

**Artigo Original**

recebido em 21/02/2001 e aceito em 13/12/2002

**Técnica de extração de regras  
para sistemas especialistas  
conexionistas**

*Technique for rules extraction for  
connexionist expert systems*

**Lourdes Mattos Brasil**

Professor Adjunto RTC e  
Diretora do Curso de Engenharia Biomédica  
Universidade Católica de Brasília (UCB)  
Pró-Reitoria de Graduação  
Centro de Ciência e Tecnologia (CTC)  
Pró-Reitoria de Pós-Graduação e Pesquisa  
Mestrado em Gestão do Conhecimento  
e da Tecnologia da Informação (MGCTI)  
  
Campus II - SGAN 916 - Módulo B - Sala A 110  
Asa Norte - Brasília - DF - 70790-160  
Tel: 0055-61-3405550/Ramal 211 Fax: 0055-61-3474797  
e-mail: lmb@pos.ucb.br

**Fernando Mendes de Azevedo**

Professor Adjunto IV e Coordenador do Instituto de  
Engenharia Biomédica (IEB-UFSC),  
Depto. de Engenharia Elétrica (EEL),  
Universidade Federal de Santa Catarina (UFSC),  
Florianópolis, SC, Brasil, 88040-900  
e-mail: azevedo@ieb.ufsc.br

**Resumo**

Explicação ao usuário é uma função importante em Sistemas Especialistas (SE) principalmente quando aplicados à área médica. Experiências com SE, sejam estes Baseados em Regras (RBES) ou em Redes Neurais Artificiais (NNES), têm mostrado que a capacidade para gerar explicações é absolutamente crucial para sua aceitação por parte de usuários médicos. Tradicionalmente, Redes Neurais Artificiais (RNA) e, conseqüentemente, os NNES têm dificuldades para a geração de estruturas de explicações. Assim, técnicas devem ser desenvolvidas de forma a se obter explicações de como os NNES chegaram a uma conclusão. Este trabalho tem como meta principal apresentar uma técnica de extração de regras para uma rede que é a base de um NNES. O algoritmo desenvolvido, denominado FUZZYRULEXT, extrai regras *Se-Então* de redes *fuzzy* com múltiplas camadas apresentando, como características particulares: a) uma topologia com neurônios *E/Ou*; b) os dados de entrada são permitidos ser booleanos, numéricos e lingüísticos; c) eliminação de regras redundantes geradas. Simulações foram realizadas para um problema real da área médica que consiste na classificação de crises epilépticas. Os resultados atingidos foram satisfatórios e encorajadores no sentido da generalização de tal algoritmo para RNA clássicas.

**Palavras-chaves:** Extração de Regras, Lógica Fuzzy, Redes Neurais Artificiais, Sistemas Especialistas

**Abstract**

*Providing explanation to the user is an important function of expert systems (ES), particularly those developed for application in the medical area. Previous experience with ES, both those based on production rules (RBES) and the ones based on Artificial Neural Networks ES (NNES) have shown that the ability of these systems to explain a given result is absolutely crucial for their acceptance by the medical users. Nevertheless, artificial neural networks (ANN) and NNES usually have difficulty in generating explanation structures. Therefore, techniques should be developed to permit the NNES to provide an adequate explanation of their conclusions. The main goal of this work is the presentation of a technique for the extraction of rules from a network that constitutes the base of the NNES. The algorithm developed, called FUZZYRULEXT, extracts If-Then rules from multi-layer fuzzy networks presenting, as main characteristics: a) an And/Or neuron topology; b) input data that can be Boolean, numerical, and linguistic; c) a redundant rule elimination process. Simulations have been accomplished based on a real problem in the medical area related to the classification of epileptic crises. The results obtained have been quite satisfactory and encouraging, according to the generalisation of algorithms for a classical ANN.*

**Keywords:** Artificial Neural Networks, Expert Systems, Fuzzy Logic, Rules Extraction.

## Extended Abstract

### Introduction

Explanation is considered a key function of the Artificial Intelligence (AI) based systems. One of its utilities is the update of knowledge structures in reasoning based on cases, e.g., when a mistake is recognized. Explanation is also used to clarify results of a reasoning process to a user (Hayashi, 1992). In some cases, the user is not an expert in a specific domain, but even so, he has the responsibility to accept or not a solution produced by an AI system. Thus, explanation is essential for the acceptance of an inference system by a user. Mainly in the health area, experts have demonstrated much resistance to Expert Systems (ES) because of the lack of explanations of the conclusions provided by the systems (Diederich et al., 1995).

Artificial Neural Networks (ANN) are black-box systems. Obtaining an explanation of the reasoning of an ANN is not a simple task (Gallant, 1988; Fu, 1993; Mitra and Pal, 1994). This is mainly due to the fact that the learned knowledge is represented by the topology of the ANN and by the weight and bias values. Some techniques have been developed to obtain explanations of how the ANN leads to a certain conclusion. One of the possibilities is the extraction of If-then rules from the ANN in which it is possible to generate a proper explanation. However, most of those rules are Boolean ones (Hayashi, 1992; Nauck et al., 1997) and real systems deal with imprecision and uncertainty. Nowadays, If-then fuzzy rules have been studied (Zimmermann, 1991).

The aim of this work is to present a technique for the extraction of rules from an And/Or fuzzy ANN.

### Artificial Neural Network ES (NNES) Description

The NNES used in this work has been developed by Brasil et al. (1999a, 2001a, 2001b), and it is illustrated in Figure 1. The idea of the development of the NNES was to propose a methodology for the Knowledge Elicitation (KE) process to minimize the time taken by both the expert and the knowledge engineering to accomplish it.

To construct the NNES it is necessary the creation of some initial rules and a set of examples obtained from a domain expert in the KE stage. These initial rules are used for the implementation of an ANN initial topology. After the extraction, the initial rules undergo some treatment according to the variables applied to the network inputs, where they represent different types of concepts, e.g. the quantitative, the linguistic, the Boolean ones or their combination.

Two sets of examples composed of the symptoms and diagnosis are also obtained from the database. The first set is used to refine the NNES through the learning algorithm called Genetic Backpropagation Based Learning (GENBACK) (Brasil et al., 1999a, 2001a, 2001b). This algorithm makes use of the Genetic Algorithm (GA) as a tool to help choose the best NNES topology and it is applied to a static feedforward ANN of multiple layers and And/Or fuzzy neurons. The other set is used to validate the refined NNES.

The initial NNES consists of one input layer, one hidden layer, and one output layer. The input layer neurons

correspond to the symptoms, i.e., the IF part of the rules. The hidden layer, which will be optimized by the GENBACK algorithm, represents the intermediate hypotheses that correspond to the initial rule conditions obtained. The output layer represents the possible diagnoses, which are the rule conclusions or the Then part. In other words, the hidden layer is defined as the And nodes, and the output layer as the Or nodes (See Figure 2).

After the NNES is refined, a reverse process is followed aiming at inferring the If-then rules together with their membership degrees. So, based on Brasil et al. (1999a, 2001a, 2001b), we have developed a new technique that can better accommodate If-then rules inferred with their membership degrees, i.e., a technique for extracting rules from a trained fuzzy NNES called Fuzzy Rule Extraction (FUZZYRULEXT), which provides explanations of the results achieved by the NNES.

### Simulations

Results related to the simulations involving two kinds of problems are presented. In the first case, the idea is to present the algorithm step-by-step. In this case, we have used the classification of cardinal numbers from 0 to 15 (binary notation) as Even, Odd, and Cousin. In the second case, a real problem applied to the medical area, e.g., the classification of clinical cases of epileptic crises, is treated.

Table 1 shows the results of some simulations that have been accomplished to test the efficacy and the performance of the technique presented here. The first column (S) corresponds to the simulation number; Column NCE shows the number of neurons in the input layer, which corresponds to the two systems developed; Column NCI shows the number of neurons in the hidden layer, which it is obtained through the GENBACK optimisation algorithm; Column NCS represents the number of neurons in the output layer, which is equal to the number of epileptic crises to be classified (e.g., 3 or 4); Column CDT corresponds to the number of cases in the set of test data, and this set differs from the training one; Column CR represents the number of rules generated containing redundancies, after the application of the FUZZYRULEXT algorithm. Column CS represents the number of rules generated by the same algorithm, but after the elimination of the redundancies; Column NA represents the number of hits, e.g., the number of hit classifications for the test set; finally, Column PA represents the system performance as to the number of positive and negative hits.

The results obtained for the classification of clinical cases of epileptic crises were between 63.6% and 83.3%.

### Conclusion

The FUZZYRULEXT algorithm has proved to be simple and efficient. One of the peculiarities of the algorithm corresponds to the redundant rule elimination process. This technique has permitted a significant reduction of the total number of rules, keeping the information provided by the NNES in the form of data as well. The application of the proposed technique to the two cases studied has presented satisfactory performance.

## Introdução

Explicação é função chave em sistemas de Inteligência Artificial (IA) e tem sido utilizada, entre outras aplicações, para atualizar estruturas de conhecimento em raciocínio baseado em casos, quando uma falha é reconhecida, isto é, aprendizado direcionado à falhas. Explicação é, também, utilizada para esclarecer os resultados de um processo de raciocínio para um usuário (Hayashi, 1992). Em alguns casos, este usuário pode não ser um especialista no domínio específico, mas tem a responsabilidade de aceitar ou rejeitar uma solução produzida por um sistema de IA. Além disso, a explicação pode ser usada para aprendizado de conhecimento intensivo sempre que uma completa teoria de domínio é dada, bem como para esclarecer, por exemplo, resultados obtidos de um Sistema Especialista baseado em Rede Neural Artificial NNES com respeito à classificação de uma dada doença (diagnóstico médico) (Brasil, 1999a).

O termo explicação se refere a uma estrutura explícita, a qual pode, internamente, ser usada para raciocínio e aprendizado e, externamente, para a explicação de resultados para um usuário. Em sistemas baseados em regras, por exemplo, explicação inclui etapas intermediárias do processo de raciocínio, como é o caso de uma estrutura de provas, quais regras foram excitadas, etc. Esta estrutura pode ser utilizada para responder questionamentos do tipo *como*. Por exemplo: como a solução  $w$  foi produzida por um sistema de inferência? Devido a certas condições  $x$  e  $y$  que foram satisfeitas após os primeiros dados de entrada e levaram às conclusões  $w$  e  $z$ , as quais satisfizeram a condição  $k$ , e assim por diante. Este tipo de explicação, dentro de uma certa limitação, é absolutamente crucial para a aceitação do usuário de um sistema de inferência (Diederich e Tickle, 1995). Na área da saúde, em particular, os especialistas têm demonstrado uma forte resistência aos SE pela falta de explicações das conclusões dos sistemas.

RNA são sistemas tipo caixa-preta. Obter uma explicação sobre o raciocínio de uma RNA não é tão fácil (Gallant, 1988; Fu, 1993; Mitra e Pal, 1994). Isto se deve, principalmente, ao fato que o conhecimento aprendido está distribuído e é representado pela topologia da rede e pelos pesos das conexões e dos "bias". Vários métodos têm sido estudados para permitir que uma RNA seja capaz de dar uma explicação a uma resposta gerada como saída. Alguns consistem em extrair regras *Se/Então*, convertendo RNA em árvores de decisões, já que tais regras são mais facilmente entendíveis pelos usuários do sistema (Hayashi, 1992; Fu, 1993).

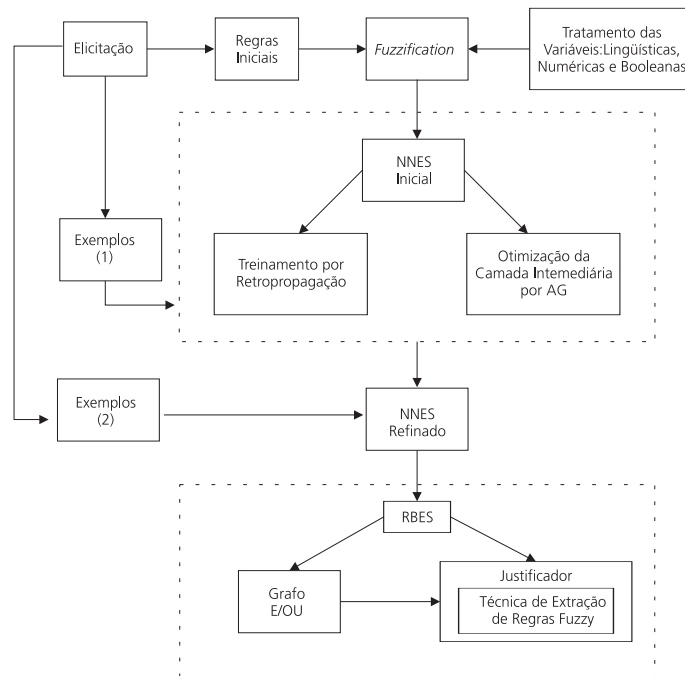
A maioria dos esforços no sentido de se extrair regras das RNA tem trabalhado com regras expressas em lógica simbólica convencional bivalente (Hayashi, 1992; Nauck *et al.*, 1997). Ultimamente, no entanto, esforços substanciais têm sido feitos para a aplicação da lógica *fuzzy* de forma que as regras a serem obtidas apresentem valores de certeza associados a "possibilidades", ou seja, representando a imprecisão e a incerteza intrínsecas aos sistemas reais (Zimmermann, 1991).

O objetivo deste trabalho é sugerir uma técnica de extração de regras de uma RNA *fuzzy E/Ou*. Propõe-se que esse conjunto de regras possa ser utilizado para a implementação de um RBES que pode ser usado para gerar, através de algum método de encadeamento para trás, as explicações do porquê de uma conclusão gerada pelo NNES que serviu de modelo para o referido RBES.

## Descrição do NNES

O NNES usado neste trabalho é aquele desenvolvido por Brasil *et al.* (1999a, 2001a, 2001b) o qual é apresentado na Figura 1.

A idéia do desenvolvimento deste NNES foi de propor uma metodologia para a Elicitação do Conhecimento (EC) que consumisse pouco tempo tanto do especialista médico quanto do engenheiro de Conhecimento. Nesta metodologia tenta-se obter um conjunto de regras *fuzzy* iniciais que possam ser utilizadas para a implementação de uma topologia inicial de RNA e que, após o tratamento das variáveis (linguísticas, numéricas e Booleanas) se constituirá no NNES inicial. Ao mesmo tempo se obtém, também do Especialista Médico, dois conjuntos de exemplos, relativos ao problema em questão, relacionando sinais e sintomas aos possíveis diagnósticos. O primeiro conjunto de exemplos é usado para refinar o NNES inicial, através de um algoritmo de aprendizado (GENBACK), que combina as características do conhecido algoritmo de Retropropagação (*Backpropagation*), com as possibilidades de otimização dos Algoritmos Genéticos (AG). O segundo conjunto de exemplos é utilizado para validar o NNES obtido através deste procedimento (NNES refinado). A rede *E/Ou*, que é a base do NNES inicial, apresenta uma topologia constituída por três camadas: a primeira, a camada de entrada, apresenta um número de neurônios (pseudo-neurônios que têm como única função a distribuição das entradas para os neurônios da camada seguinte) que é função dos sintomas e sinais considerados; a segunda camada, a intermediária, é constituída de neurônios



**Figura 1.** Sistema especialista híbrido. **Figure 1.** Hybrid Expert System. After the fuzzy treatment of variables, the ANN is trained by backpropagation and optimized with a genetic algorithm (first dashed block). The refined ANN is again trained and used for the RBES construction, whose outputs are then explained using the fuzzy technique for rules extraction (second dashed block).

do tipo *E*, que representam conceitos intermediários (a quantidade de neurônios iniciais nesta camada corresponde ao número de regras *fuzzy* iniciais obtidas) sendo que o número desses neurônios deve ou pode ser, eventualmente, otimizado pelo algoritmo GENBACK; a última camada, a de saída, é constituída de neurônios do tipo *Ou*, representando os possíveis diagnósticos. A Figura 2 apresenta um exemplo resumido de uma tal rede *fuzzy E/Ou* inicial, gerada através de regras *fuzzy* iniciais para o NNES. Observe-se que *A* a *F* correspondem aos graus de pertinência obtidos do processo de “fuzzificação” do sistema especialista híbrido, enquanto que *X* a *Z* correspondem as saídas do sistema.

Formalmente, as regras *fuzzy* utilizadas na classificação de padrões do NNES são da forma:

Se ( $k_1 \text{ é } P_1$ ) E ( $k_2 \text{ é } P_2$ ) E ... ( $k_n \text{ é } P_n$ ) Então ( $q_n \text{ é } T_n$ )  
onde  $P_1 \dots P_n$  e  $T_1 \dots T_n$  são os valores lingüísticos definidos pelos conjuntos *fuzzy* nos universos de discurso  $K_1 \dots K_n$  e  $Q_1 \dots Q_n$ , respectivamente.

Os valores *fuzzy* em uma regra *fuzzy* são dados pelos graus de pertinência definidos por uma função de pertinência. O método, utilizado neste trabalho, de combinação dos valores de pertinência é o dos operadores Max/Min (Zimmermann, 1991).

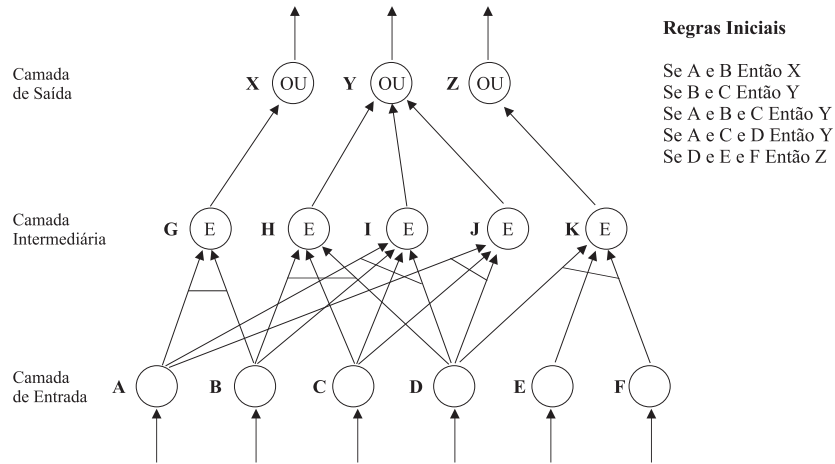
No processo de “fuzzificação” do sistema especialista híbrido, devido a este lidar inicialmente com da-

dos correspondentes a variáveis tanto Booleanas quanto numéricas e lingüísticas, os valores de entrada foram tratados da seguinte forma: se as variáveis de entrada eram *crisps* (Booleanas), estes valores foram transformados em 0s e 1s<sup>1</sup>. Quanto as variáveis numéricas, através da interpolação de valores, pode-se ter uma faixa numérica de atuação que vai depender do problema, ou seja, dos sintomas e sinais observados pelo especialista médico. No caso das variáveis lingüísticas, optou-se pelos valores lingüísticos *forte*, *moderado* e *fraco*, correspondendo, respectivamente, aos valores numéricos 0,9, 0,6 e 0,3, após o processo de “fuzzificação”.

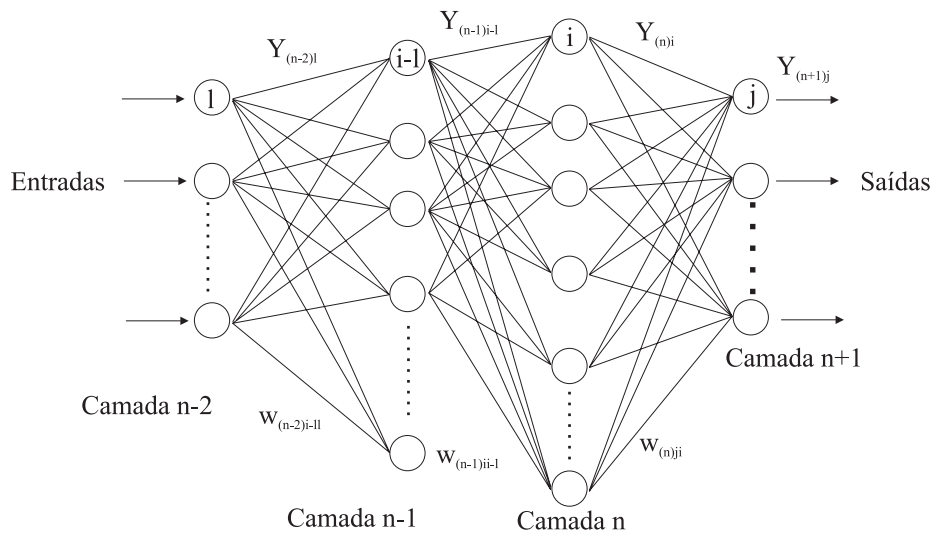
### Geração do RBES correspondente ao NNES

A grande maioria das técnicas propostas para obter explicações de um NNES, como a de Gallant (1988), considera valores *crisps*. Para sistemas *fuzzy* algumas técnicas propostas são aquelas descritas por Hayashi (1992) e Nauck *et al.* (1997). Neste trabalho optou-se por uma abordagem indireta que consiste em extrair regras *fuzzy* destas mesmas RNA de forma a se poder montar um RBES “equivalente”. Deste RBES “equivalente” pode-se obter as explicações através de al-

<sup>1</sup> Lembrando que estes também podem ser os valores extremos de um conjunto *fuzzy*.



**Figura 2.** Regras fuzzy iniciais. **Figure 2.** Initial fuzzy rules (E.g., the output X is given by the rule If A And B Then X)



**Figura 3.** Rede neural fuzzy. **Figure 3.** Fuzzy neural network. The network layers are numbered from the Input (left) to the Output (right).

gum método de encadeamento para trás. O algoritmo desenvolvido, o *Fuzzy Rule Extraction Algorithm* (FUZZYRULEXT), inspira-se em trabalhos de Fu (1993) e Mitra *et al.* (1994; 1997), mas apresenta uma série de modificações tais como: a) desde que as entradas do NNEs são valores correspondentes aos graus de pertinência, as propriedades apresentadas para se obter as cláusulas de uma regra nos algoritmos de Hayashi e Mitra foram simplificadas; b) houve a preocupação de eliminação de regras geradas como redundantes; c) a rede utilizada apresenta uma topologia com neurônios E/Ou; d) os dados de entrada do sistema conexionista são permitidos serem Booleanos, numéricos e lingüísticos.

#### Algoritmo FUZZYRULEXT

Suponha uma RNA *fuzzy* com uma camada de entra-

da, uma de saída e  $N$  intermediárias (Figura 3) usando-se  $w_{(n)ji}$  para denotar o peso entre o nó  $i$  na camada  $n$  ( $n = 0, 1, \dots$ ) e o nó  $j$  na camada  $n+1$ , bem como usando-se  $Y_{(n+1)j}$  para denotar a saída do nó  $j$  na camada de saída  $n+1$ .

#### Geração do caminho por *backtracking*

- 1 - Escolha do padrão Entrada-Saída;
- 2 - Escolha dos neurônios  $i$  com impacto positivo na conclusão de uma saída  $j$ ; Nesta etapa, selecionam-se os neurônios  $i$  na camada precedente que têm um impacto positivo na conclusão do neurônio de saída  $j$ . Então, escolhe-se o(s) neurônio(s)  $i$  na camada intermediária  $n$  se  $w_{(n)ji} > 0$ .
- 3 - Determinação do conjunto  $m_{(n)i}$  (camada intermediária  $n$ ); Seja o conjunto de  $m$  neurônios da camada intermediária  $n$ , selecionado, i.e., se  $w_{(n)ji} > 0$ :

$$m_{(n)j} = \{a_1, a_2, \dots, a_{m(n+1)j}\} \quad (1)$$

e sejam seus pesos de conexões para o neurônio  $j$  na camada de saída dados como:

$$wet_{(n)ak} = \{w_{(n)ja1}, w_{(n)ja2}, \dots, w_{(n)jam(n+1)j}\} \quad (2)$$

onde  $k$  indica o índice do número de neurônios na camada  $n$  ( $k = 1, 2, \dots, n$ ).

4 - Determinação do conjunto dos pesos acumulados  $wet_{(n-1)i}$  para o neurônio  $i$ , isto é:

$$Y_{(n)i} > 0 \quad (3)$$

$wet_{(n-1)i} = \max[wet_{(n)ak} + w_{(n-1)aki}]$  com  $w_{(n-1)aki} > 0$  e  $Y_{(n)i}$  a saída do neurônio  $i$ , ou seja, o estado do neurônio  $i$  na camada  $n$ . Caso uma RNA tenha mais de uma camada intermediária,  $Y_{(n)i}$  é determinado pelo algoritmo de treinamento do NNES na etapa de apresentação do conjunto de padrões para teste, utilizando apenas o passo para frente. Contudo, quando a RNA tiver apenas uma camada intermediária,  $Y_{(n)i}$  será o valor do padrão de entrada.

5 - Determinação do conjunto  $m_{(n-1)i-1}$  (camada intermediária  $n-1$ ); Seja o conjunto de  $m_{(n-1)i-1}$  neurônios da camada intermediária  $n-1$  assim escolhido dado por:

$$m_{(n-1)i} = \{a_1, a_2, \dots, a_{m(n)i}\} \quad (4)$$

e seus pesos de conexões acumulados para o neurônio  $j$  na camada de saída ao longo do caminho dos pesos máximos, isto é  $\{wet_{(n-1)a1}, wet_{(n-1)a2}, \dots, wet_{(n-1)am(n)i}\}$ , obtidos pela Equação 4. Esta heurística assegura que cada um dos  $m_{(n-1)i-1}$  neurônios selecionados tenha uma resposta de saída significante. Ou seja, isto ajuda a selecionar caminhos ao longo dos quais cada par de neurônios possui uma correlação positiva significante ou uma influência em cada um dos outros neurônios que são ativos para decidir a conclusão que está sendo justificada. Pode também habilitar cada neurônio  $i$  para ser mantido ao longo de um dos  $m_{(n-1)i}$  caminhos selecionados previamente, que provê o maior  $wet_{(n-1)i}$  peso da rede.

6 - Seleção do conjunto dos neurônios de entrada  $m_{(n-2)j}$ ; Seja o processo da Equação 4 resultante em  $m_{(n-1)j}$  neurônios escolhidos na camada de entrada. Estes neurônios indicam entradas que são conhecidas e têm contribuído para a conclusão final no neurônio  $j$  na camada de saída ( $n+1$ ). Pode acontecer que  $m_{(n-2)j} = 0$ , tal que nenhuma justificção clara pode ser dada em um caso particular de entrada/saída. Isto implica que nenhum caminho adequado pôde ser selecionado pela Equação 4 e o processo termina.

7 - Determinação dos pesos do caminho referente ao neurônio  $j$ ; Seja o conjunto dos neurônios de entrada  $m_{(n-2)j}$  selecionados como  $m_{(n-2)j} = \{a_1, a_2, \dots, a_{m(n-2)j}\}$  e seus pesos correspondentes ao neurônio  $j$  na camada de saída  $n+1$  dados como  $\{w_{(n-2)a1}, w_{(n-2)a2}, \dots, w_{(n-2)am(n-1)j}\}$ .

8 - Ordenação em ordem decrescente de Impacto de Rede (IR) dos elementos do conjunto de pesos obtidos no item anterior, ou seja:

$$IR_{(n-1)i} = Y_{(n-1)i} * wet_{(n-1)i} \quad (5)$$

9 - Seleção dos  $l_s$  neurônios de entrada (aqueles com  $w_{(n-2)j} > 0$ ) e os  $l_p$  neurônios restantes (aqueles com  $w_{(n-2)j} < 0$ ) para as cláusulas, tal que:

$$m_{(n-2)j} = |l_s| + |l_p| \quad (6)$$

onde  $|l_s|$  e  $|l_p|$  referem-se, respectivamente, ao número de neurônios selecionados e restantes do referido conjunto. Esta heurística permite a seleção daqueles neurônios ativos correntes contribuindo para a conclusão final como a parte antecedente de uma regra. Daí habilita as entradas dos padrões de teste ativos atuais para influenciar a base de conhecimento gerada (pesos das conexões aprendidos durante o treinamento) para produzir uma regra para justificar a inferência atual (decisão olhando o padrão de teste corrente).

## Geração das cláusulas

### 1 - Dedução dos antecedentes:

a) Determinação de  $u_{s1}$  (característica de entrada) para uma entrada Booleana ou numérica:

$$u_{s1} = (l_{s1} - 1) + 1 \quad (7)$$

onde  $l_{s1}$  é um neurônio selecionado na camada de entrada,  $-0,8 \leq l_{s1} \leq n$ ,  $-0,8 \leq u_{s1} \leq n$  e  $n$  é a dimensão do vetor do padrão de entrada.

b) Determinação de  $u_{s1}$  (característica de entrada) para uma entrada lingüística:

$$u_{s1} = (l_{s1} - 1) \text{ mod } 3 + 1 \quad (8)$$

onde  $l_{s1}$  é um neurônio selecionado na camada de entrada,  $-0,8 \leq l_{s1} \leq 3n$ ,  $-0,8 \leq u_{s1} \leq n$  e  $n$  é a dimensão do vetor do padrão de entrada, sendo  $\text{mod } 3$  o resto da divisão da expressão matemática  $(l_{s1} - 1)$  por 3.

c) Determinação do antecedente de uma regra para uma entrada Booleana ou numérica:

2 Os valores apresentados nesta propriedade são valores heurísticos definidos em função das simulações efetuadas para o desenvolvimento deste algoritmo.

$$prop = \begin{cases} -0,8 & \text{Se } u_{s_1} \leq -0,8 \\ 1 & \text{Se } u_{s_1} = 1 \\ -0,8 < x < 1 & \text{Se não} \end{cases}$$

d) Determinação do antecedente de uma regra para uma entrada lingüística<sup>2</sup>:

$$prop = \begin{cases} \text{forte} & \text{Se } u_{s_1} \geq 0,8 \\ \text{moderado} & \text{Se } -0,4 < u_{s_1} < 0,7 \\ \text{fraco} & \text{Se } u_{s_1} \leq -0,4 \end{cases}$$

e) Repetem-se os itens a) a e) até que todos os  $l_s$  sejam selecionados pela Equação (6);

## 2 - Dedução dos conseqüentes:

a) Determinação da medida de certeza  $bel_j$ ; Uma forma mais natural de decisão pode ser obtida para a saída  $j$  tendo o valor de pertinência significativa, considerando o valor de uma medida de certeza ( $bel_j$ ) como:

$$bel_j = Y_{(n+1)j} - \sum_{i \neq j} Y_{(n+1)i} \quad (9)$$

Nota-se que a dificuldade em chegar a uma decisão particular em favor da saída  $j$  é dependente, não somente do valor de pertinência  $Y_{(n+1)j}$ , mas também das diferenças dos valores de pertinência  $Y_{(n+1)i}$ , onde  $i \neq j$ . Desta forma, o valor de  $bel_j$  sendo baixo, a dificuldade em decidir uma saída  $j$  é determinada em função da escolha do maior grau de certeza da decisão de saída.

b) Determinação dos conseqüentes de uma regra através das seguintes propriedades:

$$prop = \begin{cases} -0,8 & \text{para } -5 \leq bel_j \leq -0,2 \\ 1 & \text{para } bel_j > -0,2 \\ \text{Incapaz de reconhecer} & \text{para } bel_j < -5 \end{cases}$$

c) Volta-se ao item a) caso se deseje determinar as regras *Se/Então* para justificar todas as saídas apresentadas pelos padrões de Entrada-Saída da RNA treinada.

## Eliminação das regras redundantes

Como muitas das regras geradas são redundantes, o que implica em um maior consumo de tempo computacional, desenvolveu-se o seguinte processo para auxiliar na eliminação de algumas redundâncias:

a) Determinar o número total de regras geradas; b) Criar um único vetor para os dados de entrada-saída (sintomas-sinais/diagnósticos); c) Comparar o antecedente - conseqüente de cada regra para se determinar quais são iguais; d) Guardar as regras que são distintas; e) Eliminar as regras redundantes; f) Determinar o número total de regras geradas sem redundâncias.

## Simulações

Resultados relacionados às simulações envolvendo dois tipos de problemas são apresentados. No primeiro caso, denominado *Problema 1*, o objetivo é de mostrar o algoritmo passo-a-passo, através de um problema simples. No segundo caso, *Problema 2*, utilizaremos a técnica proposta aplicada a um problema real da área médica, que consiste na classificação de crises epilépticas.

### Problema 1

Considere a RNA com três camadas apresentada na Figura 4, demonstrando uma geração de regra simples olhando as saídas  $Y_1$ ,  $Y_2$  e  $Y_3$ . Esta simulação corresponde à classificação dos números cardinais de 0 à 15 (codificados em binário) em Par, Ímpar e Primo<sup>3</sup>. Um conjunto de amostra de pesos  $w_{ji}$  e  $w_{il}$  e a ativação das entradas  $X_i$  são mostrados nessa figura.

Como este exemplo utiliza valores de entrada e saída Booleanos, não utilizaremos as variáveis  $P_1 \dots P_n$  e  $T_1 \dots T_n$  que correspondem aos valores lingüísticos definidos pelos conjuntos *fuzzy* para a entrada  $X_4$  e para as saídas Ímpar e Primo.

Exemplificando vejamos, abaixo, como obter as cláusulas considerando, para facilidade dos cálculos, que:  $m_{(n-2)l} = X_l$ ,  $Y_{(n+1)j} = Y_j$ ,  $w_{(n)aki} = w_{ji}$ ,  $w_{(n-1)aki} = w_{il}$

### Escolha do padrão entrada-saída

$$X_i = \{-0,8; -0,8; -0,8; +1\}$$

$$Y_j = \{-0,8; +1; +1\}$$

### Escolha do neurônio $i$ com impacto positivo na conclusão de uma saída $j$ ( $w_{ji} > 0$ )

$$m_i = \{a_1, a_2, \dots, a_6\}$$

### Determinação dos pesos acumulados $wet_l$ para o neurônio $l$

$$wet_l = \max(wet_{a_k} + w_{a_k l})$$

$$w_{aki} = ? (w_{ji} > 0, k = 1, \dots, 6)$$

$$w_{il} > 0, (l = 1, \dots, 4)$$

$$wet_{a_k} = \{w_{ja_1}, w_{ja_2}, w_{ja_3}\}$$

$$wet_{a_1} = \{0,59\}$$

$$wet_{a_2} = \{0,50; 0,63\}$$

$$wet_{a_3} = \{0,07; 0,25\}$$

$$wet_{a_4} = \{0,50; 0,52\}$$

$$wet_{a_5} = \{0,018\}$$

$$wet_{a_6} = \{0,59; 0,61\}$$

<sup>3</sup> Obviamente este problema não permite o teste da característica de generalização da RNA além de não ser o tipo de problema adequado para ser tratado por RNA.

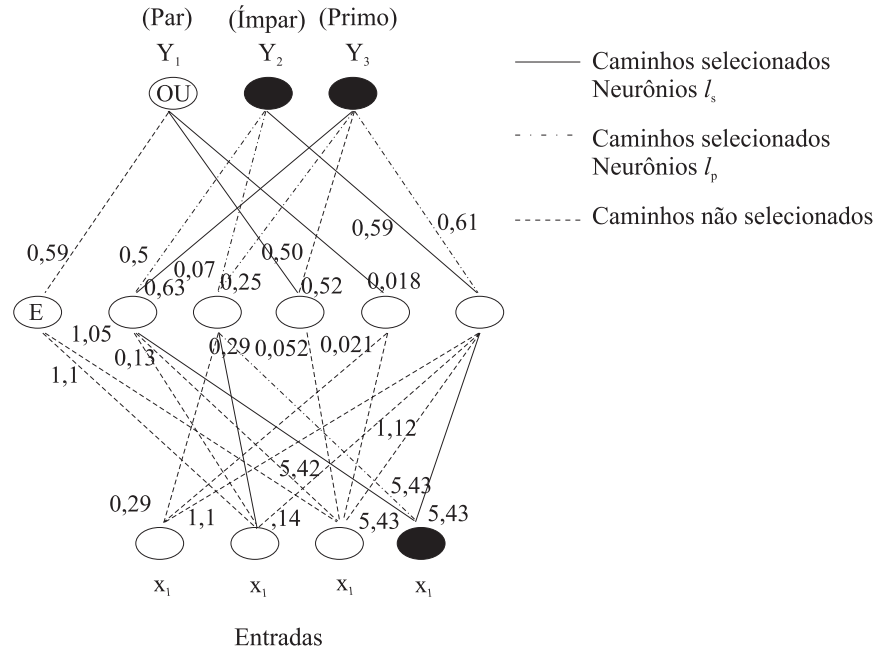


Figura 4. Simulação 1 - Geração de regra por backtracking. Figure 4. Simulation 1 – Rule generation by backtracking.

$$\begin{aligned}
 wet_1 &= \max \{0,59;0,50;0,63;0,36;0,54;0,50;0,52;0,039;1,69;1,71\} = 1,71 \\
 wet_2 &= \max \{1,59;1,53;1,66;0,20;0,38;0,50;0,52;0,16;0,52;0,61\} = 1,66 \\
 wet_3 &= \max \{1,64;1,57;1,70;0,07;0,25;5,92;5,94;0,07;1,64;1,73\} = 5,94 \\
 wet_4 &= \max \{0,59;5,93;6,06;5,50;5,95;0,50;0,52;0,18;5,95;6,04\} = 6,06
 \end{aligned}$$

**Ordenação em ordem decrescente de IR**

$$\begin{aligned}
 IR_1 &= -0,8 * 1,71 = -1,368 \\
 IR_2 &= -0,8 * 1,66 = -1,328 \\
 IR_3 &= -0,8 * 5,94 = -4,752 \\
 IR_4 &= 1 * 6,06 = 6,06
 \end{aligned}$$

e ordenando

$$\begin{aligned}
 IR_4 &= 6,06 \\
 IR_2 &= -1,328 \\
 IR_1 &= -1,368 \\
 IR_3 &= -4,752
 \end{aligned}$$

**Seleção dos  $I_s$  e  $I_p$**

$$X_i = |I_s| + |I_p|$$

onde  $I_s$  é determinado em função de  $X_i > 0,1$ .

$$I_s = \{X_4\}$$

$$I_p = \{X_4, X_2, X_3\}$$

Assim, na Figura 4, os caminhos representados pelas linhas cheias e as linhas traço-ponto (caminhos selecionados) são os que representam os neurônios  $I_s$  e  $I_p$ , respectivamente, dados pela Equação (5). As linhas pontilhadas indicam os caminhos não selecionados, os quais são determinados pela Equação (4).

**Geração das cláusulas**

**Antecedentes**

a)  $u_{sl}$  (entrada Booleana)

$$\begin{aligned}
 u_{sl} &= (I_{sl} - 1) + 1 \\
 u_{s1} &= u_{s2} = u_{s3} = -0,8 \\
 u_{s4} &= 1
 \end{aligned}$$

b) Pelas Propriedades:

Se  $x_4$ , então...

**Conseqüentes**

a)  $bel_j$

usando a expressão  $bel_j = Y_j - \sum_{l \neq j} X_l$ , obtém-se

$$bel_1 = Y_1 - \sum_{l \neq 1} (X_2, X_3, X_4) = -0,2$$

$$bel_2 = Y_2 - \sum_{l \neq 2} (X_1, X_3, X_4) = 1,6$$

$$bel_3 = Y_3 - \sum_{l \neq 3} (X_1, X_2, X_4) = 1,6$$

b) Pelas Propriedades:

Se...Então  $Y_2$

Se...Então  $Y_3$

**Determinar as regras Se/Então**

As regras geradas pelo modelo para a justificação da conclusão, olhando as saídas  $Y_2$  (Ímpar) e  $Y_3$  (Primo), podem ser:

Se  $X_4$  Então Ímpar

Se  $X_4$  Então Primo



Pode-se observar, também, que os pesos  $w_{ji} = 0,5$  e  $w_{ji} = 0,07$ , poderiam ser outros caminhos escolhidos para a formalização da justificativa da conclusão correspondente ao neurônio Ímpar, bem como os pesos  $w_{ji} = 0,25$  e  $w_{ji} = 0,61$  para o neurônio Par. Optou-se pelos caminhos com pesos maiores, por estes terem maior influência na ativação dos neurônios na saída da rede.

A simulação 2 (Figura 5) corresponde à classificação do número zero (0) em Par, Ímpar e Primo. Neste caso, observa-se que todas as entradas são negadas. Então, a Equação (4) passa a ser da seguinte forma:

$$wet_l = \min[wet_{ak} + w_{akl}]$$

com  $w_{akl} > 0$  não sendo mais considerado, ou seja, usa-se os valores de pesos  $w_{il}$  negativos para se obter as premissas negadas.

Então, neste caso, a regra gerada pelo modelo para a justificativa da conclusão, olhando as saídas  $Y_1, Y_2$  e  $Y_3$ , pode ser:

*Se não  $X_1$  E não  $X_2$  E não  $X_3$  E não  $X_4$  Então  $Y_1$*

**Problema 2**

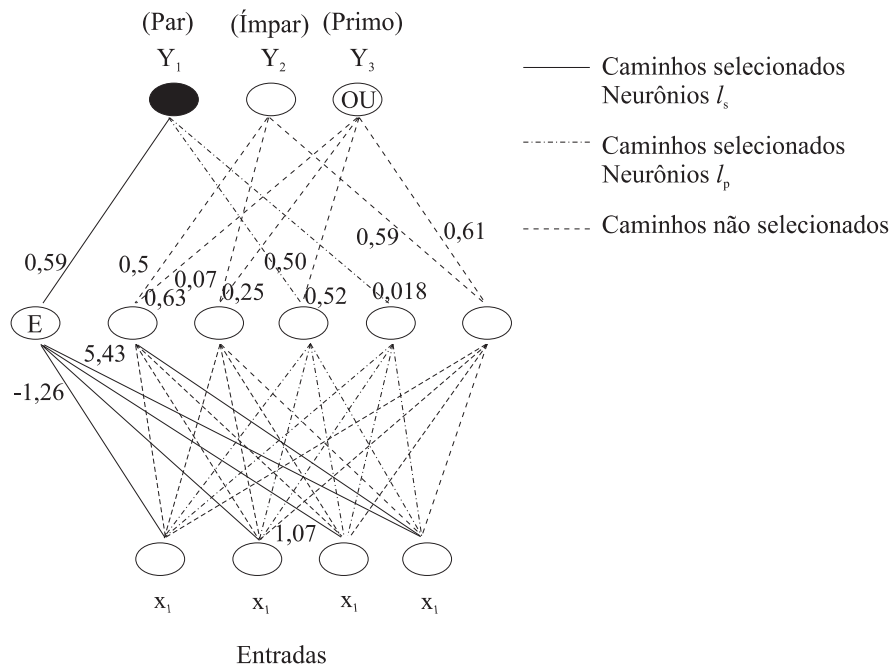
Este problema, um exemplo real de aplicação na área médica, consiste na classificação de crises epiléticas. Dois sistemas foram desenvolvidos: um que, a partir de seis (6) sintomas, classifica três (3) tipos de crises epiléticas e outro, mais completo, considerando trinta

e dois (32) sintomas para classificar quatro (4) tipos de crises epiléticas.

Para ilustração, a Figura 6 mostra a RNA para o primeiro caso. Esta RNA, com três camadas, demonstra a geração de uma regra simples olhando as saídas  $D_1, D_2$  e  $D_3$ . Além disto, tem-se um conjunto de amostra de pesos  $w_{ji}$  e  $w_{il}$ , ativação das entradas  $S_i$  e os correspondentes valores de graus de pertinência. Como se pode notar, obteve-se a seguinte regra:

De modo a avaliar a eficácia e o desempenho dos dois sistemas, uma série de simulações foi realizada, de forma a classificar crises epiléticas a partir de sinais e sintomas apresentados por pacientes submetidos à avaliação clínica do especialista médico.

Alguns resultados de simulações são mostrados na Tabela 1. Na tabela, a primeira coluna, S, corresponde ao número da simulação; a coluna NCE, ao número de neurônios na camada de entrada considerando os dois sistemas desenvolvidos; a coluna NCI, ao número de neurônios na camada intermediária, número este achado através do algoritmo de otimização GENBACK; a coluna NCS, ao número de neurônios na camada de saída, o qual é igual ao número de crises epiléticas a classificar (3 ou 4); a coluna CDT, ao número de casos no conjunto de dados de testes; a coluna CR, ao número de regras geradas com redundâncias, após a aplicação do algoritmo FUZZYRULEXT; a coluna CS, ao número de regras após a eliminação das redundâncias; a coluna NA, ao



**Figura 5.** Simulação 2: geração de regra por backtracking. **Figure 5.** Simulation 2 – Rule generation by backtracking.

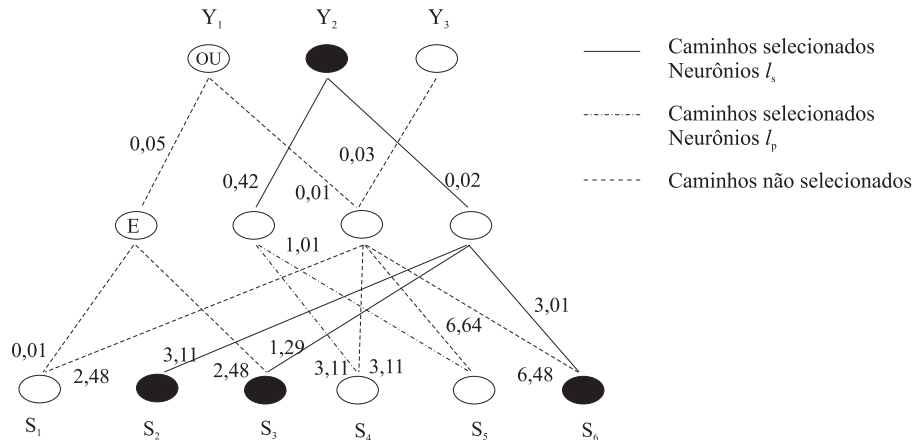


Figura 6. Simulação 3: geração de regra por backtracking. **Figure 6.** Simulation 3 – Rule generation by backtracking

Tabela 1. Problema Real – Classificação de crises epiléticas<sup>4</sup>. **Table 1.** Real Problem – Classification of Epileptic Crises.

S	NCE	NCI	NCS	CDT	CR	CS	NA	PA%
1	6	4	3	6	8	7	5	83,3
2	6	5	3	6	11	6	5	83,3
3	6	10	3	6	23	12	5	83,3
4	6	10	3	6	30	13	5	83,3
5	32	8	4	9	38	23	6	66,7
6	32	13	4	9	43	27	7	77,8
7	32	13	4	9	49	26	7	77,8
8	32	15	4	11	70	40	7	63,6
9	32	16	4	11	70	36	8	72,7

número de acertos, ou seja, o número de classificações corretas para o conjunto de teste; finalmente a coluna PA, apresenta a eficiência do sistema dada em termos do número de acertos positivos e negativos.

Pode-se observar que os resultados alcançados para os casos clínicos da classificação das crises epiléticas ficaram entre 63,6% e 83,3%.

### Conclusão

O algoritmo FUZZYRULEXT é simples e eficaz, como pode ser observado pelos resultados obtidos nas simulações. Uma das particularidades desse algoritmo diz respeito a etapa de eliminação da geração de regras redundantes. Esta técnica conseguiu reduzir consideravelmente o número total de regras, sem que se perdesse, significativamente, informações. Note-se que, com poucas modificações, o algoritmo pode ser utilizado para extração de regras *crisps*.

Quanto a validação do sistema simbólico derivado do NNES para o diagnóstico de crises epiléticas, faz-se necessário explicar que, para diferentes especialidades médicas, os resultados apresentados numa situação real em que os médicos estejam fazendo o atendimento, podem ser categorizadas em três situa-

ções, quais sejam: a) no caso de atendimento por um médico clínico geral, apesar deste ser capaz de diagnosticar, corretamente, a presença de crise epilética, o número de acertos na classificação da crise epilética não passa de 1/3; b) No caso do médico ser especialista em neurologia, o número de acertos fica em torno de 50%; c) Finalmente, no terceiro caso, correspondente ao especialista de domínio em epilepsia, a taxa de acerto fica em torno de 90%. Desta forma, observando-se que os diversos experimentos resultaram em taxas de acerto de 63,6% e 83,3%, pode-se concluir que o desempenho do sistema situa-se entre a segunda e terceira categorias, ou seja, diagnóstico realizado por um neurologista e por um especialista em epilepsia. Este último problema, tratado com auxílio de um especialista de domínio, visa mostrar as potencialidades do algoritmo proposto em solucionar problemas complexos, que possam ser, posteriormente, validados pelos especialistas médicos.

Finalizando, pode-se concluir que, para os dois casos estudados, os resultados foram satisfatórios.

<sup>4</sup> As Simulações 6 e 7 diferem apenas no CR devido à etapa de treinamento e refinamento do NNES ocorrido anteriormente, através de valores diferenciados para a taxa de aprendizado, momento, etc.

Além disso, a grande maioria dos exemplos aplicados para testar o sistema simbólico foi capaz de traduzir o conhecimento embutido nas conexões da RNA *fuzzy E/Ou* em regras *fuzzy* e, também, demonstra que os procedimentos para a redução do número de regras redundantes foram adequados.

### Agradecimentos

Os percentuais de acertos clínicos aqui apresentados são baseados em informações dadas pelo Médico e Professor Paulo C.T. Bittencourt (Med., M.Sc.), da Universidade Federal de Santa Catarina (UFSC). Sua especialidade (domínio) é na área de neurologia, mais especificamente, epileptologia clínica. Estes dados, que ainda não foram publicados, são provenientes da Clínica Multidisciplinar de Epilepsia do Sistema Único de Saúde (SUS), dirigida pelo referido especialista e que está funcionando há mais de 10 anos na Policlínica Regional I, em Florianópolis, SC.

### Referências

- Brasil, L.M., De Azevedo, F.M., Barreto, J.M. (2001a), "Hybrid expert system for decision supporting in the medical area: complexity and cognitive computing", *International Journal of Medical Informatics*, v. 63, n. 1-2, p. 19-30.
- Brasil, L.M., De Azevedo, F.M., Barreto, J.M. (2001b), "A hybrid expert system for the diagnosis of epileptic crisis", *Artificial Intelligence in Medicine*, v. 21 n. 1-3, p. 227-233.
- Brasil, L.M. (1999a), *Proposta de Arquitetura para Sistema Especialista Híbrido e a Correspondente Metodologia de Aquisição do Conhecimento*, Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica/Engenharia Biomédica, UFSC, Florianópolis, Brasil, 300 p.
- Diederich, J., Tickle, A.B. (1995), "Explanation and collective computation", *Complexity International*, v. 2, <http://journal-ci.csse.monash.edu.au/ci/vol02/index.html>.
- Fu, L.M. (1993), "Knowledge-Based Connectionist for Revising Domain Theories", *IEEE Transactions on Systems, Man and Cybernetics*, v. 23, n. 1, p. 173-182.
- Gallant, S.I. (1988), "Connectionism Expert Systems", *Communications of the ACM*, v. 31, n. 2, p. 152-169.
- Hayashi, Y. (1992), "Fuzzy neural expert system with automated extraction of fuzzy if-then rules from a trained neural network", In: *Analysis and Management of Uncertainty: Theory and Applications*, Eds.: B.M. Ayyub, M.M. Gupta, L.N. Kanal, North-Holland: North-Holland, p. 171-181.
- Mitra, S., Pal, S.K. (1994), "Logical operation based fuzzy MLP for classification and rule generation", *Neural Networks*, v. 7, n. 2, p. 353-373.
- Mitra, S., De, R.K., Pal, S.K. (1997), "Fuzzy multi-layer perceptron, inferencing, and rule generation", *IEEE Transactions on Neural Networks*, v. 8, n. 6, p. 1338-1350.
- Nauck, D., Klawonn, F., Kruse, R. (1997), *Foundations of Neuro-Fuzzy Systems*, New York: John Wiley & Sons.
- Zimmermann, H.J. (1991), *Fuzzy Set Theory and Its Applications*, Norwell: Kluwer Academic.