information the error objective and also the information about the number of nodes as a second objective. This is shown in the following equation.

$$f(z) = \left( \frac{1}{\alpha * Ranking \ (ObjV\ 1) + \beta * ObjV\ 2} \right) * 10 \qquad (4)$$

The first objective is basically the average sum of squared of errors as calculated by the predicted outputs of the MNN compared with real values of the function. This is given by the following equation.

$$f_1 = \frac{1}{N} \sum_{i=1}^{N} (Y_i - y_i) \qquad (5)$$

| CM1 | CM2 | CM3 | NC1M1 | NC2M1 | NC3M1 | NC1M2 | NC2M2 | NC3M2 | NC1M3 | NC1M3 | NC1M3 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 BITS | 3 BITS | 3 BITS | 8 BITS | 8 BITS | 8 BITS | 8 BITS | 8 BITS | 8 BITS | 8 BITS | 8 BITS | 8 BITS |

Figure 8 Basic structure of the chromosome containing the information of the MNN.

The parameters of the genetic algorithm are as follows:

Type of crossover operator: Two-point crossover
Crossover rate: 0.8
Type of mutation operator: Binary mutation
Mutation rate: 0.05
Population size per generation: 10
Total number of generations: 100

## 6 Conclusions

We described in this paper our hierarchical genetic algorithm approach for modular neural network topology design and optimization. The proposed approach was illustrated with a specific problem of recognition. The best MNN is obtained by evolving the modules (single NNs) according to the error of identification and also the complexity of the modules. The results for the problem of pattern recognition are very good and show the feasibility of the HGA approach for MNN topology optimization.

## References


[1] G. Barna and K. Kaski, “Choosing optimal network structure”, Proceedings of the International Neural Network Conference (INNC90), pp. 890-893, 1990.

[2] O. Castillo and P. Melin,"Hybrid Intelligent Systems for Time Series Prediction using Neural Networks, Fuzzy Logic and Fractal Theory", IEEE Transactions on Neural Networks, Vol. 13, no. 6, pp. 1395-1408, 2002.

[3] D. J. Chalmers, "The Evolution of Learning: An Experiment in Genetic Connectionism", Proceedings of the 1990 Connectionist Models Summer School, Morgan Kauffman, 1990.

[4] J. Feldman, “Neural representation of conceptual knowledge”, in Neural connections, mental computation (Nadel and et al., eds.), MIT Press, 1989.

[5] F. Fogelman-Soulie, “Multi-modular neural network-hybrid architectures: a review”, Proceedings of 1993 International Joint Conference on Neural Networks, 1993.

[6] B. Happel and J. Murre, “Design and evolution of modular neural network architectures”, Neural Networks, vol. 7, pp. 985-1004, 1994.

[7] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive Mixtures of Local Experts", Neural Computation, vol. 3, pp. 79-87, 1991.

[8] R. Jenkins and B. Yuhas, “A simplified neural network solution through problem decomposition: The case of the truck backer-upper”, IEEE Transactions on Neural Networks, vol. 4, no. 4, pp. 718-722, 1993.

[9] M. I. Jordan and R. A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm", Neural Computation, vol. 6, pp. 181-214, 1994.

[10] K. F. Man, K. S. Tang and S. Kwong, "Genetic Algorithms: Concepts and Design", Springer-Verlag, 1999.

[11] M. Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 1996.

[12] C. Monrocq, “A probabilistic approach which provides and adaptive neural network architecture for discrimination”, Proceedings of the International Conference on Artificial Neural Networks, vol. 372, pp. 252-256, 1993.