

UNIVERSIDADE DO EXTREMO SUL CATARINENSE

CURSO DE CIÊNCIA DA COMPUTAÇÃO

NIRIA BORGES FERREIRA

REDES *NEUROFUZZY*

PARA AVALIAÇÃO DE APARTAMENTOS EM CRICIÚMA

CRICIÚMA, NOVEMBRO DE 2006.

NIRIA BORGES FERREIRA

REDES *NEUROFUZZY*

PARA AVALIAÇÃO DE APARTAMENTOS EM CRICIÚMA

Trabalho de Conclusão de Curso para
Obtenção do Grau de Bacharel em Ciência da
Computação da Universidade do Extremo Sul
Catarinense.

Orientadora: Profa. MSc. Merisandra Côrtes
de Mattos

Co-orientadora: Profa. M. Eng. Evelise
Chemale Zancan

CRICIÚMA, NOVEMBRO DE 2006.

Aos meus maravilhosos pais, Nardi e Nivia, minha orientadora Merisandra e meus amigos pelo apoio e compreensão.

AGRADECIMENTOS

Agradeço a Deus por me guiar e iluminar o meu caminho em todos os momentos da minha vida.

Agradeço também:

Aos meus pais por serem tão maravilhosos e me proporcionarem tanto carinho, amor e ajuda nos momentos difíceis.

Ao meu amigo César por fazer parte de minha vida e estar sempre presente nos momentos difíceis e compartilhar os felizes.

A minha co-orientadora Evelise e aos professores Evânio e Ana Cláudia pela ajuda concedida.

Aos meus colegas de graduação pela amizade e colaboração proporcionadas no decorrer do curso.

Em especial, agradeço a minha professora e orientadora Merisandra, que contribui com o seu conhecimento na realização deste trabalho e também pela sua amizade e orientação.

*“Ainda acredito na possibilidade de
construirmos um modelo da realidade”*

- Albert Einstein -

RESUMO

Esta pesquisa apresenta uma abordagem híbrida para o desenvolvimento de sistemas conexionistas, por meio do protótipo de uma shell neurofuzzy, denominada *Ícaro*, que utiliza duas técnicas da Inteligência Artificial para solução de diversos tipos de problemas: as redes neurais artificiais e a lógica *fuzzy*. 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 à incorporação do conhecimento especialista pela lógica *fuzzy* e capacidade de aprendizado das redes neurais. A *shell Ícaro* foi testada desenvolvendo-se um sistema *neurofuzzy* capaz de incorporar conhecimento a respeito do comportamento do mercado imobiliário do município de Criciúma na avaliação de imóveis urbanos do tipo apartamento. A tipologia de apartamentos foi escolhida para análise nesta pesquisa por possuir maior representatividade quanto ao número de dados amostrais. Embora a quantidade adquirida para utilização na rede tenha sido baixa, os resultados obtidos foram satisfatórios, apresentando uma boa generalização para resolução do problema proposto.

Palavras-chaves: Inteligência Artificial Conexionista, Redes Neurais, Lógica *Fuzzy*, Redes *Neurofuzzy*, Avaliação de Imóveis.

LISTA DE ILUSTRAÇÕES

Figura 1. Representação dos estágios de funcionamento do sistema nervoso	21
Figura 2. Neurônio biológico	22
Figura 3. Neurônio Artificial de McCulloch e Pitts	24
Figura 4. Neurônios de McCulloch-Pitts para calcular as funções lógicas E e OU.....	25
Figura 5. Resultados do modelo de McCulloch e Pitts para as funções lógicas E e OU	25
Figura 6. Modelo geral de um neurônio.....	32
Figura 7. Exemplo de uma rede neural <i>multicamadas</i>	38
Figura 8. Classificação de valores em um conjunto <i>crisp</i>	41
Figura 9. Conjuntos <i>crisp</i> A e B em um universo de discurso U	44
Figura 10. Relações de pertinência de x aos conjuntos A e B	45
Figura 11. Conjunto fuzzy $B = \{ (x, \mu_A(x)) \mid x \in U \}$, onde $x=110$	46
Figura 12. Representação gráfica da função trapezoidal.	48
Figura 13. Representação gráfica da função triangular.....	48
Figura 14. Representação gráfica da função S	49
Figura 15. Representação gráfica da função Z	50
Figura 16. A distância que se encontra um objeto como uma variável lingüística.....	52
Figura 17. Um conjunto <i>fuzzy</i> A em um universo de discurso U	54
Figura 18. Etapas de desenvolvimento dos sistemas <i>fuzzy</i>	56
Figura 19. Aplicação do conectivo de intersecção.....	58
Figura 20. Aplicação do conectivo de união.	59
Figura 21. Modelo de sistema híbrido seqüencial.....	62

Figura 22. Modelo de sistema híbrido auxiliar.	62
Figura 23. Modelo de sistema híbrido incorporado.	63
Figura 24. Modelo da arquitetura de uma rede neurofuzzy híbrida.	65
Figura 25. Modelos da arquitetura de uma rede neurofuzzy cooperativa.	66
Figura 26. Representação gráfica do conjunto de entrada Área Total.	79
Figura 27. Representação gráfica do conjunto de entrada Número de Apartamentos.	80
Figura 28. Representação gráfica do conjunto de entrada Localização.	81
Figura 29. Representação gráfica do conjunto de entrada Qualidade.	82
Figura 30. Representação gráfica do conjunto de saída Valor.	82
Figura 31. Representação gráfica dos valores de pertinência no conjunto Área total.	84
Figura 32. Representação gráfica dos valores de pertinência no conjunto No Aptos.	84
Figura 33. Representação gráfica dos valores de pertinência no conjunto Localização.	85
Figura 34. Representação gráfica dos valores de pertinência no conjunto Qualidade.	85
Figura 35. Diagrama de caso de uso das ações do usuário.	90
Figura 36. Diagrama de caso de uso das operações do sistema.	91
Figura 37. Diagrama de atividades da <i>shell</i> ÍCARO.	93
Figura 38. Definição dos valores das entradas por meio da função trapezoidal.	94
Figura 39. Definição dos valores de saída por meio da função trapezoidal.	95
Figura 40. Definição das regras de inferência.	96
Figura 41. Arquitetura da rede <i>neurofuzzy</i> da <i>shell</i> ÍCARO.	98
Figura 42. Tela principal da <i>shell</i> ÍCARO.	99
Figura 43. Representação do modelo <i>fuzzy</i> por meio da função trapezoidal.	102
Figura 44. Inserção de dados para treinamento na <i>shell</i> ÍCARO.	102
Figura 45. Relatório de treinamento da <i>shell</i> ÍCARO.	103

Figura 46. Resultados dos testes realizados na <i>shell</i> ÍCARO	104
Figura 47. Resultados dos testes realizados na <i>shell</i> ÍCARO	104

LISTA DE SIGLAS

IA	Inteligência Artificial
IAC	Inteligência Artificial Conexionista
IAE	Inteligência Artificial Evolucionária
IAS	Inteligência Artificial Simbólica
RNA	Redes Neurais Artificiais
RNF	Redes <i>Neurofuzzy</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO GERAL	15
1.2	OBJETIVOS ESPECÍFICOS.....	16
1.3	JUSTIFICATIVA.....	16
1.4	ESTRUTURA DO TRABALHO.....	18
2	INTELIGÊNCIA ARTIFICIAL CONEXIONISTA.....	19
2.1	INTELIGÊNCIA ARTIFICIAL CONEXIONISTA E A BIOLOGIA	21
2.2	NEURÔNIO ARTIFICIAL.....	24
3	REDES NEURAIS ARTIFICIAIS	29
3.1	MODELO GERAL DE UM NEURÔNIO.....	31
3.2	PROCESSOS DE APRENDIZAGEM DA REDE NEURAL ARTIFICIAL.....	33
3.2.1	Aprendizagem Supervisionada.....	34
3.2.2	Aprendizagem Não-Supervisionada.....	35
3.3	PERCEPTRON	36
3.4	PERCEPTRON MULTICAMADAS	37
3.4.1	Algoritmo <i>Backpropagation</i>	39
4	LÓGICA <i>FUZZY</i>	41
4.1	CONJUNTOS <i>FUZZY</i>	43
4.1.1	Função de Pertinência	47
4.1.2	Variáveis Lingüísticas.....	51
4.2	OPERAÇÕES DOS CONJUNTOS <i>FUZZY</i>	52
4.3	ARQUITETURA DOS SISTEMAS <i>FUZZY</i>	56
4.3.1	<i>Fuzzyficação</i>	57
4.3.2	Inferência.....	57
4.3.3	<i>Defuzzyficação</i>	59
5	SISTEMAS DE ARQUITETURA HÍBRIDA.....	61
5.1	SISTEMA HÍBRIDO SEQUENCIAL.....	61
5.2	SISTEMA HÍBRIDO AUXILIAR.....	62

5.3 SISTEMA HÍBRIDO INCORPORADO	63
5.4 REDES <i>NEUROFUZZY</i>	64
5.4.1 Redes <i>Neurofuzzy</i> Híbridas	65
5.4.2 Redes <i>Neurofuzzy</i> Cooperativas.....	66
5.5 ALGUNS EXEMPLOS DE FERRAMENTAS <i>NEUROFUZZY</i>	68
6 ALGUNS EXEMPLOS DA UTILIZAÇÃO DAS REDES <i>NEUROFUZZY</i>	70
6.1 REDES NEURAIS <i>FUZZY</i> APLICADAS NA IDENTIFICAÇÃO E CONTROLE DE SISTEMAS	70
6.2 UMA INVESTIGAÇÃO DAS REDES <i>NEURO-FUZZY</i> APLICADAS À MINERAÇÃO DE DADOS	70
6.3 SISTEMAS INTELIGENTES HÍBRIDOS: COMO COMBINAR REDES NEURAIS ARTIFICIAIS, LÓGICA NEBULOSA E ALGORITMOS GENÉTICOS	71
6.4 UM MODELO NEURODIFUSO DE MULTIAGENTES PARA O APOIO AO ENSINO DE DISCIPLINAS DE CIÊNCIAS EXATAS.....	71
6.5 ESTUDO DA INFLUÊNCIA NA ACESSIBILIDADE NO VALOR DE LOTES ATRAVÉS DO USO DE REDES NEURAIS	72
7 SISTEMA <i>NEUROFUZZY</i> ÍCARO.....	73
7.1 AVALIAÇÃO DE IMÓVEIS	74
7.1.1 Método Comparativo de Dados de Mercado	75
7.1.2 Estatística Inferencial	76
7.2 METODOLOGIA	77
7.2.1 Levantamento bibliográfico	78
7.2.2 Aquisição do conhecimento sobre a avaliação de imóveis	78
7.2.3 Demonstração Matemática da Modelagem <i>Fuzzy</i> da Rede Neural.....	79
7.2.4 Realização da Modelagem do Protótipo da <i>Shell Ícaro</i>	88
7.2.5 Implementação do protótipo da <i>shell neurofuzzy</i>	93
7.2.5.1 Parte <i>Fuzzy</i>	94
7.2.5.2 Parte Neural.....	96
7.2.6 Treinamento da rede <i>neurofuzzy</i>	99
7.2.7 Realização de testes.....	100
7.3 RESULTADOS OBTIDOS	101

CONCLUSÃO	105
REFERÊNCIAS	107
APÊNDICE A – CONTRATO DE AUTORIZAÇÃO DE USO.....	110
APÊNDICE B – BASE DE REGRAS	111

1 INTRODUÇÃO

As Redes Neurais Artificiais (RNA) são sistemas paralelos distribuídos capazes de solucionar problemas complexos, cujo processamento é inspirado no funcionamento do cérebro humano. Uma RNA é composta por unidades de processamento simples, utilizadas para computar funções matemáticas, dispostas em uma ou mais camadas e interligadas. Sua principal característica é a capacidade de aprendizado, onde o conhecimento é adquirido por meio de treinamento (HAYKIN, 2001).

Apesar desta característica, não é possível controlar o processo de treinamento da rede, tornando difícil o acesso ao conhecimento adquirido. Desta forma, a solução do problema alcançado torna-se desconhecida e, muitas vezes, a rede pode gerar resultados inconsistentes e até mesmo incompreensíveis.

Com o objetivo de resolver este tipo de problema é possível que seja realizada uma incorporação da lógica fuzzy as RNA, para representação e processamento do conhecimento.

A lógica *fuzzy* é uma teoria que permite formalizar matematicamente a expressões da linguagem natural, por meio da qual é possível realizar operações com palavras onde os conjuntos *fuzzy* são os seus valores. Estes valores são obtidos por uma função e representam a imprecisão por meio de um grau de pertinência. Desta forma, a imprecisão referente a uma afirmação pode ser expressa por um valor que exprime a possibilidade desta ser correta (AZEVEDO; BRASIL; OLIVEIRA, 2000).

Embora a lógica *fuzzy* seja eficiente na representação do conhecimento, não é possível utilizar uma quantidade muito grande de dados, diferente das RNA, que possuem um alto poder computacional devido o seu processamento paralelo distribuído. Desta forma, dependendo do tipo de problema a ser resolvido, se faz necessária uma união destas duas técnicas.

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 apresentada nesta pesquisa realizou-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; pesquisa 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.

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 desenvolvimento de uma rede *neurofuzzy*, podendo-se assim proporcionar uma alternativa para auxílio a avaliação de imóveis.

1.1 OBJETIVO GERAL

Desenvolver o protótipo de um sistema *neurofuzzy* para avaliação de imóveis da tipologia apartamento no município de Criciúma.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa são:

- a) compreender as redes *neurofuzzy*;
- b) utilizar a lógica *fuzzy* para o tratamento da incerteza por imprecisão em redes neurais;
- c) demonstrar matematicamente a modelagem *fuzzy* da rede neural;
- d) desenvolver o protótipo de um sistema de arquitetura híbrida;
- e) proporcionar uma alternativa para a avaliação de imóveis por meio da utilização de redes *neurofuzzy*.

1.3 JUSTIFICATIVA

As redes neurais artificiais e a lógica *fuzzy* podem ser combinadas para formar sistemas de arquitetura híbrida, os quais absorvem características de ambas as técnicas provendo dois modos complementares de modelagem do conhecimento humano (AZEVEDO; BRASIL; OLIVEIRA, 2000).

Trabalhos realizados na avaliação de imóveis com a utilização de redes neurais artificiais apontam melhores resultados em relação à estatística inferencial demonstrando que a utilização do paradigma conexionista da inteligência artificial proporciona um bom desempenho neste tipo de atividade (GUEDES, 1999); (PELLI NETO; ZÁRATE, 2003); (PELLI NETO; BRAGA, 2004); (PELLI NETO; BRAGA, 2005).

A avaliação de imóveis para efeitos de cálculo dos impostos, como o Imposto Predial e Territorial Urbano (IPTU) e Imposto de Transmissão de Bens Imóveis (ITBI), visa promover a justiça fiscal nos valores atribuídos aos imóveis e a atualização das plantas de valores do município (ZANCAN, 1996).

As avaliações de imóveis são realizadas por tipologia: residencial (apartamentos e casas), comercial (salas e lojas), terrenos e industrial (galpões), sendo que nesta pesquisa escolheu-se a residencial de apartamentos devido ao número expressivo dos dados de ofertas e transações disponibilizados pelas imobiliárias e construtoras do município de Criciúma.

Neste tipo de avaliação, as técnicas atualmente utilizadas para indução do comportamento do mercado possuem um alto grau de incerteza e subjetividade, o que serve como incentivo na busca de técnicas que otimizem os resultados.

Desta forma, a Norma Brasileira de Avaliações de Imóveis (NBR 14653-2:2004) aconselha a utilização de ferramentas para indução do comportamento do mercado, tais como as redes neurais artificiais, as quais poderão ser aplicadas desde que devidamente justificadas do ponto de vista teórico e prático, com inclusão de validação quando pertinente.

Assim, esta pesquisa compreendeu o desenvolvimento do protótipo de uma *shell neurofuzzy* utilizando-se de duas técnicas da inteligência artificial como uma alternativa que pode se mostrar válida para minimizar o grau de imprecisão e subjetividade presente na avaliação de imóveis, podendo ser aplicada também na solução de outros problemas.

1.4 ESTRUTURA DO TRABALHO

Este trabalho é composto por 7 capítulos. No capítulo 1 encontra-se uma breve explanação do tema proposto, bem como os objetivos e justificativa para realização desta pesquisa.

No capítulo 2 são abordados os conceitos referentes à inteligência artificial conexionista com uma descrição de seus fundamentos biológicos e um breve histórico.

O capítulo 3 descreve os conceitos e funcionalidades das redes neurais artificiais, características e método de aprendizado.

Os fundamentos da lógica *fuzzy* são apresentados no capítulo 4 por meio da definição das funções de pertinência, variáveis lingüísticas, operações dos conjuntos e arquitetura de funcionamento. Enquanto no capítulo 5 são descritos os modelos de sistemas de arquitetura híbrida e fundamentos das redes *neurofuzzy*.

No capítulo 6 são apresentados alguns trabalhos utilizando o modelo de arquitetura híbrida *neurofuzzy*, além da utilização das redes neurais artificiais na área da avaliação de imóveis.

O capítulo 7 descreve o trabalho desenvolvido, os passos do processo de desenvolvimento desta pesquisa, bem como a exposição dos resultados obtidos.

E por fim, tem-se a conclusão, onde encontram-se também algumas sugestões para trabalhos futuros.

2 INTELIGÊNCIA ARTIFICIAL CONEXIONISTA

A Inteligência Artificial (IA) é uma especialidade da Ciência da Computação com fundamentos multidisciplinares inspirados na filosofia, matemática, economia, neurociências, psicologia, engenharia de computadores, física, matemática, lógica e lingüística (RUSSELL; NORVIG, 2004).

O estudo da IA tem como principal objetivo a criação de modelos e teorias referentes à capacidade cognitiva dos seres vivos, bem como a implementação de sistemas computacionais baseados nestes modelos. Os resultados obtidos pela IA não objetivam ser aprovados por meio da comparação com comportamentos cognitivos ou psíquicos mas sim, comprovar o desempenho de técnicas computacionais em sistemas capazes de executar tarefas por meio de ações inteligentes (RICH; KNIGHT, 1994).

Um sistema de IA deve ser capaz de armazenar e adquirir conhecimentos para resolução de problemas por meio da experiência. Além disso, é de fundamental importância em um sistema de IA a capacidade de representação por linguagens de estrutura simbólica¹, habilidade de raciocínio na resolução de problemas e a aquisição de modelos de aprendizagem (HAYKIN, 2001).

A modelagem de sistemas na IA segue três linhas principais de pesquisa: a *Inteligência Artificial Simbólica* (IAS), a *Inteligência Artificial Evolucionária* (IAE) e a *Inteligência Artificial Conexionista* (IAC). A IAS trata os processos cognitivos por meio da manipulação de símbolos, para produzir sistemas inteligentes que solucionem problemas de domínio específico. O paradigma da IAE é uma analogia à seleção natural devido ao fato de

¹ Linguagens de estrutura simbólica representam conhecimento genérico sobre o domínio de um problema e conhecimento específico sobre a solução do problema por meio da utilização de termos familiares (HAYKIN, 2001, p. 17)

utilizar processos naturais como a reprodução, mutação, cruzamento e seleção natural atuando sobre uma população específica (BITTENCOURT, 2001); (RUSSELL; NORVIG, 2004).

O principal objetivo dos sistemas conexionistas é a simulação de comportamentos inteligentes baseando-se nas estruturas funcionais do cérebro humano, onde a capacidade de processamento das informações permite a generalização dos resultados e possui a habilidade de aprender por meio de exemplos reconhecendo instâncias similares. Na realização de tarefas cognitivas, os algoritmos e arquiteturas que implementam as técnicas de um sistema conexionista são treinados ou condicionados por meio de um processo de aprendizagem, diferenciando-se dos sistemas encontrados em computadores convencionais, onde seu desenvolvimento é realizado por programação (HAYKIN, 2001); (LUGER, 2004).

O processo de aprendizagem no sistema conexionista ocorre por meio da inserção de uma coleção de dados em uma rede para que ela possa extrair as características necessárias para representação da informação requerida e encontrar a solução do problema (BRAGA; CARVALHO; LUDEMIR, 2000).

A rede de um sistema conexionista, chamada de *rede neural artificial*, é constituída por uma série de nodos interconectados conhecidos como *neurônios artificiais*, cujas funcionalidades são inspiradas nos neurônios biológicos.

A abordagem conexionista da IA tem obtido bons resultados em diversas áreas como reconhecimento de padrões, controle e identificação de processos industriais de dinâmica complexa, previsões financeiras, controle de movimento na área de robótica, processamento de sinais, diagnóstico de falhas, entre outros (BARRETO, 2001); (BITTENCOURT, 2001).

2.1 INTELIGÊNCIA ARTIFICIAL CONEXIONISTA E A BIOLOGIA

Conforme Braga, Carvalho e Ludemir (2000, p. 5), “as redes neurais artificiais tentam reproduzir as funções das redes biológicas, buscando implementar seu comportamento básico e sua dinâmica”.

O sistema nervoso humano pode ser definido como um sistema de três estágios onde o centro do sistema é o cérebro, composto pela rede neural, que recebe estímulos do meio externo por meio de receptores que os transformam em impulsos elétricos e após processá-los, os envia aos atuadores, que convertem os impulsos elétricos em respostas e como saída do sistema. Frequentemente, o percurso da informação por estes estágios retorna de um para outro a fim de realizar a *realimentação* do sistema, conforme demonstrado na Figura 1 (HAYKIN, 2001).

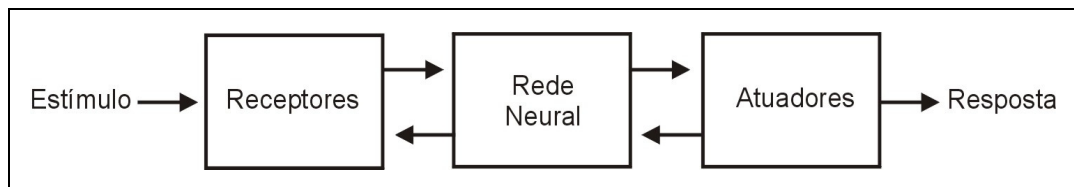


Figura 1. Representação dos estágios de funcionamento do sistema nervoso
Fonte: HAYKIN, S. (2001)

O processamento das informações no sistema nervoso é realizado por uma rede de milhares de neurônios interconectados que se comunicam continuamente e em paralelo. Estima-se que o cérebro humano contenha em torno de 10 bilhões de neurônios e 60 bilhões de conexões em sua rede neural, onde cada neurônio, ao receber impulsos nervosos processa as informações e transmite a outros neurônios da rede (BRAGA; CARVALHO; LUDEMIR, 2000); (HAYKIN, 2001).

O neurônio biológico é constituído por três partes com funções específicas e complementares (KOVÁCS, 1996); (BRAGA; CARVALHO; LUDEMIR, 2000):

- a) corpo celular: ou *soma*, que contém o núcleo da célula sendo responsável pelo processamento das informações e geração de novos impulsos que são transmitidos aos outros neurônios;
- b) dendritos: têm a função de receber impulsos nervosos de outros neurônios por meio de *sinapses*² e conduzi-los até o corpo celular;
- c) axônio: é responsável pelo recebimento e transmissão dos impulsos gerados pelo corpo celular para os outros neurônios.

A transmissão das informações de um neurônio para outro ocorre por meio de um ponto de contato entre eles definido como *sinapse*, cujo objetivo é controlar suas iterações (HAYKIN, 2001).

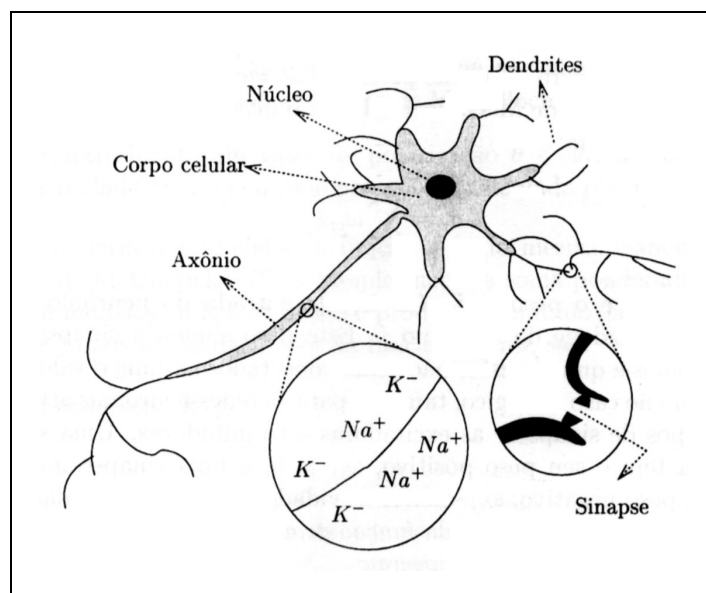


Figura 2. Neurônio biológico
Fonte: BITTENCOURT, G. (2001)

² Ponto de ligação entre os neurônios (HAYKIN, 2001, p 32).

As sinapses são controladas por substâncias chamadas *neurotransmissores*, formadas por impulsos elétricos e reações químicas. Um impulso elétrico se propaga por um dendrito ou axônio devido alteração da concentração de íons de potássio (K^+) e sódio (Na^+) como demonstradas na Figura 2 (BITTENCOURT, 2001).

Os neurotransmissores recebidos de todos os neurônios conectados são que definem a geração ou não de um impulso elétrico, ou seja, são as entradas inibitórias e excitatórias recebidas dos outros neurônios que determinam o percentual de disparo de saída. Após gerar um impulso, o axônio não pode ser novamente estimulado, passando por um período de refração onde retorna ao estado de repouso para que seja gerado um novo impulso (BRAGA; CARVALHO; LUDEMIR, 2000).

A estrutura e funcionamento apresentados são uma generalização referente ao neurônio biológico. O cérebro humano possui em sua composição diferentes estruturas de redes neurais e neurônios com uma grande variedade de formas e tamanhos (HAYKIN, 2001).

Conforme Braga, Carvalho e Ludemir (2000), embora alguns conceitos referentes à estrutura fisiológica das redes cerebrais biológicas possam ser bem definidos, seu funcionamento e a forma pela qual são realizadas suas funções, não foram ainda totalmente desvendadas.

Desta forma, o desenvolvimento de técnicas inspiradas em modelos do funcionamento das redes neurais biológicas e sua principal unidade de processamento, o neurônio, dependem da evolução científica nesta área de conhecimento.

Os conceitos descobertos inicialmente sobre as estruturas dos neurônios biológicos e do sistema nervoso inspiraram o desenvolvimento do primeiro modelo matemático simples dos neurônios em 1943, pelo neurofisiologista e filósofo americano

Warren McCulloch e o matemático Walter Pitts, baseados na neurofisiologia e lógica matemática (HAYKIN, 2001); (RUSSELL; NORVIG, 2004).

2.2 NEURÔNIO ARTIFICIAL

O funcionamento dos neurônios biológicos foi interpretado por McCulloch como sendo um circuito binário onde as entradas do neurônio possuem valores de inibição (0) ou de excitação(1). Uma *função de ativação*³ multiplica cada valor de entrada por seu peso correspondente e calcula o somatório dos resultados. A saída do neurônio é também binária, onde: se o resultado da função de ativação é maior ou igual à função de limiar θ que possui valor zero ($\theta=0$), retorna 1; caso contrário, retorna 0 (LUGER, 2004); (BARONE, 2003).

A Figura 3 mostra a estrutura do primeiro modelo de neurônio artificial desenvolvido por McCulloch e Pitts:

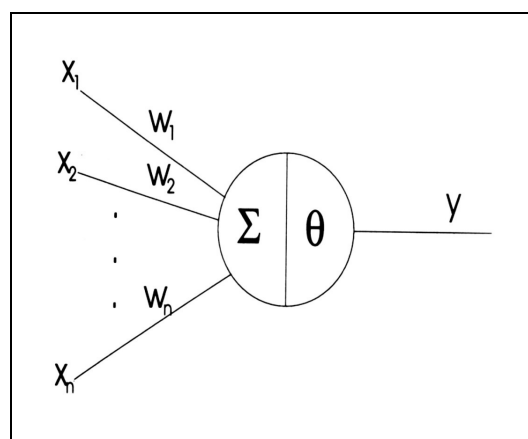


Figura 3. Neurônio Artificial de McCulloch e Pitts
Fonte: BRAGA, A. P.; CARVALHO, A.C.P.L.F.; LUDEMIR, T.B. (2000).

³ Restringe a amplitude do sinal de saída de um neurônio limitando seu intervalo. Devido a este fato é chamada também de *função restritiva* (HAYKIN, 2001, p 37).

O funcionamento do neurônio artificial de McCulloch-Pitts pode ser demonstrado na Figura 4 onde este modelo é utilizado para calcular as funções lógicas E e OU, onde, x e y são os valores lógicos com pesos de valor +1; e l é chamado de *bias*⁴ e possui valor constante de +1:

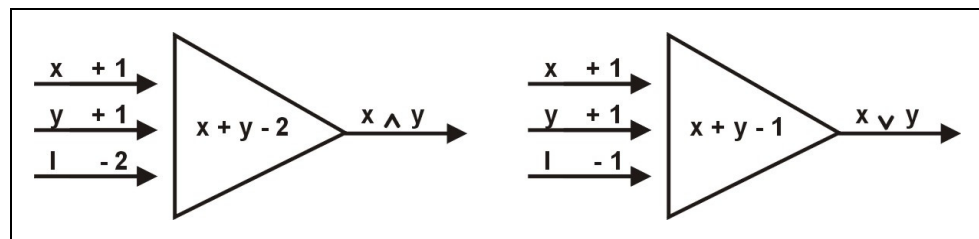


Figura 4. Neurônios de McCulloch-Pitts para calcular as funções lógicas E e OU.
Fonte: LUGER, G. F. (2004)

Os resultados gerados conforme as possíveis entradas para os valores lógicos são apresentados na Figura 5:

Função E				Função OU			
X	Y	X+Y-2	Saída	X	Y	X+Y-1	Saída
0	0	0	1	0	0	1	-1
0	1	-1	-1	0	1	0	1
1	0	-1	-1	1	0	0	1
1	1	-2	-1	1	1	-1	1

Figura 5. Resultados do modelo de McCulloch e Pitts para as funções lógicas E e OU

⁴ Também conhecido como *peso de desvio*, é utilizado para definição do limite real do neurônio (RUSSELL; NORVIG, 2004, p 714).

Com este modelo inicial de neurônio McCulloch e Pitts comprovaram que por meio de argumentos lógicos uma rede neural é capaz de calcular algumas funções computáveis. Isso se deve ao fato de poder representar funções lógicas como as dos operadores E, OU e NÃO fornecendo, portanto, um modelo computacional (BITTENCOURT, 2001); (HAYKIN, 2001); (BARONE, 2003).

Apesar de apresentar eficiência computacional, este modelo de neurônio demonstrou algumas restrições se comparado às redes neurais biológicas.

O modelo de McCulloch-Pitts considera que as camadas da rede disparam de forma síncrona, sendo que as redes biológicas não apresentam mecanismos de sincronismo para tratar a avaliação de suas entradas. As redes biológicas utilizam uma grande quantidade de camadas interconectadas onde os valores de entrada podem ser recebidos de neurônios de outras camadas, influenciando as ativações seguintes, sendo que no modelo matemático proposto por McCulloch-Pitts é considerada apenas uma camada de rede de forma que a implementação só é possível com suas entradas linearmente separadas⁵ (BRAGA; CARVALHO; LUDEMIR, 2000).

Este modelo neural trata dos valores na função matemática de forma binária devido ao fato de McCulloch e Pitts assumirem que o neurônio biológico segue a lei do “tudo ou nada” (HAYKIN, 2001).

O modelo matemático do neurônio artificial de McCulloch-Pitts foi introduzido pela publicação do artigo *A Logical Calculus of the Ideas Immanent in Nervous Activity* junto de uma discussão referente a redes com modelos computacionais para representação

⁵ Um conjunto de entradas é dito *linearmente separável* se, quando demonstrado por meio de um sistema cartesiano, pode ser separado por uma reta (RUSSELL; NORVIG, 2004, p 717).

lógica do comportamento e memória, sem tratar dos seus mecanismos de aprendizado (BRAGA; CARVALHO; LUDEMIR, 2000).

Baseados neste primeiro modelo de neurônio artificial, diversos outros estudos foram realizados para o surgimento das redes neurais artificiais.

Os mecanismos de aprendizado neurais biológicos e artificiais tornaram-se objeto de estudo no ano de 1949 no livro de Hebb intitulado *The Organization of Behavior*, sendo proposta a existência de uma modificação sináptica das conectividades das redes, conforme o aprendizado adquirido e a criação de agrupamentos neurais derivados destas modificações. Nos anos seguintes, uma série de outras pesquisas originou técnicas e teorias referentes aos sistemas neurais biológicos e artificiais (HAYKIN, 2001); (RUSSELL; NORVIG, 2004).

No ano de 1969 foi publicado o livro *Perceptrons* por Minsky e Papert, que enfatizou as limitações referentes aos modelos de redes neurais propostos até aquela época por meio da análise de suas propriedades matemáticas, como por exemplo, não serem capazes de implementar o operador *ou-exclusivo*⁶.

Devido a esta publicação, entre outros fatores como a indisponibilidade de computadores mais robustos e paralelos; algoritmos de treinamento pouco eficientes; e a quantidade insuficiente de conhecimentos referentes à estrutura neural biológica, as pesquisas de conexionismo foram abandonadas, ressurgindo uma década depois devido à superação de grande parte das limitações existentes (BITTENCOURT, 2001).

As respostas para as limitações apontadas por Minsky e Papert começaram a surgir em meados da década de 80 com a popularização do algoritmo *backpropagation*,

⁶ Conectivo lógico que resulta em falso quando ambas as sentenças são verdadeiras (RUSSELL; NORVIG, 2004)

publicado em 1986 no livro *Parallel Distributed Processing*, por Rumelhart e McClelland (BARONE, 2003).

A partir dos anos seguintes foi realizado um grande número de pesquisas nesta área, permitindo que as RNA fossem aplicadas na solução de diversos tipos de problema e originando suas diversas técnicas.

Embora a estrutura de funcionamento das redes neurais artificiais tenha sido baseada no modelo biológico, devido aos avanços obtidos, as RNA diferem estruturalmente das redes neurais biológicas, contudo, possuem semelhanças como a capacidade de comunicação paralela e distribuída por meio de conexões sinápticas modularizadas (BRAGA; CARVALHO; LUDEMIR, 2000).

A seguir são apresentadas as características, modelos e funcionalidades das redes neurais artificiais.

3 REDES NEURAIS ARTIFICIAIS

Conforme Barone (2003, p. 115), as RNA “não operam seqüencialmente como na maioria dos computadores atuais, mas sim inspiradas no paralelismo do cérebro humano”.

As redes neurais artificiais⁷ são estruturas formadas por camadas de neurônios unidirecionais interconectados que calculam funções matemáticas, onde a informação percorre três tipos de neurônios (BITTENCOURT, 2001); (HAYKIN,2001):

- a) neurônios de entrada (u_k): seu funcionamento ocorre de maneira semelhante aos neurônios sensoriais biológicos sendo responsáveis pelo recebimento das informações do meio externo;
- b) neurônios internos: também chamados de *neurônios ocultos*, podem desenvolver atividades independentes das informações recebidas pelos neurônios de entrada, sendo matematicamente importantes para a resolução de determinados problemas;
- c) neurônios de saída (y_k): seu funcionamento ocorre de forma semelhante aos neurônios motores biológicos, produzindo a saída do sistema.

As estruturas das RNA são formadas por conjuntos de neurônios, sendo denominadas *elementos de processamento*, estes podem ser vistos como grafos rotulados onde cada um dos seus nodos é um neurônio artificial (AZEVEDO; BRASIL; OLIVEIRA, 2000); (BARONE, 2003).

⁷ Também conhecidas como Sistemas de Processamento Paralelo e Distribuído (BRAGA; CARVALHO; LUDEMIR, 2000, p. 1).

Mediante a formação destas estruturas é possível identificar alguns parâmetros de classificação referentes a arquitetura das redes neurais artificiais quanto ao número de camadas da rede, tipo de conexão e conectividade entre os nodos (BRAGA; CARVALHO; LUDEMIR, 2000); (BITTENCOURT, 2001); (HAYKIN, 2001):

- a) número de camadas da rede: as RNA são denominadas de *camada única* quando entre suas entradas e saídas existe apenas um neurônio; e de *camada múltipla* ou *multicamadas* quando entre suas camadas de entrada e saída tem-se mais de um nodo. Os nodos encontrados entre estas camadas constituem-se nas camadas *ocultas* da rede;
- b) tipo de conexão: as RNA são denominadas *acíclicas* ou *feedforward* quando as informações encontradas em qualquer camada i não são transmitidas desta para camadas com rótulo menor ou igual a i , ou seja, o fluxo de informações é transmitido em um único sentido na rede. Quando as informações transmitidas na rede por um nodo i podem ser recebidas como entrada por neurônios em camadas maiores ou iguais a i estas são chamadas de redes *cíclicas*, *feedback* ou *recorrentes*.
- c) conectividade: quando uma RNA recorrente possui todos os nodos de uma camada conectados aos da camada posterior, a rede é denominada *completamente conectada*. Uma rede na qual um nodo de uma camada qualquer não possui conexões sinápticas entre todos os outros da próxima camada é chamada de *fracamente* ou *parcialmente conectada*.

Descritas estas propriedades, apresenta-se a seguir a composição matemática geral dos neurônios artificiais.

3.1 MODELO GERAL DE UM NEURÔNIO

Matematicamente um neurônio artificial é composto por n sinais de entrada ($x_1, x_2, x_3, \dots, x_n$), que possuem pesos associados ($w_1, w_2, w_3, \dots, w_n$) e produzem um sinal de saída y , conforme ilustrado na Figura 6, onde (HAYKIN, 2001) (LUGER, 2004):

- a) sinais de entrada (x_i): representam os dendritos do neurônio biológico e recebem estes valores do meio externo ou de outros neurônios. Os intervalos dos valores recebidos podem ser de diversos tipos, sendo que na maioria das aplicações são utilizados valores dos conjuntos $[0,1]$, $[-1,1]$ ou números reais;
- b) conjunto de pesos (w_i): valores reais que constituem pesos dos sinais de entrada e determinam a força das conexões exercendo influência na ativação do neurônio. Este conjunto de pesos permite valores positivos e negativos que representam as sinapses excitatórias e inibitórias;
- c) junção aditiva (v): efetua o cálculo entre um somador (u) que realiza a soma ponderada dos sinais de entrada x_i conforme seus pesos de conexão w_i com o valor de *bias*⁸ (b), sendo representada pelo par de equações:

$$u = \sum_{i=1}^n x_i w_i \quad v = u + b$$

- d) nível ou função de ativação ($\varphi(\cdot)$): função que determina a saída do neurônio por meio do resultado da junção aditiva (v). Os intervalos

⁸ Parâmetro externo do neurônio com o objetivo de aplicar uma transformação afim à saída U (HAYKIN, 2001, p. 37).

utilizados para funções de ativação mais simples são [0,1] e [-1,1]. Os três tipos básicos de função de ativação são:

a) função de limiar:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases}$$

b) função linear por partes:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq +1/2 \\ v & \text{se } +1/2 > v > -1/2 \\ 0 & \text{se } v \leq -1/2 \end{cases}$$

c) função sigmóide:

$\varphi(v) = 1/(1+\exp(-av))$, onde a representa o parâmetro de inclinação da função.

e) senal de saída (y): representa a saída do neurônio conforme a determinação dada pelo nível de ativação, sendo:

$$y = \varphi(v)$$

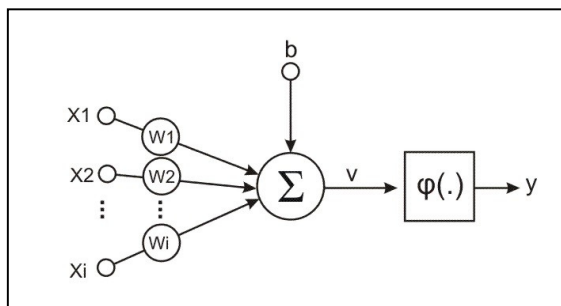


Figura 6. Modelo geral de um neurônio.
Fonte: adaptado de HAYKIN, S. (2001)

A estrutura apresentada na Figura 6, demonstra o funcionamento básico de cada neurônio que compõe uma rede neural artificial. Como foi dito anteriormente, a principal propriedade de uma rede neural artificial é a capacidade de aprender por meio de exemplos. A seguir são apresentados os modelos de aprendizagem utilizados no treinamento das redes neurais artificiais.

3.2 PROCESSOS DE APRENDIZAGEM DA REDE NEURAL ARTIFICIAL

O processo de aprendizagem de uma RNA permite que seu desempenho seja melhorado por meio de ajustes que consistem em iterações realizadas sobre seus pesos sinápticos e níveis de *bias* determinando a intensidade de conexões entre os neurônios. Desta forma, a RNA utiliza as informações relevantes na geração de padrões para criar uma representação para o problema (BRAGA; AZEVEDO; LUDEMIR, 2000); (HAYKIN, 2001).

A aprendizagem em uma RNA ocorre por meio de um conjunto de procedimentos e regras pré-definidas que adaptam seus parâmetros para a solução de um problema, formando uma função chamada de *algoritmo de aprendizagem* (HAYKIN, 2001).

Existem diversos tipos de algoritmos de aprendizagem que se diferem pela forma como o ajuste de seus pesos é realizado. A escolha do algoritmo de aprendizado depende diretamente da arquitetura de rede utilizada (AZEVEDO; BRASIL; OLIVEIRA, 2000); (BRAGA; AZEVEDO; LUDEMIR, 2000).

Conforme Russell e Norvig (2004), a arquitetura das RNA possuem duas categorias principais de estrutura de aprendizado:

- a) redes acíclicas: também chamadas de *rede de alimentação direta*, *feedforward* ou *rede alimentada adiante*. Neste tipo de arquitetura de rede as entradas são computadas até a saída sem modificações em seus pesos sinápticos;
- b) redes cíclicas: também chamadas de *redes recorrentes* ou *feedback*, diferenciam-se do modelo acíclico por efetuar um processo de *realimentação*. A realimentação da rede consiste na utilização das saídas computadas como entradas da rede, ou seja, as saídas geradas pela rede são utilizadas em um novo ciclo como entrada efetuando a atualização dos pesos sinápticos.

Conforme Haykin (2001), o modelo de estrutura utilizado em uma rede neural está diretamente ligado ao algoritmo de treinamento utilizado no processo de aprendizagem da rede. Após a escolha do modelo de arquitetura de RNA a ser implementado, é efetuada a escolha dos dados que serão utilizados neste processo de aprendizagem.

Os diversos algoritmos de aprendizagem desenvolvidos podem ser agrupados de acordo com duas abordagens principais em: *aprendizagem supervisionada* e *aprendizagem não-supervisionada*.

3.2.1 Aprendizagem Supervisionada

O aprendizado supervisionado consiste na utilização de um supervisor externo, com conhecimentos referentes ao ambiente, que forneça à rede pares de entradas e saídas que direcionam o processo de treinamento (BRAGA; AZEVEDO; LUDEMIR, 2000).

O treinamento da RNA é realizado de forma que, o supervisor forneça uma entrada e, por meio desta, a rede produza uma saída. Caso o sinal de saída resultado da rede não seja o resultado desejado, é gerado um sinal de erro para que sejam calculados os ajustes necessários nos pesos sinápticos da rede para que esta produza a saída desejada (AZEVEDO; BRASIL; OLIVEIRA, 2000).

O sinal de erro consiste na diferença entre o resultado desejado e real da rede. Desta forma, quando este erro é minimizado, a rede neural pode ser utilizada dispensando a utilização do supervisor (HAYKIN, 2001).

Este método de aprendizagem é o mais utilizado no treinamento de RNA, podendo ser implementado para que atue de duas formas (BRAGA; AZEVEDO; LUDEMIR, 2000):

- a) *off-line*: após a minimização do erro, o ajuste realizado sobre seus pesos permanece fixo. No caso de novos dados serem inseridos ao conjunto, deve ser realizado um novo treinamento incluindo os dados anteriores para que seja evitada a influência no processo realizado anteriormente;
- b) *on-line*: o conjunto de dados utilizado na rede para o ajuste dos pesos permanece em contínuo processo de adaptação mesmo após a minimização de erros e obtenção da solução mais adequada pela rede.

3.2.2 Aprendizagem Não-Supervisionada

Diferente do aprendizado supervisionado, este método não requer a utilização de um supervisor para o treinamento da rede, pois são fornecidos à rede somente os valores

de entrada, não existindo uma saída desejada, o que torna desnecessária a inserção de pares de entrada e saída como ocorre no aprendizado supervisionado (BRAGA; AZEVEDO; LUDEMIR, 2000).

Neste método de aprendizagem, o treinamento da rede consiste na organização dos valores de entrada inseridos para que sejam classificados por padrões em categorias. A saída fornecida pela rede apresenta uma indicação da classe à qual a entrada pertence, sendo que, em caso de inexistência de um padrão é gerada uma nova classe (AZEVEDO; BRASIL; OLIVEIRA, 2000).

O primeiro algoritmo de treinamento foi concebido no ano de 1958 por Frank Rosenblatt para uma rede de camada única chamada *perceptron*, utilizando uma forma simples de aprendizado supervisionado (LUGER, 2004).

3.3 PERCEPTRON

Conforme Haykin (2001) o perceptron é a estrutura mais simplificada de uma rede neural utilizada para classificação de padrões e necessita que suas classes sejam linearmente separáveis para funcionar adequadamente.

O modelo perceptron tem seus valores de entrada e níveis de ativação dados por -1 ou $+1$ e pesos de valores reais. Seu nível de ativação é dado da mesma forma que o modelo mais geral de um neurônio artificial, sendo realizado pela soma ponderada das entradas pelos seus respectivos pesos ($\sum x_i w_i$). Sua função de ativação y utiliza um valor de limiar θ onde (LUGER, 2004):

$$y = \begin{cases} 1 & \text{se } \sum x_i w_i \geq \theta \\ -1 & \text{se } \sum x_i w_i < \theta \end{cases}$$

O *perceptron* utiliza a forma mais simples de aprendizado supervisionado onde os pesos de suas entradas e o limite são modificados para minimização de erros por meio de um supervisor que fornece ao neurônio a saída correta (RICH; KNIGHT, 1994).

Este modelo básico apresentado utiliza uma rede de neurônios de uma única camada de processamento, sendo denominada *perceptron* de camada única. Esta rede de neurônios permite que sejam somente utilizados em estruturas cujos valores não são linearmente separáveis, ou seja, a restrição imposta pelos valores de entrada produz padrões de saída similares, impondo uma grande quantidade de limitações, dentre as quais pode-se mencionar a representação lógica *ou-exclusivo* (HAYKIN, 2001).

A dificuldade apresentada no uso de uma única camada de processamento no modelo *perceptron* na resolução de problemas como a representação do *ou-exclusivo* foi superada pelo uso de múltiplas camadas na estruturação destas redes, formalizando o que atualmente é conhecida como *perceptron multicamadas* (LUGER, 2004).

3.4 PERCEPTRON MULTICAMADAS

Uma rede *perceptron multicamadas* (MLP)⁹ é uma generalização do *perceptron* de camada única onde sua estrutura é constituída de k camadas, $k= 1,2,3...K$ onde estão dispostos n_k neurônios, $n_k=1,2,3...N_k$. As entradas são distribuídas por toda a rede pelo processo de propagação para frente (*feedforward*), camada por camada, até a geração da saída (HAYKIN, 2001).

⁹ Do inglês *MultiLayer Perceptron* (HAYKIN, 2001, p. 183).

Os neurônios da primeira camada da rede ($k=1$), denominada *camada de entrada*, recebem os valores de um *vetor de entradas* x de dimensão N_I de componentes x_n , $n=1,2,3,\dots,K_I$, tal que, $n_{1n}=x_n$. As saídas dos neurônios n_{1n} fornecem os valores aos neurônios n_{2n} da próxima camada e assim sucessivamente até a última camada da rede ($k=K$), denominada *camada de saída*, gerando a resposta do sistema. As camadas dispostas entre a de entrada e saída são denominadas *camadas ocultas*. Este modelo de rede está ilustrado na Figura 7 (KOVÁCS, 1996).

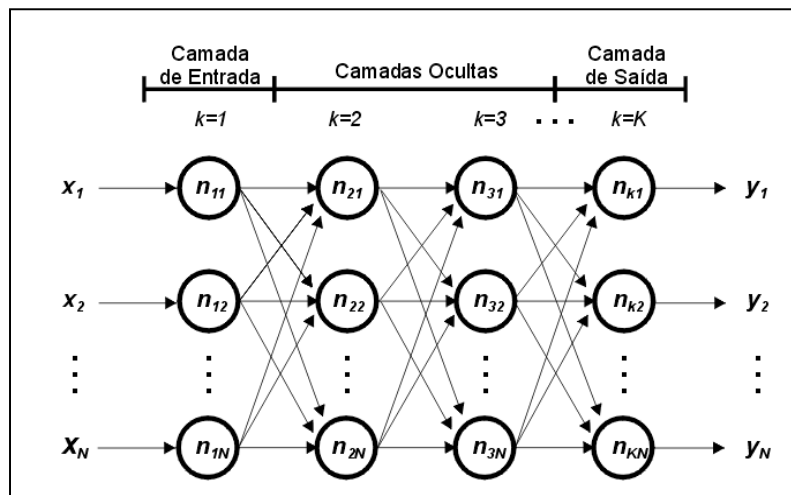


Figura 7. Exemplo de uma rede neural multicamadas.

O aprendizado em uma MLP ocorre por meio de uma retropropagação das saídas às camadas anteriores por um treinamento de forma supervisionada. Este método de aprendizado é efetuado por meio de um algoritmo conhecido popularmente como *algoritmo de retropropagação* (também chamado de *backpropagation*), criado por Rumelhart, Hilton e Williams para resolução de limitações em redes complexas (KOVÁCS, 1996); (HAYKIN, 2001).

3.4.1 Algoritmo *Backpropagation*

O algoritmo *backpropagation* efetua o aprendizado da rede por meio de dois passos, a propagação (*feedforward*) e a retropropagação (*backpropagation*). Inicialmente, os sinais de entrada são propagados até a última camada, sem que sejam modificados os valores dos pesos, gerando uma resposta aleatória da rede. A saída obtida é comparada com a desejada fornecendo um sinal de erro que é calculado por meio da seguinte equação: $e_j(\mathbf{n}) = d_j(\mathbf{n}) - y_j(\mathbf{n})$, onde j representa o neurônio de saída, d_j o valor desejado e y_j o valor resultante da rede (HAYKIN, 2001); (RUSSEL; NORVIG, 2004).

O sinal de erro é utilizado no ajuste dos pesos sinápticos dos neurônios de cada camada tal que: $\Delta w_j = \eta \cdot (e_j(\mathbf{n}) \cdot y_j(\mathbf{n}))$, onde Δw representa o *delta correção* e η é uma constante positiva que determina a *taxa de aprendizagem*. A taxa de aprendizagem representa uma constante de peso fixo que define um fator de escala para atualização dos pesos sinápticos, ou seja, o quanto o peso será atualizado. Desta forma, a modificação nos pesos é realizada tal que: $w_j = w_j + \Delta w_j$, de forma que ao peso atual w_j seja adicionado o valor de Δw_j , caso este seja positivo; ou subtraído, caso contrário (RICH; KNIGHT, 1994).

Conforme Braga, Carvalho e Ludemir (2000), embora a utilização das redes neurais artificiais obtenha ótimos resultados em diversas aplicações, existem ainda limitações e deficiências que impossibilitam que o uso exclusivo de suas técnicas seja suficiente na busca de soluções para uma grande quantidade de problemas.

Uma das limitações apresentada pelas redes neurais artificiais é a simulação do raciocínio impreciso encontrado nos seres humanos. A representação deste tipo de

raciocínio em IA é possível por meio da utilização de uma técnica conhecida como lógica *fuzzy* (BITTENCOURT, 2001).

A demonstração matemática dos cálculos realizados pela rede é apresentada no Capítulo 7, bem como as demais etapas do trabalho desenvolvido. A seguir são abordados os conceitos, estruturas e funcionalidades da lógica *fuzzy* no tratamento da incerteza por imprecisão.

4 LÓGICA FUZZY

A lógica *fuzzy*, tem como objetivo modelar os modos imprecisos de raciocínio onde informações descritas em linguagem natural são convertidas para um formato numérico. Antes que sejam introduzidos os conceitos referentes à lógica *fuzzy* é necessária uma breve abordagem sobre a teoria dos conjuntos, também conhecidos como conjuntos *crisp*.

A tradicional teoria dos conjuntos classifica uma sentença como pertencente a uma determinada categoria quando seus valores encontram-se dentro dos limites estabelecidos, permitindo que assumam apenas os estados verdadeiro e falso.

A transição dos valores de um conjunto para outro na lógica tradicional ocorre de forma instantânea, onde cada elemento pertence inteiramente ou não pertence a um conjunto, necessitando que possua propriedades bem definidas e limites claros, conforme demonstrado na Figura 8 (WANG, 1997).

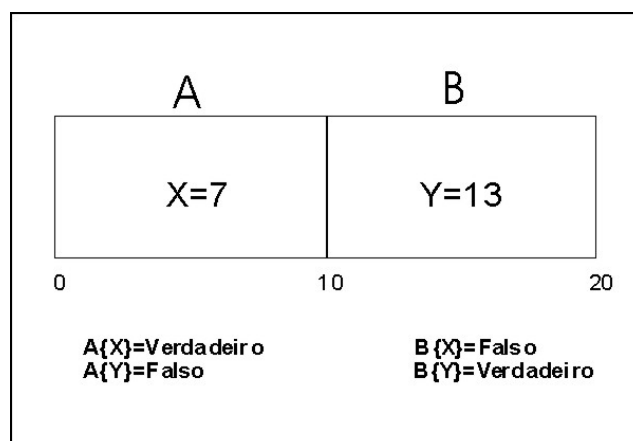


Figura 8. Classificação de valores em um conjunto *crisp*.
Fonte: Adaptado de WANG, L. (1997)

Conforme Bittencourt (2001) a lógica clássica possui como principal característica o *axioma do terceiro excluído* que indica a inexistência de uma terceira alternativa lógica para um valor além de verdadeiro (1) ou falso (0).

A precisão com a qual são classificadas as informações na teoria clássica dos conjuntos, embora apreciada na matemática, difere da forma pela qual são apresentadas no mundo real ou concebidas pelo raciocínio humano, onde algumas informações não são nem absolutamente verdadeiras nem falsas. Dependendo do tipo de aplicação a que se refere, a teoria clássica dos conjuntos fornece uma classificação imprecisa quanto a veracidade de uma determinada sentença (BITTENCOURT, 2001).

Conforme Barreto (2001, p. 58), “quanto mais se aprende sobre a cognição, mais se percebe que a habilidade em manipular conceitos mal definidos é uma das maiores qualidades e não defeito, e que constitui a chave da diferença entre a inteligência humana e da máquina”.

Como exemplo, para esclarecer melhor esta afirmação, toma-se uma distância, onde um objeto pode ser considerado perto, muito perto, mais ou menos perto, longe, razoavelmente longe, entre outras definições. Informações incertas ou vagas, como a descrita no exemplo, podem ser tratadas pela teoria dos conjuntos *fuzzy*.

O tratamento da incerteza era obtido de maneira insatisfatória em termos de imprecisão, por meio de modelos probabilísticos até o ano de 1965, quando Lofti Zadeh publicou o artigo *Fuzzy Sets* para descrever conjuntos cujos critérios de pertinência são imprecisos, formalizando a lógica *fuzzy*¹⁰ ou lógica difusa. Nesta abordagem a lógica é tratada por meio de conjuntos com definições nebulosas. A principal argumentação a respeito do uso da lógica *fuzzy* enfatiza que a teoria das probabilidades é adequada para

¹⁰ Termo inglês que significa “nebuloso” (RABUSKE, 1995, p. 182).

medir a aleatoriedade das informações, sendo inapropriada para representar a sua imprecisão (RABUSKE, 1995); (BARRETO, 2001); (LUGER, 2004).

A lógica *fuzzy* trata a imprecisão dos valores encontrados gerando um grau de possibilidade, ou seja, grau de pertinência de um valor pertencer a um conjunto. Assim é possível obter uma conclusão por meio de uma coleção de premissas imprecisas. Desta forma, a lógica *fuzzy* busca formar um modelo aproximado do raciocínio humano na tentativa de tomar decisões racionais em situações de imprecisão onde dados imprecisos são representados por meio de graus de pertinência (KLIR; YUAN, 1995).

O grau de pertinência de um elemento pode ser definido como a medida do quanto este pode pertencer a um determinado conjunto, ou seja, permite que um elemento pertença parcialmente a dois conjuntos *fuzzy*, com maior ou menor intensidade. O grau de pertinência é representado pelo intervalo real entre 0 e 1, tornando a sentença parcialmente verdadeira e falsa (LUGER, 2004).

A seguir, são abordados os principais conceitos referentes aos conjuntos *fuzzy*.

4.1 CONJUNTOS FUZZY

Conforme Wang (1997) a composição dos conjuntos *crisp* é definida tal que, U representa o *universo de discurso*, ou universo do conjunto, onde são encontrados todos os elementos de interesse em cada contexto particular. Considerando-se, por exemplo, um conjunto de todos os apartamentos no município de Criciúma; este é o universo do conjunto U . O universo de discurso (U) permite que a organização de seus componentes seja definida pelo método de pertinência. Este método utiliza uma função binária ou de

pertinência (também chamada função característica, discriminativa ou de indicação) representada por $\mu_A(x): x \rightarrow [0,1]$, onde:

$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases}$$

Esta função pode ser exemplificada pela seguinte função: todos os apartamentos em Criciúma que possuem 3 quartos.

$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A, \text{ ou seja, } x \text{ possui 3 quartos} \\ 0 & \text{se } x \notin A, \text{ ou seja, } x \text{ não possui 3 quartos} \end{cases}$$

Conforme o exemplo anterior, é possível definir o quanto um elemento pertence a um determinado conjunto, podendo assumir inclusive valores intermediários entre 0 e 1.

Outro exemplo seria a classificação de um imóvel de acordo com sua localização, onde seu valor é influenciado pela distância que se encontra de um pólo de valorização. Os conjuntos a serem considerados são: perto, representado por A , para propriedades de até 100 metros de distância; e longe, representado por B , para valores a partir de 100 metros. Dado um imóvel x localizado a 110 metros do pólo referido, de acordo com a teoria tradicional dos conjuntos, pode ser ilustrado conforme demonstrado na Figura 9:

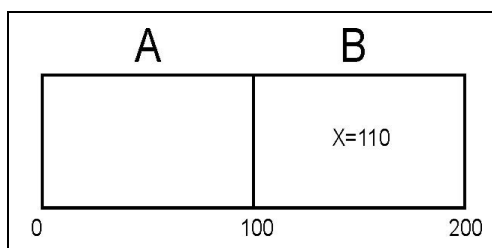


Figura 9. Conjuntos *crisp* A e B em um universo de discurso U .
Fonte: Adaptado de WANG, L. (1997)

Conforme ilustrado, o elemento x pertence a um conjunto, tal que $A = \{ x \in U \mid x \leq 100 \}$ e $B = \{ x \in U \mid x > 100 \}$, pertencendo inteiramente ou não, onde o resultado obtido é $\{0, 1\}$.

Embora estes resultados sejam válidos na teoria tradicional dos conjuntos, de acordo com o raciocínio humano, 100 e 110 metros são valores muito próximos para pertencerem a conjuntos tão distintos, gerando limites imprecisos e influenciando no resultado para formação de um valor. A imprecisão de limites encontrada requer a utilização da teoria dos conjuntos *fuzzy* em um universo U para encontrar uma generalização em um conjunto clássico.

Utilizando o exemplo anterior é possível definir o quanto x pertence aos conjuntos A e B por meio da gradação de pertinência de x ao conjunto B , sendo o conjunto de valores de domínio para longe $\{0, \dots, 200\}$ e os graus de pertinência $[0, 1]$ como demonstrado na Figura 10:

Distância	Longe(B)	Perto(A)
0	0.00	1.00
10	0.05	0.95
20	0.10	0.90
30	0.15	0.85
40	0.20	0.80
50	0.25	0.75
60	0.30	0.70
70	0.35	0.65
80	0.40	0.60
90	0.45	0.55
100	0.50	0.50
110	0.55	0.45
120	0.60	0.40
130	0.65	0.35
140	0.70	0.30
150	0.75	0.25
160	0.80	0.20
170	0.85	0.15
180	0.90	0.10
190	0.95	0.05
200	1.00	0.00

Figura 10. Relações de pertinência de x aos conjuntos A e B .

A obtenção dos valores de pertinência em conjuntos *fuzzy*, necessita que a imprecisão seja medida de uma maneira quantitativa por meio de uma *função de pertinência* $\mu_A(x)$, no intervalo 0 e 1, onde um conjunto A em U pode ser descrito como um par ordenado de um elemento x e seu valor de pertinência, sendo (WANG, 1997); (AZEVEDO; BRASIL; OLIVEIRA, 2000); (LUGER, 2004):

$$A = \{ (x, \mu_A(x)) \mid x \in U \}$$

Conforme esta abordagem, o conjunto *fuzzy* B , por exemplo, com intervalos de pertinência como mostrados na Figura 11, pode ser representado graficamente.

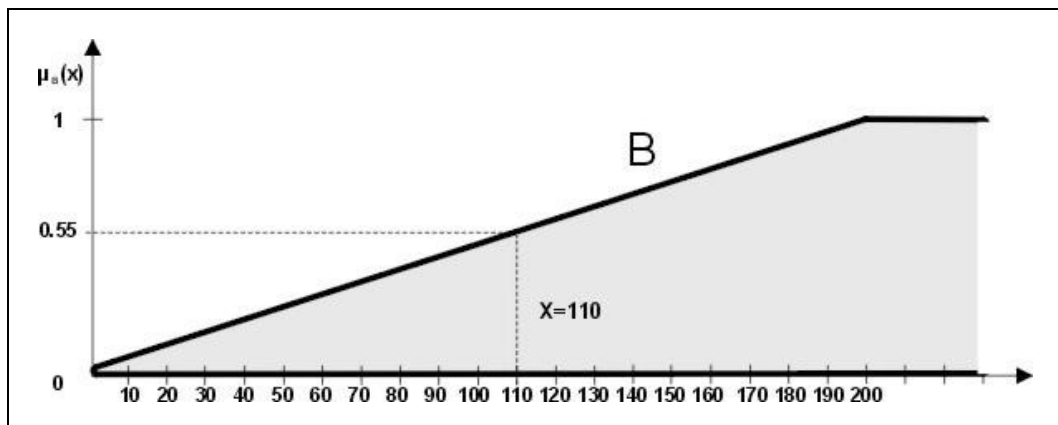


Figura 11. Conjunto fuzzy $B = \{ (x, \mu_B(x)) \mid x \in U \}$, onde $x=110$

De acordo com a Figura 11, o grau de pertinência de x em relação ao conjunto B foi obtido por meio de uma função pertinência resultando em $\mu_B(110)=0.55$.

4.1.1 Função de Pertinência

De acordo com Luger (2004), uma função de pertinência em um conjunto A , define um subconjunto X de A para medir o grau com que um elemento pertence a X . O grau de pertinência origina uma distribuição de possibilidades entre os elementos que pertencem a A , ou seja, a função de pertinência é o que especifica um conjunto *fuzzy*.

A determinação de uma função de pertinência pode ser realizada de duas formas: estatisticamente, onde a determinação da função é baseada em estudos estatísticos sobre dados coletados; ou por meio de conhecimento especializado, onde um refinamento dos parâmetros da função se faz necessário. A descrição de um elemento quanto a sua pertinência em determinado conjunto pode ser representada por diferentes funções de pertinência (BARRETO, 2001); (VELLASCO, 1999); (WANG, 1997).

Embora a função de pertinência possa ser definida a partir da experiência do especialista, é comum a utilização de funções de pertinência padrão. A seguir são apresentadas algumas destas funções juntamente com suas representações gráficas (Figuras 12, 13, 14 e 15), onde o eixo X representa o universo de discurso do sistema e o eixo $\mu(x)$ possui todos os valores possíveis no intervalo $[0,1]$, para determinar o grau de pertinência (VELLASCO, 1999):

a) função trapezoidal:

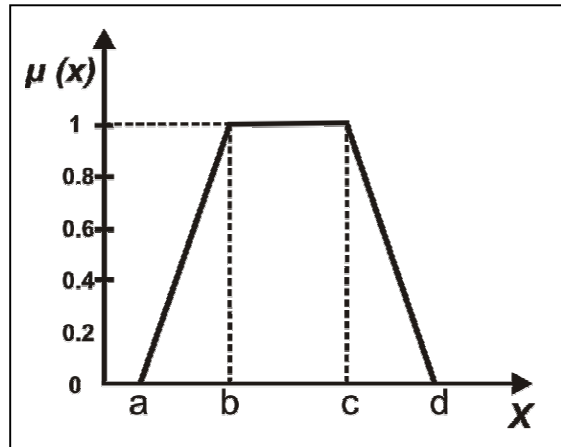


Figura 12. Representação gráfica da função trapezoidal.
Fonte: VELLASCO, M. M. B. R. (1999)

Sendo:

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

c) função triangular:

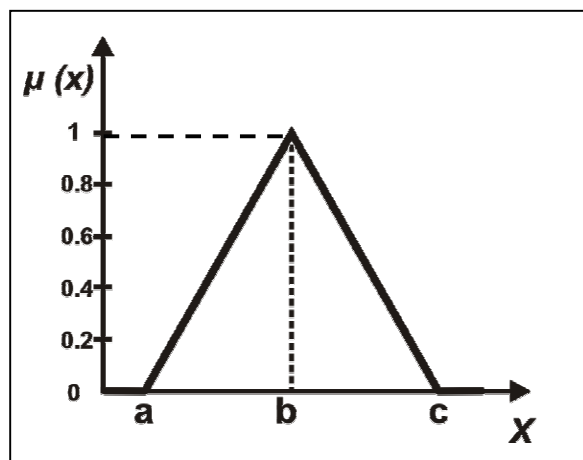


Figura 13. Representação gráfica da função triangular.
Fonte: VELLASCO, M. M. B. R. (1999).

Sendo:

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

d) função S:

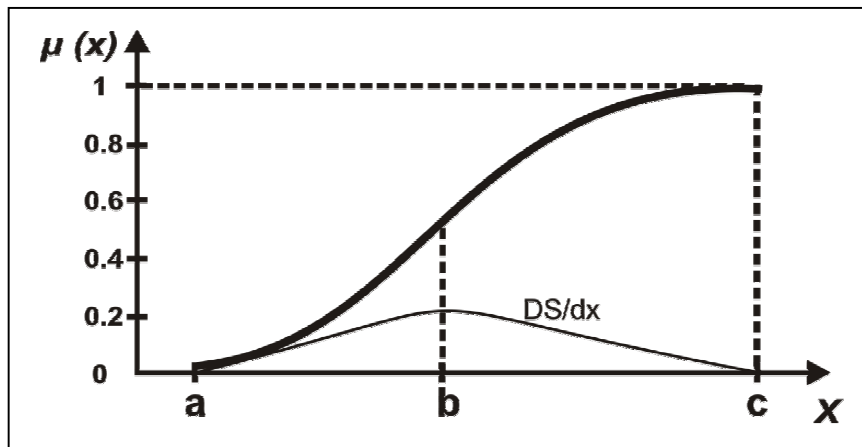


Figura 14. Representação gráfica da função S.
Fonte: VELLASCO, M. M. B. R. (1999).

Sendo:

$$S(x,a,b,c)=\begin{cases} 0 & x \leq a \\ 2[(x-a)/(c-a)]^2 & a \leq x \leq b \\ 1-2[(x-c)/(c-a)]^2 & b \leq x \leq c \\ 1 & x \geq c \end{cases}$$

c) função Z: $Z(x,a,b) = 1 - S(x,a,b)$

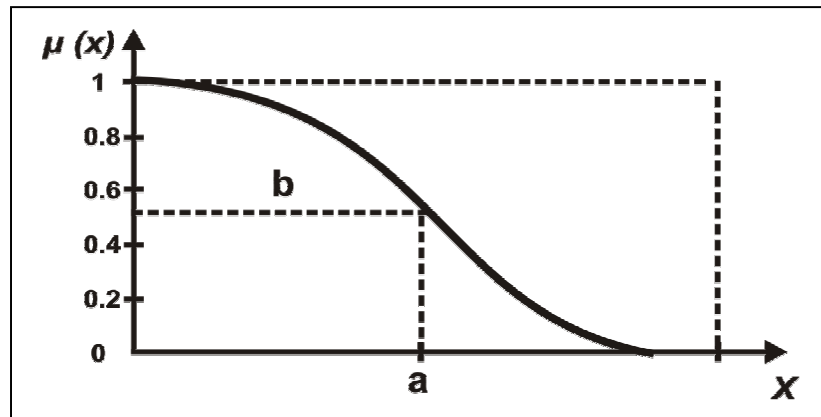


Figura 15. Representação gráfica da função Z.
Fonte: VELLASCO, M. M. B. R. (1999).

Sendo:

$$Z(x,a,b) = \begin{cases} 1 & x < a - b \\ 1 - [x - (a - b)]^2 / 2b^2 & a - b \leq x \leq a \\ [(a + b) - x]^2 / 2b^2 & a < x \leq a + b \\ 0 & x > a + b \end{cases}$$

Independente do tipo de função de pertinência a ser adotado na modelagem de um sistema *fuzzy*, para cada conjunto X em um universo de discurso U são adotadas palavras da linguagem natural para denominar tais conjuntos. Estas palavras são chamadas de *variáveis lingüísticas* onde o seu conteúdo corresponde ao próprio conjunto *fuzzy*.

A seguir são apresentados conceitos mais detalhados sobre as *variáveis lingüísticas*.

4.1.2 Variáveis Lingüísticas

As palavras da linguagem natural são utilizadas freqüentemente para descrever valores atribuídos a determinadas variáveis, como por exemplo, os valores referentes a variável “distância” podem ser definidas por uma pessoa como “perto” ou “longe”. Da mesma forma, estes valores podem ser representados numericamente como 15 metros, 43 metros, 80 metros, entre outros.

Quando a determinação do valor de uma variável se dá por meio de dados numéricos, é possível utilizar uma estrutura matemática bem definida para que estes valores sejam gerados. Quando os valores de uma variável são determinados por palavras, são utilizados conjuntos *fuzzy* para representar termos lingüísticos, onde seus valores são números reais dentro de uma escala específica (KLIR; YUAN, 1995).

Conforme a Wang (1997), uma variável que baseia-se em palavras da linguagem natural como seus valores, é chamada de variável lingüística, onde as palavras são caracterizadas por conjuntos *fuzzy* definidos em um universo de discurso U . Sendo assim, pode-se dizer que uma *variável lingüística* apresenta valores como nomes de conjuntos *fuzzy* que são representados pela *função de pertinência*.

Exemplificando esta definição, tem-se uma variável x referente a distância em que se encontra um dado objeto de um ponto de referência, onde x é uma variável lingüística que pode assumir os valores perto e longe, sendo estes valores definidos dentro do intervalo $[0,200]$, conforme ilustrado na Figura 16.

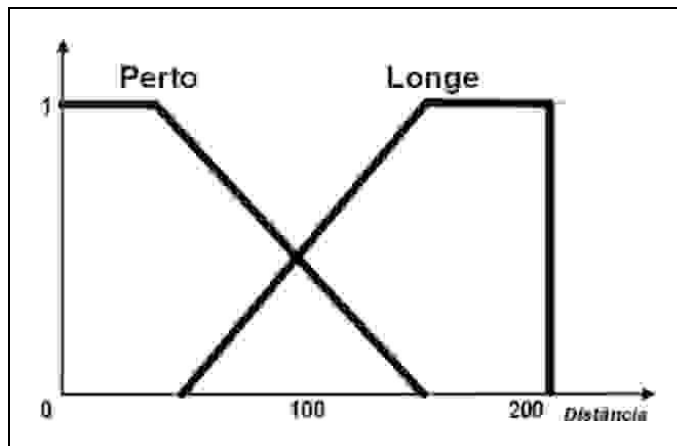


Figura 16. A distância que se encontra um objeto como uma variável lingüística que pode tomar os conjuntos *fuzzy* “perto” e “longe” como valores.

A definição das variáveis lingüísticas é o primeiro passo para realização das etapas de desenvolvimento de um sistema *fuzzy*, no entanto, da mesma forma que em conjuntos ordinários, nos conjuntos *fuzzy* existem propriedades e operações que são realizadas sobre estes conjuntos.

4.2 OPERAÇÕES DOS CONJUNTOS FUZZY

Antes que sejam introduzidos os conceitos referentes às operações sobre conjuntos *fuzzy* é necessária uma conceitualização a respeito das propriedades de um conjunto *fuzzy* (WANG, 1997); (AZEVEDO; BRASIL; OLIVEIRA, 2000):

- a) cardinalidade: é a soma dos valores de pertinência de um conjunto, tal que:

$$|A| = \sum_{x \in U} \mu_A(x);$$

- b) suporte: um conjunto *fuzzy* A em universo U contém todos os elementos $x \in$

A com grau de pertinência diferente de zero, tal que $\text{suporte}(A) = \{ x \mid \mu_A(x)$

$> 0 \}$, onde a nomenclatura *suporte* também é representada por *sup* ou

- simplesmente S . Um conjunto *fuzzy* cujo suporte é vazio, é chamado de conjunto *fuzzy* vazio. Se o suporte de um conjunto *fuzzy* é um ponto simples em U , tal que $\mu_A(x) = 1$, o conjunto é chamado de *fuzzy singleton*;
- c) núcleo: é o conjunto de todos os elementos $x \in A$ em um universo de discurso U , tal que $\text{núcleo}(A) = \{ x \mid \mu_A(x) = 1 \}$;
- d) limite: é o conjunto de todos os elementos $x \in A$ em um universo de discurso U , tal que $\text{limite}(A) = \{ x \mid \mu_A(x) \neq 0 \text{ e } \mu_A(x) \neq 1 \}$;
- e) ponto de cruzamento: ou *crossover*, é um elemento $x \in A$ em um universo de discurso U onde: $\text{crossover}(A) = \{ x \mid \mu_A(x) = 0,5 \}$;
- f) altura: é definida pelo maior valor de pertinência alcançado em qualquer ponto do conjunto, ou seja, $h(A) = \sup_{x \in U} \mu_A(x)$. Um conjunto *fuzzy* onde $h(A)=1$, é chamado *conjunto fuzzy normal*. Se $h(A)<1$, o conjunto *fuzzy* é chamado *subnormal*.
- g) α -cut: é derivado de um conjunto *fuzzy* A que contém todos os valores de pertinência maiores ou iguais a α , definido por $A_\alpha = \{ x \mid \mu_A(x) \geq \alpha \}$. Um conjunto *α -corte forte* é definido por $A_{\alpha+} = \{ x \mid \mu_A(x) > \alpha \}$.

Exemplificando estes conceitos, toma-se um conjunto A em um universo de discurso U , demonstrado na Figura 17, cujos elementos e graus de pertinência são:
 $A = \{(1,0.0), (2,0.2), (3,0.5), (4,0.7), (5,1.0), (6,1.0), (7,0.8), (8,0.5), (9,0.0)\}$:

- a) a cardinalidade de A é $|A| = \sum \mu_A(1,2,3,4,5,6,7,8,9) = 4.7$;
- b) o *suporte* de A é $S(A) = \{2, 3, 4, 5, 6, 7, 8\}$;
- c) o *núcleo* de A é $\text{Núcleo}(A) = \{5, 6\}$;
- d) o *limite* de A é $\text{Limite}(A) = \{2, 3, 4, 7, 8\}$;

- e) o ponto de cruzamento de A é $\text{crossover}(A) = \{3, 8\}$;
- f) a altura de A é $h(A) = 1$, sendo assim um conjunto *fuzzy normal*;
- g) e para $\alpha = 0.5$, o $A_\alpha = \{3, 4, 5, 6, 7, 8\}$ e $A_{\alpha+} = \{4, 5, 6, 7\}$.

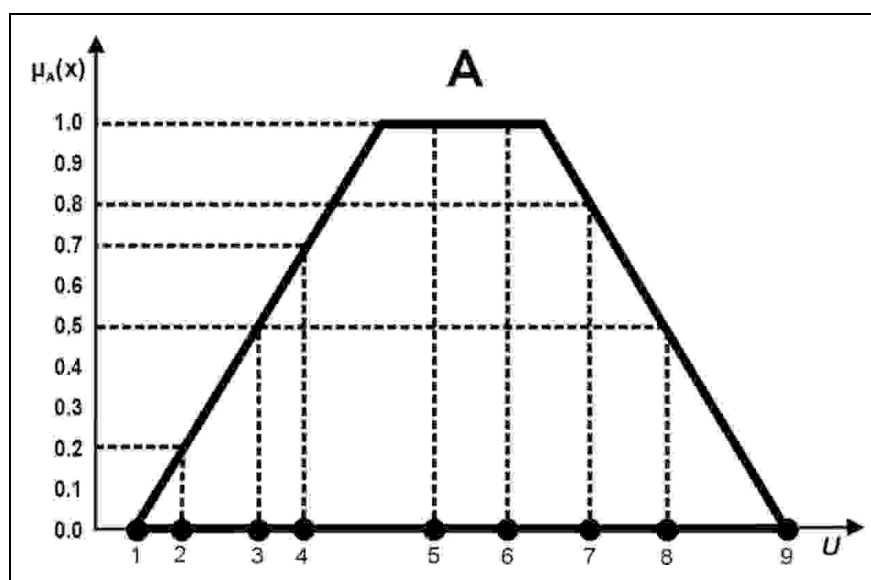


Figura 17. Um conjunto *fuzzy* A em um universo de discurso U .

Estes conceitos são aplicados a cada conjunto individualmente. Na realização de operações entre conjuntos *fuzzy*, é necessária a utilização de tratamentos matemáticos com funções de n argumentos, onde $n > 2$. Estes tratamentos são efetuados por meio de operações básicas da mesma forma que nos conjuntos *crisp*.

Conforme Klir e Yuan (1995), as operações básicas - *complemento*, *união* e *interseção* – são realizadas sobre conjuntos *fuzzy*, porém não são as únicas, sendo que outras funções podem ser apropriadas para representar estas operações em diferentes contextos.

Os conjuntos *fuzzy* são combinados e modificados pela aplicação das operações sobre sua função de pertinência. Na abordagem dos conceitos básicos sobre as principais

operações, assume-se que A e B sejam conjuntos *fuzzy* definidos no mesmo universo de discurso U , onde (WANG, 1997); (AZEVEDO; BRASIL; OLIVEIRA, 2000):

- a) o *complemento* ou *negação* de um conjunto *fuzzy* A em um universo de discurso U , representado por \bar{A} ($\neg A$, *NÃO A*) é definido para todo $x \in U$ por:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Sendo assim, o complemento do conjunto *fuzzy* A contém todos os elementos que não pertencem ao conjunto. Porém, os elementos de U em que $\mu_A(x) = \mu_{\bar{A}}(x)$ são chamados *pontos de equilíbrio*.

- b) a *união*, chamada de *t-conorma*, de A e B é um conjunto *fuzzy* em U , também, representado por $A \cup B$ (A *OU* B), cuja função de pertinência é definida por:

$$\mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)] = \mu_A(x) \vee \mu_B(x)$$

onde *max* é o maior operador, ou seja, o máximo da função de pertinência;

- c) a *interseção*, chamada também de *t-norma*, de A e B é um conjunto *fuzzy* em U , representado por $A \cap B$ (A *E* B), cuja função de pertinência é definida por:

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)] = \mu_A(x) \wedge \mu_B(x)$$

onde *min* é o menor operador, ou seja, o mínimo da função de pertinência;

Os valores manipulados por operações *fuzzy* são, em geral, valores de conjuntos *fuzzy* que possuem função de pertinência do tipo $\mu_A: \mathfrak{R} \rightarrow [0,1]$, ou seja, conjuntos definidos pelo conjunto \mathfrak{R} dos números reais, chamados também de *números* ou *intervalos*

fuzzy. Na especificação de um valor como *número fuzzy*, um conjunto A sobre \mathfrak{R} deve possuir as seguintes características (KLIR; YUAN, 1995):

- a) ser um conjunto *fuzzy normal*;
- b) A_α deve ser um intervalo fechado para todo $\alpha \in [0,1]$;
- c) o *suporte* de A deve ser limitado.

Os números *fuzzy* são atribuídos aos conjuntos como valores dentro de uma escala numérica específica para caracterizar os estados das *variáveis lingüísticas*. Por meio de números *fuzzy*, estas variáveis representam conceitos lingüísticos que permitem produzir modelos matemáticos a partir de descrições vagas da linguagem natural, sendo de fundamental importância na incorporação do conhecimento humano em sistemas computacionais por meio de regras (WANG, 1997).

Dados estes conceitos, são apresentadas a seguir as etapas que compreendem a arquitetura de um sistema que utiliza-se da lógica *fuzzy*.

4.3 ARQUITETURA DOS SISTEMAS FUZZY

As etapas do desenvolvimento de sistemas que utilizam-se da lógica *fuzzy* são constituídas pelos processos de *fuzzyficação*, *inferência das regras* e *defuzzyficação* conforme ilustrado na Figura 18.

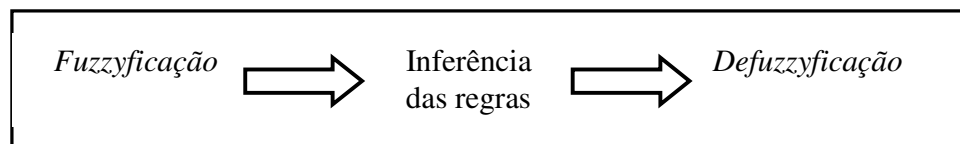


Figura 18. Etapas de desenvolvimento dos sistemas *fuzzy*.

4.3.1 Fuzzyficação

O processo de *fuzzyficação* consiste na aquisição do conhecimento do especialista para definição e formulação dos os conjuntos *fuzzy* que representam as variáveis utilizadas no problema, ou seja, as variáveis lingüísticas, bem como a determinação das funções de pertinência aplicadas sobre estes conjuntos.

Na resolução de um determinado problema, devem ser definidas as variáveis de entrada e saída do sistema para que sejam desenvolvidas as relações entre os conjuntos. A representação do conhecimento humano se dá por meio destas relações e são representadas por meio da definição de regras *fuzzy* e utilizadas no processo de inferência (WANG, 1997).

4.3.2 Inferência

A inferência em um sistema *fuzzy* é realizada por meio de proposições constituídas das regras SE-ENTÃO expressas por (WANG, 1997); (AZEVEDO; BRASIL; OLIVEIRA, 2000):

SE <antecedente>, ENTÃO <conseqüente>

Conforme Wang (1997), uma proposição *fuzzy* pode ser definida por dois tipos: proposições atômicas e proposições compostas. As proposições *fuzzy* atômicas são constituídas tal que $x \text{ é } A$, onde x representa a variável lingüística em questão e A é o valor da variável x definido por meio de uma função de pertinência. Uma proposição *fuzzy*

composta é formada por duas ou mais proposições atômicas cujas relações são definidas pelos conectivos “e”, “ou” e “não”, que representam a intersecção, união e complemento *fuzzy* respectivamente.

Como exemplo, considera-se três variáveis A, B e C cujos conjuntos *fuzzy* são dados por: A{a,b}, B{c,d,e} e C{f,g}, onde suas regras são definidas como :

Se **A=b** (0,46) e **B=c** (0,25) então **C=g**

Se **A=a** (0,20) e **B=d** (0,75) então **C=f**

Se **A=a** (0,20) e **B=e** (0,25) então **C=g**

Se **A=b** (0,46) e **B=d** (0,75) então **C=f**

Após a definição das regras que compõem a inferência *fuzzy* é realizada a interpretação das mesmas para que seja realizado o processo de *defuzzyficação*. A interpretação das regras consiste na aplicação dos conectivos lógicos apresentados anteriormente de forma que (KLIR; YUAN, 1995):

- a) são determinadas todas as proposições *fuzzy* entre as variáveis utilizadas nos conjuntos *fuzzy* cujas relações podem ser determinadas por meio do conectivo “e”. Conforme as regras de operação sobre os conjuntos *fuzzy*, são extraídos os valores mínimos entre as relações, como ilustrado na Figura 19, tal que:

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)]$$

Se A=b (0,46) e B=c (0,25) então C=g	(0,25)
Se A=a (0,20) e B=d (0,75) então C=f	(0,20)
Se A=a (0,20) e B=e (0,25) então C=g	(0,20)
Se A=b (0,46) e B=d (0,75) então C=f	(0,46)

Figura 19. Aplicação do conectivo de intersecção.

b) após a determinação dos valores mínimos das relações de interseção, são aplicados sobre estes resultados, o conectivo união para obter-se o valor máximo entre estes resultados, conforme ilustrado na Figura 20, onde: $\mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)]$. O valor final obtido na avaliação das regras é utilizado no processo de *defuzzyficação*.

C=g	(0,25)
C=f	(0,20)
C=g	(0,20)
C=f	(0,46)
C= f (0,20) ou (0,46) = (0,46)	
C= g (0,25) ou (0,20) = (0,25)	

Figura 20. Aplicação do conectivo de união.

4.3.3 Defuzzyficação

Conforme Azevedo, Brasil e Oliveira (2000), a etapa de *defuzzyficação* é o processo pelo qual as funções de pertinência são utilizadas para que sejam encontrados o grau de pertinência e uma região resultante definida. Desta forma, é encontrada a melhor representação de saída do sistema. Dentre as técnicas desenvolvidas para que sejam produzidas saídas as mais comuns são:

- a) *maximizador*: nesta técnica é selecionada a saída máxima, ou seja, o valor de maior grau de pertinência;
- b) *média do peso*: as saídas possíveis são obtidas por meio da média dos pesos;

c) centróide e suas variantes: por meio do qual o centro é localizado.

A utilização das relações entre conjuntos fuzzy permite que diversos problemas referentes às aplicações práticas sejam resolvidos quando empregadas as devidas relações. Embora sejam de grande utilidade, estas relações exigem uma grande demanda computacional. Na superação desta dificuldade, existe a possibilidade de otimização na modelagem de sistemas *fuzzy* para permitir que sejam aplicadas nas mais diversas áreas (KLIR; YUAN, 1995).

A otimização de técnicas da IA para solução de determinados tipos de problemas é realizada por meio da combinação de diferentes paradigmas para o desenvolvimento de *sistemas de arquitetura híbrida*, aumentando o potencial das aplicações. Os modelos deste tipo de sistema são apresentados a seguir.

5 SISTEMAS DE ARQUITETURA HÍBRIDA

Um sistema de arquitetura híbrida caracteriza-se pela utilização de duas ou mais técnicas da IA na modelagem de aplicações que necessitam uma otimização na simulação da complexidade do raciocínio humano. Esta união pode resultar em um sistema com menos deficiências devido a capacidade de absorver as melhores características das técnicas que a compõe (AZEVEDO; BRASIL; OLIVEIRA, 2000).

Conforme Aurélio, Lopes e Vellasco (1999), a associação de técnicas da modelagem de IA na construção de um sistema híbrido pode ocorrer de três formas distintas: *sistema híbrido sequencial*, *sistema híbrido auxiliar* e *sistema híbrido incorporado*.

5.1 SISTEMA HÍBRIDO SEQUENCIAL

O modelo de sistema híbrido sequencial apresenta a forma mais fraca de hibridização, onde um subsistema de paradigma 1 fornece as entradas de outro subsistema de paradigma 2, conforme ilustrado na Figura 21. Um exemplo deste modelo de sistema híbrido é a ocorrência de um pré-processador *fuzzy* ou estatístico acionando uma rede neural.

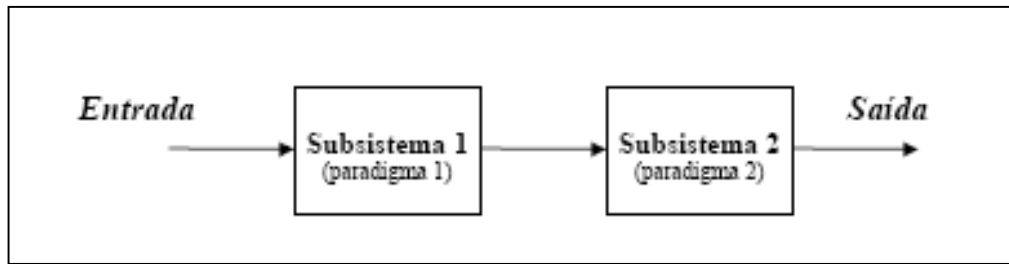


Figura 21. Modelo de sistema híbrido seqüencial.
 Fonte: AURÉLIO, M.; LOPES, C.H. ;VELLASCO, M.M.B.R.. (1999)

5.2 SISTEMA HÍBRIDO AUXILIAR

O sistema híbrido auxiliar possui um maior grau de hibridização que o modelo seqüencial, sendo que um subsistema de paradigma 1 realiza uma chamada à um subsistema de paradigma 2 para realização de uma tarefa auxiliar, conforme ilustrado na Figura 22.

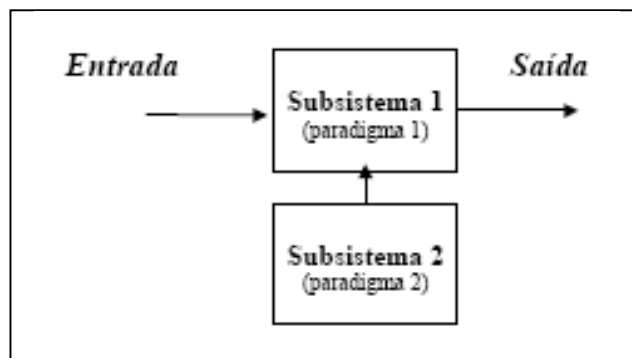


Figura 22. Modelo de sistema híbrido auxiliar.
 Fonte: AURÉLIO, M.; LOPES, C.H.; VELLASCO, M.M.B.R. (1999).

Exemplos de sistemas de arquitetura híbrida auxiliar são:

- a) neuro-genético: uma rede neural solicitando o processamento de um algoritmo genético na otimização de seus pesos, ajuste dos parâmetros de treinamento ou definição da arquitetura de rede;
- b) fuzzy-genético: associa técnicas evolucionárias com sistemas *fuzzy* na criação e base de regras e ajuste dos parâmetros;
- c) genético-estatístico: trata da associação de parâmetros por meio de algoritmos genéticos e identificação de modelos lineares.

5.3 SISTEMA HÍBRIDO INCORPORADO

O sistema híbrido incorporado possui o maior grau de hibridização possível, onde não existe uma separação entre os subsistemas que o compõem. Neste modelo um subsistema de paradigma 1 é incorporado a um subsistema de paradigma 2. Este modelo está ilustrado na Figura 23. Exemplo: um sistema de inferência *fuzzy* implementado conforme a estrutura de uma rede neural.

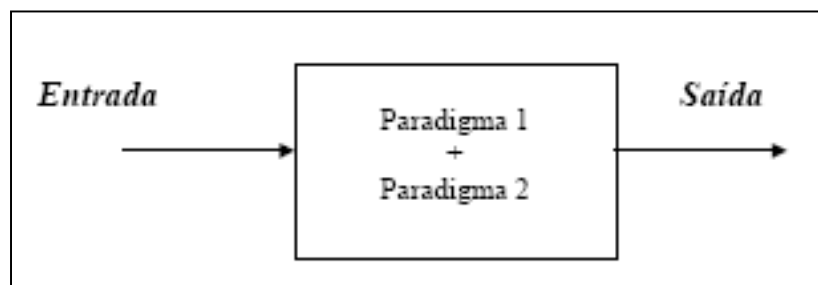


Figura 23. Modelo de sistema híbrido incorporado.

Fonte: AURÉLIO, M.; LOPES, C.H.; VELLASCO, M.M.B.R.. (1999)

Exemplos de sistemas de arquitetura híbrida incorporada são:

- a) neural-estatístico: é constituído pela união de métodos estatísticos e redes neurais para a análise e previsão de séries temporais;
- b) neuro-fuzzy: este tipo de sistema utiliza a capacidade de aprendizado das redes neurais juntamente com a possibilidade de tratamento de informações imprecisas da lógica *fuzzy*. Devido ao fato de se tratar do objeto de estudo deste trabalho, uma descrição melhor detalhada deste tipo de sistema será apresentada a seguir.

5.4 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*.

5.4.1 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 *multilayer* 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.

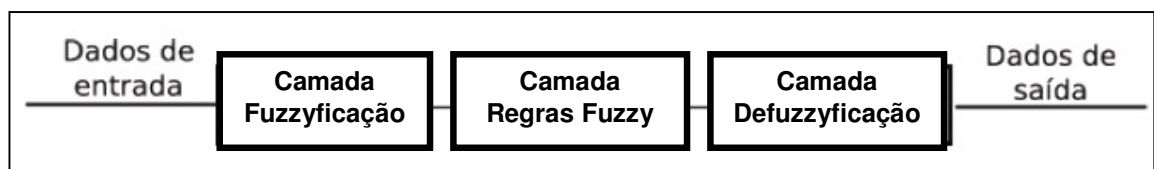


Figura 24. Modelo da arquitetura de uma rede neurofuzzy híbrida.
Fonte: RODARTE, C. A.; SANTOS, L.N. (2005)

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.

5.4.2 Redes *Neurofuzzy* Cooperativas

Nas redes *neurofuzzy cooperativas* a rede neural e o sistema *fuzzy*, mesmo integrados em sua arquitetura, processam os dados de forma independente. Neste modelo, a rede neural pode ser utilizada para pré ou pós processar as entradas ou saídas de um sistema *fuzzy*.

As redes *neurofuzzy cooperativas* apresentam quatro modelos distintos de implementação, como ilustrado na Figura 25:

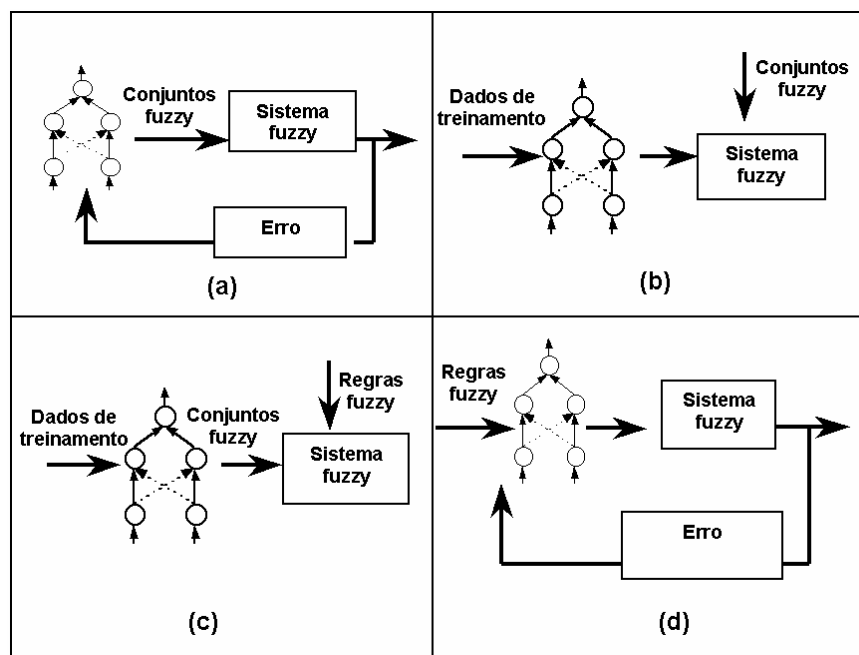


Figura 25. Modelos da arquitetura de uma rede neurofuzzy cooperativa
 Fonte: Adaptado de RAMOS, P. G. (1999)

Conforme Nauck e Kruse (1994) estes modelos consistem em:

a) primeiro modelo (a): as regras *fuzzy* devem ser previamente definidas, bem como alguns conjuntos iniciais. Os parâmetros dos conjuntos *fuzzy* são ajustados por uma rede neural tanto da forma supervisionada como não supervisionada;

b) segundo modelo (b): necessita que todos os conjuntos *fuzzy* sejam conhecidos e as regras geradas por meio da rede neural. O processo de treinamento utilizado é o não supervisionado;

c) terceiro modelo (c): as regras *fuzzy* são definidas previamente e os conjuntos gerados por meio de um treinamento não supervisionado. A rede neural deriva as funções de pertinência a partir dos dados inseridos no treinamento da rede;

d) último modelo apresentado (d): necessita que os conjuntos e as regras *fuzzy* sejam definidas inicialmente pelo especialista. A rede neural ajusta os pesos de forma que estes representem o grau de importância de cada regra.

Desta forma, é possível afirmar que o processo de aprendizagem em um sistema *neurofuzzy cooperativo* pode ser iniciado pela incorporação do conhecimento do especialista na rede e aperfeiçoado pela utilização de um algoritmo de treinamento de forma supervisionada.

No desenvolvimento da *shell Ícaro* utilizou-se a abordagem híbrida. Após conhecidos os tipos de redes *neurofuzzy*, na seção 5.5 são apresentadas algumas ferramentas para o desenvolvimento destes sistemas.

5.5 ALGUNS EXEMPLOS DE FERRAMENTAS *NEUROFUZZY*

As redes *neurofuzzy* podem ser aplicadas na resolução de diversos tipos de problema, entre eles reconhecimento de padrões, controle, previsão e interpolação de funções (FULLER, 1995).

As ferramentas *neurofuzzy*, também denominadas *shells*, mais conhecidas e desenvolvidas para auxiliar na implementação de sistemas deste tipo são (VELLASCO,1999); (NAUCK; KRUSE, 1994):

- a) FSOM: desenvolvido por Vuorimaa, este modelo aplicado para aproximação de funções e controle, calculando os graus de pertinência por meio da utilização da função triangular. Os consequentes das regras são *singletons* ou combinação linear das entradas. Este modelo foi implementado na *shell neurofuzzy* FSOM;
- b) ANFIS: da sigla “Adaptative Network-based Fuzzy Inference System”, é utilizada principalmente na implementação de sistemas de previsão e aproximação de funções. Seu modelo de arquitetura, desenvolvido por Jang, utiliza modelos de inferência desenvolvidos por Takagi-Sugeno e foi implementado como modulo para utilização na ferramenta matemática MATLAB;
- c) NEFCLASS: foi desenvolvido no ano de 1995 por Nauck e Kruse, na Universidade de Braunschweig, Alemanha. É utilizado basicamente na resolução de problemas de classificação, com a utilização de graus de pertinência gerados por meio da função triangular. É uma ferramenta de código aberto, porém sua autorização de uso não permite comercialização, sendo somente autorizado para fins acadêmicos e científicos. Esta ferramenta encontra-se disponível em: <http://fuzzy.cs.uni-magdeburg.de/nefclass/>.

d) NEFCON: desenvolvido na linguagem C++ por Nauck e implementado para UNIX por Herrmann Diekgerdes e Roland Stellmach, é para utilização em aplicações de controle. É uma ferramenta de código aberto que utiliza um controlador *fuzzy* normal e realiza os processos de aprendizado da rede por meio das regras *fuzzy*. Além das versões disponibilizadas gratuitamente em <http://fuzzy.cs.uni-magdeburg.de/nefcon/>, também apresenta um módulo implementado para o sistema de estatística e matemática MATLAB;

Estas são algumas das ferramentas *neurofuzzy* disponíveis, sendo que no próximo capítulo são abordadas algumas pesquisas que envolvem a utilização de lógica *fuzzy*, RN e redes *neurofuzzy* para a avaliação de imóveis.

6 ALGUNS EXEMPLOS DA UTILIZAÇÃO DAS REDES *NEUROFUZZY*

Neste capítulo são abordados alguns exemplos de pesquisas que envolvem a aplicação do modelo híbrido *neurofuzzy* em outros domínios de conhecimento, bem como um trabalho com a utilização das redes neurais artificiais para a avaliação de imóveis.

6.1 REDES NEURAIIS *FUZZY* APLICADAS NA IDENTIFICAÇÃO E CONTROLE DE SISTEMAS

Dissertação de mestrado desenvolvida por Huamaní (2003) na Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, apresenta um estudo comparativo entre as redes *neurofuzzy* híbridas, redes neurais e sistemas *fuzzy*, aplicados a problemas de identificação e controle de sistemas dinâmicos e não lineares.

Os resultados apresentados na pesquisa demonstraram que a utilização das redes *neurofuzzy* é uma alternativa eficaz na minimização da quantidade de informações necessária para resolução de problemas não-lineares (HUAMANÍ, 2003).

6.2 UMA INVESTIGAÇÃO DAS REDES *NEURO-FUZZY* APLICADAS À MINERAÇÃO DE DADOS

Trabalho de Conclusão de Curso de Graduação apresentado por Ramos (1999) ao Centro de Ciências Exatas da Universidade Federal de Pernambuco. Consiste na análise e aplicação das redes *neurofuzzy* em problemas de mineração de dados.

A utilização das redes *neurofuzzy* neste domínio de aplicação demonstrou resultados satisfatórios, devido a possibilidade de extração do conhecimento automaticamente, bem como a compreensão das respostas e decisões (RAMOS, 1999).

6.3 SISTEMAS INTELIGENTES HÍBRIDOS: COMO COMBINAR REDES NEURAIAS ARTIFICIAIS, LÓGICA NEBULOSA E ALGORITMOS GENÉTICOS

Trabalho de Conclusão de Curso de Graduação desenvolvido por Santos (2004) no Curso de Ciência da Computação da Universidade Estadual de Londrina, apresenta uma pesquisa referente ao estudo da utilização de técnicas de inteligência artificial para geração de sistemas inteligentes híbridos por meio da combinação das redes neurais e lógica *fuzzy*, redes neurais e algoritmos genéticos, lógica *fuzzy* e algoritmos genéticos e uma combinação destas três técnicas.

A maior integração apresentada, utilizando as três técnicas da IA, apesar de apresentar grande complexidade em seu desenvolvimento, apresentou maior eficácia na resolução de problemas de controle devido à otimização das três técnicas (Santos, 2004).

6.4 UM MODELO NEURODIFUSO DE MULTIAGENTES PARA O APOIO AO ENSINO DE DISCIPLINAS DE CIÊNCIAS EXATAS

Tese de doutorado submetida por Viali (1999) ao programa de pós-graduação em Engenharia de Produção da Universidade Federal de Santa Catarina, apresenta um modelo hipermídia para auxiliar no ensino de disciplinas da área matemática, especialmente

a Probabilidade e a Estatística, por meio da utilização de um modelo *neurodifuso* de multiagentes.

6.5 ESTUDO DA INFLUÊNCIA NA ACESSIBILIDADE NO VALOR DE LOTES ATRAVÉS DO USO DE REDES NEURAIAS

Tese de doutorado submetida por Brondino (1999) ao programa de doutorado em Engenharia de Civil - Transportes da Escola de Engenharia de São Carlos da Universidade de São Paulo. Os objetivos desta pesquisa foram a identificação das variáveis que influenciam no valor das propriedades, avaliação do uso de redes neurais artificiais para fins de avaliação e o estudo da influência de uma medida de acessibilidade no valor de terrenos urbanos.

Nos resultados obtidos, a utilização das redes neurais forneceu estimativas adequadas de valor, demonstrando um desempenho superior a metodologia de avaliação tradicional (BRONDINO, 1999).

7 SISTEMA 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 implementação do protótipo pretendia-se utilizar uma *shell neurofuzzy* integrada a um ambiente de desenvolvimento para realização deste trabalho. No decorrer da pesquisa descobriu-se que, para o tipo de problema proposto, as ferramentas gratuitas encontradas para integração com o ambiente de desenvolvimento eram de difícil compreensão e haviam sido implementadas no ano de 1993 em formato MS-DOS, o que tornava a probabilidade de sucesso na integração muito baixa, além de possuir um modelo de configuração de certa forma ultrapassado para os padrões atuais.

Devido a esse fato, optou-se pelo desenvolvimento do protótipo de uma *shell neurofuzzy* implementando-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 desenvolvimento do protótipo da *shell ÍCARO* foi realizado em ambiente de programação *Delphi* por meio da linguagem *Object Pascal*. Os testes no protótipo desenvolvido foram realizados por meio de um conjunto de dados pesquisados no Trabalho de Conclusão de Curso da acadêmica Graziela Olivo Fermo do curso de Engenharia Civil

da UNESCO e estes dados foram disponibilizados pela coordenadora do referido curso e co-orientadora desta pesquisa, Profa. M. Eng. Evelize Chemale Zancan conforme Contrato de Autorização de Uso (Apêndice A).

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.

7.1 AVALIAÇÃO DE IMÓVEIS

A Engenharia de Avaliações é uma das especialidades da Engenharia Civil que objetiva determinar com base nas normalizações da Associação Brasileira de Normas Técnicas (ABNT) e na aplicação de metodologia apropriada, uma estimativa no valor de mercado para bens de qualquer natureza, influenciando na tomada de decisões relativas ao uso e disposição destes bens quanto aos seus valores, custos e investimentos (DANTAS, 1998).

Os resultados obtidos pela aplicação das técnicas da Engenharia de Avaliações sobre um bem imóvel são de grande interesse aos diversos agentes do mercado imobiliário como compradores e vendedores de imóveis, seguradoras, bancos de crédito, imobiliárias, poder judiciário, prefeituras, construtores, entre outros.

Assim, este conhecimento é importante existe quando a necessidade de realizar a transferência de propriedade, transações de compra e venda, financiamento e crédito, justa indenização nos casos de desapropriação judicial, tomada de decisões sobre bens imóveis, balanços patrimoniais, base para cálculo de impostos e taxas, aplicações referentes

a seguros ou sobre investimentos, e valores justos nas transações de locação (ABUNAHMAN, 1999); (DANTAS, 1998).

A avaliação de imóveis consiste em analisar uma série de variáveis do mercado imobiliário a fim de estimar o valor de um imóvel em um dado momento, independente da aplicação à qual se destina, por meio da análise de dados relevantes (ABUNAHMAN, 1999).

Na análise que resultará a estimativa de valor, o Método Comparativo de Dados do Mercado é o mais utilizado para avaliação de bens imóveis como casas, apartamentos, terrenos, lojas, salas, entre outros (MENDONÇA, 1998).

7.1.1 Método Comparativo de Dados de Mercado

A aplicação do método comparativo de dados de mercado é realizada por meio da obtenção de um conjunto de dados que estatisticamente possam ser tomados como amostra do mercado imobiliário. Estes dados devem possuir características e atributos semelhantes ao bem avaliado para que, por comparação, possam exercer influência na formação do valor (DANTAS, 1998).

As etapas que constituem o desenvolvimento da aplicação deste método são:

- a) caracterização do imóvel avaliado: por meio de vistorias são analisadas as características e tudo aquilo que possa influenciar no valor de um imóvel, interna ou externamente, bem como o contexto imobiliário a que pertence, a fim de predefinir as variáveis que devem influenciar na formação de preços dos bens semelhantes;

- b) pesquisa de dados relativos a imóveis semelhantes ao avaliado: levantamento de dados do mercado imobiliário atual, com características semelhantes ao avaliado, para que seja formada uma amostragem da qual se possa iniciar o cálculo do valor do imóvel. A amostra de dados obtida deverá ser válida e possuir informações verídicas;
- c) cálculo do valor: os atributos e componentes da amostragem de dados obtida são tratados no cálculo do valor, por meio de procedimentos da estatística inferencial, que consiste na busca de modelos matemáticos para estimar o valor do imóvel em estudo.

7.1.2 Estatística Inferencial

Quando se propõe inferir, significa dizer que se forma uma conclusão a respeito de algo. Inferir estatisticamente significa gerar conclusões sobre alguma coisa com base em medidas estatísticas. Na engenharia de avaliações, o contexto de inferência estatística é utilizado para explicar o comportamento do mercado com base nos dados levantados, desta forma, ao se conhecer uma parte destes dados, é possível tratar generalizações sobre as observações levantadas no segmento do mercado que se pretende analisar (DANTAS, 1998).

Os dados tratados pela inferência estatística, são também chamados de variáveis e, ao serem definidas, devem representar da melhor forma o conjunto de dados, excluindo aquelas que não agregam valor e escolhendo as que mais influenciam no valor do imóvel (ZANCAN, 1996).

As variáveis coletadas e observadas na amostragem, denominadas *variáveis independentes*, podem ter natureza quantitativa ou qualitativa. As variáveis independentes quantitativas podem agregar valores de medida ou contagem, como exemplo: área, número de quartos ou distância de um pólo. Nas variáveis independentes qualitativas, os valores são atribuídos conforme critérios estabelecidos, classificando a variável em conjuntos valorados, como exemplo: topografia (1-alta, 2-média e 3-baixa) ou conservação (1-boa, 2-média, 3-ruim) (DANTAS, 1998); (MENDONÇA, 1998).

Para que seja encontrado o valor de um imóvel, um conjunto n de valores das variáveis independentes ($x_1, x_2, x_3, \dots, x_n$) serão utilizados para formação do valor de uma variável y denominada *variável dependente*, sendo:

$$y = x_1 + x_2 + x_3 + \dots + x_n$$

O tratamento matemático que relaciona as variáveis dependentes às independentes é definido conforme a metodologia escolhida pelo especialista, que define um modelo matemático que melhor represente o comportamento de mercado da localização submetida à avaliação.

7.2 METODOLOGIA

O desenvolvimento do protótipo de *shell neurofuzzy* ÍCARO fundamentou-se metodologicamente pelas seguintes etapas: levantamento bibliográfico; 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.

7.2.1 Levantamento bibliográfico

Consistiu no levantamento bibliográfico dos temas envolvidos na pesquisa realizada durante toda a concepção do projeto referente as técnicas utilizadas para compreensão dos paradigmas da inteligência artificial que compõem as estruturas das redes *neurofuzzy*.

Nesta etapa aprofundou-se os estudos sobre os aprendizados das redes neurais artificiais, a inferência *fuzzy* e os sistemas de arquitetura híbrida *neurofuzzy*.

Durante os estudos realizados, enfatizou-se a compreensão da arquitetura dos sistemas *neurofuzzy*, bem como a modelagem matemática para o seu treinamento e utilização.

7.2.2 Aquisição do conhecimento sobre a avaliação de imóveis

Esta etapa de desenvolvimento consistiu na realização de entrevista estruturada e desestruturada com a Profa. M. Eng. Evelize Chemale Zancan, especialista em engenharia de avaliações, referente as técnicas e utilização das variáveis a serem utilizadas 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.

7.2.3 Demonstração Matemática da Modelagem *Fuzzy* da Rede Neural

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 seguintes variáveis lingüísticas e valores *fuzzy* para o domínio de aplicação referente a avaliação de imóveis da tipologia apartamento no município de Criciúma:

- a) área total: representa o tamanho do apartamento em m². Definida no intervalo de 40,00 à 1000 m². Os conjuntos pertencentes a esta variável são: pequeno (de 40,00 à 60,00 m²), médio (de 100,00 à 150,00 m²) e grande (de 200,00 m² em diante), conforme ilustrado na Figura 26;

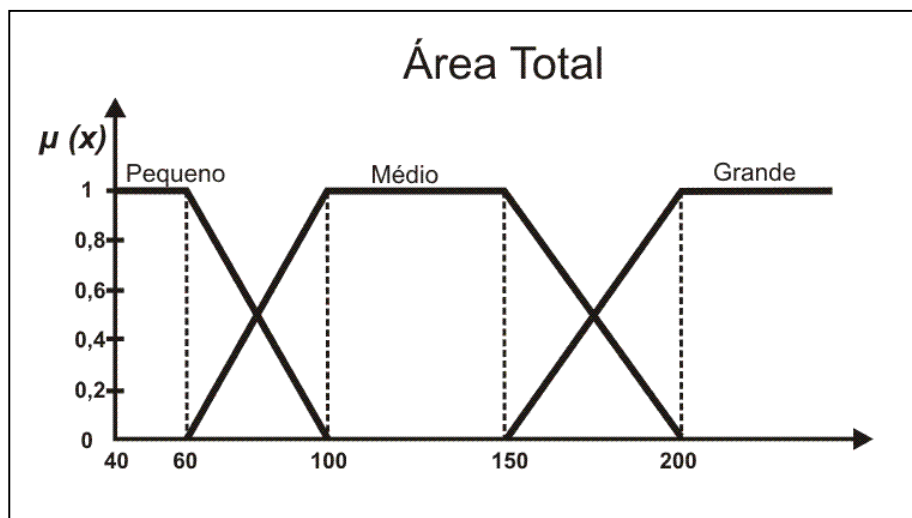


Figura 26. Representação gráfica do conjunto de entrada Área Total

b) número de dormitórios: valores referentes ao número de dormitórios por apartamento. Definida no intervalo de 1 a 6, considerando-se 0,5 para apartamentos com suíte. Os conjuntos pertencentes à esta variável são: pouco (de 1 à 1,5), normal (de 2,5 à 3,5) e muito (de 4,5 em diante) , conforme ilustrado na Figura 27;

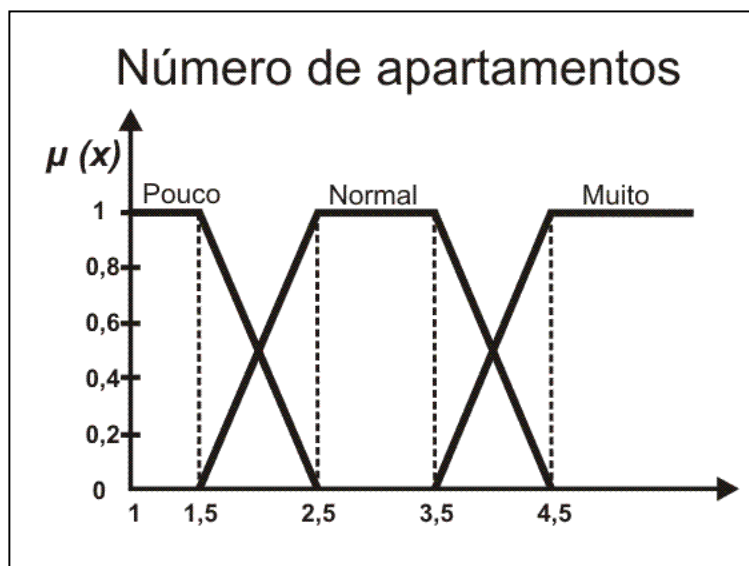


Figura 27. Representação gráfica do conjunto de entrada Número de Apartamentos

c) localização: refere-se a distância em metros (m) do imóvel em relação à um pólo de valorização, ou seja, qual a distância exata de um determinado imóvel de algum dos seguintes pontos do município de Criciúma: Praça do Congresso; Colégio Marista; Colégio Michel; Estádio Heriberto Hulse; Avenida Centenário e Parque Centenário, sendo considerada a menor distância que o imóvel se encontra de um destes pólos. A variável localização foi definida no intervalo de 0 à 1000 m. Os conjuntos pertencentes à esta variável são: perto (de 0 à 100 m),

médio (de 300 à 500 m) e longe (a partir de 1000m) , conforme ilustrado na Figura 28;

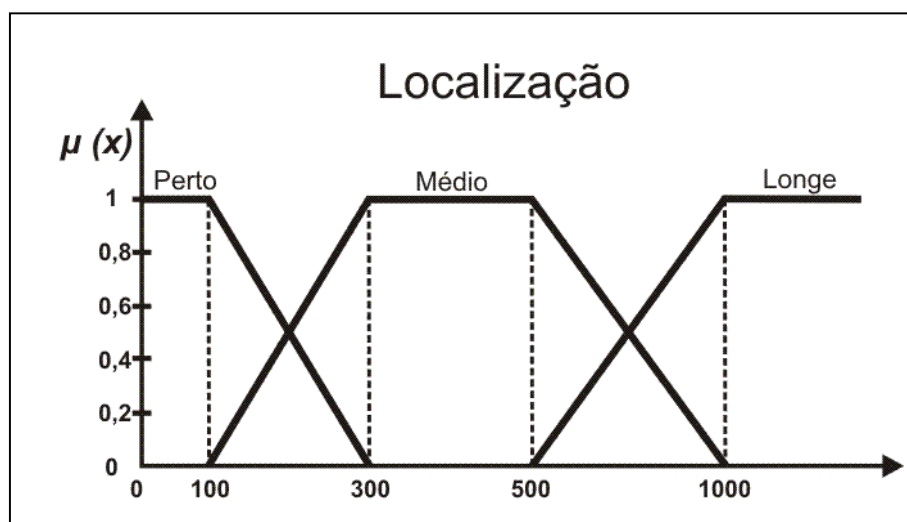


Figura 28. Representação gráfica do conjunto de entrada Localização

d) qualidade: estipula o padrão de qualidade do imóvel, com valores definidos em um intervalo entre 40 e 400 pontos. Os conjuntos pertencentes a esta variável são: inferior (de 40 a 100 pontos), média (de 200 a 300 pontos) e superior (a partir de 400 pontos) . Os valores atribuídos a esta variável foram definidos pela especialista como o produto entre o padrão do imóvel (1-Alto,2-Normal e 3-Baixo) e o índice de conservação (1-Péssimo, 2-Regular, 3-Bom e 4-Ótimo), determinado pela própria especialista para os imóveis presentes no conjunto de dados (Figura 29);

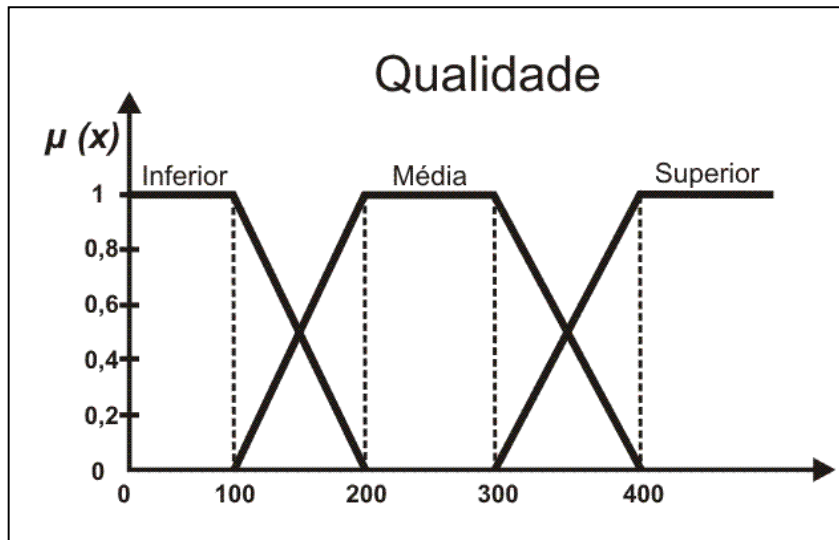


Figura 29. Representação gráfica do conjunto de entrada Qualidade

e) valor: apresenta a saída do sistema, com valores definidos em um intervalo entre R\$ 20.000,00 e R\$ 1.000.000,00. Os conjuntos pertencentes à esta variável são: baixo (de R\$20.000,00 à R\$40.000,00), médio (de R\$80.000,00 à R\$150.000,00) e alto (a partir de R\$200.000,00) , conforme ilustrado na Figura 30;

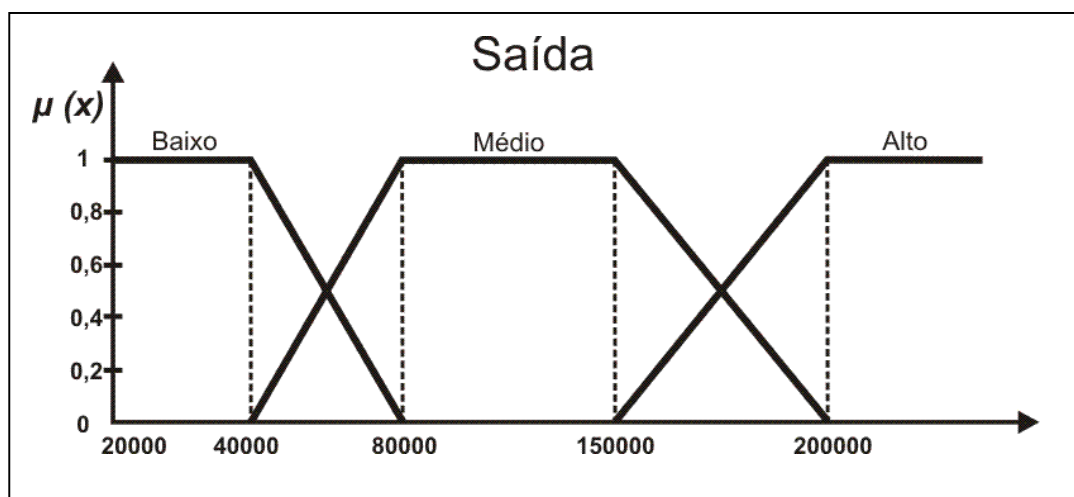


Figura 30. Representação gráfica do conjunto de saída Valor

A partir da definição dos conjuntos foram identificadas as regras de inferência a serem utilizadas para resolução do problema (Apêndice B), 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 *fuzzyficação*, *inferência* e *defuzzyficação*. A fim de exemplificar o processamento das etapas realizadas pela rede *neurofuzzy*, considerando-se os seguintes valores de entrada:

- a) área total = 164,38
- b) nº de dormitórios=2,5
- c) localização=108,94
- d) qualidade=280

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

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

- a) área total = 164,38

$$\mu_{\text{Pequeno}}(164,38) = 164,38 \geq 100 = 0$$

$$\mu_{\text{Médio}}(164,38) = 150 < 164,38 < 200 = (200-164,38)/(200-150) = 0,71$$

$$\mu_{\text{Grande}}(164,38) = 150 < 164,38 < 200 = (164,38-150)/(200-150) = 0,29$$

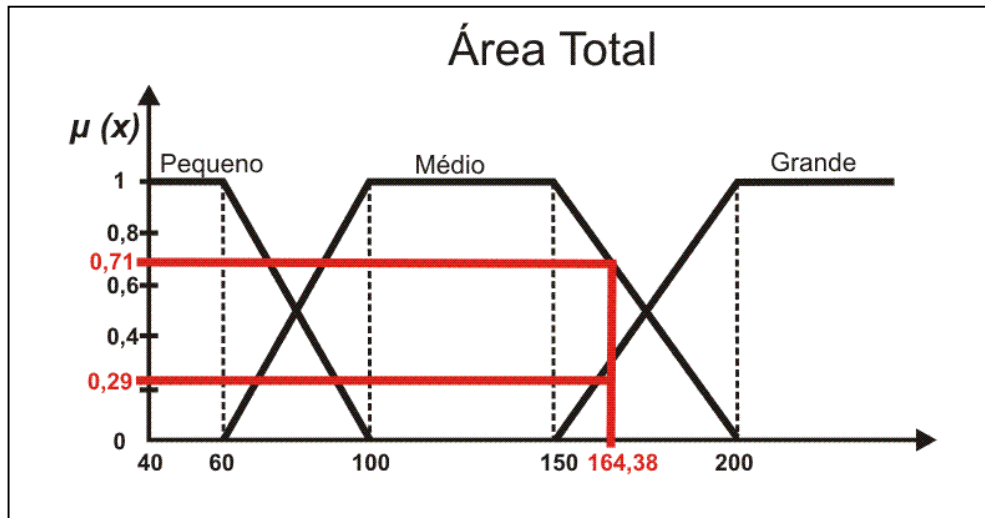


Figura 31. Representação gráfica dos valores de pertinência no conjunto Área total

b) nº de dormitórios = 2,5

$$\mu_{\text{Pouco}}(2,5) = 2,5 \geq 2,5 = 0$$

$$\mu_{\text{Normal}}(2,5) = 2,5 < 2,5 < 3,5 = 1$$

$$\mu_{\text{Muito}}(2,5) = 2,5 < 3,5 = 0$$

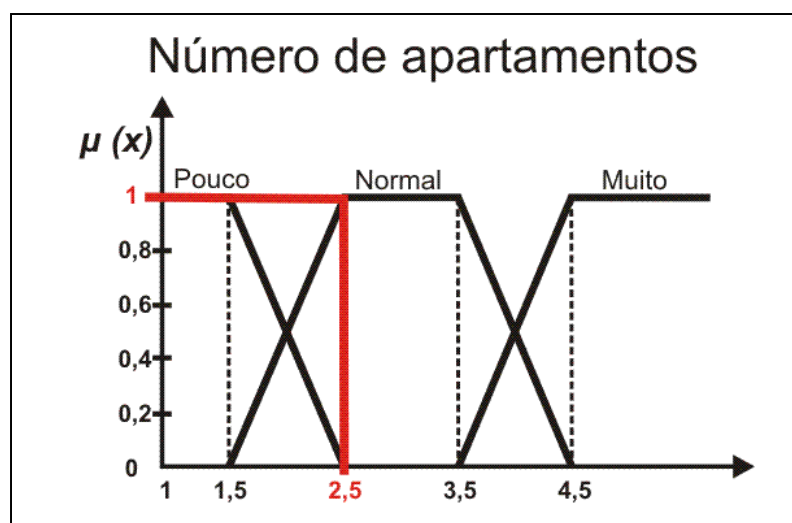


Figura 32. Representação gráfica dos valores de pertinência no conjunto Número de Apartamentos

c) localização = 108,94

$$\mu_{\text{Perto}}(108,94) = 100 < 108,94 < 300 = (300-108,94)/(300-100) = 0,96$$

$$\mu_{\text{Médio}}(108,94) = 100 < 108,94 < 300 = (108,94-100)/(300-100) = 0,04$$

$$\mu_{\text{Longe}}(108,94) = 108,94 < 500 = 0$$

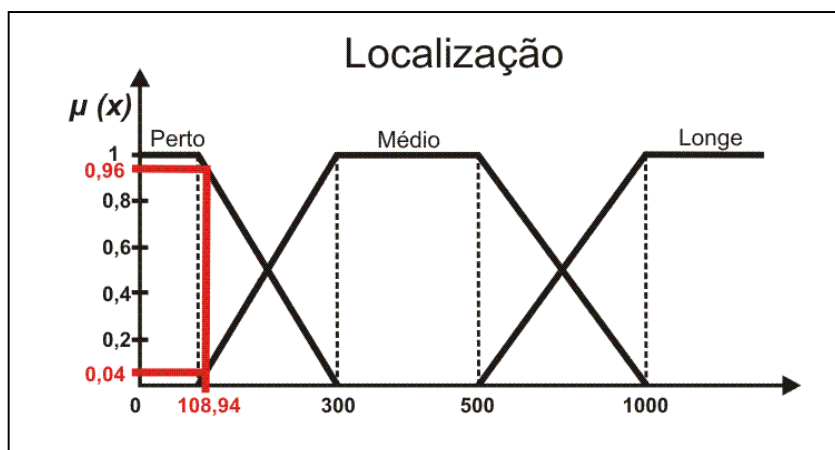


Figura 33. Representação gráfica dos valores de pertinência no conjunto Localização

d) qualidade = 280

$$\mu_{\text{Inferior}}(280) = 280 > 200 = 0$$

$$\mu_{\text{Média}}(280) = 200 < 280 < 300 = 1$$

$$\mu_{\text{Superior}}(280) = 280 < 300 = 0$$

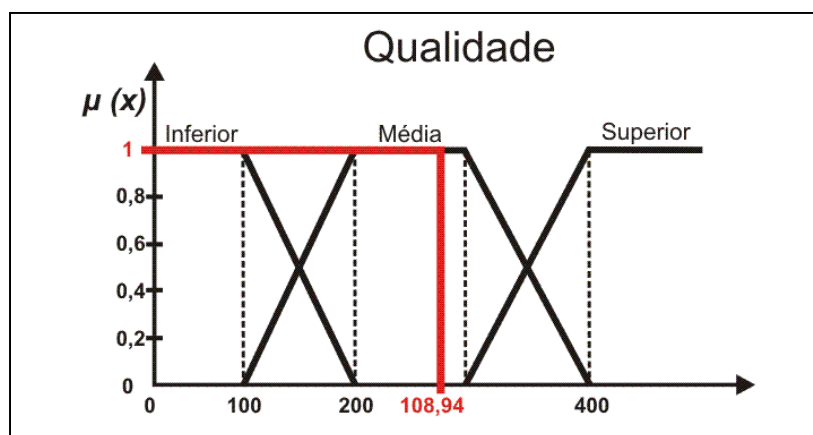


Figura 34. Representação gráfica dos valores de pertinência no conjunto Qualidade

Na próxima etapa realiza-se o processo de *inferência* por meio da interpretação das regras as quais os resultados podem ser aplicados, neste caso:

a) SE Área Total = Médio (**0,71**)

E Nro De Dormitórios = Normal (**1**)

E Localização = Perto (**0,96**)

E Qualidade = Média (**1**)

ENTÃO Valor = Médio;

b) SE Área Total = Médio (**0,71**)

E Nro De Dormitórios = Normal (**1**)

E Localização = Média (**0,04**)

E Qualidade = Média (**1**)

ENTÃO Valor = Médio;

c) SE Área Total = Grande (**0,29**)

E Nro De Dormitórios = Normal (**1**)

E Localização = Perto (**0,96**)

E Qualidade = Média (**1**)

ENTÃO Valor = Médio;

d) SE Área Total = Grande (**0,29**)

E Nro De Dormitórios = Normal (**1**)

E Localização = Média (**0,04**)

E Qualidade = Média (**1**)

ENTÃO Valor = Médio;

Dada esta interpretação, são extraídos 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 $n=1,2,3\dots N$. Desta forma, o processo de inferência resulta em:

a) $S1 = 0,71 \cap 1 \cap 0,96 \cap 1 = \text{Médio (0,71)}$

b) $S2 = 0,71 \cap 1 \cap 0,04 \cap 1 = \text{Médio (0,04)}$

c) $S3 = 0,29 \cap 1 \cap 0,96 \cap 1 = \text{Médio (0,29)}$

d) $S4 = 0,29 \cap 1 \cap 0,04 \cap 1 = \text{Médio (0,04)}$

Após a determinação dos valores mínimos das relações de interseção é realizada uma *normalização* dos valores gerados, tal que: $S'n = Sn / (S1 + S2 + S3 + \dots + Sn)$. Essa normalização apresenta os seguintes resultados:

a) $S'1 = 0,71 / (0,04+0,29+0,04) = \text{Médio (1,92)}$

b) $S'2 = 0,04 / (0,71+0,29+0,04) = \text{Médio (0,04)}$

c) $S'3 = 0,29 / (0,71+0,04+0,04) = \text{Médio (0,37)}$

d) $S'4 = 0,04 / (0,71+0,04+0,29) = \text{Médio (0,04)}$

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: $\text{Médio}(x) = \max [S'1, S'2, S'3, S'4]$. Desta forma, obtém-se:

$\text{Médio}(x) = \text{Médio (1,92)} \text{ OU } \text{Médio (0,04)} \text{ OU } \text{Médio (0,37)} \text{ OU } \text{Médio (0,04)} = 1,92$

O valor obtido na *união* é utilizado no processo de *defuzzyficação* pelo método *centróide* por meio do qual o centro é localizado, tal que $h_n = A(x) \cdot C_A$, 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, neste exemplo, resultando em apenas um conseqüente, neste caso, médio:

$h1 = 1,92 * 120.000 = 230.400$

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 $y = h_1 + h_2 + h_3 \dots h_n$. No caso do exemplo dado, apenas o conseqüente da saída “Valor=Médio” foi encontrado nas regras aplicadas, gerando o resultado:

$$y = h_1 = 230.400$$

7.2.4 Realização da Modelagem do Protótipo 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.

A Figura 35 ilustra o diagrama de caso de uso da *shell*, especificando as ações que o usuário poderá executar no protótipo, sendo:

- a) **definir parâmetros de criação da rede:** ao iniciar a criação de uma nova rede, o usuário tem a opção de definir todos os parâmetros referentes ao sistema *fuzzy* incorporado na rede neural. São eles:
 - definição das entradas: nesta primeira etapa, o usuário precisa estipular a quantidade de entradas, bem como seus nomes e valores inicial e final;
 - definição dos conjuntos das entradas: esta etapa requer que o usuário defina todos os conjuntos pertencentes as entradas,

associados as suas variáveis lingüísticas e os valores X1, X2, X3 e X4 para a criação da função trapezoidal;

- estipulação da saída: o usuário define o nome da saída da rede com seus valores inicial e final, e os conjuntos pertencentes, por meio das variáveis lingüísticas e valores da função trapezoidal;
- escolha das regras de inferência: permite que o usuário escolha as regras de inferência que serão utilizadas na rede *neurofuzzy*, produzidas pelo sistema a partir dos conjuntos definidos anteriormente, bem como seus conseqüentes.

- b) **visualizar conjuntos *fuzzy* utilizados na rede**: durante o processo de criação e mesmo após sua conclusão, ao iniciar a utilização de uma rede, é possível ao usuário realizar a visualização dos conjuntos de cada entrada e da saída, graficamente;
- c) **inserir dados na rede**: antes que a rede seja treinada, é necessário que o usuário insira no sistema os dados que serão utilizados no processo de aprendizagem, sendo que estes podem ser editados manualmente e salvos para uma utilização posterior, ou ainda, carregados de um arquivo de dados gravado em disco em formato de texto;
- d) **definir parâmetros de treinamento**: o usuário tem a opção de escolher qual será o número de épocas e o valor da taxa de treinamento para obtenção do aprendizado na rede;
- e) **visualizar relatório de treinamento**: após a conclusão do treinamento da rede *neurofuzzy*, o usuário pode visualizar o relatório de treinamento para efetuar a análise de cada etapa realizada durante o processo;

- f) **realizar testes:** antes que seja dado como concluído o treinamento, é possível que o usuário realize testes com dados ainda não apresentados à rede para que seja comprovada a eficácia no processo de aprendizado;
- g) **utilizar a rede:** quando a o processo de aprendizagem estiver concluído, bem como a etapa de testes realizada, o usuário poderá efetuar consultas referentes ao domínio da aplicação.

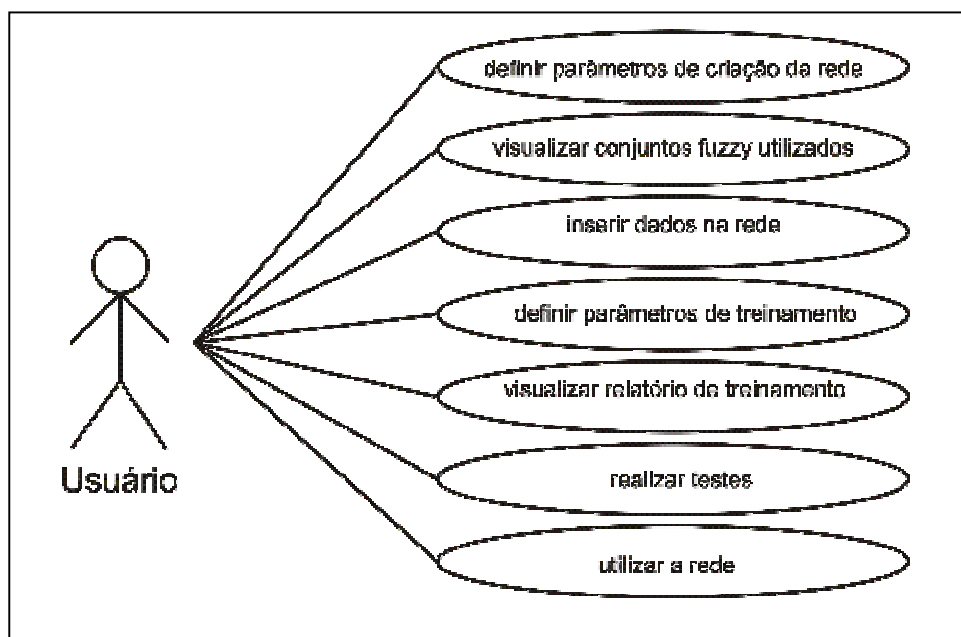


Figura 35. Diagrama de caso de uso das ações do usuário

Na Figura 36 encontra-se ilustrado o diagrama de caso de uso da *shell*, especificando as operações que o sistema executa sobre as ações do usuário, onde:

- a) **criar as regras de inferência:** o sistema realiza a criação de todas as regras de inferência possíveis a partir da combinação dos conjuntos das entradas definidos pelo usuário;
- b) **criar rede:** após o usuário definir as regras que serão utilizadas, o sistema cria a rede *neurofuzzy* baseado nos parâmetros fornecidos;

- c) **gerar a visualização da rede:** é realizada a geração da visualização da rede em tela para que o usuário possa identificar as camadas e seus respectivos nodos na estrutura;
- d) **gerar a visualização dos conjuntos *fuzzy* incorporados à rede:** o sistema realiza a geração da visualização gráfica dos conjuntos *fuzzy* das entradas e saída durante o processo de criação e/ou utilização de uma rede;
- e) **treinar a rede:** após a definição realizada pelo usuário do número de épocas e parâmetros de treinamento, o sistema executa o treinamento da rede;
- f) **emitir relatório de treinamento da rede:** durante o processo de treinamento, o sistema realiza a geração do relatório de treinamento, demonstrando cada etapa da inferência e aprendizado.

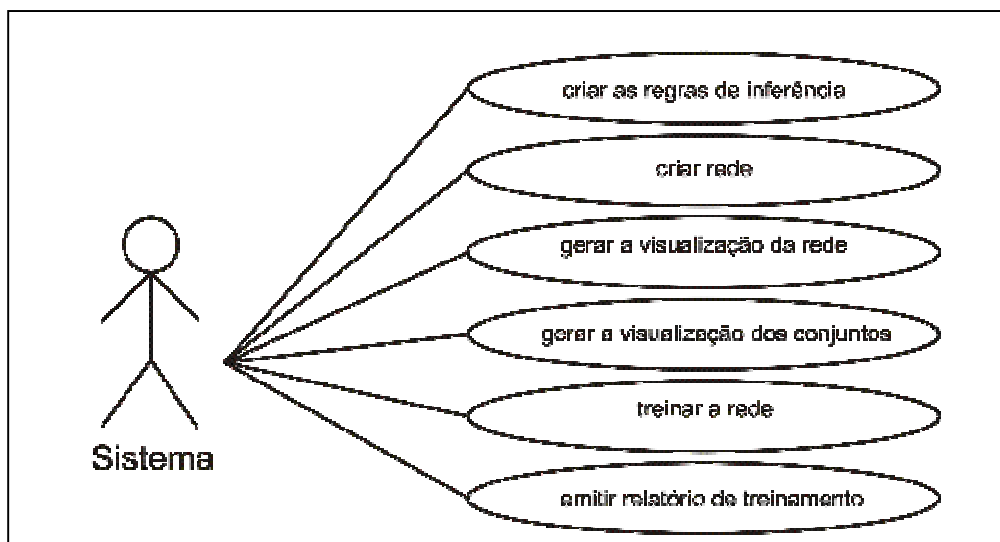


Figura 36. Diagrama de caso de uso das operações do sistema

O diagrama de atividades da *shell* ÍCARO, ilustrado na Figura 37, demonstra o aspecto dinâmico do sistema, consistindo das seguintes ações:

- a) **definir parâmetros de criação da rede:** deve-se definir todas as entradas e a saída, bem como seus limites, suas variáveis lingüísticas e valores da função trapezoidal, seguidos da escolha das regras de inferência e seus consequentes para incorporação da lógica *fuzzy* na rede neural;
- b) **criar rede:** a partir dos conjuntos *fuzzy* definidos pelo usuário, o sistema gera a rede *neurofuzzy*;
- c) **inserir dados para treinamento:** neste momento o usuário deve inserir no sistema os dados que serão utilizados para o treinamento da rede *neurofuzzy*;
- d) **definir parâmetros de treinamento:** após a inserção dos dados de treinamento, o usuário deve definir o número de épocas de treinamento e o valor da taxa de aprendizagem da rede para realização do treinamento;
- e) **treinar rede:** depois que o usuário realizou a definição dos parâmetros de treinamento, o sistema inicia o processo de aprendizado da rede *neurofuzzy*;
- f) **realizar testes:** o usuário fornece à rede novos dados, não utilizados no processo de treinamento, para que a rede seja testada;
- g) **processar os dados de teste:** o sistema processa o novo conjunto de dados inseridos pelo usuário na rede *neurofuzzy* para que, por meio dos valores obtidos, seu desempenho seja avaliado;
- h) **utilizar rede:** após a análise do desempenho realizada pelo usuário, a rede poderá ser utilizada;
- i) **processar dados inseridos na rede:** dado um conjunto de dados a rede processa o resultado.

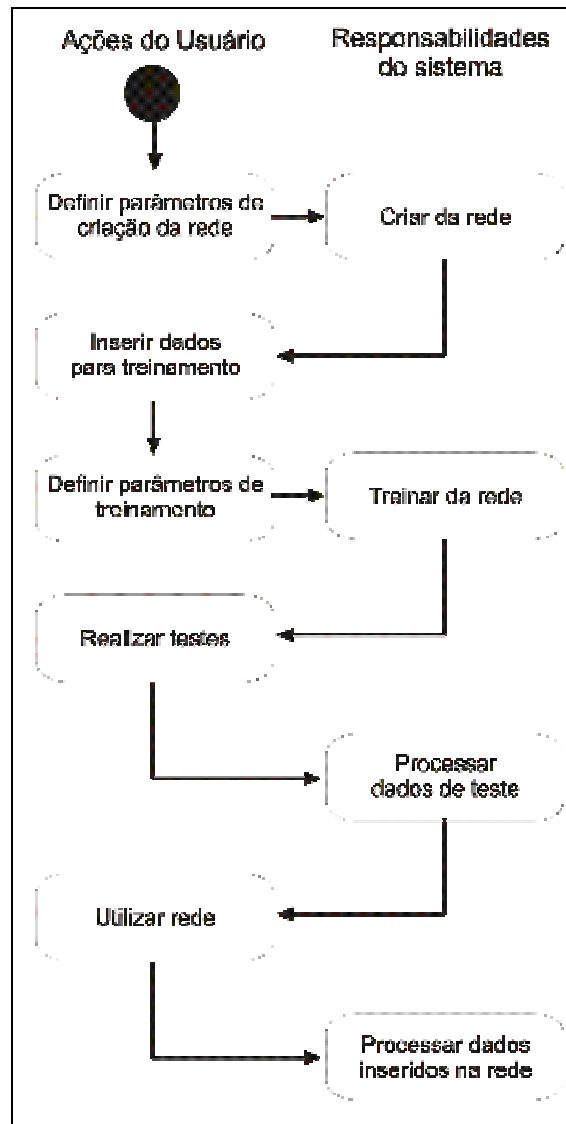


Figura 37. Diagrama de atividades da *shell* ÍCARO

7.2.5 Implementação do protótipo da *shell neurofuzzy*

O protótipo da *shell* ÍCARO foi implementado no ambiente de desenvolvimento *Borland Delphi 6.0 Professional* por meio da linguagem *Object Pascal*

por ser de fácil aprendizagem, possuir fontes para pesquisa de conteúdo abrangente e oferecer um ambiente de desenvolvimento simples e eficaz.

A implementação do protótipo de sistema pode ser definida em duas partes: *fuzzy* e neural.

7.2.5.1 Parte *Fuzzy*

Esta primeira parte da implementação da *shell* ÍCARO foi realizada 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 38.

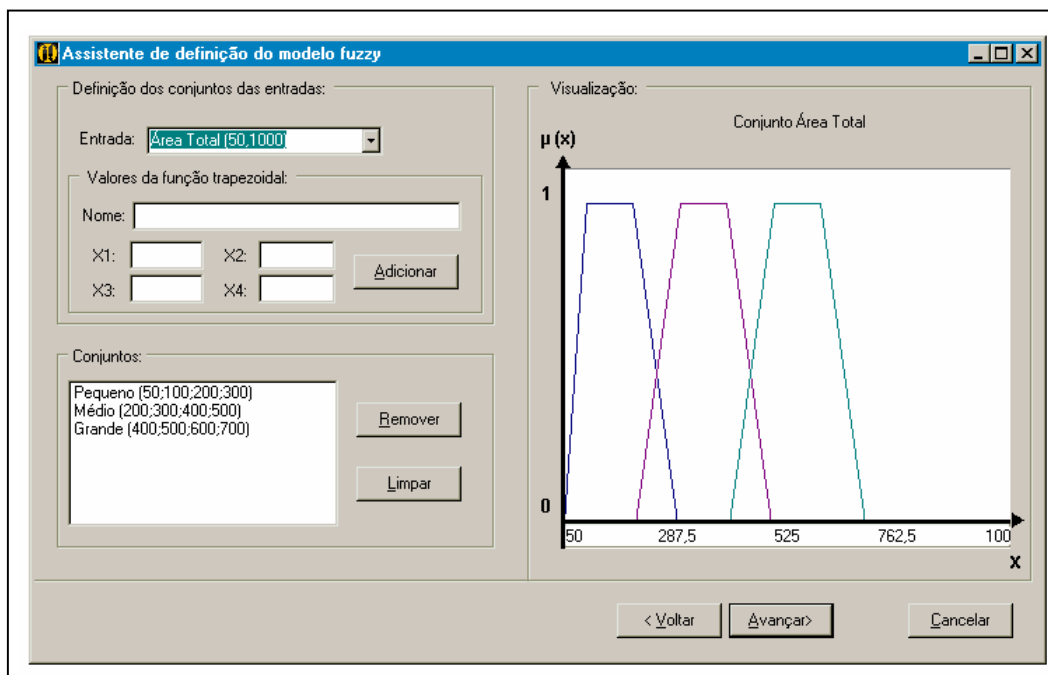


Figura 38. Definição dos valores das entradas por meio da função trapezoidal

Na próxima etapa de criação são definidos os conjuntos de saída utilizando o mesmo tipo de função, conforme ilustrado na Figura 39.

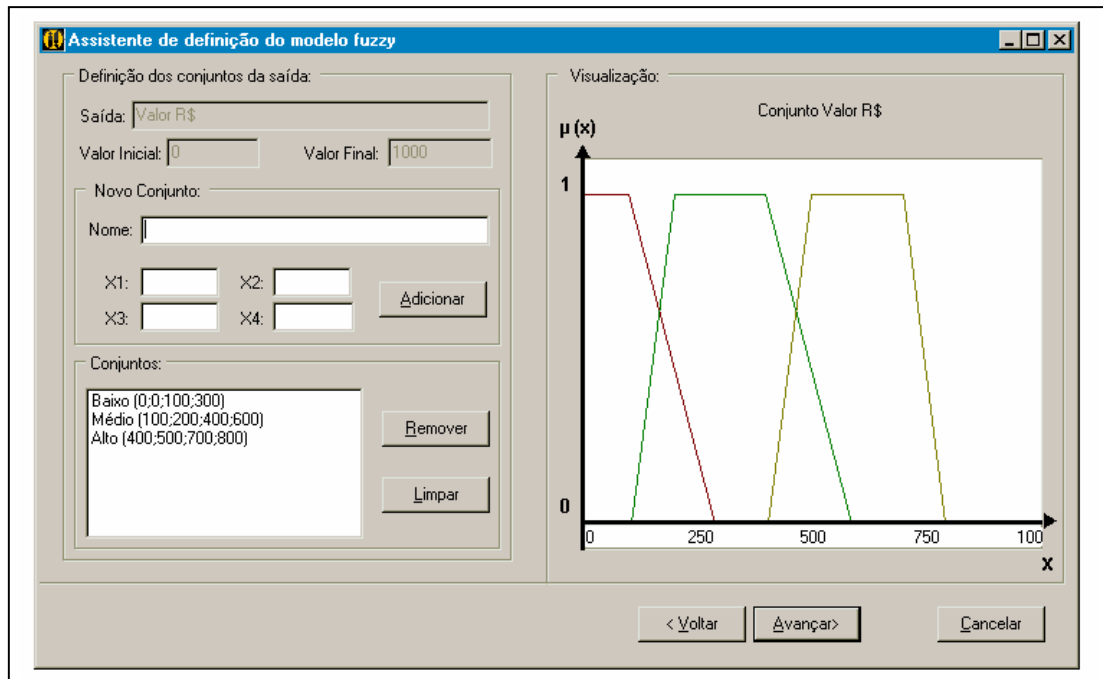


Figura 39. Definição dos valores de saída por meio da função trapezoidal

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 consequentes, de acordo com a Figura 40.

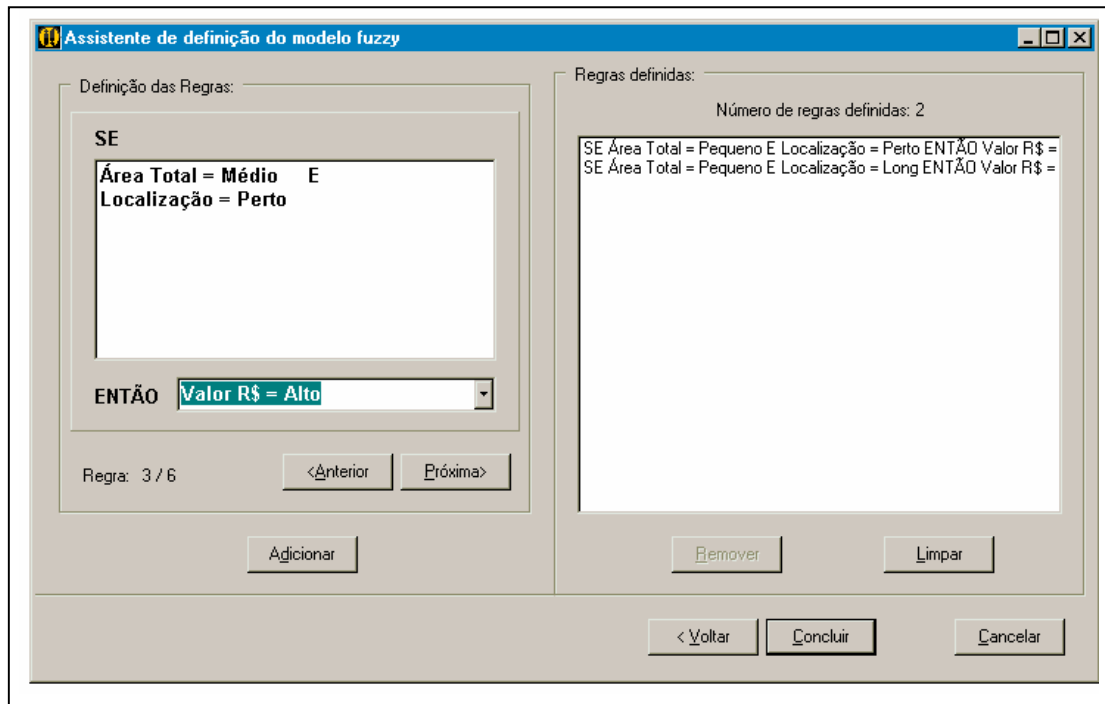


Figura 40. Definição das regras de inferência

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

7.2.5.2 Parte Neural

Nesta parte foi utilizada a estrutura de uma rede *multilayer perceptron* para desenvolvimento da rede. 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 41):

- 1) 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;

- 2) 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;
- 3) 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;
- 4) camada de normalização: os neurônios desta camada realizam o processo de normalização dos valores tal que, $S'_n = S_n / (S_1 + S_2 + S_3 + \dots + S_n)$, onde S' é o valor de saída do neurônio e S_n 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, conforme explicado na seção 4.3.3;
- 5) 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:

$h_n = S' \cdot n \cdot C_n$, onde h é a saída do neurônio, S' o maior valor recebido da camada de normalização e C o conseqüente da regra;

- 6) camada de saída: esta camada possui apenas um neurônio e fornece a saída do sistema. A saída é calculada pela soma dos valores de saída da camada anterior, tal que: $z = h_1 + h_2 + \dots + h_n$.

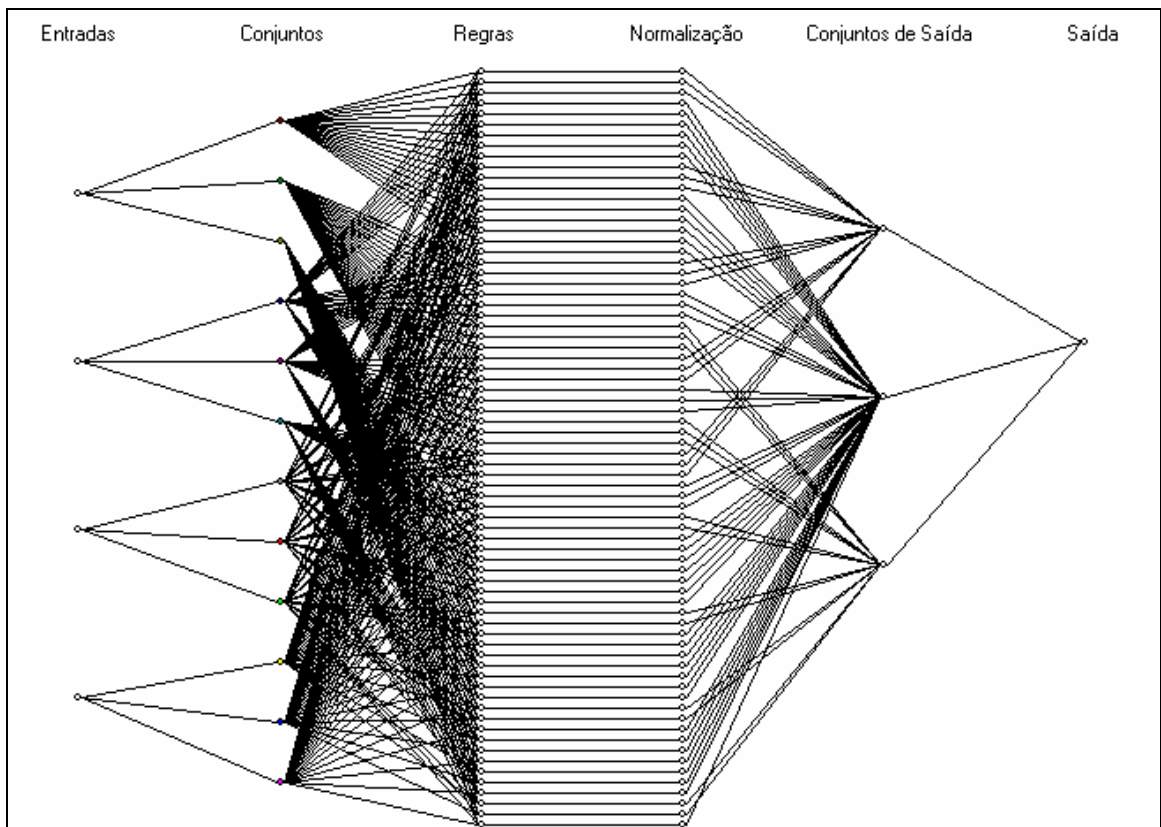


Figura 41. Arquitetura da rede *neurofuzzy* da *shell* ÍCARO

A Figura 41 representa a rede neural resultante da definição do modelo fuzzy pelo usuário. Enquanto, a Figura 42 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.

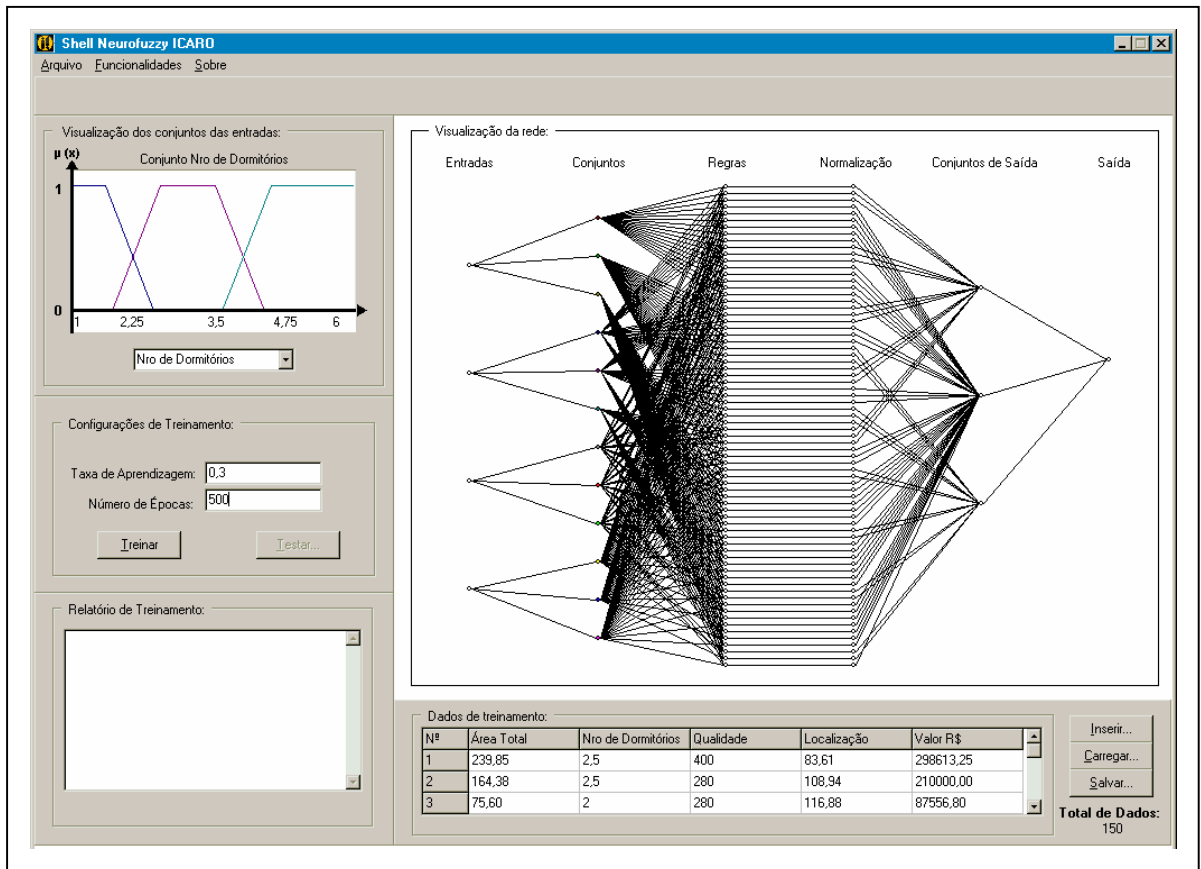


Figura 42. Tela principal da shell ÍCARO

7.2.6 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: $e_j(\mathbf{n}) = \mathbf{d}_j(\mathbf{n}) - \mathbf{y}_j(\mathbf{n})$, onde j representa o neurônio de saída, d_j o valor desejado e y_j o valor resultante da rede.

Utilizando-se o valor de saída $y = 230.400$, exemplificado na Seção 7.2.6 para as entradas área total = 164,38, número de dormitórios=2,5, localização=108,94 e qualidade=280, tendo como valor desejado de saída 210000, tem-se:

$$e_j(\mathbf{n}) = d_j(\mathbf{n}) - y_j(\mathbf{n}) = 210000 - 230400 = -20400$$

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:

$$e_j(\mathbf{n}) = (21000/120000) - (230400/120000) = 1,75 - 1,92 = -0,17$$

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

$$\Delta w_j = \eta \cdot (e_j(\mathbf{n}) \cdot y_j(\mathbf{n})) = 0,3 \cdot (-0,17 \cdot 1,92) = -0,09792$$

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 obtidos para realização desta pesquisa 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.

7.2.7 Realização de testes

Os testes da rede *neurofuzzy* na *shell* ÍCARO foram realizados pela utilização da base de dados mencionada anteriormente (Capítulo 7), que trata de valores reais

referentes à apartamentos em oferta no mercado imobiliário do município de Criciúma de janeiro à abril de 2006 para aquisição do conhecimento na parte *fuzzy* da rede neural e incorporação do conhecimento na fase de treinamento.

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 2) e normalização das saídas das regras (camada 4), por apresentares 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.

7.3 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 43).

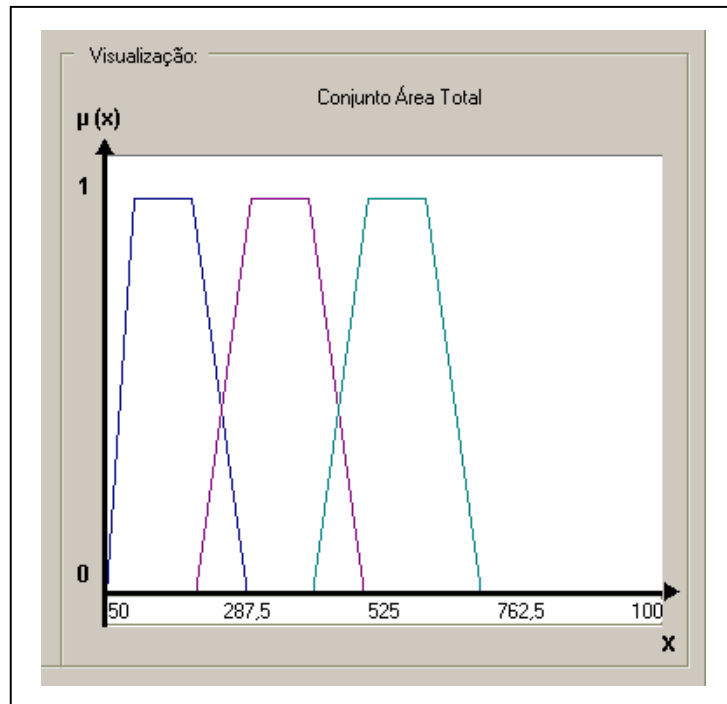


Figura 43. Representação do modelo *fuzzy* por meio da função trapezoidal

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 44) 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.

Dados de treinamento:					
Nº	Área Total	Nro de Dormitórios	Qualidade	Localização	Valor R\$
1	239,85	2,5	400	83,61	298613,25
2	164,38	2,5	280	108,94	210000,00
3	75,60	2	280	116,88	87556,80

Total de Dados:
150

Figura 44. Inserção de dados para treinamento na *shell* ÍCARO

Na Figura 45 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.

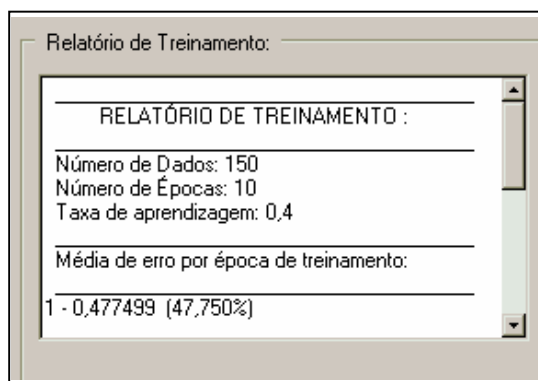


Figura 45. Relatório de treinamento da *shell* ÍCARO

Na Shell Ícaro desenvolveu-se um *help* por meio da ferramenta HelpMaker que explica sobre as redes neurofuzzy, a arquitetura implementada, o algoritmo backpropagation, bem como orientações ao usuário sobre o funcionamento da Shell.

O sistema gerado teve seus resultados originados de testes realizados na *shell* ÍCARO por meio do treinamento da rede. A rede foi criada no sistema a partir da geração dos conjuntos *fuzzy* definida junto com a especialista em avaliação de imóveis Evelize Chemale Zancan, bem como a definição das regras de inferência utilizadas (Apêndice B).

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 na época 150, com resultados conforme ilustrado na Figura 46.

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 46. Resultados dos testes realizados 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. Os testes realizados são apresentados na Figura 47:

Taxa de Aprendizado	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 47. Resultados dos testes realizados na *shell* ÍCARO

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

CONCLUSÃO

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.

A utilização da lógica *fuzzy* nesta pesquisa, permitiu a aquisição prévia do conhecimento especialista, além de diminuir a quantidade de processamento necessário para incorporação do conhecimento, bem como o tratamento da incerteza por imprecisão em redes neurais.

Por meio desta pesquisa, foi possível ampliar os conhecimentos referentes às duas técnicas da IA abordadas, bem como a sua utilização em sistemas híbridos *neurofuzzy*, sendo este um dos objetivos propostos. Dentre outros objetivos, destaca-se a demonstração matemática da modelagem *fuzzy* na rede neural que, devido ao fato de haver sido realizado o desenvolvimento de uma *shell*, foi possível a melhor compreensão do modelo de arquitetura apresentado na aplicação.

Além disso, o propósito inicial desta pesquisa foi o desenvolvimento de um protótipo de sistema de arquitetura híbrida e não de uma *shell*. Porém, houve uma grande dificuldade na compreensão e solução de determinados problemas quanto à utilização de aplicações que desenvolvessem este tipo de sistema, incentivando assim, a construção de uma ferramenta que disponibilizasse os recursos necessários para geração do sistema de forma que, seja facilmente utilizado para outras aplicações.

Outra dificuldade encontrada no decorrer da pesquisa, 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 que os outros tipos, 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.

O protótipo da *shell* implementado utiliza apenas um modo de inferência e função de pertinência, desta forma, além da utilização para geração de outros modelos de sistema, poderão ser implementados futuramente para utilização de outros tipos de função e arquitetura.

REFERÊNCIAS

- AURÉLIO, Marco; LOPES, Carlos Henrique ;VELLASCO, Marley Maria B. R..**Descoberta de Conhecimento e Mineração de Dados**. 1999. Apostila (ICA – Núcleo de Pesquisa em Inteligência Computacional Aplicada) – Curso de Engenharia Elétrica, PUC, Rio de Janeiro, 1999.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14653-2: Avaliação de Bens – Imóveis Urbanos**. Rio de Janeiro, 2004.
- AZEVEDO, Fernando Mendes de; BRASIL, Lourdes Mattos; OLIVEIRA, Roberto C. Limão de. **Redes Neurais com Aplicações em Controle e em Sistemas Especialistas**. Florianópolis: Bookstore, 2000.
- ABUNAHMAN, Sérgio. **Curso básico de engenharia legal de avaliações**. São Paulo: Pini, 1999.
- BARONE, Dante. **Sociedades artificiais: a nova fronteira da inteligência nas máquinas**. Porto Alegre: Bookman, 2003.
- BITTENCOURT, Guilherme. **Inteligência Artificial: Ferramentas e Teorias**. Florianópolis: Ed. da UFSC, 2001.
- BRAGA, Antônio de Pádua; CARVALHO, André C. P. L. Ferreira; LUDEMIR, Teresa Bernarda. **Redes Neurais Artificiais: Teoria e Aplicações**. Rio de Janeiro: LTC, 2000.
- BRONDINO, Nair Cristina Margarido. **Estudo da influência da acessibilidade no valor de lotes urbanos através do uso de redes neurais**. 1999. Tese (Doutorado em Engenharia Civil – Transportes) – Programa de Doutorado em Engenharia Civil – Transportes, Escola de Engenharia Civil da Universidade de São Paulo, São Carlos.
- DANTAS, Rubens Alves. **Engenharia de Avaliações: Uma introdução à Metodologia Científica**. São Paulo: PINI, 1998.
- FULLÉR, R. *Neural Fuzzy Systems*. 1995. Disponível em: <<http://citeseer.nj.nec.com/64350.html>> Acesso em: 03 jan. 2005.
- GUEDES, Jackson Carvalho. Aplicação de Redes Neurais na Avaliação de Bens. In: SEMINÁRIO INTERNACIONAL DA LARES, 1, 1999, São Paulo. **Anais...** São Paulo, 1999.
- HAYKIN, Simon. **Redes neurais: princípios e prática**. Tradução: Paulo Martins Engel. 2. ed. Porto Alegre: Bookman, 2001.

HUAMANÍ, Ivete R. Luna. **Redes Neurofuzzy aplicadas em Identificação e Controle de Sistemas**. 2003. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, Campinas.

KLIR, George J.; YUAN, Bo. **Fuzzy sets and fuzzy logic: theory and applications**. New Jersey: Prentice Hall, 1995.

LUGER, George F. **Inteligência artificial: estruturas e estratégias para a resolução de problemas complexos**. 4. ed Porto Alegre: Bookman, 2004.

NAUCK, Detlef, KRUSE, Rudolf. **Choosing Appropriate Neuro-Fuzzy Models**. In: FOURTH EUROPEAN CONGRESS ON INTELLIGENT TECHNIQUES AND SOFT COMPUTING, 1, 1994, Aachen. **Anais...** Aachen, 1994.

PELLI NETO, Antônio; BRAGA, Antônio de Pádua. Avaliação De Imóveis Urbanos Com Utilização De Sistemas Nebulosos (Redes Neuro-Fuzzy) e Redes Neurais Artificiais. In: CONGRESO PANAMERICANO DE VALUACIÓN CARTAGENA, 21, 2004, Colombia. **Anais...** Colombia, 2004.

PELLI NETO, Antônio; BRAGA, Antônio de Pádua. Redes Neurais Artificiais: aplicação e comparação dos resultados com a regressão linear nas avaliações de imóveis urbanos. In: CONGRESO INTERNACIONAL DE VALUACIÓN Y CATASTRO,1, CONGRESO VENEZOLANO DE VALUACIÓN Y CATASTRO,10, CONCURSO INTERNACIONAL DE VALUACION Y CATASTRO, 4, 2005, Caracas. **Anais...** Caracas, 2005.

PELLI NETO, Antônio; ZÁRATE, Luis Henrique. Avaliação De Imóveis Urbanos Com Utilização De Redes Neurais Artificiais. In: CONGRESO BRASILEIRO DE ENGENHARIA DE AVALIAÇÕES E PERÍCIAS, 12, 2003, Belo Horizonte. **Anais...** Belo Horizonte, 2003.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

RABUSKE, Renato Antônio. **Inteligência Artificial**. Florianópolis: UFSC, 1995.

RAMOS, Patrícia Gouveia. **Uma investigação das Redes Neurofuzzy aplicadas à Mineração de Dados**. 1999. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro de Ciências Exatas, Universidade Federal de Pernambuco, Recife.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. São Paulo: Manole, 2003.

RICH, Elaine. **Inteligência artificial**. 2.ed. São Paulo: Makron Books, 1994.

RODARTE, Caroline A.; SANTOS, Leandro Nunes. **Sistemas Neurofuzzy: Conceitos e Aplicações**. 2005.

Artigo– Curso de Ciência da Computação, Universidade Federal da Bahia, Salvador, 2005.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2004.

SANTOS, Davi Trindade dos. **Sistemas Inteligentes Híbridos: Como Combinar Redes Neurais Artificiais, Lógica Nebulosa e Algoritmos Genéticos**. 2004. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro de Ciências Exatas, Universidade Estadual de Londrina, Londrina.

VELLASCO, Marley Maria B. R..**Lógica Nebulosa**. 1999. Apostila (ICA – Núcleo de Pesquisa em Inteligência Computacional Aplicada) – Curso de Engenharia Elétrica, PUC, Rio de Janeiro, 1999.

VELLASCO, Marley Maria B. R..**Sistemas Neurofuzzy**. 1999. Apostila (ICA – Núcleo de Pesquisa em Inteligência Computacional Aplicada) – Curso de Engenharia Elétrica, PUC, Rio de Janeiro, 1999.

VIALI, Lori. **Um modelo neurodifuso de multiagentes para apoio ao ensino de disciplinas de Ciências Exatas**. 1999. Tese (Programa de Doutorado) – Curso de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

WANG, Li-Xin. **A course in fuzzy systems and control**. London: Prentice Hall, 1997.

ZANCAN, Evelise Chemale. **Avaliação de Imóveis em Massa para Efeitos de Tributos Municipais**. Florianópolis: Rocha, 1996.

APÊNDICE A – CONTRATO DE AUTORIZAÇÃO DE USO

Eu, **Evelize Chemale Zancan**, coordenadora do curso de Engenharia Civil / UNESC, venho por meio deste autorizar a acadêmica da 8ª fase do curso de Ciência da Computação / UNESC **Niria Borges Ferreira**, matriculada no 2º semestre de 2005, a utilizar a base de dados referente à tipologia de apartamentos do município de Criciúma/SC no treinamento de uma rede *neurofuzzy* que será desenvolvida em seu trabalho de conclusão de curso.

Criciúma, 15 de setembro de 2005.

Evelise Chemale Zancan

Coordenadora do Curso de Engenharia Civil/UNESC

APÊNDICE B – BASE DE REGRAS

01	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Baixo
02	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
03	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Superior ENTÃO Valor = Médio
04	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Baixo
05	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
06	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Superior ENTÃO Valor = Médio
07	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Baixo
08	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Média ENTÃO Valor = Médio
09	SE Area Total = Pequeno E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Superior ENTÃO Valor = Médio
10	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Baixo
11	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
12	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Superior ENTÃO Valor = Médio
13	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Baixo
14	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
15	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Superior ENTÃO Valor = Médio
16	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Baixo
17	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Média ENTÃO Valor = Médio
18	SE Area Total = Pequeno E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Superior ENTÃO Valor = Médio
19	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Baixo
20	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
21	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Superior ENTÃO Valor = Alto
22	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Baixo
23	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
24	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Superior ENTÃO Valor = Alto
25	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Baixo

26	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Média ENTÃO Valor = Médio
27	SE Area Total = Médio E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Superior ENTÃO Valor = Alto
28	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Baixo
29	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
30	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Superior ENTÃO Valor = Alto
31	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Baixo
32	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
33	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Superior ENTÃO Valor = Alto
34	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Baixo
35	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Média ENTÃO Valor = Médio
36	SE Area Total = Médio E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Superior ENTÃO Valor = Alto
37	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Baixo
38	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
39	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Perto E Qualidade = Superior ENTÃO Valor = Alto
40	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Baixo
41	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
42	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Médio E Qualidade = Superior ENTÃO Valor = Alto
43	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Baixo
44	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Longe E Qualidade = Média ENTÃO Valor = Médio
45	SE Area Total = Médio E Nro de Dormitórios = Muito E Localização = Longe E Qualidade = Superior ENTÃO Valor = Médio
46	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Médio
47	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
48	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Perto E Qualidade = Superior ENTÃO Valor = Alto
49	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Médio
50	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
51	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Médio E Qualidade = Superior ENTÃO Valor = Alto
52	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Médio
53	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Média

	ENTÃO Valor = Médio
54	SE Area Total = Grande E Nro de Dormitórios = Pouco E Localização = Longe E Qualidade = Superior ENTÃO Valor = Médio
55	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Médio
56	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
57	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Perto E Qualidade = Superior ENTÃO Valor = Alto
58	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Médio
59	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
60	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Médio E Qualidade = Superior ENTÃO Valor = Alto
61	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Médio
62	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Média ENTÃO Valor = Médio
63	SE Area Total = Grande E Nro de Dormitórios = Normal E Localização = Longe E Qualidade = Superior ENTÃO Valor = Médio
64	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Perto E Qualidade = Inferior ENTÃO Valor = Médio
65	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Perto E Qualidade = Média ENTÃO Valor = Médio
66	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Perto E Qualidade = Superior ENTÃO Valor = Alto
67	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Médio E Qualidade = Inferior ENTÃO Valor = Médio
68	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Médio E Qualidade = Média ENTÃO Valor = Médio
69	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Médio E Qualidade = Superior ENTÃO Valor = Alto
70	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Longe E Qualidade = Inferior ENTÃO Valor = Médio
71	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Longe E Qualidade = Média ENTÃO Valor = Médio
72	SE Area Total = Grande E Nro de Dormitórios = Muito E Localização = Longe E Qualidade = Superior ENTÃO Valor = Médio