

## Extração de Regras de Redes Neurais via Algoritmos Genéticos

Raul T. Santos<sup>1</sup>, Júlio C. Nievola<sup>2</sup>, Alex A. Freitas<sup>3</sup>, Heitor S. Lopes<sup>4</sup>

<sup>1,4</sup>CEFET-PR/CPGEI, Av. Sete de Setembro, 3165, Curitiba-PR, 80230-901

<sup>2,3</sup>PUC-PR/CCET-PPGIA, Av. Imaculada Conceição, 1155 Curitiba- PR, 80215-901

raul@dainf.cefetpr.br, nievola@ppgia.pucpr.br, alex@ppgia.pucpr.br, hslopes@cpgei.cefetpr.br

### Abstract

*In knowledge extraction from databases, a frequently found problem is noisy data. Neural networks are usually tolerant a noise in the training set, but they are have a poor performance in explaining how a solution is found. In this work it is presented a method for obtainiy correct and comprehensible kwnledge by using rule extraction from trained neural networks. Such a method uses genetic algorithm to find a suitable topology for a neural network that allows the RX algorithm to extract a rule set as accurate and comprehensible as possible. The proposed system was tested with two publicly available data sets and results were very satisfactory.*

### 1. Introdução

Com o desenvolvimento da tecnologia de armazenamento de dados, surge o interesse em transformar os dados armazenados em conhecimento. Em termos ideais, o conhecimento deveria ser tanto correto quanto compreensível para o usuário [1]. Uma das dificuldades para a extração de conhecimento correto é que os dados armazenados podem conter ruídos. Para lidar com este problema, será utilizado neste trabalho o paradigma de descoberta de conhecimento, ou *data mining*, baseado em Redes Neurais, devido à sua tolerância a ruídos na base de dados.

Por outro lado, redes neurais têm sido freqüentemente criticadas devido à não apresentar as razões de suas conclusões. Então surge o interesse em extrair conhecimento compreensível de redes neurais treinadas, aplicando-se para isso algum algoritmo de extração de regras. As regras extraídas são do tipo IF...THEN, onde a parte IF especifica um conjunto de condições sobre valores de atributos previsores e a parte THEN especifica um valor previsto para o atributo classe. Este trabalho aborda a tarefa de classificação, amplamente investigada na literatura.

O problema de extração de regras de redes neurais treinadas é tratado por Lu e colaboradores [2] em 3 fases: (a) treinando a rede neural; (b) efetuando em seguida a poda da rede neural enquanto a precisão preditiva da rede não diminui significativamente; (c) aplicando o algoritmo RX para extração das regras.

Neste trabalho propõe-se a idéia de utilizar uma adaptação do algoritmo RX em conjunto com

algoritmos genéticos. Algoritmos genéticos são um método de busca global robusto [3] e são adequados para problemas difíceis, com grande espaço de busca. Nosso objetivo é utilizar um algoritmo genético para encontrar uma topologia de Rede Neural que possibilite a extração de um conjunto de regras com maior taxa de acerto e menor complexidade.

### 2. Fundamentação teórica

#### 2.1. Algoritmos genéticos

Algoritmos genéticos são algoritmos de busca baseados na seleção natural dos seres vivos. A cada geração, novos indivíduos (*strings*) são gerados a partir dos indivíduos velhos. Cada indivíduo representa os parâmetros para solução do problema e possui também um valor de *fitness*, o qual indica o quanto ele é bom como solução do problema.

Os principais operadores genéticos são *crossover* e *mutação*. A seleção é um processo que ocorre antes da aplicação dos operadores genéticos e consiste na escolha de indivíduos dentre a população. A seleção dos indivíduos é feita baseada no seu valor de *fitness*. Indivíduos com melhor *fitness* possuem maior probabilidade de serem selecionados.

*Crossover* é o processo que combina dois indivíduos selecionados. O *crossover* efetua escolha de uma posição aleatória na *string* e troca partes correspondentes das duas *strings* selecionadas, criando dois novos indivíduos. O operador de *mutação* apenas efetua a troca de bits de uma *string* (indivíduo) substituindo 0 por 1 e vice-versa. Ele é importante para tentar recuperar material genético útil que pode ter sido perdido com as operações de seleção e *crossover* ao longo das gerações. Para um estudo mais detalhado recomenda-se a leitura de [3].

#### 2.2 Data mining via redes neurais

*Data mining* objetiva extrair conhecimento correto e compreensível de bases de dados. Dentre as tarefas e métodos de *data mining* encontra-se a classificação via redes neurais, onde os neurônios na camada de entrada correspondem aos atributos previsores da base de dados e a saída da rede representa a classificação do registro associado aos valores dos atributos na camada de entrada. Se o problema possui apenas 2 classes um neurônio na camada de saída é suficiente para fazer a

classificação; caso contrário tem-se um neurônio para cada classe existente.

O conhecimento do sistema tem sua representação distribuída entre os pesos das conexões, mas esta representação não pode fornecer uma explicação compreensível sobre a razão pela qual uma classe foi escolhida. Conseqüentemente, no contexto de *data mining* geralmente é desejável converter esta representação em regras do tipo IF-THEN que possam ser compreensíveis para o usuário [4],[2].

### 3. O método proposto para a extração de regras de redes neurais via algoritmos genéticos.

Conforme mencionado anteriormente, este trabalho propõe a utilização de algoritmos genéticos para encontrar uma topologia de rede neural que conduza à extração de regras de alta qualidade a partir da rede treinada. Nesta abordagem, cada indivíduo da população é associado com uma topologia de rede neural. Para cada topologia, a rede é treinada e regras são extraídas usando-se uma adaptação do algoritmo RX [2]. Portanto, no método aqui desenvolvido, cada indivíduo da população é associado com um conjunto de regras e a função de *fitness* do algoritmo genético avalia a qualidade deste conjunto de regras como um todo, em vez de avaliar regras individuais. Esta abordagem tem a vantagem de considerar a interação entre as várias regras extraídas.

Será utilizado para evolução da rede neural o *software* ENZO [5], cujo ciclo de evolução é mostrado na figura 1. Porém este *software* não possui um algoritmo de extração de regras de redes neurais. Desta forma, o primeiro passo do trabalho é a implementação de um novo módulo para ser inserido no ciclo de evolução do ENZO que efetue a extração de regras de todas as redes neurais da população. Este módulo também efetua o cálculo do *fitness* do conjunto de regras extraído e atribui tal *fitness* a rede neural. Assim sendo, o novo ciclo de evolução do ENZO tem o aspecto apresentado na figura 2.

A rede evoluída pelo ENZO é uma rede neural *feedforward* completamente conectada com apenas uma camada oculta. Ao contrário do que é feito em [2], esta Rede Neural é fornecida ao ENZO sem nenhuma poda.

A base de dados é dividida em 3 partes: um conjunto de treinamento, um conjunto de validação, e um conjunto de teste. Apenas os dois primeiros são usados durante a evolução do algoritmo genético. Os dados do conjunto de treinamento são usados para treinar a rede neural. Uma vez que uma rede é treinada, regras são extraídas a partir dela, o algoritmo de extração de regras também usa os dados de treinamento. Entretanto, uma vez que as regras foram extraídas, é necessário avaliar a qualidade dessas regras, a qual será usada como função de *fitness* do indivíduo (rede neural). Essa avaliação usa os dados do conjunto de validação, que é independente do conjunto

de treinamento, para evitar um *overfitting* das regras aos dados de treinamento.

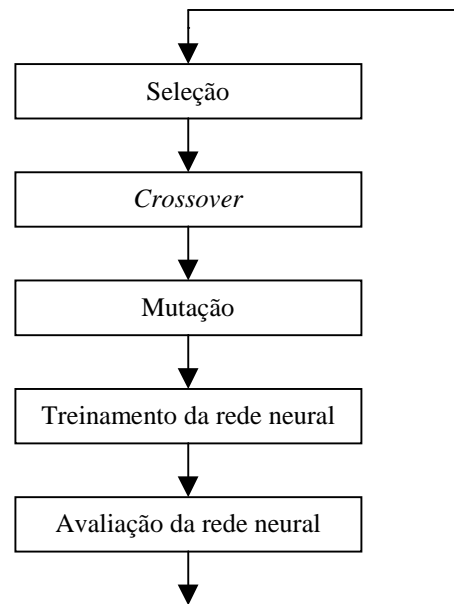


Figura 1: Ciclo de evolução do ENZO

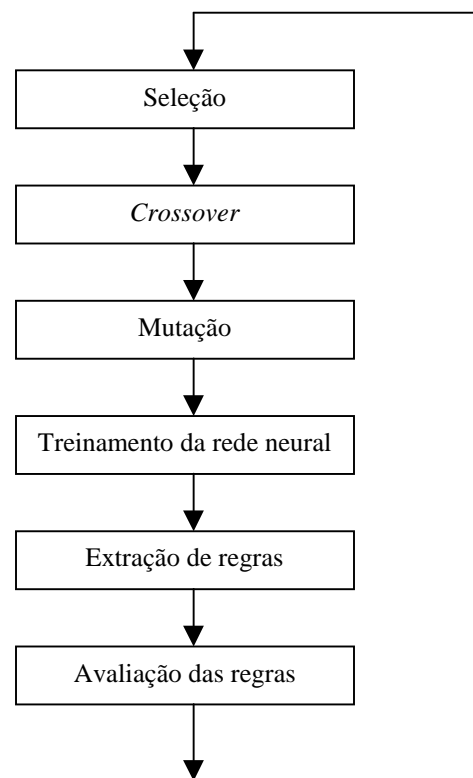


Figura 2: Ciclo de Evolução do ENZO com o novo módulo

Finalmente, após a evolução do algoritmo genético ser terminada, o conjunto de regras extraídas do melhor indivíduo da última geração é considerado o conjunto final de regras extraído e mostrado ao usuário. Este conjunto tem a sua qualidade avaliada com base nos

dados de teste, que são independentes dos dados de treinamento e validação.

### 3.1 Extração de regras

Utilizou-se para extração das regras uma adaptação do algoritmo RX apresentado em [2]. Em particular, o método proposto avalia a qualidade das regras extraídas com relação à precisão preditiva e compreensibilidade, enquanto o algoritmo RX avalia as regras apenas com relação à precisão preditiva.

Após a obtenção de um conjunto de regras faz-se necessário o cálculo do fator de confiança de cada uma das regras devido ao fato de que mais de uma regra pode ser aplicável a um mesmo exemplo. Neste caso o fator de confiança indica qual regra é a mais confiável, e portanto deve ser escolhida. O fator de confiança é dado pela equação (1).

$$FC = \frac{A}{ETC} \quad (1)$$

onde, FC é o fator de confiança da regra, A é o número de acertos e ETC é o número de exemplos cobertos pela regra.

Por exemplo, se a parte IF da regra cobre 10 exemplos de treinamento e 8 destes exemplos são corretamente classificados pela parte THEN da regra, então o fator de confiança é de 80%.

### 3.2 Cálculo do *fitness*

O cálculo do *fitness* do conjunto de regras extraídas da rede neural é composto da taxa de acerto do conjunto de regras e da compreensibilidade deste mesmo conjunto de regras. A taxa de acerto é obtida utilizando-se a equação (2), lembrando-se que quando mais de uma regra é aplicável a um dado exemplo, utiliza-se a regra com maior fator de confiança para classificar aquele exemplo.

$$TC = \frac{A}{ECV} \quad (2)$$

onde, TC é a taxa de acerto do conjunto de regras, A é o número de acertos e ECV é o número de exemplos no conjunto de validação.

A complexidade do conjunto de regras segundo [6] é dada pela equação (3). Normalizando (3) e calculando-se o complemento obtemos a compreensibilidade do conjunto de regras que é dada pela equação (4).

$$\text{complexidade} = 2 * R + C \quad (3)$$

$$CP = 1 - \frac{2 * \frac{R}{Max\_R} + \frac{C}{Max\_C}}{3} \quad (4)$$

onde, R é o número de regras, C é o número de condições, Max\_R é o maior número de regras extraídas até o momento e Max\_C é o maior número de condições extraídas até o momento.

Como o *software* ENZO considera o melhor indivíduo o que possui o menor *fitness*, a seguinte função de *fitness* foi definida,

$$Fitness = 1 - (pta * TC + pc * CP) \quad (5)$$

onde pta e pc são pesos definidos pelo usuário.

### 3.3 Método de seleção e operadores

O método de seleção utilizado gera um número aleatório que freqüentemente selecionará o primeiro quarto da população para gerar os filhos. Isto permite manter uma maior diversidade genética. Os operadores utilizados são *crossover* que insere uma conexão que é encontrada em apenas um dos pais, mutação de *links*, mutação de unidades ocultas e mutação de unidades de entrada. Para maiores detalhes sobre as operações de seleção, *crossover* e mutação consulte [5].

### 3.4 Bases de dados utilizadas

Neste trabalho foram utilizadas duas bases de dados disponíveis publicamente no repositório Machine Learning (<http://www.ics.uci.edu/AI/Machine-Learning.html>).

A primeira base utilizada foi Hayes-Roth [7]. Esta base possui 4 atributos previsores e um atributo meta como apresentado a seguir:

- A - hobby: Valores nominais no intervalo de 1 a 3.
- B - Idade: Valores nominais no intervalo de 1 a 4.
- C - Nível Educacional: Valores no intervalo de 1 a 4.
- D - Estado Civil: Valores no intervalo de 1 a 4.
- E - Atributo meta: Valores no intervalo de 1 a 3.

Esta base apresenta 132 exemplos de treinamento e 28 exemplos de teste. Para o presente trabalho foram separados 28 exemplos dos 132 exemplos de treinamento para validação durante a execução do algoritmo genético. A classificação para esta base é apresentada a seguir.

Somente atributos B-D são diagnosticados; valor A é ignorado.

Nenhuma das Classes: Se um 4 ocorre para qualquer atributo B-D.

Classe1: Se (número de 1's) > (número de 2's) para atributos B-D.

Classe2: Se (número de 2's) > (número de 1's) para atributos B-D.

Classe 1 ou 2: Se (número de 1's) = (número de 2's) para atributos B-D.

Criou-se para esta base uma rede neural *feedforward* totalmente conectada, com 15 neurônios na camada de entrada, 5 neurônios na camada oculta e 3 neurônios na camada de saída. Para a evolução com o algoritmo genético foram efetuadas 50 gerações, com população de 30 indivíduos e 10% de mutação.

A segunda base utilizada foi a Zoo. Esta base possui 16 atributos previsores e um atributo meta como apresentado a seguir:

- A - Pelo: 0 ou 1.
- B - Penas: 0 ou 1.
- C - Ovos: 0 ou 1
- D - Leite: 0 ou 1
- E - Voador: 0 ou 1.
- F - Aquático: 0 ou 1.
- G - Predador: 0 ou 1.
- H - Dentado: 0 ou 1.
- I - coluna vertebral: 0 ou 1.
- J - Respira: 0 ou 1.
- K - Venenoso: 0 ou 1
- L - Barbatana: 0 ou 1.
- M - Pernas: Valores possíveis [0,2,4,5,6,8].
- N - Rabo: 0 ou 1.
- O - Doméstico: 0 ou 1.
- P - Catsize: 0 ou 1.
- Q - Atributo Meta: 7 classes.

Esta base apresenta 101 exemplos. Para este trabalho os exemplos foram divididos em 3 conjuntos: (a) 61 exemplos de treinamento; (b) 21 exemplos de validação; e (c) 19 exemplos de testes. Para esta base foi criada uma rede neural *feedforward* totalmente conectada, com 24 neurônios na camada de entrada, 4 neurônios na camada oculta e 7 neurônios na camada de saída. Para evolução foram efetuadas 60 gerações, com população de 50 indivíduos e 20% de mutação.

#### 4. Resultados

A figura 3 apresenta a evolução do melhor *fitness*, do pior *fitness* e da média dos *fitness*, para o problema Hayes-Roth. As figuras 4 e 5 apresentam a evolução da taxa de acerto do conjunto de regras e a compreensibilidade do conjunto de regras obtido, respectivamente. A taxa de acerto do conjunto de regras no conjunto de exemplos de teste foi de 89,28%. A compreensibilidade foi 0,923 numa escala de 0 a 1 obtida através de (4).

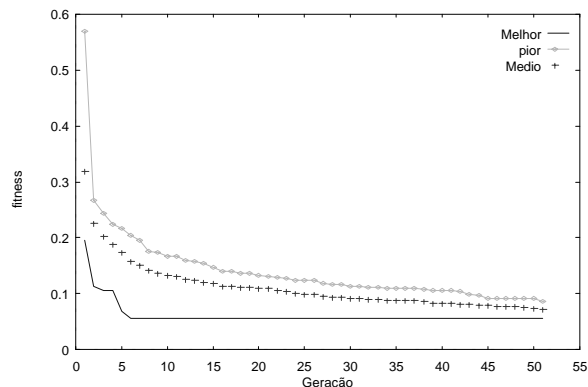


Figura 3 – Evolução do *fitness* da base Hayes-Roth.

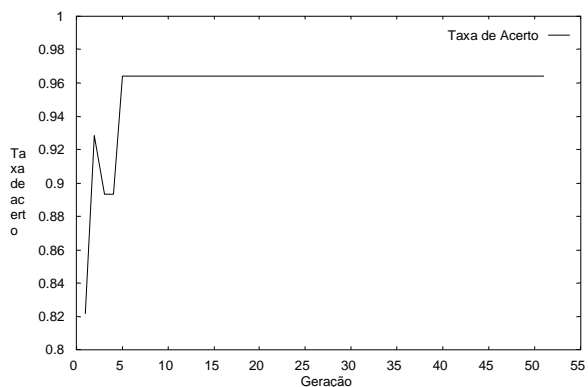


Figura 4 – Evolução da taxa de acerto das regras da base Hayes-Roth.

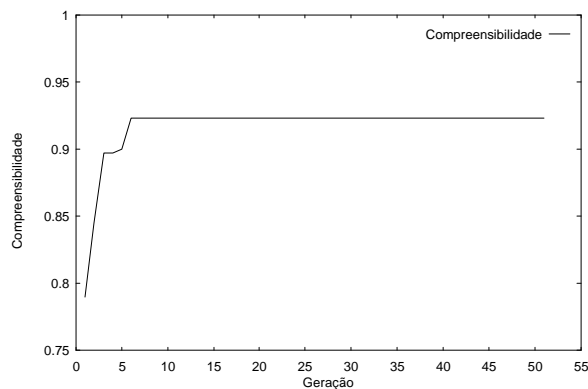


Figura 5 – Evolução da compreensibilidade das regras da base Hayes-Roth.

A seguir é apresentado o conjunto de regras obtido pela evolução do algoritmo genético em conjunto com o algoritmo de extração de regras RX. Na tabela 1 encontra-se o percentual de acerto de cada regra obtida.

- Regra 1: Se (B =1) E (C=1) E (D =2) então Classe 1
- Regra 2: Se (B ≠ 4) E (C = 1) E (D ≠ 2) E (D ≠ 4) então Classe 1
- Regra 3: Se (B = 1) E (C ≠ 4) E (D ≠ 2) E (D ≠ 4) então Classe 1

Regra 4: Se  $(B \neq 1) \text{ E } (B \neq 4) \text{ E } (C = 1) \text{ E } (D \neq 2) \text{ E } (D \neq 4)$  então Classe 2

Regra 5: Se  $(B \neq 4) \text{ E } (C \neq 1) \text{ E } (C \neq 4) \text{ E } (D = 2)$  então Classe 2

Regra 6: Se  $(B \neq 1) \text{ E } (B \neq 4) \text{ E } (C \neq 1) \text{ E } (C \neq 4) \text{ E } (D \neq 2) \text{ E } (D \neq 4)$  então Classe 2

Default – Classe 3.

Tabela 1 – Hayes-Roth: Taxa de acerto das regras obtidas.

Regra	Dados de teste	
	Exemplos cobertos pela regra.	Percentual de acerto.
1	1	100%
2	6	100%
3	4	100%
4	2	100%
5	6	83%
6	8	75%

A figura 6 apresenta a evolução do melhor *fitness*, do pior *fitness* e a média dos *fitness*, para o a base zoo. As figuras 7 e 8 apresentam a evolução da taxa de acerto do conjunto de regras e da compreensibilidade do conjunto de regras obtido, respectivamente. A taxa de acerto do conjunto de regras no conjunto de exemplos de teste foi de 78,94%. A compreensibilidade foi 0,946 numa escala de 0 a 1 obtida através de (4).

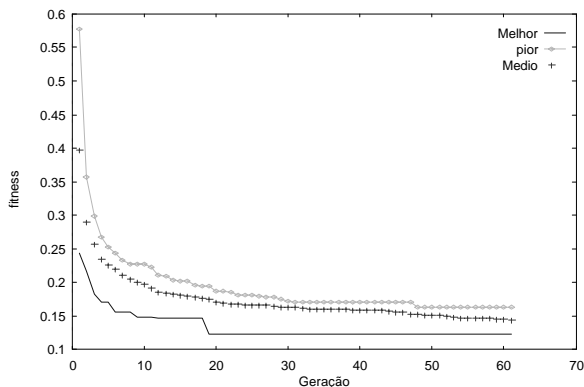


Figura 6 – Evolução do *fitness* base zoo.

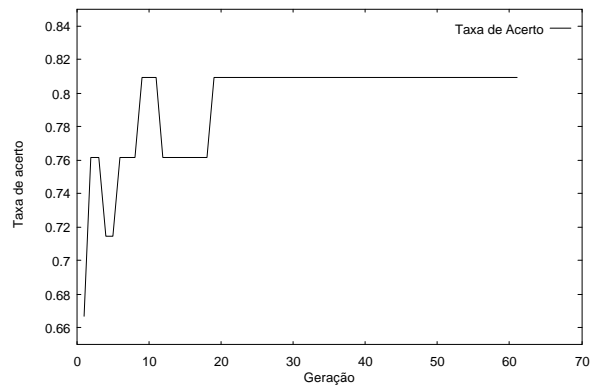


Figura 7 – Evolução da taxa de acerto das regras da base zoo.

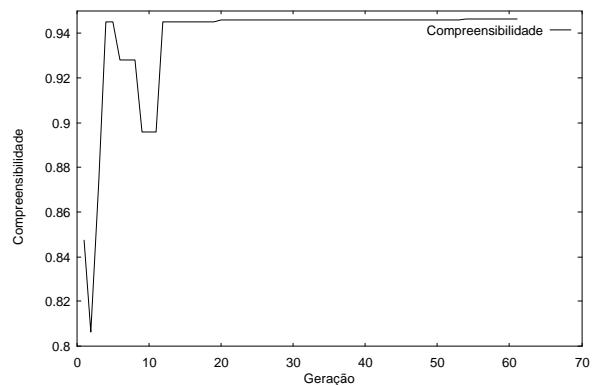


Figura 8 – Evolução da compreensibilidade das regras da base zoo.

A seguir é apresentado o conjunto de regras obtidos pela evolução do algoritmo genético em conjunto com o algoritmo de extração de regras RX. Na tabela 2 encontra-se o percentual de acerto de cada regra obtida.

- Regra 1: Se D então Classe 1
- Regra 2: Se  $(\text{Not}(D)) \text{ E } (M \neq 2) \text{ E } (M \neq 6)$  então Classe 4
- Regra 3: Se  $(\text{Not}(D)) \text{ E } (M = 6) \text{ E } (\text{Not}(H))$  então Classe 6
- Default – Classe 2

Tabela 2 – Zoo database: Taxa de acerto das regras obtidas.

Regra	Dados de teste	
	Exemplos cobertos pela regra.	Percentual de acerto.
1	8	100%
2	6	33%
3	1	100%

## 5. Discussão e análise dos resultados

Comparando-se a taxa de acerto do conjunto de regras obtidos da base Hayes-Roth citada na sessão

anterior, a saber 89,28%, com o percentual de ocorrência da classe *default* (classe da maioria) no conjunto de testes, que é de 50%, observa-se boa precisão do conjuntos de regras. Resultado semelhante também foi obtido para a base zoo. Apesar da precisão obtida ser um pouco menor, a saber 78,94%, este valor ainda é maior que o percentual de ocorrência da classe *default* no conjunto de teste, cujo percentual é de 42,1%.

Pode-se observar nas tabelas 1 e 2 os percentuais de acerto individual de cada uma das regras obtidas, sendo bastante satisfatório com exceção da regra 2 na tabela 2. Deve-se observar que estes percentuais foram calculados sobre dados de teste não vistos durante o processo de evolução.

As regras descobertas também são compreensíveis, consistindo de um pequeno número de condições, conforme foi visto anteriormente.

## 6. Conclusões e trabalhos futuros

Através do uso de algoritmos genéticos foi possível encontrar uma topologia de rede neural artificial (número de conexões e número de unidade na camada intermediária) que possibilitou a extração de um conjunto de regras que fornecessem uma boa taxa de acerto e ao mesmo tempo fossem tão compreensíveis quanto possível.

Na mesma linha deste trabalho poderiam ser implementados outros algoritmos de extração de regras de redes neurais, bem como otimização do algoritmos de extração de regras para possibilitar sua utilização com grandes bases de dados.

## Referências Bibliográficas

- [1] Fayyad, V.M.et.al. From data mining to Knowledge Discovery: an overview. In:Gayyad, V.M.et.al. (Eds.) *Advances in Knowledge Discovery and Data Mining*,1-34. AAAI/MIT,1996.
- [2] Lu, H., Setiono, H., Liu, H. NeuroRule: a connectionist approach to data mining. *Proc. 21<sup>st</sup> Conf. on Very Large Databases*. Zurich, 1995.
- [3] Goldberg, D. E. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison Wesley, 1989.
- [4] Freitas, A. A., Lavington, S. H. *Mining very large databases with parallel processing*. London: Kluwer Academic Publisher; 1998.
- [5] Braun, H., Ragg, T., *ENZO – User Manual and Implementation Guide*, Version 1.0, University of Karlsruhe; 1995.
- [6] Janikow, C. Z. A knowledge intensive genetic algorithm for supervised learning. *Machine Learning*, 13, 189-228, 1993.
- [7] Hayes-Roth, B., Hayes-Roth, F. Concept learning and the recognition and classification of exemplars, *Journal of Verbal Learning and Verbal Behavior*, 16, 321-338, 1977.