

# FERRAMENTA PARA O DESENVOLVIMENTO DE REDES NEUROFUZZY

NÍRIA B. FERREIRA, MERISANDRA C. DE MATTOS, EVELISE C. ZANCAN, PRISCYLA WALESKA T. A. SIMÕES

*Grupo de Pesquisa em Inteligência Computacional Aplicada, Unidade Acadêmica de Ciências, Engenharias e Tecnologias, Curso de Ciência da Computação, Curso de Engenharia Civil, Universidade do Extremo Sul Catarinense*

*Av. Universitária 1105, Bairro Universitário 88806-000, Criciúma, SC, Brasil*

*E-mails: niriaborges@hotmail.com, mem@unesc.net, evelise@unesc.net, pri@unesc.net*

**Abstract**— This article presents a hybrid boarding for the development of intelligents systems, by means of the one archetype shell neurofuzzy called Ícaro. The main objective of the use of the neurofuzzy networks is the development of systems where the knowledge is represented and processed of explicit form with easy interpretation. This occurs due the incorporation of the knowledge specialist for the logic fuzzy and capacity of learning of the neural networks. Shell Ícaro was tested developing a referring system neurofuzzy the evaluation of urban property of the type apartment in the city of Criciúma.

**Keywords**— Artificial Intelligence, Neural Networks, Fuzzy Logic, Neurofuzzy Networks, Evaluation of Property.

**Resumo**— Este artigo apresenta uma abordagem híbrida para o desenvolvimento de sistemas inteligentes, por meio do protótipo de uma *shell neurofuzzy* denominada Ícaro. O principal objetivo da utilização das redes *neurofuzzy* é o desenvolvimento de sistemas onde o conhecimento seja representado e processado de forma explícita com fácil interpretação. Isto ocorre devido a incorporação do conhecimento especialista pela lógica *fuzzy* e capacidade de aprendizado das redes neurais. A *shell* Ícaro foi testada desenvolvendo-se um sistema *neurofuzzy* referente a avaliação de imóveis urbanos do tipo apartamento no município de Criciúma.

**Palavras-chave**— Inteligência Artificial, Redes Neurais, Lógica *Fuzzy*, Redes *Neurofuzzy*, Avaliação de Imóveis.

## 1 Introdução

A união de duas ou mais técnicas da inteligência artificial constitui-se no desenvolvimento de um sistema de arquitetura híbrida que compensa as deficiências de uma com os benefícios da outra. A pesquisa apresentada consistiu no desenvolvimento de um sistema de arquitetura híbrida onde a lógica *fuzzy* é incorporada a estrutura de uma rede neural artificial, conhecida como rede *neurofuzzy*.

O sistema desenvolvido consiste no protótipo de uma *shell neurofuzzy*, que pode ser aplicada na solução de diversos tipos de problema. A realização dos testes nesta pesquisa deu-se por meio de dados referentes a avaliação de imóveis urbanos do tipo apartamento no município de Criciúma.

A avaliação de imóveis é uma atividade realizada pelos Engenheiros de Avaliações e Perícias, que propõem estimativas de mercado para determinar o valor de moradia e outros tipos de propriedade onde, a primeira aproximação, é a comparação de venda com a de propriedade similar que tenha sido vendida recentemente, com ajuste dos respectivos preços (ZANCAN, 1996).

Este método comparativo dos dados de mercado, deve ser utilizado preferencialmente em uma avaliação onde, o valor da propriedade avaliada ou de algumas de suas partes, é determinado por meio de comparações entre dados de mercado relativos a outros imóveis de características semelhantes. A aplicação deste método desenvolve-se por meio das seguintes etapas: caracterização do imóvel avaliado; pes-

quisa de dados relativos a imóveis semelhantes ao avaliado e cálculo do valor (ZANCAN, 1996).

A determinação do valor de uma propriedade pode ser feita baseando-se em tratamento estatístico, por meio da estatística descritiva ou inferencial.

A estatística inferencial utiliza variáveis qualitativas e quantitativas na verificação de evidências para formação dos valores dos imóveis. As variáveis qualitativas identificam características que não são medidas por uma escala numérica definida ou que se resume a existência ou não de determinados atributos, tais como: suíte, infra-estrutura e conservação. As variáveis quantitativas são medidas diretamente das grandezas em estudo numa escala numérica, tais como: área, número de dormitórios, distâncias e idade.

Devido à possibilidade de utilização das variáveis quantitativas e qualitativas da estatística inferencial como variáveis lingüísticas, a pesquisa aqui apresentada aplicou as redes *neurofuzzy* para realizar o tratamento da incerteza por imprecisão presente nesta área, com a possibilidade de incorporação do conhecimento na forma de aprendizagem.

As redes neurais e a lógica *fuzzy* foram combinadas para formar um sistema de arquitetura híbrida, pois de acordo com Azevedo, Brasil e Oliveira (2000) estes absorvem características de ambas as técnicas provendo dois modos complementares de modelagem do conhecimento humano.

Mediante isso, esta pesquisa compreendeu o desenvolvimento do protótipo de um sistema de arquitetura híbrida, utilizando-se das técnicas de redes neurais artificiais e lógica *fuzzy*, para o desenvolvi-

mento de uma rede *neurofuzzy*, podendo-se assim proporcionar uma alternativa para auxílio a avaliação de imóveis, podendo também ser aplicada na solução de outros tipos de problemas.

## 2 Redes Neurofuzzy

Desde o início da década de 80, diversos autores buscam reunir a capacidade da lógica *fuzzy* de incorporar o conhecimento de especialistas e representar conceitos imprecisos, com a habilidade de aprendizado e poder de processamento das RNA para geração de um sistema de arquitetura híbrida denominado rede *neurofuzzy* (REZENDE, 2003).

Desta forma, o principal objetivo da utilização das redes *neurofuzzy* é o desenvolvimento de sistemas onde o conhecimento seja representado e processado de forma explícita e de fácil interpretação, além de possuir a capacidade de aprendizado das redes neurais.

Conforme Fullér (1995) as redes *neurofuzzy* possuem duas abordagens distintas que diferem-se pela arquitetura e funcionamento de seus modelos, podendo-se classificar estes sistemas como *híbrido* ou *cooperativo*.

O modelo de rede *neurofuzzy* utilizado para o desenvolvimento da *shell* Ícaro foi o híbrido.

### 2.1 Redes Neurofuzzy Híbridas

A arquitetura de uma rede *neurofuzzy* híbrida é composta por  $n$  camadas de neurônios onde cada camada constitui-se de uma etapa do raciocínio *fuzzy*, conforme ilustrado na Figura 24. Basicamente, uma rede *neurofuzzy* híbrida pode ser vista como redes *multi-layer* de três camadas, sendo estas as camadas de *fuzzyficação*, regras *fuzzy* e *defuzzyficação*. O processamento dos neurônios de cada camada, denominado neurônio *fuzzy*, calcula os resultados da seguinte forma (NAUCK; KRUSE, 1994):

- a) camada de *fuzzyficação*: cada neurônio recebe os valores de entrada no sistema e, por meio de uma função de pertinência, calcula o grau de pertinência das entradas fornecendo o valor do antecedente das regras *fuzzy* para a próxima camada;
- b) camada de regras *fuzzy*: representa a base de regras do sistema. Desta forma, cada neurônio desta camada corresponde a uma regra *fuzzy*. Sua função é calcular o valor do conseqüente da regra por meio de uma t-norma, fornecendo o resultado à camada seguinte;
- c) camada de *defuzzyficação*: fornece a saída ao sistema calculando o produto entre as somas dos níveis de disparo de cada neurônio da camada anterior e os valores dos conseqüentes das regras.

Este modelo de arquitetura *neurofuzzy híbrida* apresenta uma total integração das redes neurais com o sistema *fuzzy* e o aprendizado da rede é incorporado utilizando-se algum tipo de algoritmo de treinamento aplicável aos modelos das redes neurais convencionais.

## 3 Shell Neurofuzzy Ícaro

A pesquisa consistiu na implementação do protótipo de sistema híbrido *neurofuzzy*, ou seja, uma ferramenta para realizar a implementação de sistemas onde a modelagem das variáveis fosse obtida por meio de conjuntos *fuzzy* aplicados a uma rede neural.

Na *shell neurofuzzy* Ícaro implementou-se inicialmente somente um modo de inferência e de aprendizado, havendo assim, a possibilidade da modelagem de outros tipos de problemas além dos referentes à avaliação de imóveis. Desta forma, o protótipo de *shell neurofuzzy* implementado, poderá ser concluído em trabalhos futuros até que se obtenha um sistema completo e gratuito para utilização em outras pesquisas, universidades e instituições.

O protótipo da *shell* Ícaro foi desenvolvido em ambiente de programação *Borland Delphi 6.0* por meio da linguagem *Object Pascal*. Os testes foram realizados por meio de um conjunto de dados pesquisados e disponibilizados no Trabalho de Conclusão de Curso da acadêmica Graziela Olivo Fermo do curso de Engenharia Civil da UNESC.

Os dados utilizados para testes na Ícaro são referentes ao valor de imóveis do tipo apartamento, em oferta no mercado imobiliário do município de Criciúma, em conjunto com as variáveis que influenciam a geração de valores na avaliação de imóveis.

O desenvolvimento do protótipo de *shell neurofuzzy* Ícaro fundamentou-se metodologicamente pelas seguintes etapas: aquisição de conhecimento sobre a avaliação de imóveis; demonstração matemática da modelagem *fuzzy* da rede neural; realização da modelagem do protótipo; implementação do protótipo; treinamento da rede *neurofuzzy* e realização de testes.

### 3.1 Aquisição de Conhecimento

Esta etapa de desenvolvimento consistiu na realização de entrevista estruturada e desestruturada com a especialista em engenharia de avaliações, referente as técnicas e utilização das variáveis para obtenção dos resultados.

De acordo com os critérios estabelecidos pela engenheira avaliadora foram definidas as variáveis que possuíam maior influência na obtenção do valor de um imóvel para que fossem empregadas na pesquisa. As variáveis utilizadas na modelagem da rede *neurofuzzy* foram: área total do apartamento, número de dormitórios, localização e qualidade.

### 3.2 Demonstração Matemática da Modelagem Fuzzy

Na demonstração matemática da modelagem *fuzzy* foram definidos os conjuntos para cada variável de entrada e saída, bem como os seus respectivos intervalos, com a utilização da função trapezoidal, devido a maior facilidade de implementação e por representar adequadamente as teorias da lógica *fuzzy*, já que possuem um intervalo de valores do universo de discurso com pertinência completa ao conjunto.

Desta forma, obteve-se as variáveis lingüísticas e valores *fuzzy* para o domínio de aplicação referente a avaliação de imóveis da tipologia apartamento no município de Criciúma.

A partir da definição dos conjuntos foram identificadas as regras de inferência a serem utilizadas para resolução do problema, como por exemplo:

- SE Área Total = Pequeno
- E Nro De Dormitórios = Pouco E
- Localização= Perto
- E Qualidade= Inferior
- ENTÃO Valor=Baixo

Dada a definição dos conjuntos e das regras, foram realizadas as etapas de *fuzzyficação*, *inferência* e *defuzzyficação*.

O processo de *fuzzyficação* das entradas na *Shell Ícaro* ocorreu por meio do cálculo da função *trapezoidal*:

$$\text{Trap}(x,a,b,c,d) = \begin{cases} 0 & \text{se } x \leq a \\ (x - a) / (b - a) & \text{se } a < x < b \\ 1 & \text{se } b \leq x \leq c \\ (d - x) / (d - c) & \text{se } c < x < d \\ 0 & \text{se } x \geq d \end{cases}$$

Na próxima etapa realizou-se o processo de *inferência* por meio da interpretação das regras, extraindo-se os resultados das regras, ou seja, valores mínimos entre suas relações, tal que:  $S_n = \min [\mu_{\text{ÁREA TOTAL}}(x), \mu_{\text{NRO DE DORMITÓRIOS}}(x), \mu_{\text{LOCALIZAÇÃO}}(x), \mu_{\text{QUALIDADE}}(x)]$ , para  $n=1,2,3...N$ .

Após a determinação dos valores mínimos das relações de interseção é realizada uma *normalização* dos valores gerados, tal que:

$$S'_n = S_n / (S_1 + S_2 + S_3 + \dots + S_n) \quad (1)$$

O último processo na etapa de *inferência* é realizado por meio da aplicação da operação de *união* padrão entre os resultados de cada conseqüente para obtenção do valor máximo, tal que, por exemplo:

$$\text{Médio}(x) = \max [S'_1, S'_2, S'_3, S'_4] \quad (2)$$

O valor obtido na *união* é utilizado no processo de *defuzzyficação* pelo método *centróide* por meio do qual o centro é localizado, tal que:

$$h_n = A(x) \cdot C_A \quad (3)$$

Onde  $A(x)$  é o valor resultante da operação *união* e  $C$  é o valor do conseqüente, que na *função trapezoidal* é obtido por meio do cálculo do valor médio entre o inicial e final de cada conjunto.

Na etapa de *defuzzyficação* a saída do sistema é calculada, atribuída a uma variável  $y$ , e consiste no somatório dos valores de  $h_n$ , sendo:

$$y = h_1 + h_2 + h_3 \dots h_n \quad (4)$$

### 3.3 Modelagem da Shell Ícaro

A modelagem do protótipo da *shell* ÍCARO foi realizada com o objetivo de simplificar e padronizar a interface, além de facilitar seu desenvolvimento, bem como a interação do usuário com o protótipo. Na realização da modelagem foi elaborada por meio da *Unified Modeling Language* (UML), com a utilização dos diagramas de caso de uso, atividades e seqüência.

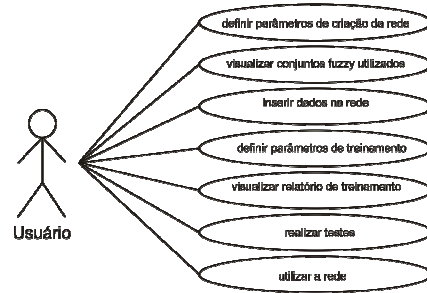


Figura 1. Diagrama de Caso de Uso das Ações do Usuário

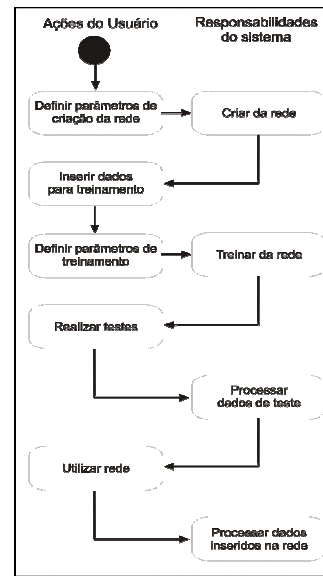


Figura 2. Diagrama de Atividades da Shell Ícaro

### 3.4 Implementação da Shell Neurofuzzy Ícaro

A implementação do protótipo de sistema foi definida em dois modelos: *fuzzy* e neural.

O modelo *fuzzy* foi realizado por meio da utilização da função trapezoidal e a *defuzzyficação* centróide. A função trapezoidal realiza a *fuzzyficação* das entradas no sistema composta pelas seguintes

variáveis: área total, número de apartamentos, padrão de qualidade, localização e valor.

Ao iniciar a criação de uma rede o usuário define os conjuntos *fuzzy* de entrada por meio da função trapezoidal, conforme ilustrado na Figura 3.

Na próxima etapa de criação são definidos os conjuntos de saída utilizando o mesmo tipo de função (Figura 4).

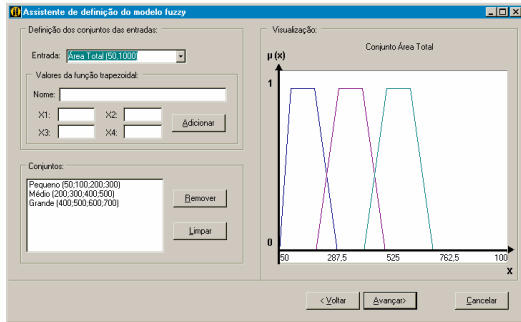


Figura 3. Definição dos Conjuntos *Fuzzy* de Entrada

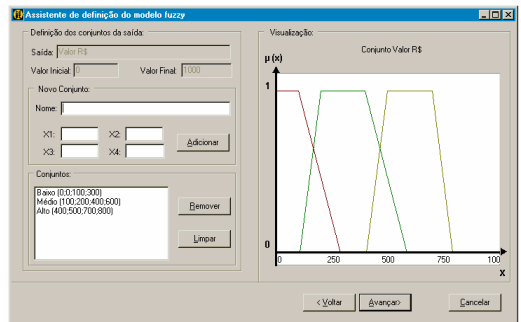


Figura 4. Definição dos Conjuntos *Fuzzy* de Saída

Após a definição dos conjuntos *fuzzy* que serão utilizados na arquitetura da rede, o sistema gera todas as possibilidades das regras de inferência para estes conjuntos, para que sejam escolhidas pelo usuário, bem como seus conseqüentes, de acordo com a Figura 5.

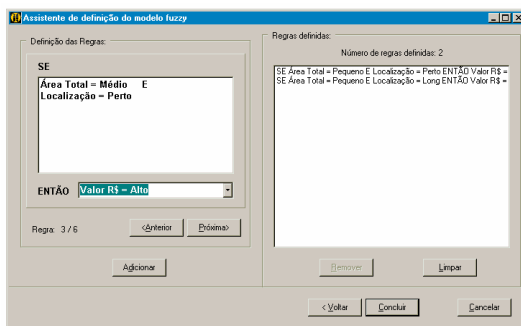


Figura 5. Definição das Regras de Inferência

A partir desta etapa, o sistema inicia o processo de criação da parte neural para geração da rede *neurofuzzy*.

No modelo neural foi utilizada a estrutura de uma rede *multilayer perceptron*. A partir dos conjuntos de entrada e saída *fuzzy* fornecidos pelo usuário, o

sistema gera uma rede com seis camadas onde a rede possui a seguinte estrutura (Figura 6):

- camada de entrada:** recebe os valores fornecidos pelo usuário para treinamento, teste ou utilização. Esta camada não processa os dados recebidos, sua função consiste apenas em enviar os dados para os neurônios que deverão receber estes valores na próxima camada;
- camada de conjuntos:** ao receber os valores da primeira camada, cada neurônio calcula o grau de pertinência dos valores recebidos para que sejam enviados à próxima camada como antecedente das regras. Como foi dito anteriormente, o grau de pertinência é calculado no sistema por meio da função trapezoidal;
- camada de regras:** cada neurônio desta camada corresponde a uma regra *fuzzy*. O grau com que o conseqüente será atendido é calculado por meio da execução da *interseção* padrão dos valores dos antecedentes, que neste caso, são os valores recebidos da camada anterior. A saída dos neurônios desta camada são enviados a todos os neurônios da próxima;
- camada de normalização:** os neurônios desta camada realizam o processo de normalização dos valores tal que,  $S'n = S'n / (S1 + S2 + S3 + \dots + Sn)$ , onde  $S'$  é o valor de saída do neurônio e  $Sn$  os valores provenientes de todos os neurônios da camada anterior. A saída deste neurônio é enviada a apenas um neurônio da próxima camada, ou seja, para aquele que corresponde ao conseqüente do neurônio na camada de regras, gerado pelo método centróide;
- camada de conjuntos da saída:** nesta camada é executada uma operação de *união* padrão entre todos os valores provenientes da camada anterior. O neurônio fornece a entrada calculando o produto entre o maior valor proveniente da camada anterior e o conseqüente da regra, tal que:  $h_n = S'n * C_n$ , onde  $h$  é a saída do neurônio,  $S'$  o maior valor recebido da camada de normalização e  $C$  o conseqüente da regra;
- camada de saída:** possui apenas um neurônio e fornece a saída do sistema, sendo calculada pela soma dos valores de saída da camada anterior, tal que:  $z = h1 + h2 + \dots + hn$ .

A Figura 6 representa a rede neural resultante da definição do modelo *fuzzy* pelo usuário. Enquanto, a Figura 7 apresenta a tela principal do sistema, onde são inseridos os dados para o treinamento da rede, bem como as taxas de aprendizado e o número de épocas para posterior realização dos testes, demonstrando-se ao usuário a arquitetura da rede *neurofuzzy* gerada.

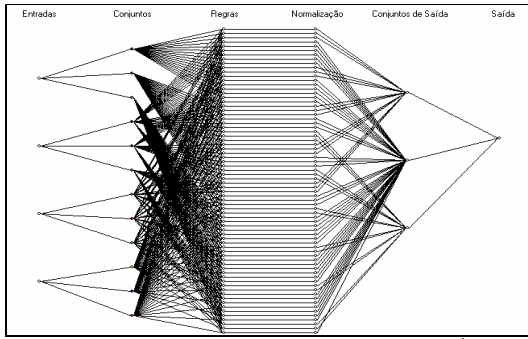


Figura 6. Arquitetura da Rede Neurofuzzy da Shell Ícaro

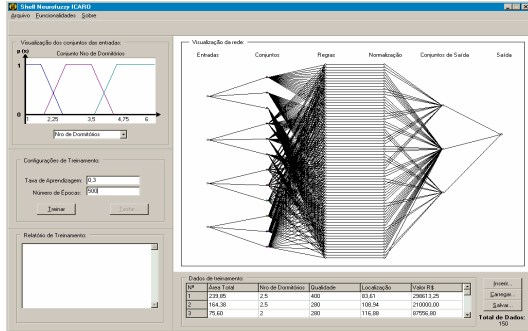


Figura 7. Tela principal da Shell Neurofuzzy Ícaro

### 3.5 Treinamento da Rede Neurofuzzy

O treinamento da rede *neurofuzzy* na *Shell Ícaro* foi realizado de forma supervisionada por meio do algoritmo *backpropagation*. Ao receber o valor de saída da rede no processo de propagação dos valores é calculado o erro por meio da equação:

$$e_j(n) = d_j(n) - y_j(n) \quad (4)$$

Onde  $j$  representa o neurônio de saída,  $d_j$  o valor desejado e  $y_j$  o valor resultante da rede.

Utilizando-se, por exemplo, o valor de saída  $y = 230.400$  para as entradas área total = 164,38, número de dormitórios=2, localização=108,94 e qualidade=280, tendo como valor desejado de saída 210000, tem-se:

$$e_j(n) = d_j(n) - y_j(n) = 210000 - 230400 = -20400$$

Na rede *neurofuzzy* gerada, o valor resultante e desejado são divididos pelo valor do conseqüente para que seja minimizado o custo da rede, desta forma:

$$e_j(n) = \frac{21000}{210000} - \frac{230400}{210000}$$

$$e_j(n) = 1,75 - 1,92 = -0,17$$

O sinal de erro é utilizado no ajuste dos pesos tal que:  $\Delta w_j = \eta \cdot (e_j(n) \cdot y_j(n))$ , onde  $\Delta w$  representa o *delta correção* e  $\eta$  determina a taxa de aprendizagem definida pelo usuário. Neste exemplo, com a atualização de uma taxa de aprendizagem de 0,3, resultando em:

$$\Delta w_j = \eta \cdot (e_j(n) \cdot y_j(n))$$

$$\Delta w_j = 0,3 \cdot (-0,17 \cdot 1,92) = -0,09792$$

Desta forma, a modificação nos valores dos pesos é realizada tal que:  $w_j = w_j + \Delta w_j$ , onde  $w$  é o peso atual do neurônio  $j$ , tendo-se:

$$w_j = w_j + \Delta w_j = 1 - 0,09792 = 0,90208$$

Este processo é realizado para o ajuste dos pesos das camadas de normalização e entrada dos conjuntos durante várias épocas de aprendizado para que seja obtida uma generalização dos dados do conjunto de treinamento.

Os dados referentes a avaliação de imóveis do tipo apartamento em Criciúma foram separados em duas partes: a primeira para a realização do treinamento da rede *neurofuzzy* composta por 150 dados; e a segunda, utilizada na realização dos testes da rede, com 55 dados.

### 3.6 Realização de Testes

Os testes da rede *neurofuzzy* na *Shell Ícaro* foram realizados pela utilização da base de dados mencionada anteriormente, que trata de valores reais referentes a apartamentos em oferta no mercado imobiliário do município de Criciúma de janeiro à abril de 2006.

Inicialmente, a arquitetura utilizada apresentou diversas dificuldades na parte de aprendizado. Para que a rede apresentasse o melhor resultado possível, foram testadas as atualizações de pesos de diferentes formas para obter-se uma maior generalização das amostras apresentadas.

Mediante isto, para aquisição do modelo mais eficaz de atualização dos pesos sinápticos, escolheu-se a realização da modificação dos pesos somente nas camadas de saída dos graus de pertinência (camada b) e normalização das saídas das regras (camada d), por apresentarem os melhores resultados.

Os resultados obtidos no treinamento da rede *neurofuzzy* com a utilização deste modelo por meio da base de dados é descrita a seguir.

## 4 Resultados Obtidos

A *shell* desenvolvida permite a geração de sistemas *neurofuzzy*, portanto, para exemplificar a sua utilização, considerando-se para uma situação de entrada, onde Área Total {pequeno, médio, grande}, juntamente com seus respectivos limites para determinação dos valores na função trapezoidal, o modelo *fuzzy* gerado é apresentado ao usuário (Figura 8).

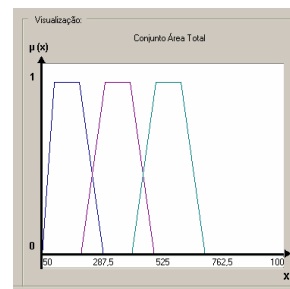


Figura 8. Representação do Modelo Fuzzy

## 5 Conclusão

A partir da inserção de todos os conjuntos das entradas cria-se a parte referente a rede neural, onde inserem-se os dados de treinamento (Figura 9) juntamente com a taxa de aprendizado e o número de épocas de treinamento desejado. Este conjunto de dados pode ser salvo e carregado posteriormente para que seja utilizado em outros treinamentos de mesmo domínio de aplicação, ou seja, avaliação de imóveis do tipo apartamento em Criciúma.

Nº	Área Total	Nro de Dormitórios	Qualidade	Localização	Valor R\$
1	228,05	2,5	400	83,81	238613,25
2	164,38	2,5	290	108,94	210000,00
3	75,60	2	280	116,88	87556,80

Figura 9. Inserção de dados para treinamento na Shell Ícaro

Na Figura 10 tem-se parte de tela principal que demonstra o desempenho da rede, por meio do relatório de treinamento, que pode ser verificado a fim de se conhecer o percentual de erro obtido durante o aprendizado. Após a rede treinada, é possível executar a realização dos testes para posterior realização de consultas.

Figura 10. Relatório de Treinamento da Shell Ícaro

Foram inseridos valores diferentes para as taxas de aprendizado, no intervalo entre 0,1 e 0,5, bem como o número de épocas de treinamento fixado em 2000 para verificação da melhor generalização do conhecimento apresentado pelos dados de treinamento. Durante o processo de treinamento da rede, foram detectados os menores erros médios com a taxa de aprendizado de 0,5 (Figura 11).

Taxa de Aprendizado	Erro médio total	Percentual de Erro médio
0,1	0,121258	12,12%
0,2	0,088933	8,89%
0,3	0,077115	7,71%
0,4	0,079634	7,96%
0,5	0,070602	7,06%

Figura 11. Resultados dos testes na Shell Ícaro

Além da realização de testes com todos os dados obtidos na pesquisa, foram inseridos na rede números menores de dados a fim de realizar uma comparação da influência que a quantidade de amostras exerce no treinamento da rede, com a taxa de aprendizado de 0,5 apresentada anteriormente como o melhor resultado (Figura 12).

Épocas	Erro médio total	Percentual de Erro médio
20	0,079799	7,97%
50	0,078590	7,85%
70	0,077453	7,74%
100	0,076894	7,68%
150	0,070602	7,06%

Figura 12. Testes na Shell Ícaro

A utilização de sistemas de arquitetura híbrida tem se mostrado uma alternativa válida na solução de diversos tipos de problema. Isto se deve ao fato de uma técnica servir como complemento da outra e ambas compensarem suas deficiências.

No que se refere a *Shell Neurofuzzy Ícaro*, conforme os resultados apresentados, foi constatado que a arquitetura definida no sistema por meio das variáveis utilizadas, obtém um melhor desempenho com uma taxa de treinamento de 0,5 em um número de 150 épocas, obtendo assim, uma taxa de erro de 0,0706, ou seja, 7,06%.

No desenvolvimento da pesquisa uma das dificuldades encontradas refere-se ao aprendizado da rede devido a baixa quantidade de dados disponível para treinamento. Embora a tipologia de apartamentos possua um número maior de dados amostrais, existe uma grande dificuldade no que se refere à aquisição de dados reais completos, atualizados e que não apresentem redundâncias. Mesmo com uma quantidade relativamente baixa de dados, foi possível a construção de um modelo de sistema com uma boa taxa de acerto, podendo ser utilizada em trabalhos futuros para realização de outros testes na área de avaliação de imóveis, além de outras aplicações.

## Referências Bibliográficas

- Azevedo, F. M., Brasil, L. M. e Oliveira, R. C. L. (2000). *Redes Neurais com Aplicações em Controle e em Sistemas Especialistas*. Florianópolis: Bookstore.
- Fullér, R. *Neural Fuzzy Systems*. (1995). Disponível em: <<http://citeseer.nj.nec.com/64350.html>>.
- Nauck, D. e Kruse, R. (1994). Choosing Appropriate Neuro-Fuzzy Models, *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing*, 1, Aachen.
- Rezende, S. O. (2003). *Sistemas inteligentes: fundamentos e aplicações*. São Paulo: Manole.
- Zancan, E. C. (1996). *Avaliação de Imóveis em Massa para Efeitos de Tributos Municipais*. Florianópolis: Rocha.