A Parsimonious Generation of Combinatorial Neural Model

Hércules A. Prado*, Sandra R.Frigeri**, Paulo M. Engel***

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Av. Bento Gonçalves, 9500 – Bairro Agronomia
Porto Alegre / RS - Brasil
Caixa Postal 15.064 – CEP 91.501-970
Fone: +55(051)316-6829
Fax: +55(051)319-1576
e-mail: prado, rovena, engel +{@inf.ufrgs.br}

Abstract

This paper presents a new approach to reduce the space problem due to combinatorial explosion of CNM (Combinatorial Neural Model) method. First we show a description of CNM, proposed by Machado and Rocha [MAC 91], [MAC 92], [MAC 92a], [MAC 97], as a variation of fuzzy neural network introduced as an alternative to meet many requirements, such as expressiveness, inteligibility, plasticity and flexibility. Our approach represents an alternative to generate the CNM network with certainty factors for each hypothesis. We demonstrate by means of a simple practical example that the number of combinations can be really reduced.

Keywords: Data Mining, Knowledge Discovery from Databases, Supervised Learning, Hybrid Systems, Neural Networks

* Lecturer at Universidade Católica de Brasília and researcher at EMBRAPA – Empresa Brasileira de Pesquisa Agropeucária
** Lecturer at Universidade de Caxias do Sul
*** Professor at Universidade Federal do Rio Grande do Sul

A Parsimonious Generation of Combinatorial Neural Model

1. Introduction

Classification systems based on symbolic-connectionist hybrid architectures have been proposed, e.g. [HUD 92], [KNA 92] and [GUA 94], as a way of obtaining benefits from the specific characteristics of both models. The associative characteristics of artificial neural networks (ANN) and the logical nature of symbolic systems have led to easier learning and the explanation of the acquired knowledge.

This work addresses one of such architectures, the Combinatorial Neural Model, introduced by Machado and Rocha [MAC 91], [MAC 92], [MAC 92a], [MAC 97], presenting an alternative to cope with one of its major problems: the combinatorial explosion of CNM network as the number of attributes increases. This approach is illustrated through an example of application in agricultural research. By using a real training set, the total space of original CNM network is shown and then we present the possible reduction of this space as a consequence of using our approach.

2. Description of CNM

CNM is a hybrid architecture for intelligent systems that integrates symbolic and connectionist computational paradigms. It has some significant issues, such as the ability to build a neural network from background knowledge; incremental learning by examples, solving the plasticity-stability dilemma [FRE 92]; a way to cope with the diversity of knowledge; knowledge extraction of an ANN; and the ability to deal with uncertainty. CNM is able to recognize regularities from high-dimensional symbolic data, performing mappings from this input space to a lower dimensional output space.

CNM uses supervised learning and a feedforward topology with: one input layer, one hidden layer - here called combinatorial - and one output layer (FIGURE 2.1). Each neuron of the input layer corresponds to a concept - a complete idea about an object of the domain, expressed by an object-attribute-value form, they represent the evidences of the domain application. On the combinatorial layer there are aggregator type neurons, each one connected to one or more neurons of the input layer by fuzzy AND arcs that represent logical concepts. The output layer contains one neuron for each possible class (also called hipothesis), linked to one or more neurons on the combinatorial layer by fuzzy OR arcs that also represent concepts. The synapses may be excitatory or inhibitory and they are characterized by a strength value (*weight*) between zero (not connected) to one (fully connected synapses), that can express the logical relations. For the sake of simplicity, we will work with the learning of crisp relations, thus with strenght value of synapses equal to one, when the concept is present, and zero, when the concept is not present. However, this option does not affect the approach to fuzzy relations learning.

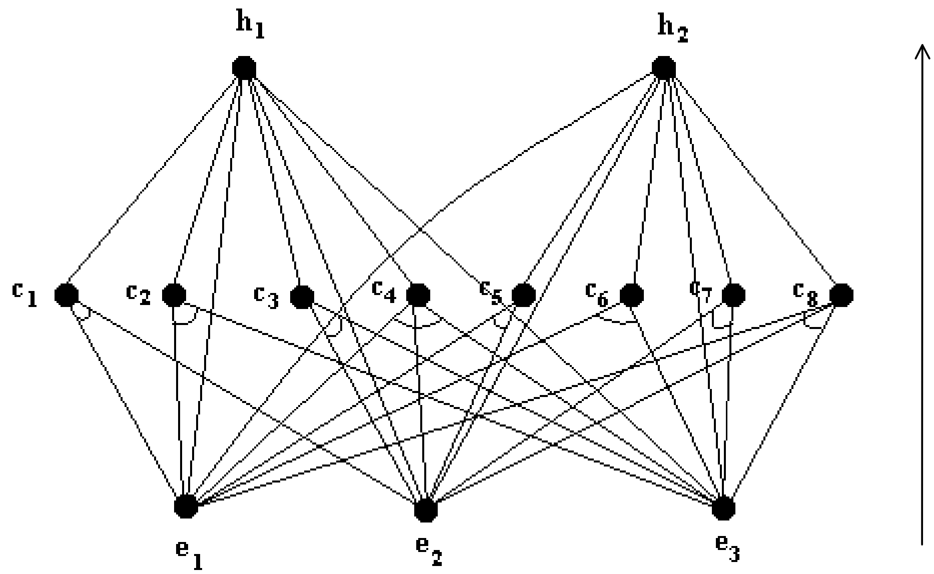

FIGURE 2.1 - The complete version of the combinatorial network for 3 input evidences and 2 hypotheses [MAC 91]

The network is created completely empty, according to the following steps: (a) one neuron in the input layer for each evidence in the training set; (b) a neuron in the output layer for each class in the training set; and (c) for each neuron in the output layer, there is a complete set of hidden neurons in the combinatorial layer which correspond to all possible combinations (lenght between two and nine) of connections with the input layer. There is no neuron in the combinatorial layer for length one connections. In this case, input neurons are connected directly to hypotheses.

The learning mechanism works in only one iteration, and it is described bellow:

PUNISHMENT_AND _REWARD_LEARNING_RULE

- *Set* to each arc of the network an accumulator with initial value zero;
- **For each** example case from the training data base, **do**:

  Propagate the evidence beliefs from input nodes until the hypotheses layer;

  For each arc reaching a hypothesis node, **do**:

    If the reached hypothesis node corresponds to the correct class of the case

Then *backpropagate* from this node until input nodes, increasing the accumulator of each traversed arc by its evidencial flow (Reward)

Else *backpropagate* from the hypothesis node until input nodes, decreasing the accumulator of each traversed arc by its evidencial flow (Punishment).

After training, the value of accumulators associated to each arc arriving to the output layer will be between [-T, T], where T is the number of all cases present in the training set.

The last step is the prunning of network; it is performed by the following actions: (a) remove all arcs whose accumulator is lower than a threshold (specified by a specialist); (b) remove all neurons from the input and combinatorial layers that became disconnected from all hypotheses in the output layer; and (c) make weights of the arcs arriving at the output layer equal to the value obtained by dividing the arc accumulators by the largest arc accumulator value in the network. After this prunning, the network becomes operational for classification tasks.

3. The Problem

Despite CNM is a simple model, it has many worthy features, as seen in the previous section. However, it has some weaknesses that limit its use, like:

- in the initial phase, the generation of the network completely empty, representing all possible combinations for each hipothesis, is clearly unfeasible as recognized by the author of the model.
- the full generation of all combinations of attribute-values may create many unreal hypotheses in respect to majority applications.
- as a consequence of its knowledge representation form, CNM has its expressivity limited to Propositional Logic.

In the first paper [MAC 89], the authors suggest the control of combinatorial explosion of the nodes in the hidden layer by incrementally building of the network. The mechanism starts with a low combination order and increases the order to an upper one until an arbitrary limit. The author suggests a criterium based on the “magic number” of Miller [MIL 56], seven plus or minus two, to stablish the upper bound to the order of combinations. Some works [LEA 93] and [FEL 97] address the same problem - the combinatorial explosion. Although they reach combinations of higher order, the search in the solution space is, as a rule, limited by the rapid growing of the network. Our approach is addressed to this problem too and may be seen as an alternative that can increase expressively the order of generated combinations and reduce the growing of the network.

4. Our approach for building CNM network

This section presents our approach to generate the CNM, that may reduce the cost of the algorithm in terms of space, and that we call *parsimonious generation of CNM network*. By this approach, the neurons and the connections are created only by contingence, i.e., only when required by an example in the training set. Moreover, during the training phase, it is only computed the rewarding of the arcs arriving at the correct class. There is no punishment. The computation of the effect of misclassifications is done by calculating the difference between the value of each accumulator at the end of the training, for each combination, and the value of the other accumulators for the same combination related to different classes.

Let us take the example of the training set used in the original proposal [MAC 89], shown in TABLE 4.1.

TABLE 4.1 **-** Patients with diseases and associated symptoms

Name	Symptoms	Disease
John	s1, s2, s3	d1
Diana	s1, s2, s4	d1
Mary	s1, s3, s4	d2
Peter	s2, s3, s4	d2

In the original approach, the expansion of the network based on this training set produces the combinations shown in TABLE 4.2. During the initial phase - creation of the empty network – twenty eight combinations were generated. After prunning with threshold one, ten combinations remain, and with two as threshold, two remain. Using our approach, according to TABLE 4.3, twenty two combinations are generated, and the same quantities remain - ten and two - after the prunning with threshold one and two, respectively.

The algorithm proposed for generation of the CNM takes the following form:

- **For each** example in the training set, **do**:

Compute all possible combinations based on the example

For each computed combination, **do:**

If there is an equal combination in the network arriving to the same class

Then *add* one to the accumulator of the arc arriving (Reward)

Else *include* an arc corresponding to the actual combination, setting the accumulator to one (Reward)

To compute the final value of the accumulators, the result of the folllowing operations is taken: for each accumulator of each combination, take its value as ACC; for all combinations equal to the precedent one pointing to classes different from the precedent, sum their accumulators, calling it SUM; the final result of ACC is given by ACC = ACC - SUM. It is equivalent to punishments of the original algorithm, in only one passing. Both training and accomplishment of the final value of accumulators are easily traced through TABLES 4.1 and 4.3.

TABLE 4.2 - Effects of training and prunning of the CNM

	Symptoms					Accumulators				Thrshld/ Prun.	
Disease	s1	s2	s3	s4	Begin	Jo	Di	Ma	Pe	1	2
d1	X				0	1	2	1	1	1	-
		X			0	1	2	2	1	1	-
			X		0	1	1	0	-1	-	-
				X	0	0	1	0	-1	-	-
	X	X			0	1	2	2	2	2	2
	X		X		0	1	1	0	0	-	-
	X			X	0	0	1	0	0	-	-
		X	X		0	1	1	1	0	-	-
		X		X	0	0	1	1	0	-	-
			X	X	0	0	0	-1	-2	-	-
	X	X	X		0	1	1	1	1	1	-
	X	X		X	0	0	1	1	1	1	-
	X		X	X	0	0	0	-1	-1	-	-
		X	X	X	0	0	0	0	-1	-	-
d2	X				0	-1	-2	-1	-1	-	-
		X			0	-1	-2	-2	-1	-	-
			X		0	-1	-1	0	1	1	-
				X	0	0	-1	0	1	1	-
	X	X			0	-1	-2	-2	-2	-	-
	X		X		0	-1	-1	0	0	-	-
	X			X	0	0	-1	0	0	-	-
		X	X		0	-1	-1	-1	0	-	-
		X		X	0	0	-1	-1	0	-	-
			X	X	0	0	0	1	2	2	2
	X	X	X		0	-1	-1	-1	-1	-	-
	X	X		X	0	0	-1	-1	-1	-	-
	X		X	X	0	0	0	1	1	1	-
		X	X	X	0	0	0	0	1	1	-

Jo=John, Di=Diana, Ma=Mary, Pe=Peter

TABLE 4.3 - Parsimonious generation of CNM

	Symptoms					Accumulators					Thrshd/Prun	
Disease	s1	s2	s3	s4	Begin	Jo	Di	Ma	Pe	Acc	1	2
	X				0	1	2			1	1	-
		X			0	1	2			1	1	-
			X		0	1	1			-1	-	-
	X	X			0	1	2			2	2	2
	X		X		0	1	1			0	-	-
d1		X	X		0	1	1			0	-	-
	X	X	X		0	1	1			1	1	-
				X	0		1			-1	-	-
	X			X	0		1			0	-	-
		X		X	0		1			0	-	-
	X	X		X	0		1			1	1	-
	X				0			1	1	-1	-	-
			X		0			1	2	1	1	-
				X	0			1	2	1	1	-
	X		X		0			1	1	0	-	-
d2	X			X	0			1	1	0	-	-
			X	X	0			1	2	2	2	2
	X		X	X	0			1	1	1	1	-
		X			0				1	-1	-	-
		X	X		0				1	0	-	-
		X		X	0				1	0	-	-
		X	X	X	0				1	1	1	-

5. Example

In this example we use data related to the use of pesticides in São Paulo*, during 1994, according to FIGURE 5.1. Training is accomplished over attributes *city, crop, disease, pesticide,* and *quantity*, described below:

city: Code of the city where the pesticide was applied.
There are 120 cities.

crop: Code of the crop that received the pesticide.
There are 27 crops.

disease: Code of the disease being treated.
There are 55 diseases.

pesticide: Code of the pesticide applied.
There are 140 pesticides.

* Data obtained by agreement between EMBRAPA Environment and CREA-SP.

quantity: It is the target attribute and indicates pesticides level applied in one city. Domain={High, Medium, Low}.

According to the original version of CNM, disregarding combinations between different values from the same attribute, combinations shown in TABLE 5.2 are generated.

TABLE 5.2 - Generated combinations for each hypothesis through CNM

Combinations of 2 attributes:	
city and *crop:*	3,240
city and *disease:*	6,600
city and *pesticide:*	16,800
crop and *disease:*	1,485
crop and *pesticide:*	3,780
disease and pesticide:	7,700
Combinations of 3 attributes:	
city, crop and disease:	178,200
city, crop and *pesticide:*	453,600
city, disease and pesticide:	924,000
crop, disease and pesticide:	207,900
Combinations of 4 attributes:	
city, crop, disease and pesticide:	24,948,000
Total of combinations for each hypothesis:	26,751,305

Considering that we have 3 hypotheses, the total amount of generated combinations, with empty network, is 80,253,915. The parsimonious generation of the network, with the same training set, produced only 5,152 combinations, representing a drastic redution on the number of generated combinations. FIGURE 5.1 shows the summary of the training, listing total combinations generated. In other training sets this gain may be lower, but it is possible that in almost all cases a considerable gain will be obtained.

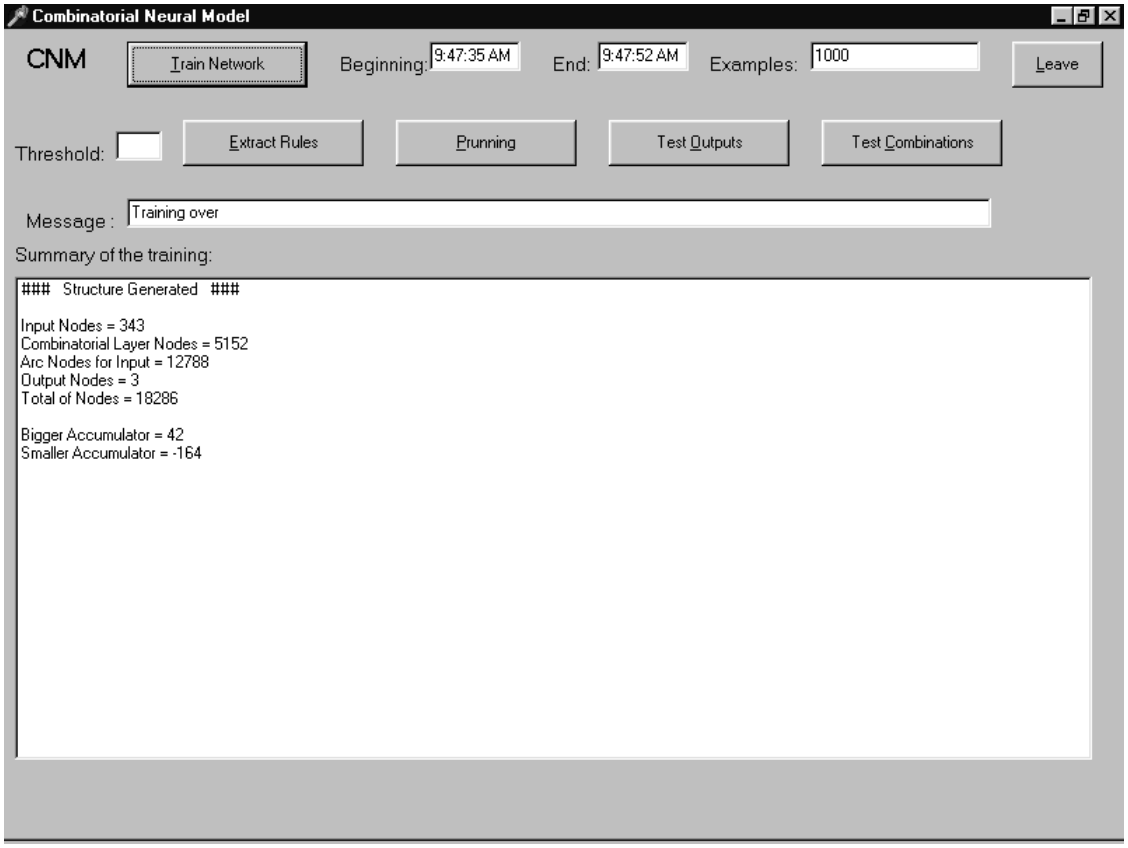

FIGURE 5.1 - Partial outputs of the parsimonious generation of CNM

6. Conclusions

By the presented approach it is never created unnecessary arcs in the network; this fact leads to the generation of trained networks smaller than the original proposal [MAC 91], [MAC 92], [MAC 92a], [MAC 97], and other approachs [LEA 93] and [FEL 97]. On the other hand, the final network obtained after prunning phase is the same in all alternatives. The main problem of space occurs in the training phase and our approach reduces this problem.

During training phase, gain in terms of space, provided by this proposal, presents a compensation through cost increase to compute the final value of accumulators, since it is necessary to identify equal combinations for different hypotheses. However, such way of building the network may be considered an alternative when the main restriction is space.

The space complexity of this proposal will be, at worst case, equal to that of the original one. In other words, if all possible combinations are associated to all possible hypotheses in the training set, the required space to build the CNM in both alternatives, will be the same. In any other situations, the present proposal will generate a smaller network.

Bibliographic References:

[FEL 97] FELDENS, M. A. & CASTILHO, J. M. V. Data mining with the combinatorial rule model: an application in a health-care relational database. In: CLEI, 23., 1997, Valparaízo, Chile. **Proceedings ...** Valparaízo: CLEI, 1997.v. 1, p. 135–145.

[FRE92] FREEMAN, James A.; SKAPURA, David M. Adaptive Resonance Theory. In: **Neural Networks, Algorithms, Applications, and Program Techniques.** Reading: Addison-Wesley, 1992. 401p. p.292-339.

[GUA 94] GUAZZELLI, A. **Aprendizagem em Sistemas Híbridos.** Porto Alegre: CPGCC da UFRGS, 1994. Dissertação de mestrado.

[HUD 92] HUDSON, D. L. *et al.* Medical diagnosis and treatment plans derived from a hybrid expert system. In: KANDEL, A. & LANGHOLZ, G**. Hybrid architectures for intelligent systems**, Boca Raton, FL: CRC Press, 1992.

[KNA 92] KNAUS, R. Representing expert knowledge in neural nets. In: KANDEL, A. & LANGHOLZ G. **Hybrids Architectures for Intelligent Systems**, Boca Raton, FL: CRC Press, 1992.

[LEA 93] LEÃO, B. F.; REÁTEGUI, E. B. A hybrid connectionist expert system to solve classification problems. **Proceedings of Computer in Cardiology,** IEEE Computer, IEEE Computer Society, London, 1993.

[MAC 89] MACHADO, R. J.; ROCHA, A. F. **Handling knowledge in high order neural networks: the combinatorial neural network.** Rio de Janeiro: IBM Rio Scientific Center, 1989. (Technical Report CCR076) .

[MAC 91] MACHADO, R. J. & ROCHA, A. F. da. The combinatorial neural network: a conectionist model for knowledge based systems. In: BOUCHON, B.; YAGER, R. R.; ZADEH, L. A. **Uncertainty in Knowledge Bases**, Berlin, Germany: Springer Verlag, 1991, p.578-587.

[MAC 92] MACHADO, R. J. & ROCHA, A. F. da. **Evolutive fuzzy neural networks**. In: IEEE International Conference on Fuzzy Systems, San Diego, CA, p.493-500. Mar.1992.

[MAC 92a] MACHADO, R. J.; ROCHA, A. F. da. A Hybrid architecture for fuzzy connectionist expert systems. In: KANDEL A. & LANGHOLZ G. **Hybrids Architectures for Intelligent Systems**, Boca Raton, FL: CRC Press, 1992. p. 136-152.

[MAC 97] MACHADO, R. J.; ROCHA, A. F. da. Inference, inquiry, evidence censorship, and explanation in connectionist expert systems. **IEEE Transactions on Fuzzy Systems**, New York, v. 5, n. 3, p.443-459, Aug. 1997.

[MIL 56] MILLER, G. A., The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. **Psycological Review**, [s.l.], 63, p.81-97, 1956.