USING FUZZY HETEROGENEOUS NEURAL NETWORKS TO LEARN A MODEL OF THE CENTRAL NERVOUS SYSTEM CONTROL

Lluís A. Belanche Julio J. Valdés
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona Salgado, 1-3, Barcelona 08034, Spain.
Phone: (34-93) 401-5644 Fax: (34-93) 401-7014
{belanche, valdes}@lsi.upc.es

Keywords: Heterogeneous Neural Networks; Fuzzy Logic; Genetic Algorithms; Time-Series Prediction

ABSTRACT: Fuzzy heterogeneous networks based on similarity are recently introduced feed-forward neural network models composed by neurons of a general class whose inputs are mixtures of continuous (crisp and/or fuzzy) with discrete quantities, admitting also missing data. These networks have activation functions based on similarity relations between inputs and neuron weights. They can be coupled with classical neurons in hybrid network architectures, trained with genetic algorithms. This paper compares the effectivity of this fuzzy heterogeneous model based on similarity with the classical feed-forward one (scalar-product driven and using crisp quantities) in a time-series prediction setting. The results obtained show a remarkable increasing performance when departing from the classical neuron and a comparable one when confronted with other current powerful techniques, such as the FIR methodology.

INTRODUCTION

The notion of heterogeneous neurons was introduced in (Valdés *et al.*, 97) as a model accepting as inputs vectors composed by a mixture of continuous real-valued and discrete quantities, possibly also containing missing data. The other feature of this model departuring from the classical was its definition as a general mapping from which different instance models could be derived. In particular, when the model is constructed as the composition of two mappings, different instance models can be derived by making concrete choices of the transfer and activation functions. In this special case, whereas the classical neuron model uses dot product as transfer, and logistic (or hyperbolic tangent) as activation functions, the heterogeneous model uses as transfer a *similarity* or *proximity relation* (Chandon *et al.*, 81) between the input and the weight tuples, and a sigmoid-like automorphism of the reals in $[0, 1]$ as activation. The choice of the specific similarity function should account for the heterogeneous nature of neuron inputs and the presence of missing data. This showed to be a reasonable brick for constructing layered network architectures mixing heterogeneous with classical neurons, since the outputs of these neurons can be used as inputs for the classical ones. Such type of hybrid networks was composed of one hidden layer of heterogeneous networks and one output layer of classical neurons. These networks were able to learn from non-trivial data sets with an effectivity comparable, and sometimes better, than that of classical methods, and exhibited a remarkable robustness when information degrades due to the increasing presence of missing data.

A step further in the development of the heterogeneous neuron model was the inclusion of fuzzy quantities within the input set, extending the former use of real-valued quantities of crisp character. This way, uncertainty and imprecision can be explicitly considered within the model, making it more flexible. In the context of a real-world application it was found that such neurons performed better by treating data with its natural imprecision than considering them as crisp quantities, as is usually the case. Also, the hybrid networks (with or without fuzzy inputs) outperformed those with classical neurons, even when trained with sophisticated procedures like a combination of gradient techniques with simulated annealing (Valdés *et al.*, 98).

The purpose of this paper is then twofold: first, the introduction and study of different hybrid architectures (composed of classical and heterogeneous neurons) and full heterogeneous architectures, not studied before in the aforementioned work. Second, the application of these architectures to a problem of different flavour than the pure classification tasks addressed so far. For the present study, the availability of cardiology data from a patient and the knowledge of previous attempts to induce accurate models out of these data using *Fuzzy Inductive Reasoning* (Nebot *et al.*, 98; Cueva *et al.*, 97) brought an opportunity to confront the Heterogeneous Neural Network (HNN) with them.

THE FUZZY HETEROGENEOUS NEURON MODEL

A fuzzy heterogeneous neuron was defined in (Valdés *et al.*, 98) as a mapping $h : \hat{\mathcal{H}}^n \to \mathcal{R}_{out} \subseteq \mathbb{R}$, satisfying $h(\phi) = 0$ (ϕ is the empty set). Here $\mathbb{R}$ denote the reals and $\hat{\mathcal{H}}^n$ is a cartesian product of an arbitrary number of *source sets*. Source sets may be families of extended reals $\hat{\mathcal{R}} = \mathbb{R} \cup \{\mathcal{X}\}$, extended fuzzy sets $\hat{\mathcal{F}}_i = \mathcal{F}_i \cup \{\mathcal{X}\}$, and extended finite sets of the form $\hat{\mathcal{O}}_i = \mathcal{O}_i \cup \{\mathcal{X}\}$, $\hat{\mathcal{M}}_i = \mathcal{M}_i \cup \{\mathcal{X}\}$, where each of the $\mathcal{O}_i$ has a full order relation, while the $\mathcal{M}_i$ have not. In all cases, the special symbol $\mathcal{X}$ denotes the unknown element (missing information) and it behaves as an incomparable element w.r.t. any ordering relation. According to this definition, neuron inputs are possibly empty arbitrary tuples, composed by n elements among which there might be reals, fuzzy sets, ordinals, nominals and missing data. The form of the resulting input set for a fuzzy heterogeneous neuron is $\hat{\mathcal{H}}^n = < \hat{\mathcal{R}}^{n_r}, \hat{\mathcal{F}}^{n_f}, \hat{\mathcal{O}}^{n_o}, \hat{\mathcal{M}}^{n_m} >$, with $\hat{\mathcal{R}}^0 = \hat{\mathcal{F}}^0 = \hat{\mathcal{O}}^0 = \hat{\mathcal{M}}^o = \hat{\mathcal{H}}^o = \phi$, $n = n_r + n_f + n_o + n_m$ and $n > 0$ (n_r, n_o and n_m are the lengths of the corresponding cartesian products formed using the source sets $\hat{\mathcal{R}}_i, \hat{\mathcal{F}}_i, \hat{\mathcal{O}}_i, \hat{\mathcal{M}}_i$). Heterogeneous neurons will be classified according to the nature of its image set (which must not be necessary restricted to a subset of the reals). In the present study, a model with an image set given by $\mathcal{R}_{out}$ will be called of the *real kind*. The reason to consider in a first place neuron models of the real kind is their natural coupling with other classical neuron models (i.e. accepting only real inputs), thus leading to hybrid networks in a straightforward way.

A particular class of heterogeneous submodels of the real kind was constructed by considering h as the composition of two mappings, that is, $h = f \circ s$, such that $s : \hat{\mathcal{H}}^n \to \mathcal{R}' \subseteq \mathbb{R}$ and $f : \mathcal{R}' \to \mathcal{R}_{out} \subseteq \mathbb{R}$. The mapping h can be considered as a $n - ary$ function parameterized by a $n - ary$ tuple $\hat{\vec{w}} \in \hat{\mathcal{H}}^n$ representing neuron's weights, i.e. $h(\hat{\vec{x}}, \hat{\vec{w}}) = f(s(\hat{\vec{x}}, \hat{\vec{w}}))$. In particular, function s represents a *similarity* and f a squashing non-linear function with its image in $[0, 1]$. Accordingly, the neuron is sensitive to the degree of similarity between its input, composed in general by a mixture of continuous and discrete quantities possibly with missing data. More precisely, s is understood as a *similarity index*, or proximity relation (transitivity considerations are put aside). That is, a binary, reflexive and symmetric function $s(x, y)$ with image on $[0, 1]$ such that $s(x, x) = 1$ (strong reflexivity). The semantics of $s(x, y) > s(x, z)$ is that object y is more similar to object x than z. Clearly, there are many possible choices for the s function, and in particular some of them are currently under investigation. The concrete instance of the model under study in the present paper uses as transfer function a *Gower-like* similarity index (Gower, 71) in which the computation for heterogeneous entities is constructed as a weighted combination of partial similarities over subsets of variables. This coefficient has its values in the real interval $[0, 1]$ and for any two objects i, j given by tuples of cardinality n, is given by

$$s_{ij} = \frac{\sum_{k=1}^{n} g_{ijk}\, \delta_{ijk}}{\sum_{k=1}^{n} \delta_{ijk}}$$

where g_{ijk} is a similarity *score* for objects i, j according to their value for variable k. These scores are in the interval $[0, 1]$ and are computed according to different schemes for numeric and qualitative variables. The factor δ_{ijk} is a binary function expressing whether objects i, j are comparable or not according to their values w.r.t. variable k. It is 1 iff both objects have values different from $\mathcal{X}$ for variable k, and 0 otherwise. Additional weights representing the differential importance of variables can be introduced in the computation of this similarity, but they will not be considered here. Rather, Gower's original definitions for real-valued and discrete variables are kept. For variables representing fuzzy sets, similarity relations from the point of view of fuzzy theory have been defined elsewhere (Klir *et al.*, 88; Zimmermann, 92; Dubois *et al.*, 97) and different choices are possible. In our case, if $\mathcal{F}_i$ is an arbitrary family of fuzzy sets from the source set, and $\tilde{A}$, $\tilde{B}$ are two fuzzy sets such that $\tilde{A}$, $\tilde{B} \in \mathcal{F}_i$, the following similarity relation is used

$$g(\tilde{A},\, \tilde{B}) = \max_{x}\, (\mu_{\tilde{A} \cap \tilde{B}}(x))$$

where $\mu_{\tilde{A} \cap \tilde{B}}(x) = min(\mu_{\tilde{A}}(x),\, \mu_{\tilde{B}}(x))$. For the activation function, a modified version of the classical logistic is used, which is an automorphism of the real interval $[0, 1]$.

$$f(x, p) = \begin{cases} \frac{-p}{(x-0.5)-a(p)} - a(p) & \text{if } x \leq 0.5 \\ \frac{-p}{(x-0.5)+a(p)} + a(p) + 1 & \text{otherwise} \end{cases}$$

where $a(p)$ is an auxiliary function given by $a(p) = \frac{-0.5+\sqrt{0.5^2+4*p}}{2}$ and p is a real-valued parameter controlling the curvature, set in the experiments to 0.1. This family of functions is displayed in figure 1. The training procedure for the HNN is based on genetic algorithms (GAs), since the heterogeneity of the variables involved and the non-differentiability of the similarity function prevent the use of gradient-based techniques.

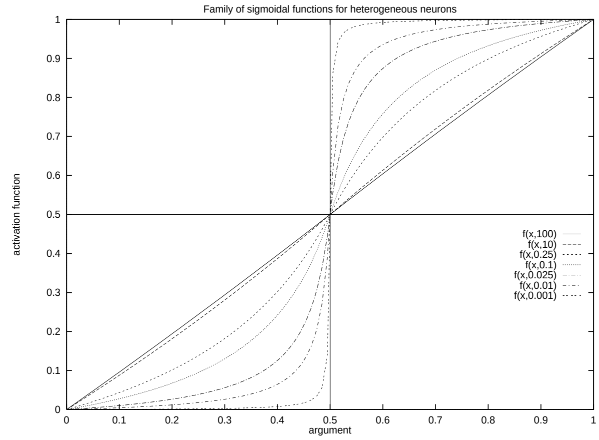

Figure 1: The family of sigmoidal functions $f(x, p)$.

DESCRIPTION OF THE PROBLEM

The cardiovascular system is composed of the hæmodynamical system and the Central Nervous System (CNS) control. Whereas the structure and functioning of the hæmodynamical system are well known and a number of quantitative models have already been developed that capture its behavior fairly accurately, the CNS control is, at present, still not completely understood and no good deductive models exist able to describe the CNS control from physical and physiological principles. The use of other approaches –like qualitative methodologies or neural networks– may offer an interesting alternative to classical quantitative modeling approaches –such as differential equations and NARMAX techniques– for capturing the behavior of the CNS control. The CNS control model is composed of five separate controllers: the heart rate controller, the peripheral resistance controller, the myocardial contractility controller, the venous tone controller, and the coronary resistance controller. All five controller models are single–input/single–output (SISO) models driven by the same input variable, the *Carotid Sinus Pressure*. In the present study, we will concentrate on the first of these signals, the *heart rate*. The input and output signals of the CNS control were recorded with a sampling rate of 0.12 seconds from simulations of the purely differential equation model. The model had been tuned to represent a specific patient suffering from an at least 70% coronary arterial obstruction, to agree with the measurement data taken from the patient. The full set of data consists of 7869 timed measurements. From these, the first 1500 were used as training set and the immediately following 1000 as the test set to be forecasted. To give a graphical impression, the input and output variables of the *heart rate* controller subsystem are displayed in figures 2 and 3. Note that both signals exhibit high-frequency oscillations modulated by a low-frequency signal.

EXPERIMENT SETUP AND RESULTS

The use of the heterogeneous neuron as a brick for configuring network architectures can be done in several ways. In this paper, several architectures will be explored, in an attempt to clarify the effect of combining such neurons with classical ones. In order to keep things manageable, however, we will restrict ourselves to networks with one or no hidden layers. To this end, let us introduce the notation q_x to denote a single layer of q neurons, where possibilities for x are: n (classical, scalar product and logistic activation), h (heterogeneous) and f (fuzzy heterogeneous). Similarly, $p_x q_x$ denotes a network composed of a hidden layer of p neurons and an output layer of q neurons. In the present study, there is just one output to be predicted (the *heart rate*) so the output layer is always composed of a single neuron and the hidden layer (if any) will always have 3 neurons[1]. The

[1]It should be noted that there has been no attempt to find better architectures (different number of hidden-layer neurons and/or more than one hidden layer) nor to improve GA performance on this particular problem by tuning its parameters or devising

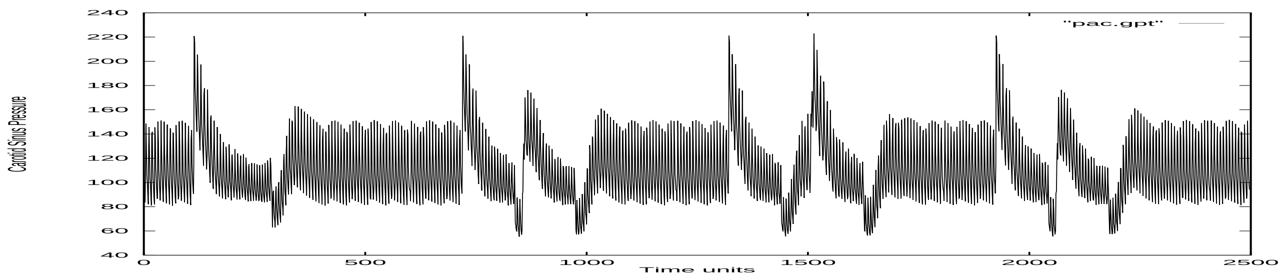

Figure 2: Input signal: *Carotid Sinus Pressure.*

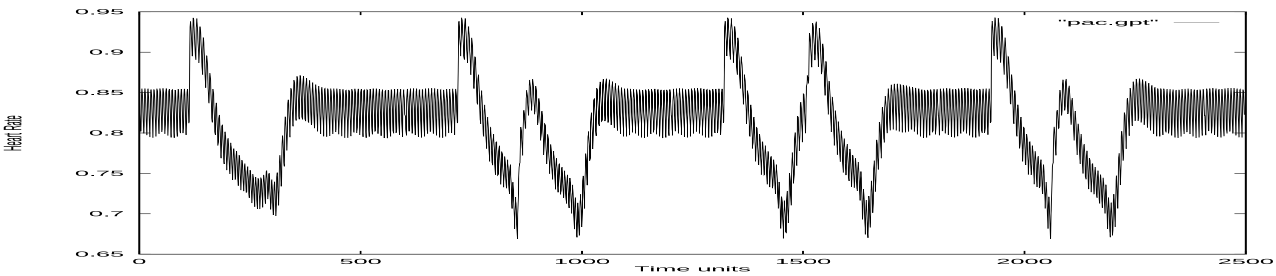

Figure 3: Output signal: Heart Rate Controller, measured in seconds between beats.

architectures under study will then be: $1_n, 1_h, 1_f, 3_n1_n, 3_h1_n, 3_f1_n, 3_n1_h, 3_n1_f, 3_h1_h$ and 3_f1_f. All of them were trained with the same GA, in order to eliminate this source of variation from the analysis. The only difference between h and f neurons is that –according to the fuzzy HNN model presented– the latter have their inputs and weights fuzzyfied. In this experiment, original crisp data were converted into (triangular) fuzzy numbers in the form of a 5% of imprecision w.r.t. the original value[2]. In order to assess the performance of the different HNN architectures, two other soft-computing approaches were also employed to infer a model for the task at hand: a feed-forward neural network working in the complex plane and the aforementioned FIR methodology. A brief comment on both is due.

The complex neural network (CNN) is an advanced model (Birx *et al.*, 92,93) which operates in the complex plane, having inputs, weights and outputs given by complex numbers. They have been used very successfully in the analysis of many complex dynamic systems and in difficult classification problems. In these networks, the transfer function is a direct translation of the scalar product to complex arithmetic. Let $z = x + iy \in \mathbb{C}$ be the complex neuron net output as given by the scalar product. The squashing function used is given by $f(z) = f(x+iy) = p\,x + i\,p\,y$ where $p = \frac{tanh(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$. Following our terminology, we will use a 3_c1_c architecture, where c denotes a complex neuron. In this case, the output neuron will use a linear activation function. The training procedure chosen is a powerful combination of simulated annealing with conjugate gradient-descent.

The inductive reasoning methodology was first developed (Klir, 85) as a tool for general system analysis. Fuzzy measures were introduced in the late eighties, giving way to the FIR methodology. In the FIR approach, the qualitative systems are modeled by a special class of finite state machines defined by an *optimal mask* and a *behavior matrix*, and the episodical behavior of the system is simulated by a technique called *fuzzy forecasting*. In previous studies of the data at hand, FIR capabilities were greatly increased by performing a Markov analysis in search of single-dependency variable-order significant time delays (Nebot *et al.*, 98). It was found that in both input $x(t)$ and output $y(t)$ signals there were two specific time delays (1 and 6 sampling intervals), highly significant from the point of view of exhibiting a Markov chain behavior when the continuous process is discretized. According to this, a training set consisting of four inputs $x(t-1), x(t-6), y(t-1), y(t-6)$ and one output $y(t)$ was constructed. This hybrid technique has been very successfully applied to the task at hand though using data from different patients and controllers and different training and test set regimes. Hence, for the purpose of direct comparison, new runs of the FIR methodology were performed.

For each neural architecture, five different training trials were run using different random initial populations,

specialized operators. Although it is reasonable to believe that this would probably improve the results obtained with the complex network (presented later on) and (specially) the HNN, our main concern was to compare both networks in a crude (perhaps more fair) way, using reasonable (although maybe not optimal) settings.

[2] This percentage is probably an upper-bound for modern measuring devices.

Arch.	1_n	1_h	1_f	$3_n 1_n$	$3_h 1_n$	$3_f 1_n$	$3_n 1_h$	$3_n 1_f$	$3_h 1_h$	$3_f 1_f$
Avg.	2.150e-02	9.855e-04	5.723e-04	1.640e-03	1.114e-04	7.817e-05	2.621e-03	1.594e-03	1.661e-04	7.683e-05
Best	9.965e-04	9.657e-04	3.510e-04	1.216e-03	9.405e-05	6.652e-05	2.603e-03	1.036e-03	9.424e-05	6.527e-05

Table 1: MSE errors for the different HNN architectures.

in an attempt to reduce the effect of a specially lucky (or unlucky) strike by the GA. Average and best MSEs on *test* set were then calculated. The CNN was given 50 different annealing tries and the final (overall best) result is the one displayed (tables 1 and 2). Notice the decrease of MSE in orders of magnitude due to the increasing presence of heterogeneous neurons, until a comparable (and slightly better) performance to the obtained by the FIR and the CNN is reached. Although this final error is not the best, it is probably very close.

Architecture	**FIR**	$3_c 1_c$
Best	2.095e-04	8.136e-05

Table 2: MSE errors for FIR and the CNN.

CONCLUSIONS AND FUTURE WORK

The study and prediction of time-varying processes is a fundamental problem with a long tradition in the literature. In this paper, we have shown how the use of fuzzy heterogeneous networks can significatively increase the accuracy of the models obtained. These networks have been compared to the FIR methodology and a complex neural network for the task at hand, and the overall results are promising. Clearly, a full study with a richer set of data from other patients and controllers deserves future work. Most important, the use of GAs to find the weights of the networks is not perhaps the best choice. As general and versatile function optimizers, these algorithms constituted our initial choice. Although a finer tuning of the GA parameters and a clever design of specialized genetic operators would definitely improve performance (as a consequence of introducing problem-specific knowledge) we believe that other evolutionary techniques (namely, a form of evolution strategies) would better suit the problem of minimizing a (non-differentiable) error function that has such an heterogeneity in its constituting variables (recall both inputs and weights are heterogeneous), many of them continuous, difficult for the binary coding of the GA. Another possibility would be to devise a differentiable similarity function, so the bunch of gradient-descent methods would be available, and a definite comparison with the best of classical neural networks could be made. These two points are the subject of our future research.

REFERENCES

Birx, D., Pipenberg, S. 1992. *Chaotic oscillators and complex-mapping feedforward networks for signal detection in noisy environment.* Int. Joint Conf. on Neural Networks, Baltimore, MD.

Birx, D., Pipenberg, S. 1993. *A complex-mapping network for phase sensitive classification.* IEEE Trans. on Neural Networks, **4**:1, pp 127-135.

Chandon, J.L, Pinson, S. 1981. Analyse Typologique. Théorie et Applications. Masson, pp. 254.

Cueva J., Alquézar R., Nebot A. 1997. *Experimental comparison of fuzzy and neural network techniques in learning models of the central nervous system control.* Proc. EUFIT'97, 5th European Congress on Intelligent Techniques and Soft Computing, pages 1014-1018, Aachen, Germany.

Dubois D., Prade H., Esteva F., García P., Godo L., López de Mántaras R. 1997. Fuzzy set modeling in case-based reasoning. *Int. Journal of Intelligent Systems* (to appear).

Gower, J.C. 1971. A General Coefficient of Similarity and some of its Properties. *Biometrics 27, 857-871.*

Klir, G.J. 1985. *Architecture of Systems Problem Solving.* Plenum Press, New York.

Klir, G.J., Folger T.A. 1988. Fuzzy Sets, Uncertainty and Information. Prentice Hall Int. Editions.

Nebot A., Valdés J.J., Guiot M, Alquézar R., Vallverdú M. 1998. *Fuzzy inductive reasoning approaches in the identification of models of the central nervous system control.* Proc. EIS'98, Int. Conf. on Engineering of Intelligent Systems, Tenerife.

Valdés J.J, Belanche, Ll. 1998. Fuzzy Heterogeneous neurons for Imprecise Classification Problems. LSI Research Report LSI-98-33-R. Universitat Politècnica de Catalunya. Dept. of Languages and Information Systems.

Valdés J.J, García R. 1997. A model for heterogeneous neurons and its use in configuring neural networks for classification problems. *International World Conference on Artificial and Natural Neural Networks* IWANN'97, Lecture Notes in Computer Science 1240, Springer Verlag, pp 237-246.

Zimmermann H.J. 1992. Fuzzy set theory and its applications. Kluver Academic Publishers, p. 399.