Modularity: a natural choice for improving generalization

LEONARDO FRANCO[1] JOSE M. JEREZ[2]
[1] Cognitive Neuroscience Sector
International School for Advanced Studies (SISSA)
Via Beirut, Trieste, 34014
ITALY
[2] Departamento de Lenguajes y Ciencias de la Computación
University of Málaga
Complejo Tecnológico, Campus de Teatinos
Málaga, 29071
SPAIN

Abstract: - One of the principal motivations for constructing artificial modular neural networks comes from biological neural systems, where modularity can be observed at different levels of organization. Starting from some theoretical results that roughly state that modular architectures generalize better than their monolithic counterparts, we test through simulations this assertion on three problems: the parity function and other two "real world" problems as face expression recognition and diabetes diagnosis. We also analyze how the size of the networks influences the generalization ability. From the results we extract some general recommendations on how to build and train modular architectures.

Key-Words: - Artificial neural networks, generalization performance, modularity, back propagation, parity function, face expression recognition, medical diagnosis.

1 Introduction

There is clear evidence that a modular organization exists at different anatomical scales of the brain of living organisms, ranging from a global functional specialization to the level of cortical columns, and maybe even deeper [1][2]. The main advantage of modular systems might be the reduction of mutual interference between simultaneous processing. Also, even it is not completely clear, seems that modularity enhance learning and generalization performance, explaining, for example, that many vital tasks in living organisms require minimal exposure of relevant stimuli to be learned. Other potential benefits of modular architectures include fault tolerance, good scaling properties and better fitness to hardware constraints [2][3].

From this perspective, applying biological ideas to the construction of artificial neural networks seems promising, but practice has revealed that the process of achieving the high capabilities of biological systems is not quite simple and straight.

Neural networks, in particular feed forward ones, have been successfully applied to several classes of problems but still many aspects of their implementation and learning capabilities in a general context remain unclear. For example, selecting a neural network architecture for a determined problem is a complex task, involving many factors as complexity of the problem, size of the training set, training algorithm, etc, that affect the network performance in a not completely understood form [3].

In this work we construct modular architectures and analyze their generalization properties for three different problems: parity function, face expression recognition and diabetes diagnosis. In next subsection some general results about generalization properties of feed forward neural networks are presented; in section 2 the implementation and results of the simulations are shown, to finally discuss them and give some recommendations on how to implement modular networks.

1.2 Motivation

Generalization is the property to respond with accuracy to inputs signals that have never seen before by the system, being one of the most attractive properties of neural networks. Answers to general questions such as: How many examples are necessary to obtain valid generalization? or What size neural

nets gives best generalization? can be addressed within the context of the VC dimension results [3][4]. This theory suggests that smaller networks generalize better than larger ones as a consequence of a reduced number of free parameters, and also that smaller networks should be less sensitive to overfitting (for a detailed description on these issues we refer the reader to [4] and references therein). However, empirical results contradict the previous hypothesis showing that in some cases larger neural nets *trained with backpropagation* perform better than smaller ones. One reason, explaining partially this behavior, can be the early stopping of the learning procedure preventing an excessive growth of the synaptic values and keeping the state of the system in a reduced exploratory space [5].

On the other hand, in [6] a method was developed for selecting examples in feed forward networks, and through its application good generalization can be obtained with reduced set of examples. The general results agree with those from the VC theory, in the sense that modular neural networks, having less number of weights than fully connected architectures, would need less examples to generalize; but also it was shown that the scaling properties of modular networks are much better than those for monolithic architectures [7].

With the previous ingredients as background, we decided to test on three different problems the generalization properties of modular vs. fully connected networks exploring some of the above mentioned issues.

2 Simulation studies

We select three different problems to carry on our study: the parity function was chosen because some of the previous results were obtained through its study [7]; the other two problems were selected as they are considered "real world" problems. From the particular features of each problem, we distinguish the following characteristics: first, parity has boolean input values while the others two problems have real ones; second, a salient difference concerns the size of the training set, better evaluated comparatively with the number of synaptic weights: for the parity function the *whole exact* set of examples is available, in the face expression recognition task the training set is very small and in the third case, diabetes diagnosis, the number of available examples is somewhat large but includes some missing data that has been replaced with fixed values, introducing a certain level of error [8]. In table 1 these features of the problems are shown, while in table 2 the number of weights used for the different architectures are presented.

Problem	Inp.	Outp.	Inp.Type	Examples
Parity	8	1	boolean	256
Face Expr. Recogn.	48	3	real	56
Diabetes Diagn.	8	1	real	768

Table 1. Some features of the three analyzed problems.

2.1 The parity function

The parity function is one of the most used functions for testing learning algorithms, both for historical and complexity reasons [7][9], being a function considered “hard”, as learning algorithms does not perform very well on it.

The simulations were carried on a modular architecture with 8 input neurons, five hidden layers containing 20 neurons and a fixed number of synapsis per neurons equals to 2. In figure 1 the whole structure of the network its shown, being possible to observe the constituting modules with the structure 2-2-1, normally used to solve the XOR function. To build the modular network we use the property that the parity of an arbitrary number of bits can be computed by dividing them into groups and computing the parity of every group independently.

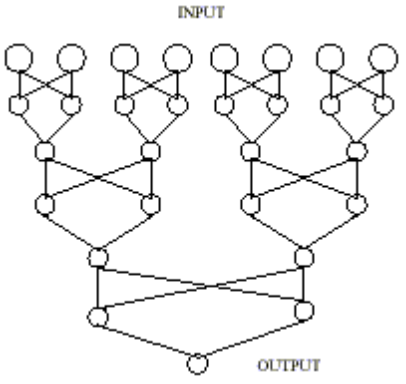

Figure 1. Tree-like modular network used to compute the 8-bit parity function.

We first use backpropagation as learning algorithm but it was not possible to find solutions to the problem, despite the fact that this architecture is able to compute the parity function [7]. The problem of backpropagation in computing parity seems to be more general, as previous results found that neutral statistical problems, which parity belongs to, are very difficult to learn [10]. We also tested different problems on the same architecture and the

performance of backpropagation did not improve much, except for the case of trivial functions. Thus, we change the learning algorithm to simulated annealing and also restrict the values of the synaptic weights to discrete values [+1, -1], obtaining results in clear agreement with the theoretical predictions mentioned before [7]. For comparison we also analyze the performance on a monolithic network with a single hidden layer containing 8 neurons fully connected. In figure 2 the generalization error as a function of the number of examples in the training set is shown for both architectures.

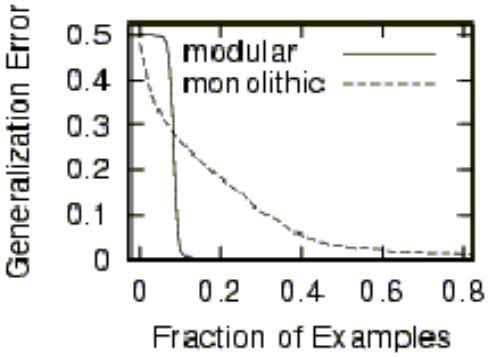

Figure 2. Generalization error vs. fraction of training of examples in the training set for modular and monolithic architectures used to compute the 8-bit parity function.

2.2 Face expression recognition

Face expression recognition is a very important task of human cognition playing an important role in several activities where human interacts. An increasing interest has raised in last years for constructing artificial systems that could interact with humans in a more natural way [11].

Following recent results in which was shown that face expression recognition involves different specialized areas of the brain [12], we construct a modular system where each of the modules specializes in a different emotion. This type of modular networks, also known as mixtures of experts, has been successfully used in many applications [2][3].

We use as training set, images taken from the Yale face database [13], selecting a set of 56 examples of 14 subjects displaying four different expressions: neutral, happy, sad and surprise. In figure 3 a sample subject displaying the four emotions is shown, being also indicated in the rightmost expression (surprised face), through a white rectangle, the area (24x8 pixels) cropped from the original images and used as input of the networks.

Figure 3. Sample subject showing the four full faces expressions (neutral, happy, sad and surprised). The white rectangle inside the rightmost figure corresponds to the area cropped and used as input for the neural networks.

Being the training set quite small compared to the number of pixels per image and consequently to the possible number of weights of an architecture, we reduce the dimension of the input through a hebbian unsupervised process, implemented in the network with one layer of neurons. For the scope of this work, we consider this first layer as a preprocessing stage that reduces the number of inputs from 192 down to 48. Thus, the modular network structure has 48 input units connected to three modules specialized on the expressions, those different from the neutral one. Each module has one hidden layer and a single output that has to be activated when the input image displays the same expression in which the module is specialized on. The optimal number of hidden units was found to be 3 for happy and surprise modules, while 4 units were optimal for the sad one. A cross validation training procedure was used; first, the network was trained with 12 subjects and the performance was validated on a 13 one, obtaining a measure of when to stop the training process to avoid overfitting and get a better generalization. Then the network was retrained with 13 subjects and the generalization error was measured on the left out one, averaged over all the 14 possible cases and over 10 initial conditions in each case. Backpropagation with learning constant $\eta = 0.05$ was used.

To compare the performance of the modular architecture, a fully connected neural network were implemented, with a number of hidden units varying from 3 to 15, finding the best results with the larger network. In table 2 results are presented for both modular and monolithic architectures, showing an improvement of generalization around 20% when using the modular one. Also in the table, two results are indicated for the case of monolithic networks to appreciate the effect of network size.

2.3 Diabetes diagnosis

In this problem taken from the Proben1 database [8], the diabetes state of Pima Indians has to be diagnosed based in 8 parameters including personal data (age, number of times pregnant, etc.) and the results of medical examinations (glucose tolerant test, blood pressure, etc.). The examples set (diabetes1.dt) contains 768 cases, from which 384 ones are used for training, 192 for the validation step and the remaining 192 for testing the generalization performance, as indicated in the database protocol, that we follow to allow further comparisons. The input values are all real, normalized between [0,1] and the diabetes state of the patients is indicated by a boolean variable.

Construct a modular architecture for this problem is by no means clear, in the sense that the problem is not intrinsically modular, as parity, neither is possible to apply the approach used in the expression recognition case as this problem has a single output. The criteria used for selecting the modules was based on a study of the correlation between pairs of inputs, clustering in a same module those with higher values. We use as correlation function the quadratic distance between the input values averaged over all the training examples, as the results were easier to interpret than those obtained with other measures. From the analysis we obtained two groups of three inputs bits clearly defined by the correlation values; for the remaining two inputs, a first idea was to cluster them in another group, but as their correlation was not significant we test two possible architectures containing two modules of four units each, to select the one that gives a best performance. In that way the structure of the network has one hidden layer with two modules comprising four units each, followed by a second hidden layer that contains one output neuron for each module both converging to a single output unit.

In table 2 the results obtained with the modular architecture and also those obtained with two fully connected network with 4 and 8 hidden units are shown. For comparison, it is worth noting that in [8] a generalization error of 24.1% was the lowest generalization error obtained, using a fully connected network. All the networks were trained using backpropagation with learning constant $\eta = 0.05$ and momentum constant $\alpha = 0.5$.

Problem	Architect.	Weigths	Error
Parity	modular	42	0.0 %
	monolithic	72	21.7 %
Face Expr. Recogn.	monolithic	490	15.1 %
	monolithic	255	18.9 %
	monolithic	765	18.6 %
Diabetes Diagn.	modular	44	21.4 %
	monolithic	36	23.2 %
	monolithic	72	24.0 %

Table 2. Generalization error and number of weights of the architectures used for solving the parity function, a face expression recognition task and a diabetes diagnosis case with modular and monolithic networks.

3 Discussion

We analized the generalization ability of modular and fully connected networks on performing three different tasks: parity, face expression recognition and diabetes diagnosis. The initial idea was, using backpropagation as learning algorithm, analyze the behavior of the generalization error for different network sizes. However, it was not possible to obtain solutions for the case of the parity function, despite the fact that the used architecture can compute it. Instead, simulated annealing was used to train the parity networks obtaining a much better generalization ability with the modular architecture than with the monolithic one, confirming the theoretical results [8]. In the other two cases we successfully use backpropagation, obtaining also better results with the modular networks. In table 2 the obtained results are presented. For the face recognition problem a network of the type known as mixture of experts was used, with each module specialized on one expression. The results show an improvement on the generalization ability around 20% compared with fully connected networks, noting that the best values with monolithic architectures were obtained with the largest network. This result support previous empirical ones about the good performance of backpropagation combined with early stopping in large networks [5]. The results are shown in table 2, where for monolithic architectures two values corresponding to different network sizes are displayed, one corresponds to the optimal value found. In the diabetes diagnosis case, at least up to our knowledge, was no clear how to construct a modular network, so we developed an architecture based on the correlations of input values. We compare its performance with the results obtained on similar architectures, but with a different arrangement of the inputs, obtaining better results

with the designed architecture. It will be interesting to validate this method on other problems to see its general applicability. The performance of modular architectures in this case were somewhat greater than when using monolithic architectures, but comparatively lower than in the previous cases. Both in modular and monolithic architectures an optimal size network was found, but being fully connected networks less sensitive to size increasing.

We can conclude saying that restricting the size of a network architecture does not seem to improve much the generalization ability, given that backpropagation combined with early stopping is quite effective with large size networks, but if this reduction on the number of weights is done within a modular architecture a better generalization ability could be achieved. An interesting point for further research is to look for learning algorithms, that perform better than backpropagation on modular networks with many hidden layers, permitting a deeper exploration of their abilities.

Acknowledgments:
L.F. acknowledges partial support from the Human Frontiers Program Grant RG0110/1998-B.

References:

[1] Happel, B.L.M. & Murre, J.M.J. The Design and Evolution of Modular Neural Networks Architectures. *Neural Networks*, No.7, 1994, pp. 985-1004.

[2] Auda, G. & Kamel, M. Modular neural networks: a survey. *International Journal of Neural Systems*, Vol.9, No.2, 1999, pp. 129-151.

[3] Haykin, S. *Neural Networks: A Comprehensive Foundation*, Macmillan/IEEE Press, 1994.

[4] Lawrence, S., Giles, C. L., Tsoi, A. C. *What Size Neural Network Gives Optimal Generalization? Convergence Properties of Backpropagation*. In Technical Report UMIACS-TR-96-22 and CS-TR-3617, Institute for Advanced Computer Studies, Univ. of Maryland, 1996.

[5] Caruana, R., Lawrence, S., Lee Giles, C. *Overfitting in Neural Networks: Backpropagation, Conjugate Gradient, and Early Stopping*. In Advances in Neural Information Processing Systems, Denver, Colorado, 2000.

[6] Franco, L. & Cannas, S.A. Generalization and Selection of Examples in Feedforward Neural Networks. *Neural Computation*, Vol.12, No.10, 2000, pp. 2405-2426.

[7] Franco, L. & Cannas, S.A. Generalization Properties of Modular Networks: Implementing the Parity Function. *IEEE Transactions in Neural Networks*, in press, 2001.

[8] Prechelt, L. PROBEN1 - *A Set of Benchmarks and Benchmarking Rules for Neural Network Training Algorithms*. Technical Report 21/94, Fakultat fur Informatik, Universitat Karlsruhe, Germany, 1994.

[9] Hecht-Nielsen, R. *Neurocomputing*. Addison-Wesley, 1989.

[10] Thornton, C. Parity: the problem that won't go away. In *Proceedings of AI-96*, Toronto, 1996, pp. 362-374.

[11] Pantic, M. & Rothkrantz, L.J.M. Automatic Analysis of Facial Expressions: The State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22, No.12, 2000, pp. 1424-1445.

[12] Kanwisher, N., McDermott, J., & Chun, M.M. The fusiform face area: A module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, No.17, 1997, pp. 4302-4311.

[13] Belhumeur, P.N. & Kriegman, D.J. *The Yale Face Database* URL: http://cvc.yale.edu/projects/yalefaces/yalefaces.html, 1997.