UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANA PAULA SCOTTI

O MÉTODO DE REDES NEURAIS COM FUNÇÃO DE ATIVAÇÃO DE BASE RADIAL PARA A TAREFA DE CLASSIFICAÇÃO NA *SHELL* ORION *DATA MINING ENGINE*.

CRICIÚMA, JUNHO DE 2010

ANA PAULA SCOTTI

O MÉTODO DE REDES NEURAIS COM FUNÇÃO DE ATIVAÇÃO DE BASE RADIAL PARA A TAREFA DE CLASSIFICAÇÃO NA *SHELL* ORION *DATA MINING ENGINE*.

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

Orientadora: Profa. MSc. Merisandra Côrtes de Mattos

CRICIÚMA, JUNHO DE 2010

ANA PAULA SCOTTI

O Método de Redes Neurais com Função de Ativação de Base Radial para a Tarefa de Classificação na *Shell* Orion *Data Mining Engine*

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

Profa. MSc. Merisandra Côrtes de Mattos (UNESC)
Orientador

Prof. MSc. Cristian Cechinel (Universidade Federal do Pampa, Bagé/RS)

Prof. MSc. Kristian Madeira (UNESC)

Aos meus pais, Mauro e Marlene, meus irmãos Maurício e Patrícia e meus queridos amigos pelo apoio concedido.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, por todo amor, carinho e apoio concedidos, pelos sacrifícios em razão da minha educação ou simplesmente por me aturarem com paciência..

Agradeço também:

Ao meu amigo Ademar pelo auxilio concedido no decorrer desta pesquisa e pela amizade ao longo da graduação e as minhas queridas amigas Camily, Joana e Bruna por estarem ao meu lado nos momentos difíceis e me proporcionarem momentos de diversão.

A todos os meus colegas de graduação, por compartilharem suas experiências e momentos de diversão e descontração durante todo o curso.

Ao professor Dr. Guilherme de Alencar Barreto por gentilmente esclarecer muitas dúvidas no decorrer desta pesquisa.

A todos os professores que passaram por minha vida, especialmente a grande incentivadora desta pesquisa, minha professora e orientadora Merisandra que contribuiu com seu conhecimento e amizade.

Enfim, a todos que de alguma forma contribuíram com meu crescimento profissional e pessoal, e com o desenvolvimento desta pesquisa.

“A vitória pertence ao mais perseverante”.

(Napoleão Bonaparte)

RESUMO

O surgimento de novas tecnologias para análise de informações e extração de conhecimento, iniciou-se devido à formação de grandes bases de dados ocasionada pelos avanços computacionais no que se refere ao armazenamento de dados. Neste contexto, destacam-se as técnicas de *data mining,* que são aplicadas no processo de descoberta do conhecimento por meio de ferramentas computacionais e consideradas a principal etapa deste processo. Estas ferramentas são em sua grande maioria comerciais e de alto custo, por este motivo, encontra-se em desenvolvimento pelo Grupo de Pesquisa em Inteligência Computacional da UNESC a *Shell* Orion *Data Mining Engine,* propondo disponibilizar uma ferramenta gratuita que implemente os diversos métodos de *data mining*. Esta pesquisa tem como objetivo ampliar as funcionalidades já existentes na *Shell* Orion fundamentando-se na modelagem matemática e implementação de uma rede neural com função de ativação de base radial para a tarefa de classificação, que consiste em identificar registros de uma base de dados com características comuns entre si e relacioná-los a uma determinada classe. A fim de analisar o funcionamento do algoritmo, no final desta pesquisa foram realizados testes com bases de dados que comprovaram o funcionamento correto do módulo desenvolvido.

Palavras-chave: Inteligência Artificial; Data Mining; Classificação; Redes Neurais Artificiais; Radial Basis Function.

ABSTRACT

The appearing of new technologies for information analysis and knowledge discovery starts due to the increase of databases caused by the improvements of computing and storage models. In this context, it is distinguished the data mining techniques that are applied on the knowledge discovery process through computing tools, and considered the main stage of this process. These tools are in its majority commercial, therefore, it is found in development by the Computational Intelligence Research Group at UNESC the project called Shell Orion Data Mining Engine that proposes to release a free tool that implements different methods of data mining. The objective of this research is to increase the features of this tool founded on the mathematical demonstration and implementation of a radial basis function neural network for the classification task that has as objective to identify the similar elements in databases and relate these records to some class. In order to verify the implemented module, at the end of this research some tests with different databases has been carried through that proves the satisfactory operation of the module developed.

Palavras-chave: Artificial Intelligence; Data Mining; Classification; Artificial Neural Networks; Radial Basis Function.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

CART	*Classification and Regression Trees*
DM	*Data Mining*
GK	*Gustafsson-Kessel*
GPL	*General Public License*
IA	Inteligência Artificial
IEEE	*Institute of Electrical and Eletronics Engineers*
ID3	*Iterative Dichotomiser 3*
JDBC	*Java Data Base Conectivity*
KDD	*Knowledge Discovery in Databases*
LMS	*Last Mean Square*
MLP	*Multilayer Perceptron*
RBF	*Radial Basis Function*
RNA	Redes Neurais Artificiais
SQL	*Structured QueryLanguage*
TCC	Trabalho de Conclusão de Curso
UFRN	Universidade Federal do Rio Grande do Norte
UFSC	Universidade Federal de Santa Catarina
UML	*Unified Modeling Language*
UNESC	Universidade do Extremo Sul Catarinense

SUMÁRIO

1 INTRODUÇÃO

Com os avanços tecnológicos, o armazenamento de informações tornou-se um processo fácil, no entanto, analisar e extrair conhecimento útil de grandes bases de dados tornou-se um problema complexo para as organizações contemporâneas. A fim de facilitar esta análise foram desenvolvidas técnicas de *data mining*, capazes de identificar padrões e ralações entre os dados.

O *data mining* utiliza conceitos de diferentes áreas como inteligência artificial, estatística, aprendizagem de máquina e banco de dados. Esse processo envolve a aplicação de algoritmos específicos, definidos conforme o problema e as características dos dados. Pode-se encontrar variadas tarefas de *data mining* disponíveis em ferramentas exclusivas para descoberta de conhecimento em bases de dados (GOLDSCHIMDT; PASSOS, 2005).

O Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da UNESC mantém em desenvolvimento o projeto de uma ferramenta específica para *data mining* chamada *Shell* Orion *Data Mining Engine*. A ferramenta é organizada em módulos referentes a cada tarefa e já possui implementados os algoritmos A Priori para associação; ID3, CART e C4.5 para classificação e K-means, Kohonen, Gustafson-Kessel e Gath-Geva para clusterização.

A classificação é uma das tarefas mais populares no processo de descoberta de conhecimento em bases de dados. Classificar consiste em encontrar propriedades comuns em um conjunto de registros de uma base de dados e relacioná-los a um único rótulo categórico pré-definido, denominado classe (HAN; KAMBER, 2001, tradução nossa).

Em geral, os algoritmos de classificação têm problemas de desempenho no reconhecimento de padrões em grandes quantidades de dados, por possuírem poder de predição limitado (OLSON; DELEN, 2008, tradução nossa). A capacidade de se adaptar a

problemas específicos por meio do treinamento da rede utilizando exemplos, e produzir respostas coerentes para dados não-conhecidos, faz das redes neurais um paradigma computacional muito utilizado em problemas de classificação de padrões de grande dimensão (BRAGA; CARVALHO; LUDERMIR, 2000).

De acordo com Bishop (1995) as redes neurais artificiais são uma abordagem alternativa para resolver o problema de classificação de padrões em bases de grandes dimensões. Neste contexto, esta pesquisa consiste no desenvolvimento da tarefa de classificação pelo método de redes neurais com função de ativação de base radial na *Shell* Orion *Data Mining Engine*.

1.1 OBJETIVO GERAL

Desenvolver o método de Redes Neurais com Função de Ativação de Base Radial para a tarefa de classificação da *Shell Orion Data Mining Engine.*

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa são:

a) compreender os conceitos de *data mining* e da tarefa de classificação;
b) entender o método de redes neurais e o funcionamento das redes neurais com função de ativação funções de ativação de base radial;
c) aplicar o algoritmo de função de ativação de base radial na tarefa de classificação de *data mining;*
d) demonstrar matematicamente o funcionamento do algoritmo de função de ativação de base radial;

e) testar o funcionamento do algoritmo de função de ativação de base radial na tarefa de classificação da *Shell* Orion *Data Mining Engine;*

f) analisar o desempenho do módulo desenvolvido.

1.3 JUSTIFICATIVA

O *data mining* explora grandes quantidades de dados em busca de padrões aplicando processos de generalização, ou seja, mantém somente dados genéricos o que permite prever a ocorrência de padrões (GOLDSCHIMDT; PASSOS, 2005).

A grande competitividade entre as organizações aumentou o interesse em adquirir ferramentas capazes de auxiliar na análise de informações e na tomada de decisões, destacando-se neste contexto, as s*hells* de *data mining.*

Devido ao fato de que grande parte dessas ferramentas são comerciais, o projeto acadêmico da *Shell* Orion *Data Mining Engine* propõe o desenvolvimento de uma ferramenta gratuita, tornando sua aquisição viável para as organizações. Esta pesquisa objetiva ampliar as funcionalidades da *Shell* Orion*,* desenvolvendo a tarefa de classificação por meio de redes neurais com função de ativação de base radial.

De acordo com Carvalho (2001) a classificação é uma das tarefas mais utilizadas no *data mining*. Até o momento, a *Shell* Orion possui três algoritmos de classificação implementados, ID3, CART e C4.5, todos utilizando o método de árvores de decisão[1].

O problema da classificação resume-se em particionar os dados de entrada em classes, de acordo com suas características. Na primeira etapa de um algoritmo classificador ocorre o processo de aprendizagem onde um modelo pré-definido de dados é aplicado, e

[1] Método utilizado para dividir grande volume de dados em subconjuntos por meio de regras de decisão (BERRY; LINOFF, 2004).

posteriormente utilizado pelo algoritmo com o objetivo de predizer a classe de dados não pertencentes ao conjunto aplicado como exemplo (OLSON; DELEN, 2008, tradução nossa).

Redes neurais tornaram-se um método atrativo para tarefas de *data mining* por apresentarem desempenho superior aos modelos convencionais (HAYKIN, 2001). Segundo Pal e Mitra (2004) essa arquitetura é bastante adequada para a tarefa de classificação devido a sua capacidade adaptativa, robustez e velocidade para predição.

As Redes Neurais com Função de Base Radial, em inglês Radial Basis Function (RBF), são uma variação das *Multi Layer Perceptron* (MLP), ou Redes Neurais de Múltiplas Camadas, apresentando como principal vantagem a capacidade de aprender rapidamente padrões complexos e tendências presentes nos dados, ganhando desempenho em relação a redes MLP e apresentando igual poder preditivo (BISHOP, 1995, tradução nossa).

O aprendizado de uma rede neural de função de base radial é considerado híbrido (combina-se um método supervisionado com um não supervisionado). Essa arquitetura destaca-se entre os modelos de redes neurais devido à simplicidade do processo de treinamento e à eficiência computacional (AZEVEDO; BRASIL; OLIVEIRA, 2000).

1.4 ESTRUTURA DO TRABALHO

Este trabalho foi divido em capítulos sendo que o Capítulo 1 apresenta uma visão geral da pesquisa, descrevendo também os objetivos, e justificativa do trabalho.

O Capítulo 2 aborda conceitos sobre descoberta de conhecimento em base de dados e *data mining*, fundamentais para o desenvolvimento da pesquisa.

No Capítulo 3 são detalhadas as características, métodos e funcionalidades desenvolvidas até o momento para a ferramenta acadêmica *Shell* Orion *Data Mining Engin*e.

A tarefa de classificação no processo de *data mining* é descrita no Capítulo 4 e o método de redes neurais para esta tarefa é apresentado no Capítulo 5.

O Capítulo 6 aborda detalhadamente o funcionamento de redes neurais com função de base radial desenvolvido para tarefa de classificação na *Shell* Orion *Data Mining Engine.* Alguns exemplos da utilização deste modelo de rede neural são demonstrados no Capítulo 7.

No Capítulo 8 são descritas as etapas para do trabalho desenvolvido, bem como a metodologia e os resultados obtidos.

Finalizando, tem-se a conclusão desta pesquisa e sugestões de trabalhos futuros.

2 *DATA MINING*

Com a automatização das empresas o armazenamento de dados tornou-se uma tarefa trivial, proporcionando um aumento significativo da quantidade de informações. Por esse motivo, a extração de conhecimento útil de grandes repositórios de dados por meio de métodos convencionais, tornou-se uma tarefa difícil, porém necessária para auxiliar as instituições na tomada de decisões. A análise destes dados possibilita um melhor planejamento e execução das práticas de negócio, garantindo a competitividade das organizações atuais (GOLDSCHIMDT; PASSOS, 2005).

A exploração de bases de dados é inviável quando feita empregando-se métodos estatísticos convencionais, pois limita a capacidade de extração de conhecimento devido a grande quantidade de informação armazenada. Neste contexto, surgiu o *Data Mining* (DM), que objetiva, por meio de técnicas computacionais inteligentes, extrair conhecimento a partir dos dados auxiliando o homem no processo decisório (CARVALHO, 2005).

Segundo Olson e Delen (2008) o uso desta técnica não é restrito as empresas, *data mining* também oferece vantagens em áreas como medicina, economia, geologia dentre outras, auxiliando os profissionais com o desenvolvimento de melhores práticas.

Han e Kamber (2006) abordam o *data mining* como resultado da evolução natural da tecnologia de informação devido ao aprimoramento dos processos de coleta, armazenamento e gestão de dados.

Alguns fatores que contribuíram à aceitação da técnica de *data mining* são descritos a seguir (HAN; KAMBER, 2001, tradução nossa):

a) **aplicabilidade em grande volume de dados:** grandes repositórios de dados como de empresas de telefonia, bancos, supermercados, entre outras, são adequados à aplicação do *data mining***;**

b) **potencialização dos recursos computacionais:** os avanços tecnológicos possibilitaram o uso do DM que necessita de recursos computacionais potentes para executar seus algoritmos. As inovações na área de banco de dados, como *data warehouses*[2], são propícios para a aplicação do *data mining*.

Apesar da Inteligência Artificial (IA) ser muito utilizada, a exploração de dados por meio de métodos estatísticos ainda é a base do processo de *data mining* (OLSON; DELEN, 2008, tradução nossa).

DM é definido com um processo de reconhecimento de padrões, atuando também como uma ferramenta de previsões, através da extração do conhecimento implícito em bases de dados, sendo abordado por diversos autores como sinônimo do processo de descoberta de conhecimento em bases de dados, chamado de *Knowledge Discovery in Databases* (KDD). (WITTEN; FRANK, 2005, tradução nossa).

2.1 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

Diante das limitações dos sistemas gerenciadores de banco de dados atuais surgiu o conceito de descoberta de conhecimento em bases de dados, com o objetivo de desenvolver novas técnicas computacionais que auxiliassem os seres humanos na manipulação dos dados armazenados, a fim de extrair conhecimento novo que seja vantajoso ao contexto do problema (GOLDSCHIMDT; PASSOS, 2005).

O KDD engloba diversas áreas como aprendizado de máquina, reconhecimento de padrões, banco de dados, estatística e inteligência artificial. É considerado um método interativo, pois exige uma relação com o especialista humano e também iterativo, pois é executado repetidamente, cada vez aprimorando o resultado obtido. O processo de KDD é

[2] Conjunto de banco de dados integrados, que armazena informações corporativas de sistemas para suporte à decisão (KANTARDZIC, 2003).

composto por diferentes etapas e não implica somente na aplicação de algoritmos para reconhecimento de padrões, também considera os métodos de armazenagem e acesso dos dados, e a maneira com que os resultados serão visualizados e interpretados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, tradução nossa).

2.1.1 Etapas da Descoberta de Conhecimento em Bases de Dados

O processo de KDD pode ser dividido em três etapas operacionais básicas, descritas a seguir (HAN; KAMBER, 2006, tradução nossa):

a) **pré-processamento:** consiste na preparação dos dados com o objetivo de obter um melhor desempenho e um resultado mais preciso no processo de extração de conhecimento. O tratamento dos dados é por meio das seguintes funções:
   - **seleção e redução:** analisar os dados que realmente são relevantes para a resolução do problema, com isso tem-se uma otimização do uso de memória e tempo de processamento;
   - **limpeza:** a qualidade dos dados influencia diretamente o resultado do processo de *data mining* e consequentemente o processo de tomada de decisão. Nesta etapa são removidos os dados incompletos e inconsistentes provenientes da má coleta, eliminando redundâncias e erros (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, tradução nossa);
   - **transformação:** consiste em padronizar as informações contidas na base de dados. Esta padronização é específica para cada algoritmo, podendo envolver a transformação de valores numéricos em categóricos ou vice-versa e o preenchimento de valores nulos (HAN; KAMBER, 2006, tradução nossa);

b) ***data mining:*** refere-se efetivamente à busca por conhecimento e extração de padrões. Segundo Rezende (2005) esta etapa envolve a escolha da tarefa de *data mining* e de algoritmos específicos para cada problema, que serão executados de forma iterativa;

c) **pós-processamento:** nesta etapa os especialistas humanos analisam o conhecimento adquirido no *data mining*, sendo este interpretado e disponibilizado para o usuário por meio de algumas operações:

- **avaliação:** identifica os padrões potencialmente úteis para representação do conhecimento (HAN; KAMBER, 2006, tradução nossa);
- **simplificação:** remove detalhes técnicos com o intuito de tornar o modelo menos complexo para o usuário, sem a perda de informações;
- **apresentação:** são utilizados modelos de representação do conhecimento para facilitar a visualização do conhecimento obtido.

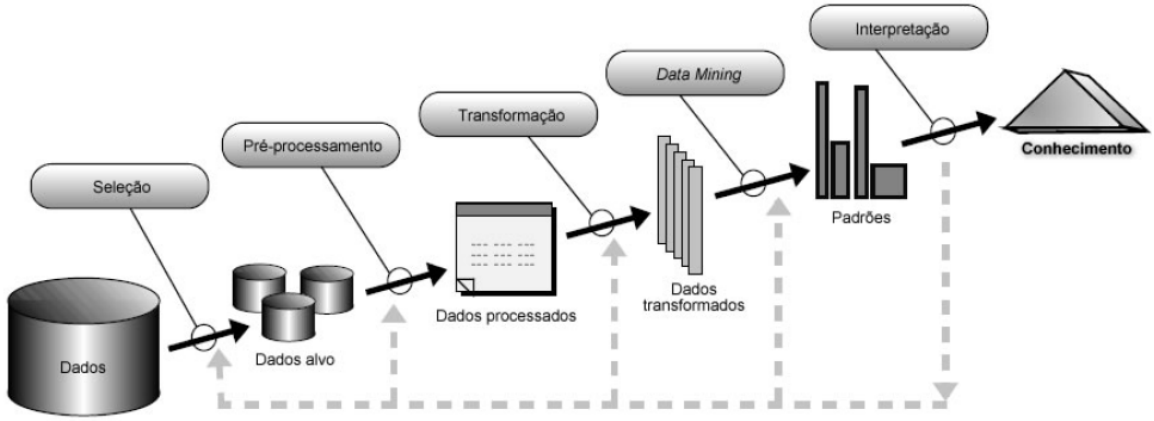

Figura 1. Etapas da descoberta de conhecimento
Fonte: Adaptado de FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH,P. (1996)

A descoberta de conhecimento é considerada um processo complexo, onde o sucesso depende não somente da eficiência computacional, como também do profissional da área de aplicação do KDD que influencia diretamente a qualidade do pré-processamento de dados e interpretação dos resultados obtidos.

2.1.2 Aplicações da Descoberta de Conhecimento em Bases de Dados

Conforme comentado anteriormente, o KDD pode ser aplicado em diversas áreas, sob qualquer domínio que ofereça volume de dados suficiente. Carvalho (2005) apresenta alguns exemplos de aplicações que utilizam *data mining:*

a) **supermercados:** coletar dados sobre os hábitos de compra dos clientes, e analisar essa informação para identificar preferências, vendas casadas e o nível de consumo, permitindo que a empresa organize prateleiras de forma estratégica e ofereça ofertas. De acordo com Berry e Linoff (2004) alguns supermercados vendem este tipo de informação para que os fornecedores também invistam em propaganda;

b) **instituições do governo:** o governo dos Estados Unidos utiliza ferramentas de *data mining* para identificar padrões em contas bancárias que caracterizem crimes. Segundo Goldschmidt e Passos (2005) o KDD também é utilizado para análise do histórico de pagamento de impostos, identificando possíveis inadimplentes;

c) **medicina:** segundo Witten e Frank (2005) o KDD é uma ferramenta de suporte que permite a descoberta de associações entre doenças e as características regionais, sociais e hábitos pessoais, fornecendo conhecimento para estudos de epidemiologia, análise de diagnósticos, tratamentos e sintomas de cada paciente, objetivando a redução de falhas e o aperfeiçoamento dos profissionais;

d) **identificação do perfil dos consumidores:** o *data mining* pode ser utilizado para analisar o perfil dos clientes, criando planos de retenção adequados para evitar a perda do cliente para o concorrente. Também é apropriado para

identificar maus clientes que não geram retorno para a empresa (WITTEN; FRANK, 2005, tradução nossa);

e) **análise financeira:** considerando-se histórico de pagamentos dos clientes, é possível aplicar métodos de KDD que identificam os bons e maus pagadores e os valores adequados para limite de crédito . Além disso, a descoberta de conhecimento pode ser utilizada para prever tendências do mercado financeiro, visando aumentar o retorno e diminuir o risco de perda (CARVALHO, 2005).

Considerando as diversas aplicações citadas e que o *data mining* é composto por tarefas, é necessário analisar o domínio de cada problema para identificar quais métodos são mais adequados para a resolução do problema e otimização do conhecimento gerado.

2.2 TAREFAS DE *DATA MINING*

Conforme Kantardzic (2003) o *data mining* é composto por tarefas, e envolve a aplicação de algoritmos específicos para o domínio de cada problema. Estas tarefas podem ser dividas em preditivas, as quais prevêem informações a partir de um conjunto de exemplos de dados, e descritivas, capazes de identificar padrões que possam ser interpretados por seres humanos (Figura 2).

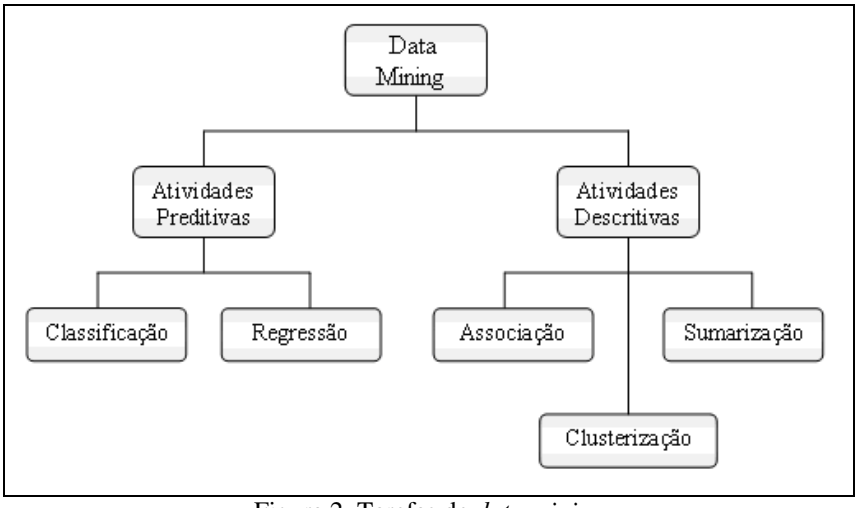

Figura 2. Tarefas de *data mining*

Fonte: Adaptado de REZENDE, S. (2005)

A seguir são descritas as características e exemplos de aplicações das principais tarefas de *data mining:*

a) **associação:** busca por padrões e relações entre os dados, com ocorrência simultânea e freqüente, que possam caracterizar uma tendência. Pode ser aplicada para identificar itens que são consumidos em conjunto, permitindo uma disposição estratégica dos mesmos nas prateleiras, que induz à compra de outros produtos (BERRY; LINOFF, 2004, tradução nossa);

b) **classificação:** segundo Carvalho (2004) é uma das técnicas mais utilizadas no *data mining,* que consiste na busca por uma regra que permita associar cada registro da base de dados à uma classe. Conforme Berry e Linoff (2004) esta tarefa cria funções de classificação que podem ser aplicadas a conjuntos de dados não classificados, sendo capaz de classificar clientes em baixo, médio ou alto risco de empréstimo;

c) **clusterização:** é uma tarefa descritiva utilizada quando não existem classes predefinidas como na classificação, e tem como objetivo identificar conjuntos de elementos que compartilhem características similares entre si, e distintas em relação aos objetos de outros grupos (HAN; KAMBER, 2001, tradução nossa). Esta tarefa pode ser aplicada para agrupar clientes de acordo com as preferências de compra, ou como pré-processamento para aplicação de outros métodos;

d) **previsão de séries temporais:** prevê futuros valores de um índice por meio da análise do comportamento passado (BERRY; LINOFF, 2004). Um exemplo de aplicação desta técnica é determinar o índice de vendas de um produto no decorrer de um período de tempo;

e) **regressão:** mapeia os registros de uma base de dados em valores reais, divergindo da classificação apenas por estar restrita à valores numéricos (KANTARDZIC, 2003, tradução nossa). A regressão pode ser utilizada na definição do limite de crédito dos clientes de um banco.

Muitos problemas podem ser solucionados utilizando *data mining,* sendo que cada tarefa possui suas particularidades, por isso deve ser criteriosamente escolhida pelo especialista de acordo com o problema e com os objetivos do usuário (KANTARDZIC, 2003, tradução nossa).

Diferentes métodos podem ser aplicados às tarefas de DM de acordo com os objetivos do usuário e as características do problema. Alguns destes métodos são descritos a seguir:

a) **redes neurais:** uma rede neural artificial é um paradigma computacional inspirado na estruturas neurais do cérebro humano, capazes de aprender por experiência, generalizar conhecimento, associar padrões e abstrair características relevantes dos dados de forma análoga aos seres humanos. Este método é aplicado principalmente em problemas de classificação, clusterização, associação e aproximação de funções[3] (MEHROTRA; MOHAN; RANKA, 1996, tradução nossa);

b) **árvores de decisão:** de acordo com Berry e Linoff (2004) são estruturas hierárquicas utilizadas para dividir uma grande coleção de dados em pequenos sub-conjuntos. É um método bastante empregado na tarefa de classificação, pois a base de dados é sucessivamente dividida por meio de regras de decisão;

[3] Tarefa que consiste na descoberta de uma função de mapeamento de pares entrada-saída (MEHROTRA; MOHAN; RANKA, 1996, tradução nossa);

c) **algoritmos genéticos:** são modelos computacionais baseados na teoria da evolução das espécies[4] e reprodução genética[5]. São capazes de convergir para soluções ótimas ou aproximadamente ótimas por meio de um processo adaptativo do conjunto de dados, sendo muito úteis na resolução de problemas de classificação, otimização e na avaliação de outros algoritmos de DM (HAN; KAMBER, 2006, tradução nossa).

d) **lógica *fuzzy:*** conforme Rezende (2005) na lógica tradicional, os elementos pertencem ou não a um determinado conjunto de dados, já a lógica *fuzzy* permite que um elemento pertença a um ou mais conjuntos, porém com graus de pertinência diferentes, utilizando um raciocínio aproximado de forma semelhante a capacidade humana de tratar imprecisões.

As tarefas e métodos descritos anteriormente encontram-se disponíveis em ferramentas específicas para o processo de descoberta de conhecimento, chamadas *shells.* Na Tabela 1 são descritas algumas ferramentas de *data mining* disponíveis;

[4] Indivíduos mais aptos possuem maiores chances de sobrevivência e de gerar descendentes (GOLDSCHMIDT; PASSOS, 2005).
[5] União de genes que resultam em um indivíduo com as características dos seus ascendentes (GOLDSCHMIDT; PASSOS, 2005).

Tabela 1. Exemplos de ferramentas de *data mining*

Nome	Técnicas Disponíveis	Fabricante	Tipo
Clementine	Classificação, regras de associação, clusterização, e padrões seqüenciais	Data-Miner PTy Ltd www.data-miner.com	Comercial
Cubist	Regressão	Rule Quest www.rulequest.com	Comercial
Darwin	Classificação, regressão e clusterização	Oracle Corp. www.oracle.com	Comercial
DataMite	Regras de associação	Dr. Phillip Vasey do LPA Prolog	Comercial
Intelligent Miner	Regras de associação, padrões seqüenciais, classificação, clusterização, sumarização e modelagem de dependência	IBM Corp. www.ibm.com	Comercial
Microsoft Data Analyzer	Classificação e clusterização	Microsoft Corp. www.microsoft.com	Comercial
Oracle Data Mining	Classificação e regras de associação	Oracle Corp.	Comercial
Orion Data Mining Engine	Associação, classificação e clusterização	Grupo de Pesquisa em Inteligência Computacional Aplicada – Unesc	Gratuita
PolyAnalyst	Classificação, regressão, associação, clusterização, sumarização e modelagem de dependência	Megaputer Intelligence www.megaputer.com	Comercial
WEKA	Classificação, regressão e regras de associação	University of Waikato www.cs.waikato.ac.nz	Gratuita

Fonte: Adaptado de REZENDE, S. (2005)

A aquisição destas ferramentas pode se tornar inviável para as organizações, pois em sua maioria são comerciais resultando em uma carência de *shells* gratuitas. Dentre as ferramentas gratuitas disponíveis, encontra-se em desenvolvimento a *Shell* Orion *Data Mining Engine,* um projeto acadêmico iniciado em 2005 pelo Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da Unesc que aplica os conceitos de descoberta de conhecimento descritos anteriormente.

3 *SHELL ORION DATA MINING ENGINE*

A *Shell* Orion *Data Mining Engine* é uma ferramenta que está sendo desenvolvida pelos acadêmicos e professores do Grupo de Pesquisa em Inteligência Computacional do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, que tem como objetivo disponibilizar uma *Shell* de *data mining* gratuita no mercado.

O projeto iniciou-se no ano de 2005 e possui atualmente três tarefas implementadas: associação, classificação e clusterização, e cada método foi desenvolvido por um acadêmico em seu Trabalho de Conclusão de Curso (TCC). Na Tabela 2 pode-se acompanhar a evolução do desenvolvimento da *Shell* Orion, bem como os algoritmos já implementados.

Tabela 2. Evolução da *Shell* Orion

Ano	Tarefa	Método	Algoritmo	Tipo de atributo	Referência
2005	Associação	Regras de associação	*Apriori*	Numéricos	(CASAGRANDE, 2005)
2005	Classificação	Àrvores de decisão	ID3	Nominais	(PELEGRIN, 2005)
2007	Classificação	Árvores de decisão	CART	Nominais e numéricos	(RAIMUNDO, 2007)
2007	Clusterização	Particionamento	*K-means*	Numéricos	(MARTINS, 2007)
2007	Clusterização	Redes Neurais	*Kohonen*	Numéricos	(BORTOLOTTO, 2007)
2008	Clusterização	Lógica *fuzzy*	Gustafson-Kessel	Numéricos	(CASSETARI JUNIOR, 2008)
2009	Clusterização	Lógica *fuzzy*	Gath-Geva	Numéricos	(PEREGO, 2008)
2009	Classificação	Árvores de decisão	C4.5	Nominais e numéricos	(MONDARDO, 2009)

No desenvolvimento da *Shell* Orion utiliza-se a linguagem de programação *Java,* pois segundo Pelegrin (2005) esta tecnologia é multiplataforma, permite a reutilização de código e as suas ferramentas de desenvolvimento são gratuitas.

Para facilitar o acesso do usuário aos métodos desenvolvidos e a outras funcionalidades da *Shell* Orion*,* foi desenvolvida uma interface principal que centraliza todas as funções da ferramenta. Esta interface permite conexões com qualquer banco de dados desde que possua um *driver Java Data Base Connectivity* (JDBC) devidamente instalado.

Após conectar com a base de dados, na etapa de *data mining,* o usuário pode acessar os algoritmos descritos anteriormente na Tabela 2.

3.1 TAREFAS E MÉTODOS DA SHELL ORION

Atualmente a *Shell* Orion está organizada em diferentes módulos para as tarefas de associação, classificação e clusterização, sendo que cada algoritmo desenvolvido necessita de informações peculiares de acordo com a tarefa em que é aplicado.

Para o módulo de associação foi desenvolvido o algoritmo *Apriori,* muito utilizado para encontrar relacionamentos entre os itens de dados gerando um conhecimento prévio das características de conjuntos freqüentes (HAN; KAMBER, 2001, tradução nossa).

No módulo de classificação foram implementados três algoritmos, o ID3 (Figura 3), CART e C4.5. Segundo Kantardzic (2003) o algoritmo ID3 desenvolvido por John Ross Quinlan é aplicado recursivamente e consiste em escolher o melhor atributo para cada nó de decisão da árvore, gerando uma árvore de decisão de cima para baixo (*top-down*).

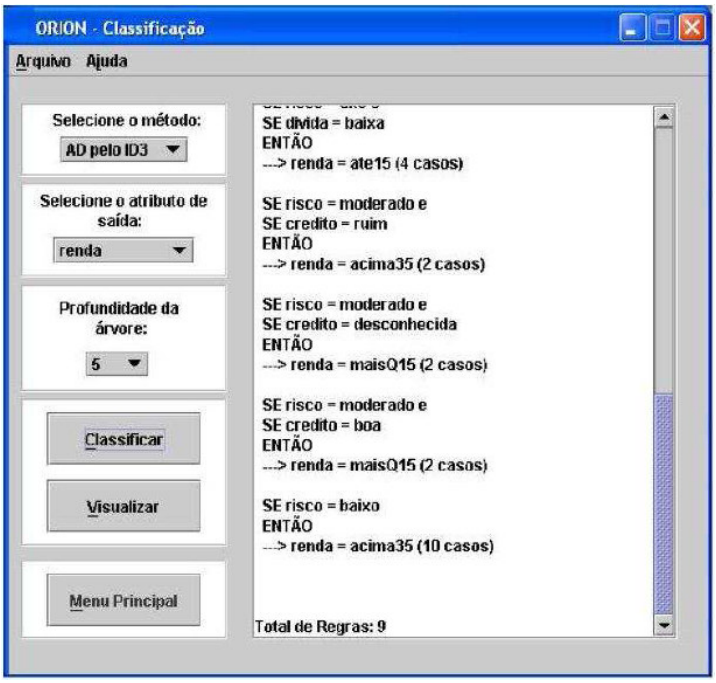

Figura 3. Módulo de classificação pelo algoritmo ID3
Fonte: PELEGRIN, D. (2005)

A representação do conhecimento por meio de árvores de decisão (Figura 4) facilita a visualização do conhecimento adquirido e do atributo que mais influenciou no resultado do modelo (PELEGRIN, 2005).

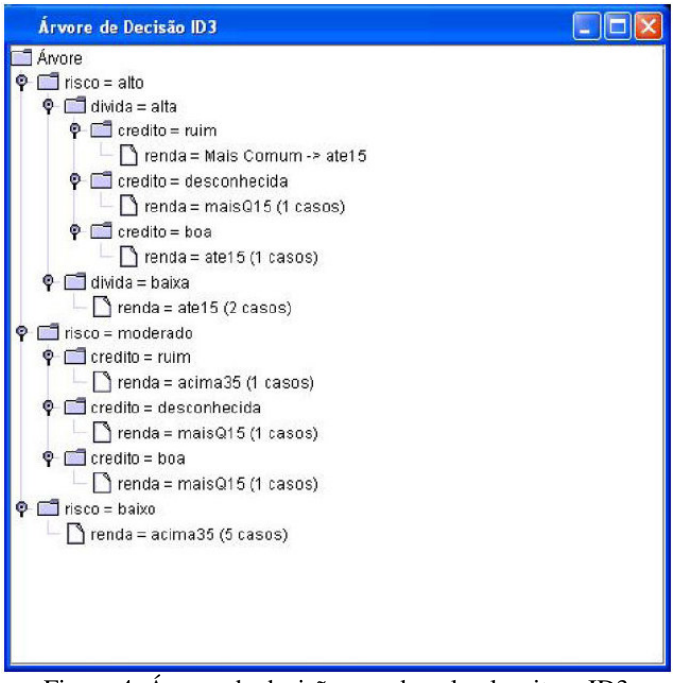

Figura 4. Árvore de decisão gerada pelo algoritmo ID3
Fonte: PELEGRIN, D. (2005)

O CART é um exemplo de algoritmo de partição binária recursiva, que divide os nós em dois subconjuntos utilizando regras de classificação (Figura 5) executando recursivamente em cada subconjunto gerado (FONSECA, 1994).

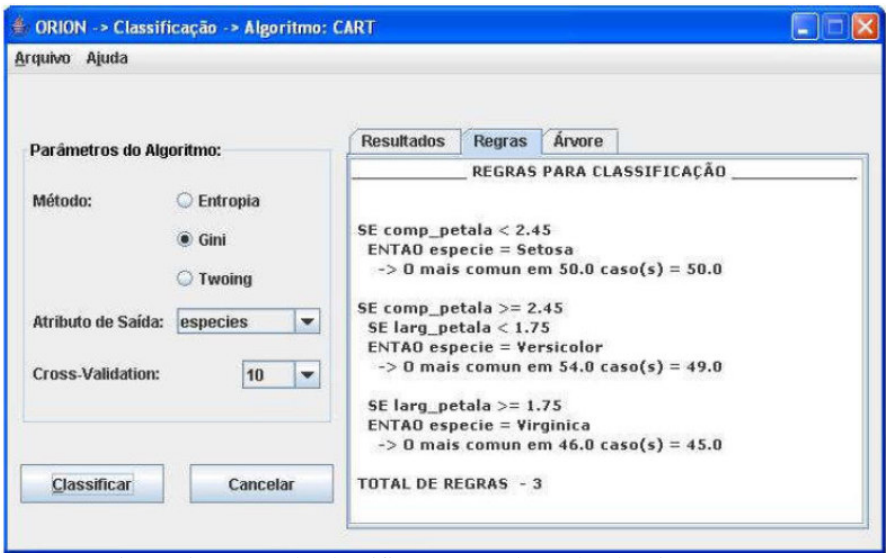

Figura 5. Regras de classificação geradas pelo algoritmo CART
Fonte: RAIMUNDO, L. (2007)

Segundo Fonseca (1994) este algoritmo possui grande capacidade de descoberta de relações entre os dados e também utiliza árvores de decisão para representação dos resultados (Figura 6).

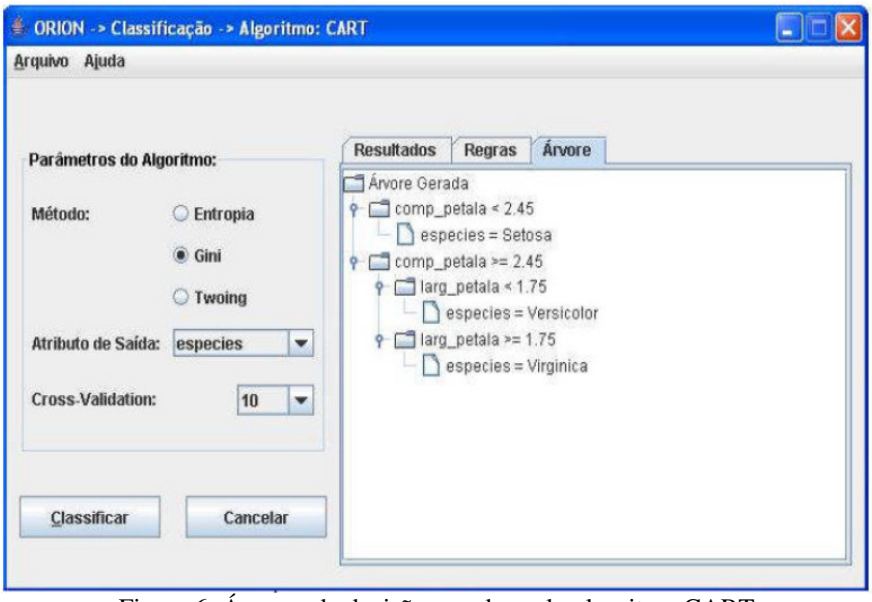

Figura 6. Árvores de decisão geradas pelo algoritmo CART
Fonte: RAIMUNDO, L. (2007)

Publicado também por John Ross Quinlan em 1987 como uma evolução do algoritmo ID3, o algoritmo C4.5 é a mais recente funcionalidade da *Shell* Orion para a tarefa de classificação. Este algoritmo permite a poda da árvore de decisão, ou seja, pode-se excluir as regras insignificantes para o problema de classificação.

O usuário deve informar uma série de parâmetros como pode-se observar nas Figuras, e os resultados são disponibilizados no formato de regras e de árvore de decisão.

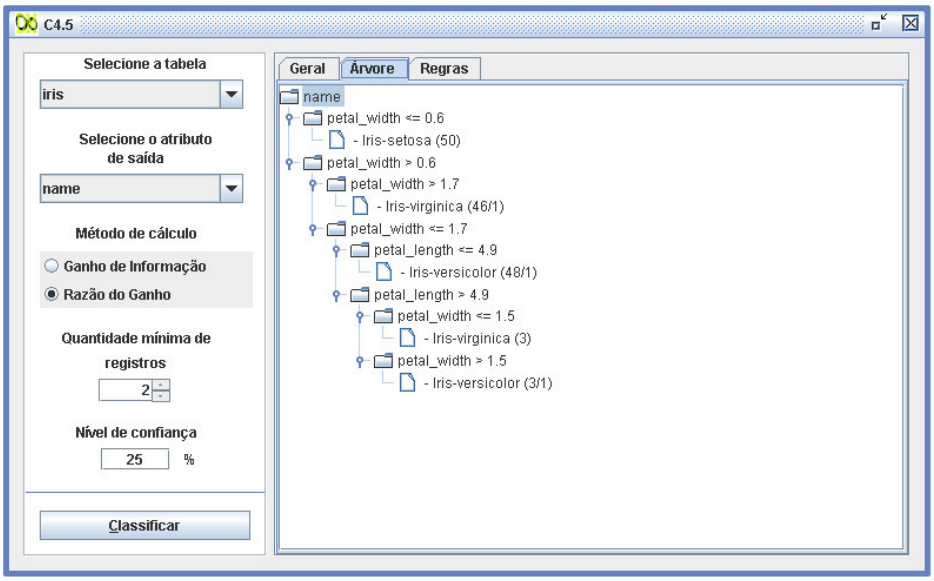

Figura 7. Resultado gerado pelo algoritmo C4.5 por meio de árvore de decisão
Fonte: MONDARDO, R. (2009)

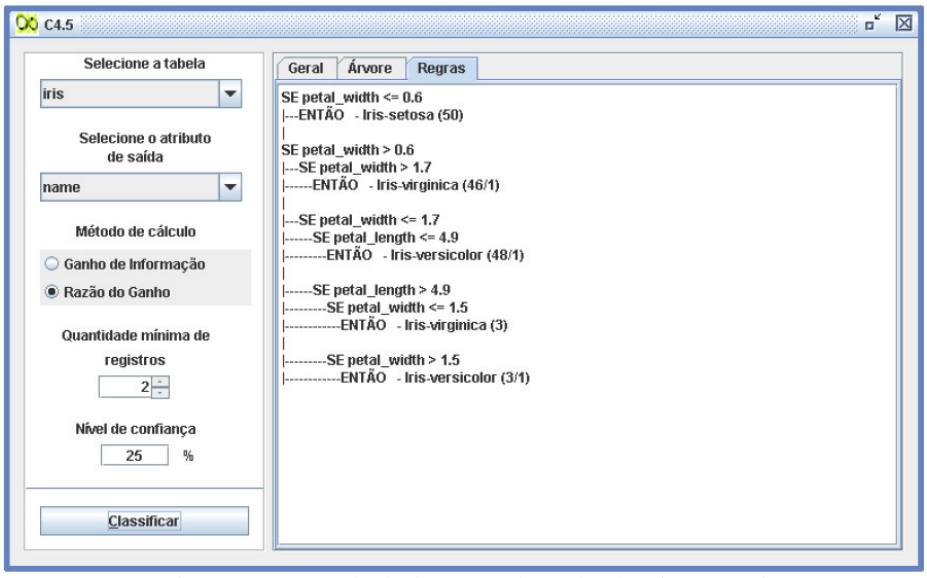

Figura 8. Regras de decisão geradas pelo algoritmo C4.5
Fonte: MONDARDO, R. (2009)

Além das tarefas citadas anteriormente, também foi desenvolvido na *Shell* Orion o módulo de clusterização, no qual foi implementado inicialmente o algoritmo *K-means.* Segundo Kantardzic (2003) este algoritmo seleciona aleatoriamente *k* pontos de dados que serão o centro inicial de cada *cluster,* posteriormente calcula as distâncias de um ponto em relação a cada elemento, e cada ponto é atribuído ao grupo que apresentar menor distância.

Encontra-se implementado também no módulo de clusterização da *Shell* Orion, o algoritmo de *Kohonen* pelo método de redes neurais auto-organizáveis, nas quais o treinamento é não-supervisionado e competitivo. Não-supervisionado, pois a rede adquire conhecimento sem a indicação de exemplos, e competitivo porque os neurônios competem entre si resultando em um vencedor. O algoritmo, por meio deste aprendizado, organiza os elementos em grupos com características similares entre si e diferente de outros grupos (GOLDSCHIMDT; PASSOS, 2005).

Em 2008 foi adicionado na *Shell* Orion o método de lógica *fuzzy* para tarefa de clusterização, por meio do algoritmo Gustafson-Kessel. Este algoritmo busca grupos de formas geométricas diferentes utilizando uma matriz de indução *fuzzy* que contém as medidas de afastamento (distância) de cada conjunto (GUERRA, 2006).

O algoritmo Gustafson-Kessel trabalha somente com operações matemáticas, por este motivo processa somente valores numéricos. Se for necessário utilizar atributos nominais deve-se realizar a conversão destes para valores decimais (CASSETARI JUNIOR, 2008).

Também foi desenvolvido para o módulo de clusterização pelo método de lógica *fuzzy*, o algoritmo Gath-Geva. Este algoritmo propõe modificações no algoritmo Gustafson-Kessel com o objetivo de melhorar o desempenho do processo de clusterização.

O algoritmo Gath-Geva também trabalha com uma matriz de indução *fuzzy,* porém, seu diferencial está na distância utilizada que envolve um termo exponencial baseada na probabilidade de seleção de cada grupo (GUERRA, 2006).

Com base no objetivo de desenvolver novas funcionalidades para a *Shell* Orion, esta pesquisa consiste na implementação do método de redes neurais com função de ativação de base radial para o módulo de classificação.

4 A TAREFA DE CLASSIFICAÇÃO NO PROCESSO DE DATA MINING

Considerada uma das tarefas mais comuns do processo de data mining, o método de classificação consiste em mapear os registros de uma base de dados em uma quantidade finita de conjuntos, atribuindo cada elemento a uma categoria pré-definida (PAL; MITRA, 2004).

De acordo com Motta (2004) é uma atividade preditiva que busca por relações entre os dados de entrada e as classes pré-definidas, extraindo padrões que identifiquem tendências futuras.

Conforme mostra a Figura 9, a tarefa de classificação pode ser entendida como a busca por uma função que associe cada registro X_i de uma base de dados, a um único rótulo categórico Y_j, denominado classe ou objeto de saída (GOLDSCHMIDT; PASSOS, 2005).

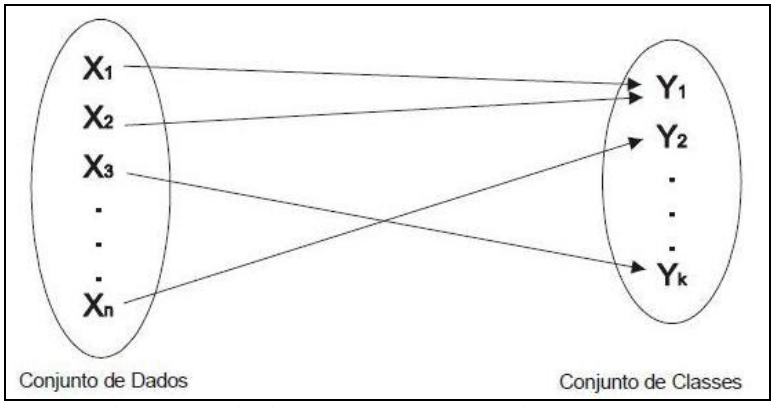

Figura 9. Atribuição de classes para registros de dados
Fonte: Adaptado de GOLDSCHMIDT, R.; PASSOS, E. (2005)

O método de classificação recebe como entrada um conjunto de registros de uma base de dados, formado por valores de atributos inerentes ao domínio do problema, distinguindo-se de outros métodos, pela presença de um atributo especial denominado classe (BERRY; LINOFF, 2004, tradução nossa). As variáveis de entrada exercem influência direta

sobre o conhecimento adquirido, podendo assumir valores categóricos (alfanuméricos) ou contínuos (numéricos) (GOLDSCHMIDT; PASSOS, 2005).

O conjunto de atributos da base de dados a ser explorada pode ser classificado em dois grupos: dados de treinamento, composto pelos registros utilizados na fase de aprendizagem, e dados de teste, utilizados na avaliação do modelo gerado (RUSSEL; NORVIG, 2004).

Neste contexto, segundo Han e Kamber (2001), o processo de extração do conhecimento ocorre em duas etapas:

a) **aprendizagem:** conforme pode-se observar na Figura 10, os dados de treinamento, para os quais as classes são conhecidas, são utilizados para criação de um modelo de aprendizagem ou classificador, que posteriormente é aplicado a dados desconhecidos a fim de classificá-los (BERRY; LINOFF, 2004, tradução nossa);

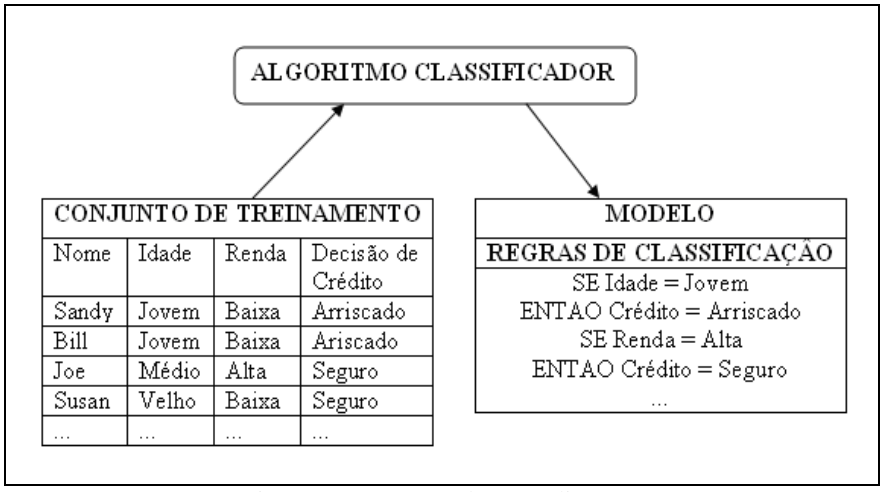

Figura 10. Processo de aprendizagem
Fonte: Adaptado de HAN, J.; KAMBER, M. (2001)

b) **teste:** o conjunto de teste é utilizado como estimativa da exatidão do classificador (Figura 11). De acordo com Han e Kamber (2001) a precisão do modelo é estimada empregando-se critérios como:

- **precisão da predição:** capacidade do modelo para classificar objetos ainda não conhecidos,
- **custo computacional**: velocidade de processamento,
- **robustez:** habilidade do classificador de atribuir cada registro à classe correta.

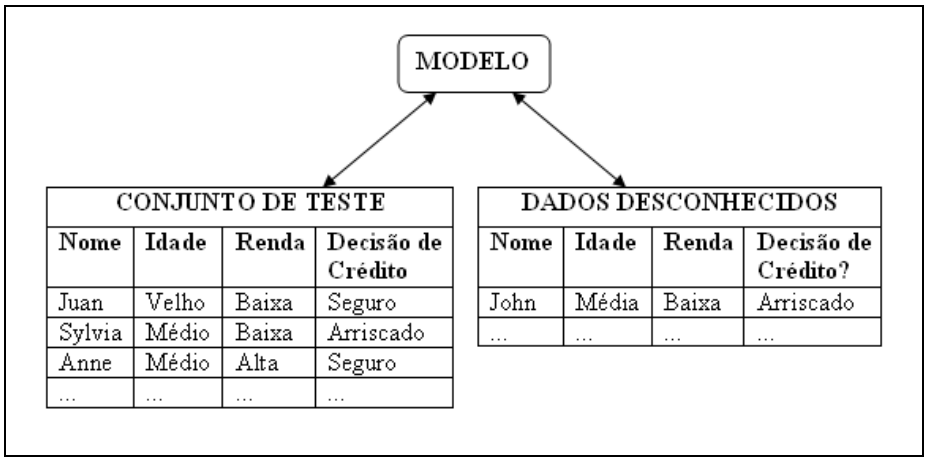

Figura 11. Teste e Classificação
Fonte: Adaptado de HAN, J.; KAMBER, M. (2001)

Pode-se observar que o objeto de saída selecionado é o atributo *Decisão de Crédito*, sendo assim, o conjunto de teste é aplicado ao modelo, e posteriormente cada registro de um conjunto de dados desconhecidos será classificado como, *seguro* ou *arriscado*, sendo estes os possíveis valores que o atributo *Decisão de Crédito* poderá assumir.

De acordo com Han e Kamber (2001) quando o classificador se ajusta em excesso ao conjunto de treinamento e incorpora algumas particularidades dos dados de exemplo, tende a apresentar bom desempenho (*overfitting*). Por este motivo, na fase de aprendizagem (Figura 10), a seleção dos registros deve ser aleatória e, independente dos registros de teste, não interferindo na construção do modelo classificador.

Diversas abordagens são aplicáveis à tarefa de classificação, como por exemplo, árvores de decisão, redes neurais e algoritmos genéticos. Considerando que esta pesquisa aplica o método de redes neurais, o mesmo será descrito detalhadamente no próximo capítulo.

5 O MÉTODO DE REDES NEURAIS PARA CLASSIFICAÇÃO

As Redes Neurais Artificiais (RNA) são sistemas paralelos distribuídos compostos por unidades de processamento simples (neurônios) capazes de mapear funções matemáticas e armazenar conhecimento (BRAGA; CARVALHO; LUDERMIR, 2000).

Wang et al (1991) afirma que as RNA apresentam a habilidade de classificar padrões desconhecidos adequando-se à resolução de problemas onde se tem pouco conhecimento das relações entre atributos e classes. As RNA são capazes de adquirir conhecimento por meio de um conjunto reduzido de exemplos e produzir respostas consistentes para dados não conhecidos, diferenciando-se de outros métodos computacionais aplicados ao processo de tomada de decisão.

Em uma RNA clássica os neurônios são dispostos em camadas e interligados por conexões conhecidas como pesos sinápticos. Estes pesos representam o conhecimento da rede, que é produzido de forma análoga ao cérebro humano empregando um processo de aprendizagem e criando uma representação relativa ao domínio do problema (HAYKIN, 2001).

A capacidade de aprender a partir de seu ambiente e melhorar seu desempenho pelo processo de aprendizagem, é a principal propriedade de uma rede neural. As RNA também apresentam outras propriedades similares ao cérebro humano, que são atrativas para resolução das tarefas de DM como (KANTARDZIC, 2003, tradução nossa):

a) **busca paralela:** o conhecimento de uma RNA fica distribuído pela rede, deste modo a busca por informação ocorre de forma paralela e não seqüencial;
b) **adaptabilidade:** as RNA possuem capacidade de adaptar seus pesos sinápticos de acordo com as alterações no ambiente externo, tornando-se uma ferramenta útil para classificação adaptativa de padrões e processamento adaptativo;

c) **tolerância a falhas:** RNA são capazes de realizar computação robusta, ou seja, a perda de um conjunto de neurônios não danifica seriamente o conhecimento e funcionamento da rede devido a natureza distribuída da informação, apresentando assim uma degradação suave do seu desempenho;

d) **generalização:** a capacidade de generalização permite que a rede responda corretamente a dados ruidosos e distorcido;

e) **abstração:** as RNA possuem habilidade de identificar a relevância dos dados de entrada, podendo extrair informações dos padrões a partir de dados ruidosos;

f) **resposta a evidências:** no contexto da classificação de padrões, refere-se a confiabilidade da RNA para tomada de decisão, essa característica permite que a rede seja capaz de rejeitar padrões ambíguos e consequentemente melhorar seu desempenho;

g) **não-linearidade:** capacidade que as redes neurais possuem para identificação de relações entre dados de entrada não-lineares.

5.1 PARADIGMAS DE APRENDIZAGEM

A principal propriedade de uma rede neural artificial é a sua capacidade de aprender e melhorar seu desempenho ajustando seus pesos sinápticos iterativamente de acordo com a resposta da rede ao ambiente (ativação dos neurônios). Este processo é conhecido como aprendizagem da rede ou algoritmo de aprendizagem.

A maneira como ocorre este ajuste de pesos é que determina o tipo do aprendizado, que pode ser classificado em supervisionado ou não-supervisionado.

5.1.1 Aprendizagem Supervisionada

O processo de aprendizagem supervisionada em uma rede neural ocorre com o auxílio de exemplos fornecidos por um supervisor externo, por este motivo também é chamado de aprendizado com um professor.

O aprendizado supervisionado para classificação de padrões, caracteriza-se por um mapeamento de entrada e saída que ajusta iterativamente os valores dos pesos sinápticos por meio da aplicação de exemplos de dados de treinamento, conforme mostra a Figura 12. Cada exemplo é composto por uma amostra de entrada relacionada a uma saída desejada, deste modo, os pesos sinápticos são modificados convergindo para uma diferença mínima entre as respostas desejadas e a da rede (HAYKIN, 2001).

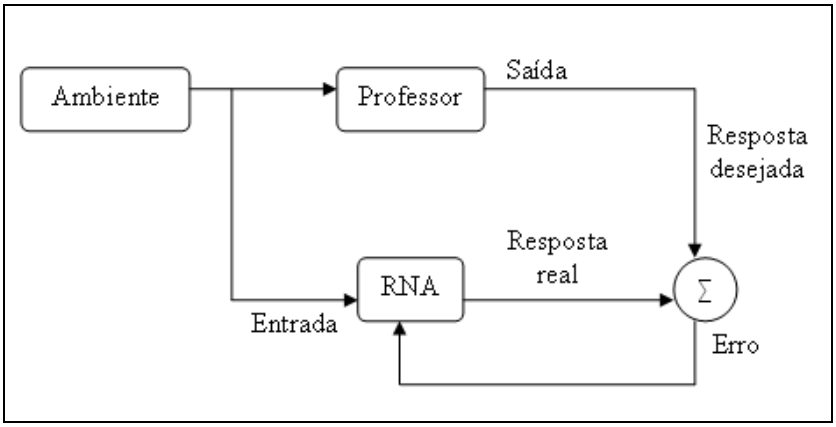

Figura 12. Aprendizagem supervisionada
Fonte: Adaptado de HAYKIN, S. (2001)

Neste paradigma, a função do supervisor é fornecer à RNA as respostas desejadas para o vetor de treinamento proveniente do ambiente. Os parâmetros da rede são ajustados de acordo com o sinal de erro, transmitindo o conhecimento do ambiente para a RNA. Após este processo atingir sua completude, a rede torna-se capaz de processar dados do ambiente independentemente de um supervisor (HAYKIN, 2001).

5.1.2 Aprendizagem Não-Supervisionada

Como o próprio nome sugere, no aprendizado não-supervisionado, também chamado auto-organizado, não há um supervisor que forneça as saídas desejadas, ou seja, a rede adquire conhecimento a partir de exemplos (BRAGA; CARVALHO, LUDERMIR, 2000).

A aprendizagem em redes auto-organizáveis utiliza apenas os dados de entrada como mostra a Figura 13. Neste contexto, os neurônios da rede atuam como classificadores dos padrões de entrada, por meio de um processo de competição e cooperação (ORTEGA, 2008).

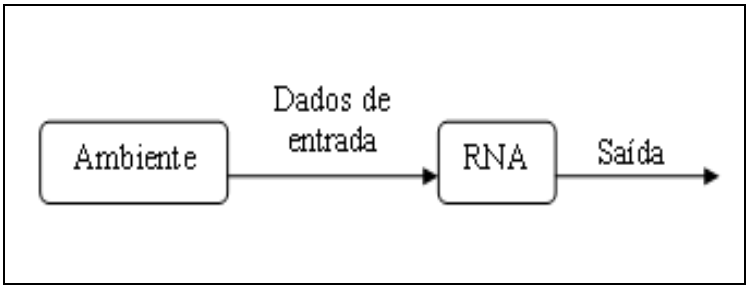

Figura 13. Aprendizagem não-supervisionada
Fonte: Adaptado de HAYKIN, S. (2001)

Uma RNA auto-organizável organiza arbitrariamente os dados de entrada recebidos, ajustando somente os pesos sinápticos do neurônio vencedor (aquele que está mais próximo do vetor de entrada), inativando os demais neurônios. Deste modo, entradas com propriedades semelhantes ativam o mesmo neurônio, tornando a RNA capaz de definir a classe de cada entrada, e de criar automaticamente novas classes para entradas que não se encaixarem em nenhuma classe (AZEVEDO; BRASIL; OLIVEIRA, 2000).

5.2 REDES NEURAIS DE MÚLTIPLAS CAMADAS

As Redes Neurais de Múltiplas Camadas (MLP) caracterizam-se pela existência de uma ou mais camadas ocultas entre as camadas de entrada e saída (Figura 14).

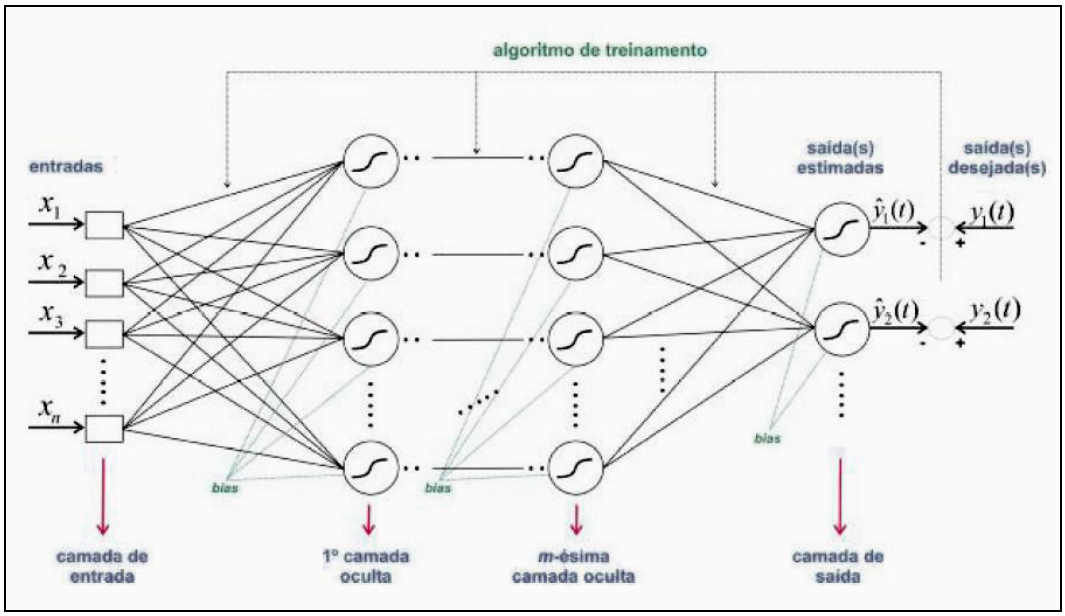

Figura 14. Arquitetura de uma rede neural de múltiplas camadas
Fonte: GUERRA, F. (2006)

Haykin (2001) afirma que as redes MLP têm sido utilizadas com sucesso na resolução de problemas complexos por meio de treinamento supervisionado pelo algoritmo de retropropagação de erro[6] (*backpropagation*) e apresentam maior poder computacional, quando comparadas com redes sem camadas ocultas.

De acordo com Braga et al (2000) as redes neurais de uma só camada são capazes de resolver apenas problemas linearmente separáveis, ou seja, que podem ser satisfeitos por uma reta ou hiperplano como fronteira de decisão (Figura 15). Já a resolução de problemas de classificação não-lineares necessita de redes neurais com uma ou mais camadas ocultas.

[6] Algoritmo baseado na regra de aprendizagem por correção de erro (HAYKIN, 2001).

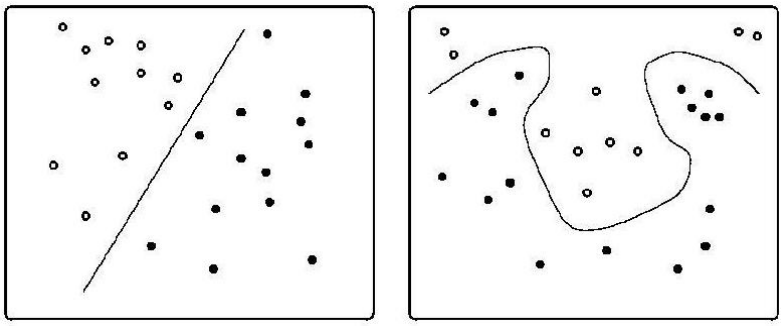

Figura 15. Classificação linear e não-linear
Fonte: Adaptado de BISHOP, C. (1995)

Neste contexto, destacam-se as redes neurais com função de ativação de base radial por serem capazes de realizar o mapeamento não-linear do espaço de entrada.

5.3 REDES NEURAIS COM FUNÇÃO DE ATIVAÇÃO DE BASE RADIAL

Broomhead e Lowe em 1998 estão entre os primeiros a explorar o uso de funções de base radial em RNA. Estas redes são capazes de aprender rapidamente padrões complexos e tendências, apresentando rápida adaptação a mudanças (HAYKIN, 2001).

Uma Rede Neural com Função de Ativação de Base Radial (RN-RBF) pode ser entendida como um problema de ajuste de curva (aproximação de funções) em um espaço de alta dimensionalidade, nas quais a ativação de um neurônio da camada oculta é determinada pela distância entre os vetores de entrada e peso, e produz uma resposta localizada para o estímulo de entrada (THEODORIDIS; KOUTROUMBAS, 2006, tradução nossa).

As RN-RBF são consideradas aproximadores universais, assim como as redes MLP, entretanto, as redes possuem arquiteturas bem distintas. Redes RBF podem possuir mais de uma camada intermediária, contudo, é comum associá-las a redes com apenas uma camada oculta que transforma um conjunto de padrões de entrada não linearmente separáveis em um conjunto de saídas linearmente separáveis (BRAGA; CARVALHO; LUDERMIR, 2000).

Considerando o objetivo geral desta pesquisa, e a fim de facilitar o entendimento do algoritmo, as características e funcionalidades deste modelo de rede neural são apresentadas com detalhes no Capitulo 6.

6 O MÉTODO DE REDES NEURAIS COM FUNÇÃO DE ATIVAÇÃO DE BASE RADIAL

Uma rede neural com função de ativação de base radial consiste em um modelo neural multicamadas, capaz de aprender padrões complexos e resolver problemas não-linearmente separáveis. Assim, assume uma significativa posição dentro do domínio de redes neurais devido ao ganho de tempo no processo de treinamento e sua eficiência computacional (HAYKIN, 2001).

Conforme pode ser observado na Figura 16, a arquitetura de uma rede RBF é dividida em três camadas distintas (FERNANDES; DORIA NETO; BEZERRA, 1999):

a) **camada de entrada:** esta camada permite a conexão da rede com o meio, na qual são apresentados os vetores de entrada;
b) **camada oculta ou escondida:** constituída por neurônios que representam funções de base radial os quais aplicam uma transformação não-linear do espaço de entrada. Cada neurônio desta camada cria um campo receptivo local, chamado de centro da unidade radial, sendo que a distância entre o vetor de entrada e este centro é que determina a ativação do neurônio;
c) **camada de saída:** camada linear que fornece a resposta da rede ao padrão apresentado na camada de entrada, podendo ser composta por um ou mais neurônios de saída.

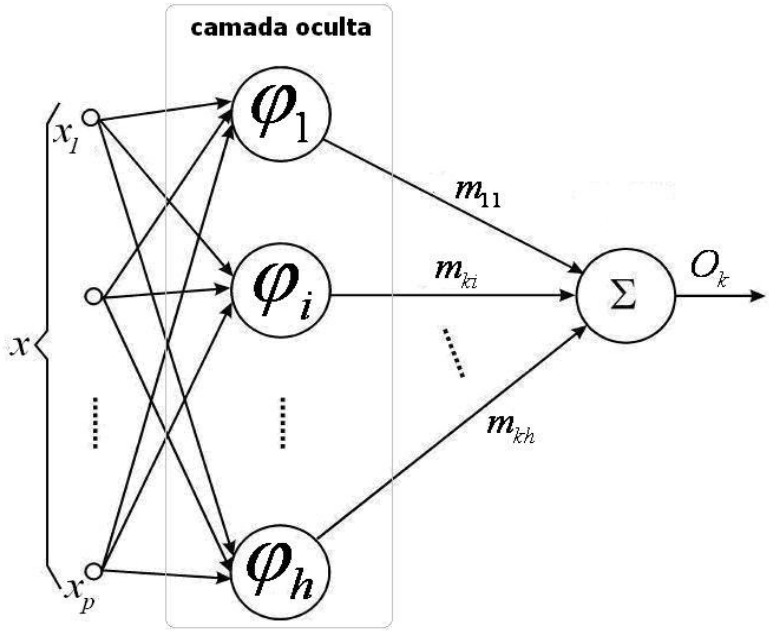

Figura 16. Arquitetura de uma rede RBF para classificação de padrões.
Fonte: Adaptado de FERNANDES, M. A. C. et al (1999)

Onde, m_{ki} é o peso sináptico entre o neurônio k da camada de saída e o neurônio i da camada oculta; x_p é o p-ésimo vetor de entrada de um conjunto de treinamento x; φ_i é a função de base radial para o neurônio i e O_k é o k-ésimo componente do vetor de saída O, que será igual à quantidade de classes pré-definidas. Em problemas de classificação com apenas duas classes, um neurônio de saída é suficiente.

O processo de aprendizado das redes RBF equivale ao ajuste de uma superfície multidimensional ao conjunto de dados, transformando um problema de classificação não-linear em um problema linear. Esta transformação justifica-se no Teorema de Cover (1965), o qual afirma que um problema de classificação de padrões disposto em um espaço de alta dimensão tem maior probabilidade de ser linearmente separável do que em um espaço de baixa dimensionalidade (MICHIE; SPIEGELHALTER, 1994, tradução nossa).

Na prática, o funcionamento correto de uma rede RBF, implica em definir os valores adequados para os parâmetros da rede, para que o processo de aprendizado atinja a maior aproximação possível do conjunto de estradas *x* em relação às saídas desejadas *d*.

O treinamento da rede RBF desenvolvida pode ser dividido nas seguintes etapas:

a) **seleção dos centros:** define-se os centros de cada função de base radial da camada oculta da rede;
b) **definição do raio da função de base:** determina-se o raio de abrangência da função de base em relação ao seu centro;
c) **mapeamento do espaço não-linear:** na camada oculta da rede, as funções de base realizam a transformação dos dados de entrada não-lineares;
d) **camada de saída:** aplica-se um método de classificação linear.

6.1 DEFINIÇÃO DOS CENTROS DA REDE RBF

A abordagem mais simples para determinação do vetor de centros é definir valores fixos para cada função gaussiana. Estes valores podem ser obtidos aleatoriamente a partir do conjunto de treinamento, desde que os dados de treinamento estejam distribuídos de uma forma representativa para o domínio do problema. No entanto, esta abordagem requer um grande conjunto de treinamento para obter um desempenho satisfatório (GUERRA, 2006).

6.2 DEFINIÇÃO DO RAIO DA FUNÇÃO DE BASE RADIAL

O valor definido para o raio σ das funções de base radial, influencia diretamente na precisão da rede RBF. Se este parâmetro assumir um valor muito alto, a resposta da rede aumenta e sua precisão diminui, ou seja, a rede generaliza demais (*underfitting)*. Entretanto, se o valor do raio for muito pequeno, a precisão é elevada apenas para padrões de entrada muito próximos dos centros, ou seja, a rede não generaliza bem (CASTRO, 2001).

O valor do raio pode ser obtido por meio da equação (1) que consiste em uma fração da maior distância euclidiana entre os centros de todos os neurônios (HAYKIN, 2001).

$$\sigma = \frac{dist_{\max}(c_i, c_j)}{\sqrt{2H}}, \forall i \neq j \tag{1}$$

Onde,

$$dist_{\max}(c_i, c_j) = \max_{\forall i \neq j} \left\{ \left\| c_i - c_j \right\| \right\} \tag{2}.$$

Cada uma das funções de base pode assumir diferentes valores para o raio, entretanto, optou-se por utilizar o mesmo raio para cada neurônio da camada, pois de acordo com Castro (2001) esta abordagem permite que a rede mapeie os padrões de entrada desde que haja uma quantidade suficiente de funções de base.

6.3 MAPEAMENTO DO ESPAÇO NÃO LINEAR

O conjunto de funções de base radial dos neurônios da camada oculta é responsável pela transformação do espaço de entrada, o qual permite que a rede seja capaz de mapear com exatidão o vetor de entrada para a saída desejada (THEODORIDIS; KOUTROUMBAS, 2006, tradução nossa).

Para cada vetor de entrada x apresentado à rede na iteração t, calcula-se a ativação da *h*-ésima função de base por meio da expressão:

$$u_i(t) = \| x(t) - c_i(t) \|, i = 1, \ldots H \tag{3}$$

Onde H é o número de funções de base da camada oculta e c_i representa o vetor de centros de cada função i, definido por:

$$c_i = \lfloor c_{i1}, c_{i2}, c_{ij} \ldots c_{ip} \rfloor \tag{4}$$

Em que c_{ij} é o peso que conecta a *j*-ésima entrada a i-ésima função de base.

A função $\|\bullet\|$ utilizada representa a norma euclidiana definida pela raiz quadrada da soma dos quadrados das diferenças entre dois pontos.

$$u_1(0) = \sqrt{\sum_{i=1}^{H} [x(t) - c_i(t)]^2} \tag{5}$$

Segundo Bishop (1995) a saída de cada neurônio da camada oculta pode ser obtida pela função Gaussiana, que é a função de base mais comumente utilizada em redes RBF. Esta função é estritamente positiva e definida por:

$$\varphi_i(t) = \exp\left(-\frac{u_i^{\,2}(t)}{2\sigma_i^{\,2}}\right) \tag{6}$$

Onde, σ_i representa o raio da função de base e define o espalhamento dos dados representados pela função de base radial em torno do seu centro. Sendo assim, a função gaussiana que é uma função local, que fornece respostas significativas apenas para as entradas inclusas no campo receptivo σ de cada centro (GUERRA, 2006).

Pode-se observar que de acordo com a equação (6), o neurônio i fornece resposta máxima ($\varphi_i(t) \approx 1$), para vetores de entrada próximos do seu centro c_i. Deste modo, cada neurônio da camada oculta tem seu próprio campo receptivo no espaço de entrada, que é uma região com centro c_i de tamanho proporcional a σ_i sendo ativado sempre que o vetor de entrada estiver perto o suficiente de seu centro (BISHOP, 1995, tradução nossa).

6.4 PROJETO DA CAMADA DE SAÍDA

A camada de saída é responsável por fazer uma combinação linear das funções de base não-lineares e determinar os pesos de saída m. Segundo Haykin (2001) em problemas

de classificação, pode-se optar pelo uso da regra de aprendizagem do *perceptron*[7] simples, onde a saída do k-ésimo neurônio é dada por:

$$o_k(t) = \begin{Bmatrix} 1, & u_k(t) \geq 0 \\ 0, & u_k(t) < 0 \end{Bmatrix} \tag{7}$$

Onde $u_k(t)$ é definido pela equação (10):

$$U_k(t) = \sum_{i=1}^{H} m_{ki}(t)\varphi_i(t) \tag{8}$$

H representa a quantidade de funções de base e considerando que as saídas das funções gaussianas são as entradas para os neurônios da camada de saída, a equação (9) é utilizada para atualização dos pesos de saída, que só ocorre quando o valor obtido para o erro calculado por meio da equação (10) for diferente de zero (HAYKIN, 2001):

$$m_{ki}(t+1) = m_{ki}(t) + \eta e_k \varphi_i(t) \tag{9}$$

Sendo que η corresponde à taxa de aprendizagem da rede e $e_k(t)$ representa o erro na saída do *k*-ésimo neurônio. A equação (10) descreve o cálculo do erro utilizado na atualização dos pesos sinápticos (BISHOP, 1995, tradução nossa).

$$e_k(t) = d_k(t) - o_k(t) \tag{10}$$

Onde $d_k(t)$ é uma componente do vetor de saídas desejadas. A cada apresentação de um vetor de entrada, as saídas das funções φ_i são recalculadas e os seus valores são utilizados na equação (11) para atualizar os pesos de saída m_{ki}.

Após a apresentação de todos os vetores de treinamento à rede, o que define o final de uma época de treinamento, deve-se encontrar o valor para o erro médio quadrático dado pela equação:

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(e_k(t)\right)^2 \tag{11}$$

[7] Modelo neural proposto por Frank Rosenblatt, em 1957, utilizado para reconhecimento de padrões (AZEVEDO; BRASIL; OLIVEIRA, 2000).

Sendo que *N* representa a quantidade total de vetores de treinamento.

Durante a fase de teste do classificador, o vetor de entrada apresentado de classe desconhecida, é associado à classe representada pelo neurônio y_k que gerar maior valor de saída o_k. Neste contexto, as redes RBF podem ser entendidas como modelos matemáticos que realizam aproximação de funções por meio da combinação linear de funções de base gaussianas (BISHOP, 1995, tradução nossa).

Entendido o funcionamento das redes neurais de base radial, alguns exemplos da utilização deste modelo computacional para a tarefa de classificação e reconhecimento de padrões são apresentados.

7 EXEMPLOS DA UTILIZAÇÃO DE REDES NEURAIS COM FUNÇÃO DE BASE RADIAL

Redes neurais têm sido amplamente utilizadas com sucesso em diferentes áreas do conhecimento como, Economia, Medicina, Engenharia, Robótica, entre outras, devido a sua capacidade de aprendizado.

Por este motivo, são apresentadas algumas aplicações nos quais foram utilizadas redes neurais com função de base radial para tarefas de *data mining*.

7.1 CLASSIFICAÇÃO DE CROMOSSOMOS HUMANOS

Esta pesquisa foi defendida em 1995, por Luis Leomar Todesco como Tese de Doutorado do Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

A pesquisa desenvolvida consiste em uma rede neural com função de ativação de base radial para diminuir a taxa de erro na classificação de cromossomos. Foram testadas diferentes arquiteturas de redes neurais comparando os resultados obtidos com outras abordagens (TODESCO, 1995).

A rede implementada classifica os cromossomos humanos em 24 classes, compondo duas etapas. Na primeira os cromossomos são classificados em 7 grupos (grupos de Denver[8]), e na segunda etapa utilizam-se as saídas da primeira, juntamente com 20 amostras de cromossomos para realizar a classificação em 24 grupos (TODESCO, 1995).

[8] De acordo com o sistema Denver, criado em 1960, os cromossomos humanos são numerados de 1 a 22 e divididos em grupos de A a G (TODESCO, 1995).

De acordo com os testes realizados a rede neural desenvolvida apresentou uma redução considerável da taxa de erro de classificação quando comparada com outros modelos como, por exemplo, redes MLP (TODESCO, 1995).

7.2 DETECÇÃO E DIAGNÓSTICO DE FALHAS EM ROBÔS MANIPULADORES

Renato Tinós desenvolveu esta pesquisa em 1999, como Dissertação de Mestrado da Escola de Engenharia de São Carlos para obtenção do título de Mestre em Engenharia Elétrica. O objetivo da pesquisa era propor um sistema de detecção e diagnóstico de falhas baseado em redes neurais artificiais para robôs manipuladores.

O sistema desenvolvido utiliza duas redes neurais: a primeira é uma rede MLP que tem como função reproduzir o comportamento dinâmico do manipulador, e a segunda tem por objetivo classificar os sinais produzidos pela diferença entre as saídas da primeira rede utilizando redes RBF (TINÓS, 1999).

O sistema desenvolvido é capaz de detectar e isolar as falhas que ocorrem em trajetórias não-vistas e em instantes diferentes dos padrões treinados, porém deve-se observar que os resultados poderiam ser melhores se o número de padrões utilizados na rede RBF fosse maior (TINÓS, 1999).

7.3 DETECÇÃO INTELIGENTE DE SINAIS DIGITAIS

Este artigo foi publicado em 1999 na IV Conferência Brasileira de Redes Neurais na cidade de São José dos Campos (São Paulo) e foi desenvolvido por Marcelo Augusto Costa Fernandes, Adrião Duarte Doria Neto e João Batista Bezerra, sendo que todos faziam

parte do Laboratório de Engenharia da Computação e Automação da Universidade Federal do Rio Grande do Norte.

A aplicação desenvolvida utiliza uma rede neural com função de base radial para deteção de sinais em sistemas de comunicação digital, chamada detetor RBF. Este detector está dividido em dois blocos: o primeiro é o bloco classificador que utiliza uma rede RBF para classificar as coordenadas estimadas, já o segundo, chamado bloco decisor, transforma as saídas da rede RBF nas respectivas palavras binárias associadas aos sinais (FERNANDES; DORIA NETO; BEZERRA, 1999).

Foram realizados testes com simulações de ruídos e constatou-se que a grande vantagem da rede RBF está na velocidade de treinamento, concluindo-se que o detetor RBF pode ser um bom substituto dos detetores MLP (FERNANDES; DORIA NETO; BEZERRA, 1999).

7.4 CLASSIFICAÇÃO DE IMAGENS

Este artigo foi publicado em 1998, no Simpósio Brasileiro de Sensoriamento Remoto em Santos (São Paulo). Foi desenvolvido por Waleska Nishida e Lia C. Bastos, ambas do Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Universidade Federal de Santa Catarina.

Foi desenvolvida uma aplicação de um classificador híbrido de imagens utilizando redes RBF treinadas com o algoritmo de Kohonen, sendo que a entrada da rede são os vetores de níveis de cinza da imagem e para inicialização dos pesos da rede foram utilizados os *pixels* de cada classe (NISHIDA; BASTOS, 1999).

Duas imagens foram utilizadas para testar o funcionamento da rede. A arquitetura da rede proposta diminuiu o tempo de processamento e apresentou resultados satisfatórios

para pequenas amostras de treinamento, devido a sua capacidade de generalização que reconhece padrões nunca apresentados à rede (NISHIDA; BASTOS, 1999).

7.5 RECONHECIMENTO FACIAL

Meng Joo Er, Shiqian Wu, Juwei Lu e Hock Lye Toh publicaram este artigo em 2002, para a Conferência de Redes Neurais do *Institute of Electrical and Electronics Engineers* (IEEE) propondo a utilização de classificadores RBF no reconhecimento facial.

Optou-se pelo uso de redes neurais de base radial devido ao seu melhor desempenho diante de pequenos conjuntos de treinamento de alta dimensão, que é o caso do problema de reconhecimento de faces, o qual fornece poucos padrões de amostras, porém cada padrão possui muitas características (THOMAZ; FEIOSA; VEIGA, 2002, tradução nossa).

Foi utilizada uma base de dados com 250 imagens faciais de 25 pessoas para a fase de treinamento e outra de igual tamanho para testar o desempenho do protótipo com diferentes expressões e detalhes faciais dos indivíduos. Os resultados mostraram que o classificador RBF apresentou taxa de reconhecimento de 98% para uma pequena quantidade de imagens e melhor desempenho quando comparados a outros classificadores convencionais (THOMAZ; FEIOSA; VEIGA, 2002, tradução nossa).

8 O MÉTODO DE REDES NEURAIS COM FUNÇÃO DE ATIVAÇÃO DE BASE RADIAL NA SHELL ORION DATA MINING ENGINE

A *Shell* Orion é um projeto acadêmico que desenvolve uma ferramenta gratuita de *data mining,* agregando conhecimentos à comunidade acadêmica, relativos à descoberta de conhecimento.

O desenvolvimento do algoritmo de rede neural com função de ativação de base radial contribuiu com o aumento das funcionalidades da ferramenta para a tarefa de classificação para a tarefa de classificação.

A fim de testar o funcionamento do algoritmo desenvolvido, foram utilizadas diferentes bases de dados escolhidas conforme a pesquisa desenvolvida por cada acadêmico. Nos testes realizados neste trabalho, utilizou-se a base de dados contendo registros referentes a plantas da família das Íridáceas.

8.1 BASE DE DADOS

A base de dados utilizada para testes na rede RBF implementada na *Shell* Orion *Data Mining Engine* é composta de dados não-lineares contendo dados referentes a três tipos de plantas da família das Iridáceas: setosa, versicolor e virgínica (Figura 17).

Íris Setosa Íris Versicolor Íris Virgínica

Figura 17. Imagens da iridáceas: setosa, versicolor e virgínica

Esta base de dados foi utilizada em três outras pesquisas relacionadas à *Shell* Orion: na a tarefa de classificação pelo algoritmo CART implementado pela Bacharel em

Ciência da Computação Lidiane Rosso Raimundo em 2005 e pelo algoritmo C4.5 desenvolvido pelo Bacharel em Ciência da Computação Ricardo Linemburger Mondardo em 2009; e na tarefa de clusterização pelo algoritmo Gath-Geva implementado pelo também Bacharel em Ciência da Computação Daniel Perego em 2009.

A base de dados das iridáceas está disponível gratuitamente no site: http://archive.ics.uci.edu/ml/datasets.html da Universidade da Califórnia na cidade de Irvine, Estados Unidos, sendo que totaliza 150 registros distribuídos igualmente entre as três classes de plantas e contendo os seguintes atributos numéricos:

a) ***sepal_lenght:*** comprimento da sépala;

b) **sepal_width:** largura da sépala;

c) **petal_lenght:** comprimento da pétala;

d) **petal_width:** largura da pétala.

8.2 METODOLOGIA

A metodologia de desenvolvimento do algoritmo de redes neurais com função de ativação de base radial na *Shell* Orion *Data Mining Engine* é composta pelas seguintes etapas: levantamento bibliográfico; modelagem do módulo de Redes Neurais RBF na *Shell* Orion; demonstração matemática do funcionamento do algoritmo; implementação e realização de testes, que serão descritas a seguir.

8.2.1 Modelagem do Módulo de Redes RBF na *Shell* Orion

A modelagem do módulo de classificação com redes RBF teve início com a construção dos diagramas de caso de uso, atividades e seqüência utilizando os padrões da *Unified Modeling Language*[9] (UML) por meio da ferramenta gratuita JUDE[10].

No diagrama de caso de uso (Figura 18) é possível observar as atividades realizadas pelo usuário e pelo sistema.

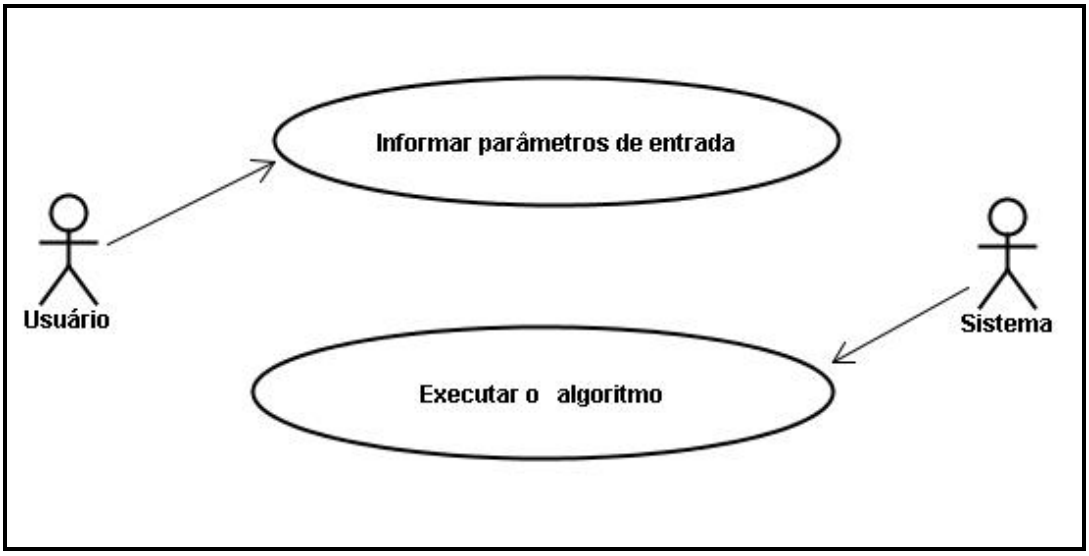

Figura 18. Diagrama de caso de uso

O diagrama de atividades demonstra o fluxo de tarefas executadas pelo usuário e pelo sistema (Figura 19):

a) **informa parâmetros de entrada:** o usuário informa os parâmetros de entrada necessários para execução do algoritmo;

b) **solicita execução do algoritmo:** após definir os parâmetros, o usuário solicita ao sistema a execução da rede RBF;

c) **processa a rede RBF:** o sistema inicia o processamento dos dados e execução do algoritmo;

[9] Linguagem desenvolvida para modelagem visual de classes, relacionamentos e métodos de uma aplicação, que visa facilitar a documentação e simplificar o entendimento de seus módulos (GUEDES, 2008).
[10] Disponível para download no site http://jude.change-vision.com/jude-web/download/index.html

d) **visualiza os resultados:** os resultados gerados pelo algoritmo são apresentados ao usuário.

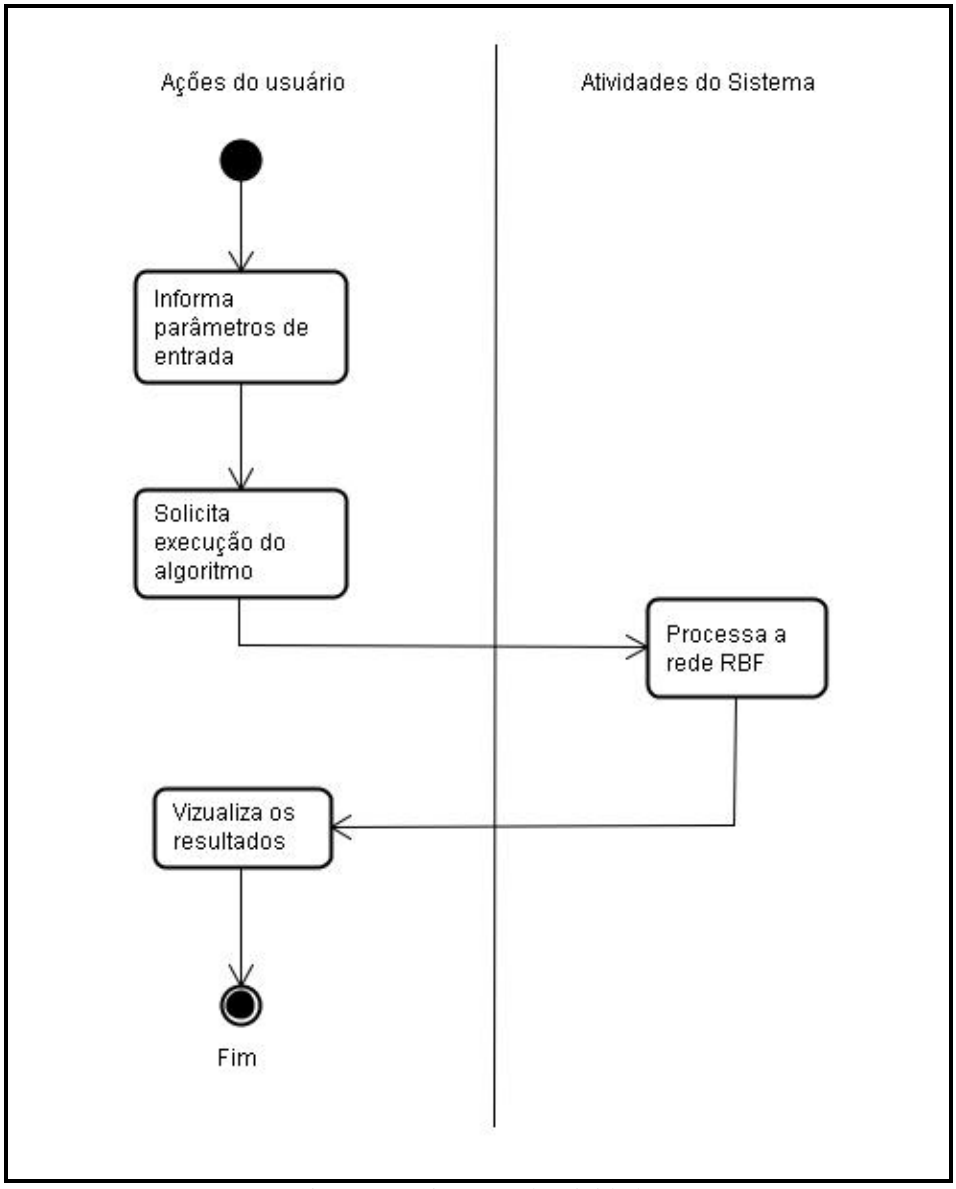

Figura 19. Diagrama de atividades

Na Figura 20 podem ser observadas as interações entre usuário e sistema representadas no diagrama de seqüência. O usuário solicita a interface do método de redes RBF e, após inserir os parâmetros de entrada, o algoritmo é executado gerando os resultados para o usuário.

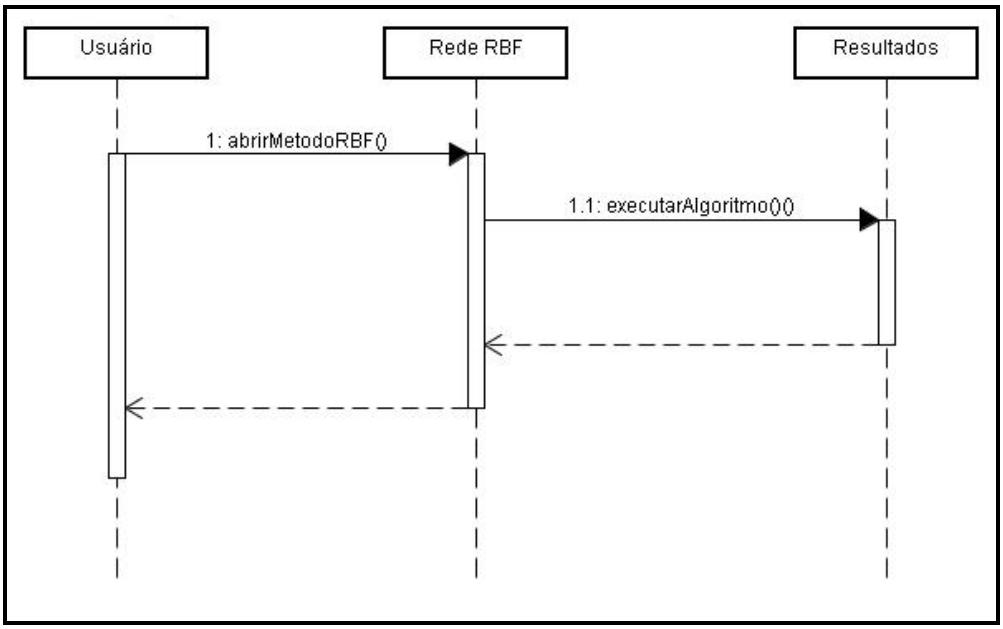

Figura 20. Diagrama de seqüência

Concluído o projeto UML do módulo de redes RBF para classificação na *Shell* Orion, iniciou-se o desenvolvimento da modelagem matemática.

8.2.2 Demonstração Matemática do Funcionamento das Redes com Função de Ativação de Base Radial

Nesta etapa da pesquisa desenvolveu-se a demonstração matemática do funcionamento de uma rede neural com função de ativação de base radial para a tarefa de classificação de padrões, visando facilitar o entendimento dos cálculos realizados durante o processamento do algoritmo e também simplificar a etapa de implementação do módulo.

O Capítulo 5 do livro *Neural Networks for Pattern Recognition* (BISHOP, 1995), e do livro Redes Neurais: princípios e práticas (HAYKIN, 2001), foram as principais fontes de embasamento para os conceitos e formalismos matemáticos descritos nesta modelagem.

A fim de demonstrar a transformação do espaço de entrada não-linear que a rede RBF executa, a base de dados utilizada nesta demonstração é referente ao problema do

XOR[11]. Esta base é composta por quatro padrões (x_1, x_2, x_3, x_4) constituídos por três atributos (x, y, d) cada um, como mostra a Tabela 3.

Tabela 3. Base de dados utilizada na modelagem do algoritmo

Atributos	x_1	x_2	x_3	x_4
x	1	0	0	1
y	1	1	0	0
d[12]	0	1	0	1

Considerando que a modelagem completa do algoritmo é muito extensa, nesta seção foram apresentados apenas os cálculos parciais, porém estes proporcionam a compreensão do mesmo.

Definidos os padrões de entrada, para que se possa iniciar a execução do algoritmo, é necessário determinar os parâmetros de entrada descritos a seguir:

a) **taxa de aprendizagem:** determina a capacidade de aprendizado da rede e a intensidade de alteração dos pesos sinápticos que influencia na velocidade do aprendizado. Uma taxa muito baixa torna o aprendizado lento, bem como uma taxa muito alta impede a convergência do processo de treinamento. Deste modo, este parâmetro compreende um valor entre 0 e 1, neste caso optou-se pelo valor médio 0.5 apenas para demonstração do cálculo (HAYKIN, 2001);

b) **quantidade de classes:** este parâmetro define o número de classes existentes no problema, e corresponde à quantidade de neurônios da camada de saída. Nesta demonstração matemática aborda-se um problema com somente duas classes, nestes casos segundo Bishop (1995) apenas um neurônio de saída é suficiente, pois se um vetor de entrada não pertence à classe representada pelo neurônio de saída, conseqüentemente pertencerá à outra;

[11] Este problema consiste em construir um classificador que produza saída 0 para os padrões (1,1) e (0,0), e saída 1 para (0,1) e (1,0) (HAYKIN, 2001).
[12] Saída desejada para o padrão de entrada apresentado.

c) **quantidade de centros:** representa o número de funções de base radial da camada oculta da rede, neste caso foram utilizados dois neurônios ocultos. Deve-se considerar que a quantidade de funções de base deve ser sempre menor que a quantidade total de padrões, a fim de evitar *overfitting* (THEODORIDIS; KOUTROUMBAS, 2006, tradução nossa).

A demonstração matemática da rede RBF foi dividida em cinco etapas a fim de facilitar o entendimento dos cálculos realizados pelo algoritmo:

a) **seleção dos centros das funções de base:** um subconjunto dos dados de treinamento é atribuído aos vetores centro, sendo que a quantidade de funções é um parâmetro definido pelo usuário;

b) **definição do raio de abrangência:** calcula-se a área de sensibilidade da função de base em relação ao seu centro;

c) **cálculo da ativação dos neurônios ocultos:** define-se o grau de ativação de cada neurônio da camada oculta;

d) **mapeamento do espaço não-linear:** as funções de base radial executam a transformação dos dados de entrada não-lineares em saídas lineares;

e) **ajuste dos pesos de saída:** os pesos de saída da rede são atualizados e utilizados na próxima iteração.

8.2.2.1 Seleção dos Centros das Funções de Base

Na primeira fase do processamento do algoritmo, deve-se atribuir valores aos vetores centro das unidades de base radial. A localização dos centros é definida aleatoriamente a partir do conjunto de treinamento, ou seja, é um subconjunto dos vetores de entrada.

Seguindo a abordagem de seleção aleatória, o primeiro passo é definir valores randômicos para os centros a partir do conjunto de entrada (Tabela 3). Neste caso utilizaram-se dois vetores centros descritos na Tabela 4.

Tabela 4. Inicialização dos centros

Atributos	$c_1(0)$	$c_2(0)$
x	1	0
y	1	0

Após a inicialização dos centros, deve-se calcular o raio das funções de base radial, e posteriormente iniciar o processamento dos neurônios ocultos.

8.2.2.2 Definição do Raio de Abrangência

Nesta etapa é necessário determinar os raios das funções de base radial. O método utilizado para tal, define o mesmo raio para cada neurônio e pode ser encontrado por meio da seguinte equação:

$$\sigma = \frac{dist_{\max}(c_{i,}c_j)}{\sqrt{2H}}$$

De acordo com a equação, o raio σ é equivalente à maior distância entre os centros de todos os neurônios ocultos, dividida pela raiz quadrada de duas vezes a quantidade de neurônios H. Esta distância pode ser obtida por meio do cálculo da distância euclidiana entre dois vetores centros c_i e c_j, definida pela seguinte equação:

$$dist_{\max}(c_i, c_j) = \max\left\{\left\|c_i - c_j\right\|\right\}$$

Calculando a distância entre o centro 1 (c_1) e o centro 2 (c_2), tem-se:

$$dist = \left\{\left\|c_1 - c_2\right\|\right\}$$

$$dist = \left\|\begin{bmatrix}1\\1\end{bmatrix} - \begin{bmatrix}0\\0\end{bmatrix}\right\|$$

$$dist = \left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|$$

$$dist = \sqrt{\sum_{i=1}^{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^2}$$

$$dist = \sqrt{(1^2 + 1^2)}$$

$$dist = 1.41421356$$

Considerando que o exemplo utilizado só possui dois centros, a distância máxima encontrada foi:

$$dist_{\max} = 1.41421356$$

Substituindo os valores encontrados, o cálculo do raio é definido por:

$$\sigma = \frac{1.41421356}{\sqrt{2 \cdot 2}}$$

$$\sigma = 0.707168$$

Após definir o raio de sensibilidade da função de base radial, deve-se definir a ativação de cada neurônio oculto.

8.2.2.3 Ativação dos Neurônios Ocultos

Cada atributo da base de dados possui um respectivo valor para o centro, estes valores foram definidos aleatoriamente e podem ser observados na Tabela 4. A ativação u dos neurônios ocultos é calculada para cada vetor de entrada apresentado à rede e obtida por meio da distância euclidiana definida por:

$$u_i(t) = \| x(t) - c_i(t) \|$$

Esta equação define a distância euclidiana entre o vetor de entrada x e o centro c na iteração t que representa cada vetor de entrada apresentado, portanto para cada neurônio oculto tem-se:

$$u_1(0) = \| x(0) - c_1(0) \|$$

$$u_1(0) = \left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|$$

$$u_1(0) = \left\| \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\|$$

$$u_1(0) = \sqrt{\sum_{i=1}^{2} \begin{bmatrix} 0 \\ 0 \end{bmatrix}^2}$$

$$u_1(0) = \sqrt{(0^2 + 0^2)}$$

$$u_1(0) = 0$$

Utilizando o mesmo procedimento obtêm-se seguinte valor de ativação do segundo neurônio oculto.

$$u_2(0) = \| x(0) - c_2(0) \|$$

$$u_2(0) = \left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\|$$

$$u_2(0) = \left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|$$

$$u_2(0) = \sqrt{\sum_{i=1}^{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^2}$$

$$u_2(0) = \sqrt{(1^2 + 1^2)}$$

$$u_2(0) = 1.414214$$

8.2.2.4 Mapeamento do Espaço Não-Linear

Definidos os vetores centro, ativação dos neurônios e raio das funções de base, é possível calcular as saídas de cada neurônio oculto por meio da equação:

$$\varphi_i(t) = \exp\left(-\frac{u_i^{\,2}(t)}{2\sigma_i^{\,2}}\right)$$

Este cálculo executa o exponencial da ativação u do neurônio oculto ao quadrado, dividida por duas vezes o raio σ do neurônio ao quadrado. Substituindo os valores, tem-se para o primeiro padrão de entrada x_1:

$$\varphi_1(0) = \exp\left(-\frac{0^2}{2 \cdot 0.707108^2}\right)$$

$$\varphi_1(0) = \exp(0)$$

$$\varphi_1(0) = 1$$

Realizando o mesmo procedimento para os outros neurônios, obtém-se:

$$\varphi_2(0) = \exp\left(-\frac{1.414214^2}{2 \cdot 0.707108^2}\right)$$

$$\varphi_2(0) = \exp(-1)$$

$$\varphi_2(0) = 0.135335$$

A Tabela 5 representa as saídas das funções de base radial dos dois neurônios ocultos para cada padrão de entrada da base de dados.

Tabela 5. Saídas das funções gaussianas para o problema do XOR

Padrão de entrada x	φ_1	φ_2
[1,1]	1	0,1353
[0,1]	0,3678	0,3678
[0,0]	0,1353	1
[1,0]	0,3678	0,3678

Com base nas saídas dos neurônios ocultos, pode-se observar que após a transformação do espaço de entrada não-linear realizado pela rede RBF, o problema do XOR tornou-se linearmente separável como demonstra a Figura 21.

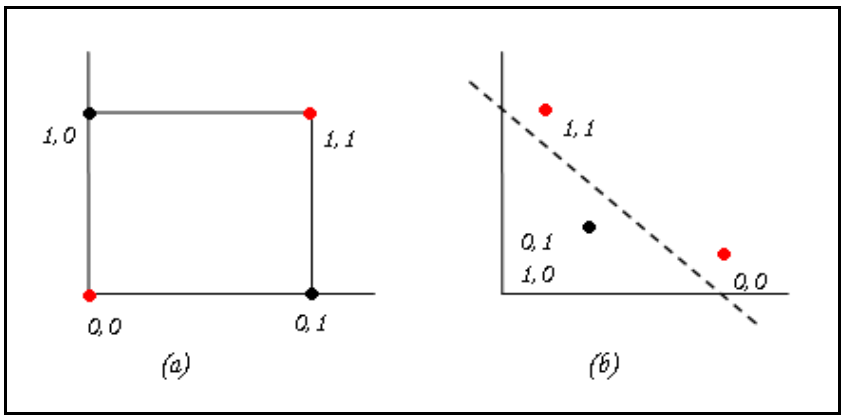

Figura 21. Espaço não-linear (a) e linear (b) após processamento da rede RBF
Fonte: Adaptado de Haykin, S. (2001)

As saídas lineares geradas pelo mapeamento do espaço não-linear, permitem que o ajuste de pesos da camada de saída seja atualiza por meio da regra do *perceptron* simples.

8.2.2.5 Ajuste dos Pesos de Saída

Após a transformação do espaço de entrada, a execução da rede segue com a determinação dos pesos de saída da rede. Considerando que a camada de saída processa um problema linearmente separável, optou-se pelo uso da regra de aprendizagem do *perceptron* simples (HAYKIN, 2001). Neste caso a resposta o de cada neurônio k da camada de saída pode ser obtida por meio da seguinte regra:

$$o_k(t) = \left\{ \begin{array}{l} 1, \quad u_k(t) \geq 0 \\ 0, \quad u_k(t) < 0 \end{array} \right\}$$

Onde:

$$u_k(t) = \sum_{i=1}^{H} m_{ki}(t)\varphi_i(t)$$

Considerando que os pesos m são inicializados com zero, e substituindo os valores encontrados para as funções gaussianas φ, tem-se:

$$u_1(0) = \left[\left(m_{11}(0)\varphi_1(0)\right) + \left(m_{12}(0)\varphi_2(0)\right)\right]$$

$$u_1(0) = \left[\left(0 \cdot 1\right) + \left(0 \cdot 135335\right)\right]$$

$$u_1(0) = 0$$

Encontrado o valor de u pode-se definir a saída o_k, considerando que $u_1 \geq 0$, então:

$$o_k(t) = 1$$

Durante a fase de treinamento, a saída do neurônio associado à classe a qual o vetor de entrada atual pertence deverá ser 1, enquanto as saídas de outros neurônios deverá ser 0. A Tabela 6 demonstra a matriz de saídas desejadas para o problema demonstrado.

Tabela 6. Matriz de saídas desejadas

Padrões de entrada x	**Classe 1**	**Classe 2**
[1,1]	1	0
[0,1]	0	1
[0,0]	1	0
[1,0]	0	1

A atualização dos pesos sinápticos é realizada de acordo com o valor obtido pra o erro e na saída de cada neurônio o, calculado por meio da diferença entre a saída desejada d e a saída real da rede o. Para o padrão x_1 tem-se:

$$e_k(t) = d_k(t) - o_k(t)$$

$$e_1(0) = d_1(0) - o_1(0)$$

$$e_1(0) = (0 - 1)$$

$$e_1(0) = -1$$

Deste modo, o ajuste dos pesos de saída é realizado por meio da seguinte equação:

$$m_{ki}(t+1) = m_{ki}(t) + \eta e_k \varphi_i(t)$$

$$m_{11}(0+1) = m_{11}(0) + \eta e_1 \varphi_1(0)$$

$$m_{11}(1) = 0 + 0.5 \cdot (-1) \cdot 1$$

$$m_{11}(1) = -0.5$$

Onde η representa a taxa de aprendizado da rede definida como 0.5 para esta demonstração. Executando o mesmo procedimento para outros pesos sinápticos, obtém-se:

$$m_{12}(0+1) = m_{12}(0) + \eta e_1 \varphi_2(0)$$

$$m_{11}(1) = 0 + 0.5 \cdot (-1) \cdot 0.1353$$

$$m_{12}(1) = -0.06766$$

Após o ajuste dos pesos sinápticos serem executados, a iteração deve ser incrementa em um. Neste momento um novo padrão de entrada é apresentado à rede, que repete todo o seu processamento utilizando os parâmetros atualizados.

Ao final de cada época de treinamento o erro médio de aproximação deve ser calculado a partir da taxa de erro de cada iteração, neste caso, os valores encontrados para a primeira época de treinamento [13]podem ser observados na Tabela 7.

Tabela 7. Taxas de erro da rede na primeira época

Entrada	**Taxa de Erro (e)**
x1	-1
x2	1
x3	-1
x4	0

Pode-se então encontrar o valor do erro médio da primeira época de treinamento, por meio da soma de todos os erros ao quadrado, dividida pela quantidade total de padrões apresentados.

[13] Uma época de treinamento consiste na execução completa de todos os padrões de treinamento pela rede (HAYKIN, 2001).

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(e_k(t)\right)^2$$

$$E = \frac{1}{4}\left((-1)+1+(-1)+0\right)^2$$

$$E = 0.25$$

O erro médio quadrado deve ser calculado ao final de cada época até a convergência, situação em que o valor obtido para o erro médio quadrático seja menor que o valor máximo permitido, ou até que o algoritmo execute o máximo de épocas permitidas informadas pelo usuário.

8.2.3 Implementação e Realização de Testes

A rede RBF foi implementada no módulo de classificação da *Shell* Orion *Data Mining Engine* por meio da linguagem de programação Java e ambiente de programação Netbeans 6.8.

A *Shell* Orion possibilita a conexão com *drivers* de diferentes bancos de dados que podem ser adicionados na ferramenta por meio do menu *Arquivos,* submenu *Conexões.* Nos testes realizados nesta pesquisa optou-se pelo uso do MySQL 5.1, disponível gratuitamente para download em: http://dev.mysql.com/downloads/mysql.

Concluída a inserção do *driver* pode-se configurar a conexão com a base de dados pelo menu *Arquivo,* submenu *Conectar.* Neste caso, para testar o funcionamento do classificador RBF, foi utilizada a base de dados das Iridáceas. Após efetuada a conexão com a base, torna-se possível acessar os algoritmos implementados na ferramenta, pelo menu *Data Mining,* submenu *Classificação* e método *RBF* (Figura 22).

Figura 22. Acesso ao classificador RBF na *Shell* Orion

Para executar a tarefa de classificação utilizando RN-RBF é necessário definir alguns parâmetros da rede (Figura 23):

a) **quantidade de classes:** número de classes que o algoritmo irá identificar, o valor informado não pode ser maior que a quantidade real de classes;

b) **quantidade de épocas:** quantidade máxima de vezes em que o conjunto de treinamento é apresentado à rede. O algoritmo executa a quantidade máxima de épocas informadas apenas quando não atinge a convergência pelo erro médio. Este valor varia de acordo com o tamanho da base de dados, quando maior a quantidade de dados de treinamento maior é o tempo de convergência;

c) **taxa de aprendizagem:** taxa de atualização dos pesos sinápticos que corresponde ao grau de aprendizagem da rede, quando este valor é muito baixo o aprendizado se torna muito lento, enquanto que um valor muito alto impede a convergência do algoritmo;

d) **quantidade de centros ou funções de base:** número de funções de base radial que irá compor a rede, este valor não pode ser muito baixo para não

comprometer o tempo de aprendizado da rede e nem muito alto para que a rede não memorize os dados de treinamento;

e) **atributos de entrada:** atributos da base de dados que serão utilizados como valores de entrada da rede neural.

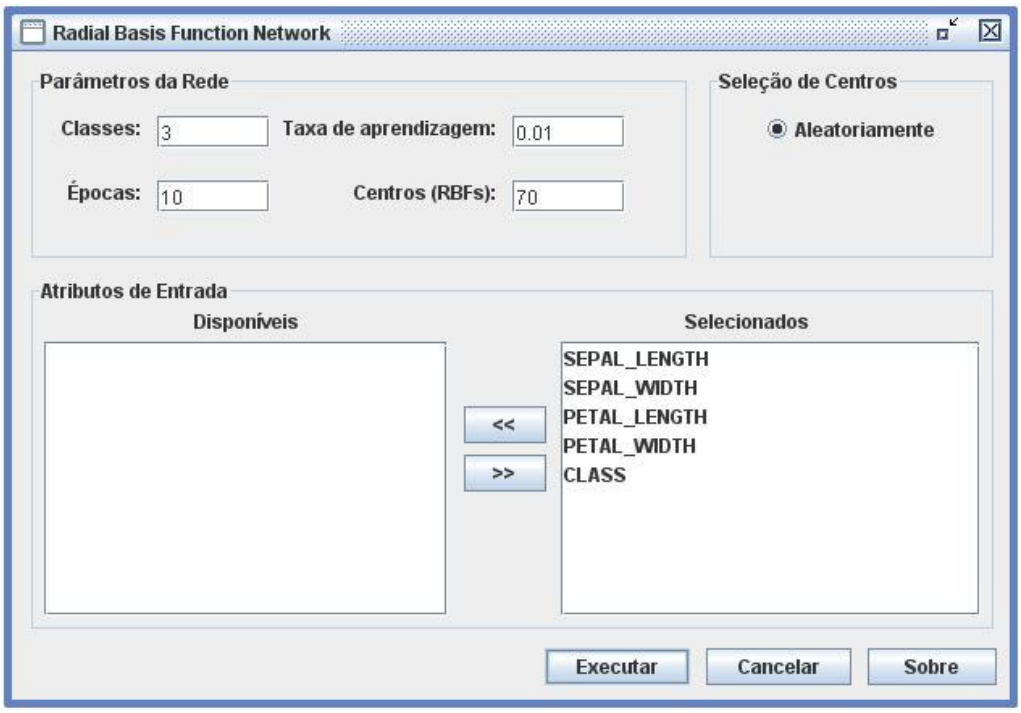

Figura 23. Parâmetros de entrada do classificador RBF da Shell Orion

Os valores informados para os parâmetros de entrada descritos acima interferem diretamente no resultado do algoritmo, sendo possível obter diferentes resultados para uma mesma base de dados de acordo com os valores escolhidos pelo usuário.

A *Shell* Orion permite que os resultados obtidos pelo algoritmo possam ser analisados por meio de resumo, árvore e gráfico. Nas Figura 24 e 25 observa-se o relatório gerado pelo algoritmo contendo um resumo das com informações da execução do mesmo.

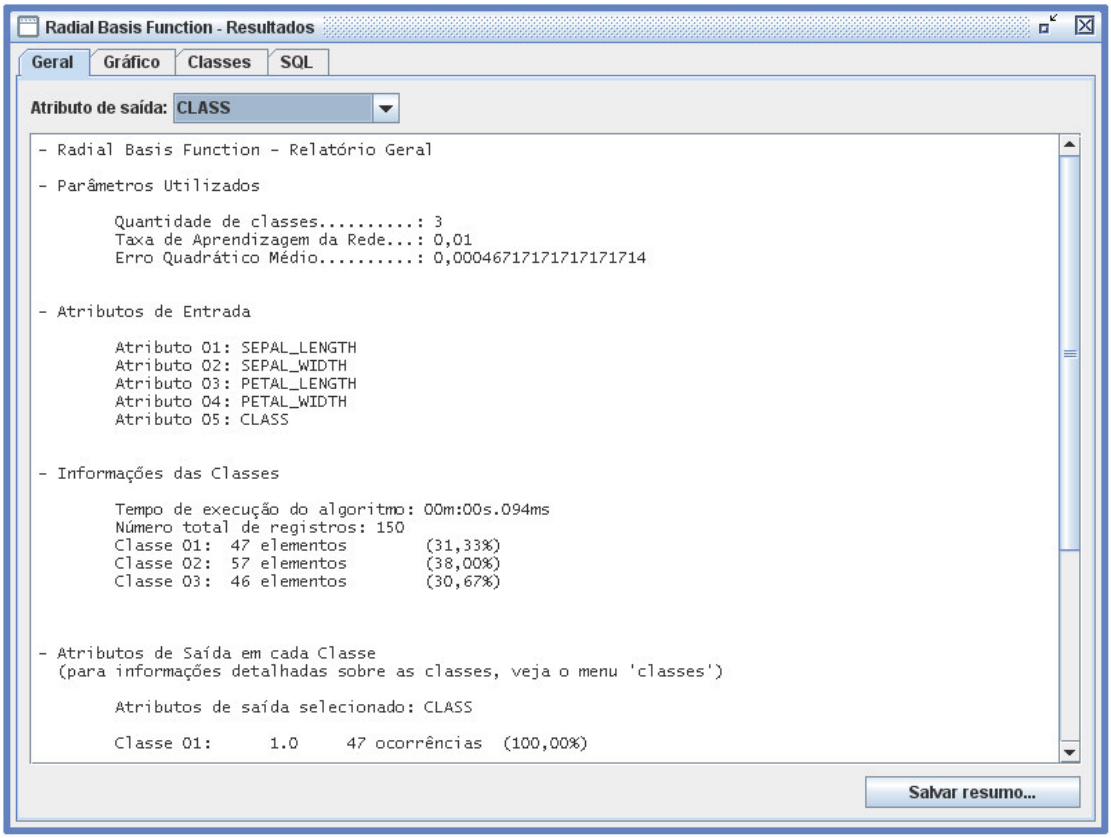

Figura 24. Resumo da classificação por meio da Rede RBF

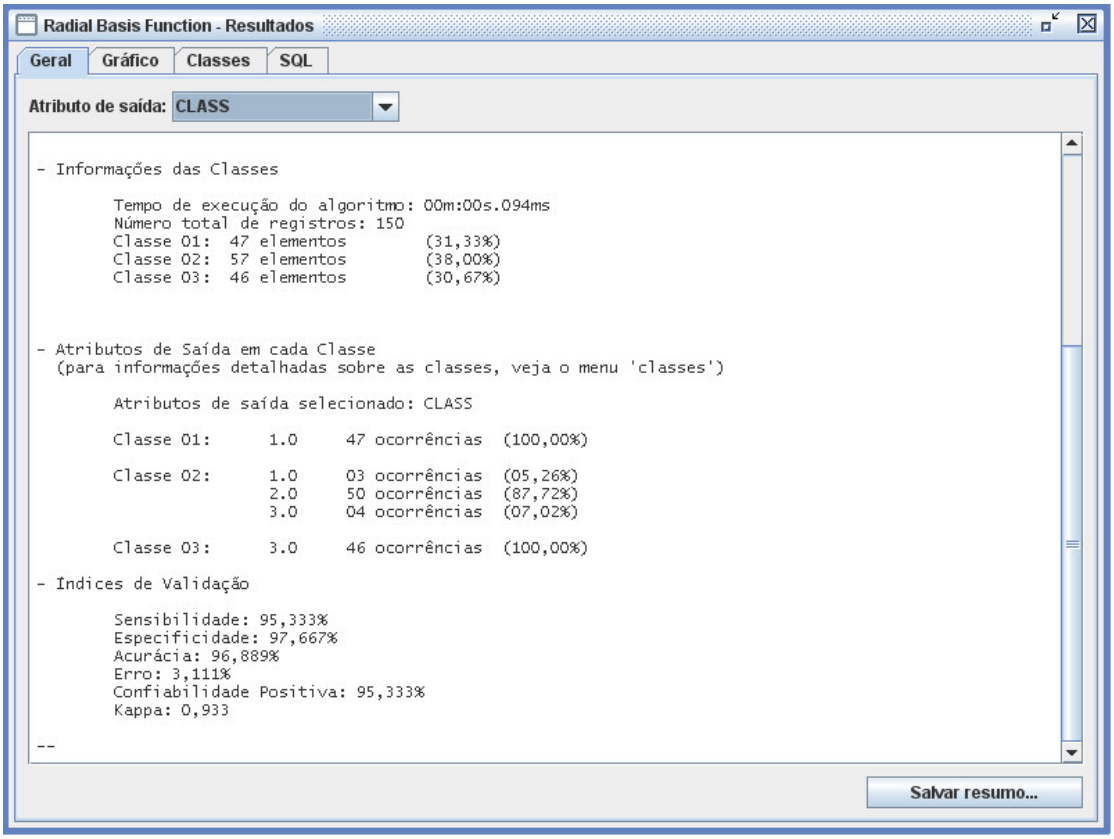

Figura 25. Resumo da classificação por meio da Rede RBF

O resumo mostra a identificação de três classes com 49, 55 e 46 elementos respectivamente, bem como outros detalhes da execução como, por exemplo, a taxa de erro médio, tempo de execução, atributos de entrada utilizados e índices de validação do modelo. A ferramenta também disponibiliza a exportação deste relatório para um arquivo de texto por meio do botão *Salvar resumo.*

A atribuição dos registros para cada classe pode ser facilmente visualizada também em forma gráfica como mostra a Figura 26, onde as classes identificadas são representadas por meio de *Principal Component Analysis*[14] (PCA). O método PCA transforma uma base de dados de *n* dimensões em uma matriz de duas dimensões, possibilitando a projeção dos dados graficamente.

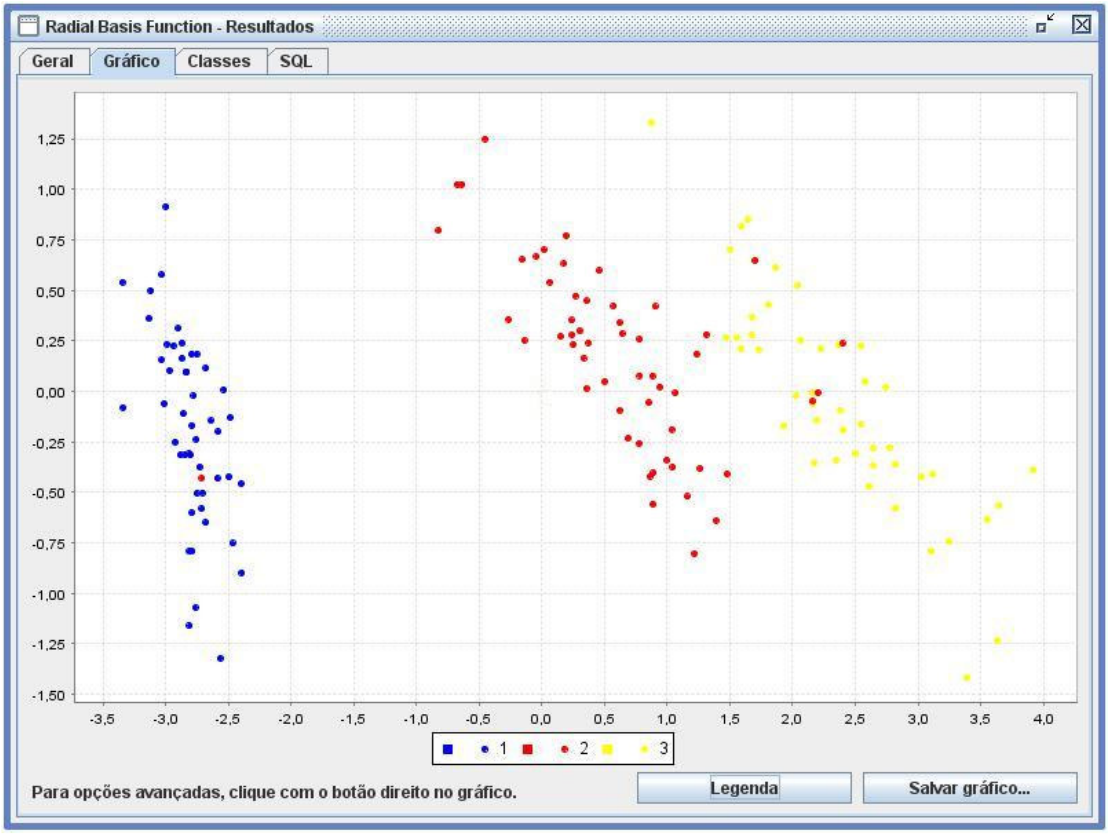

Figura 26. Gráfico gerado pelo classificador RBF

[14] Guia sobre as diferentes implementações do PCA está disponível em: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Informações dos elementos contidos em cada classe podem ser analisadas individualmente por meio de uma estrutura em árvore. Esta estrutura permite a visualização detalhada dos atributos de entrada de cada registro, como demonstra a Figura 27.

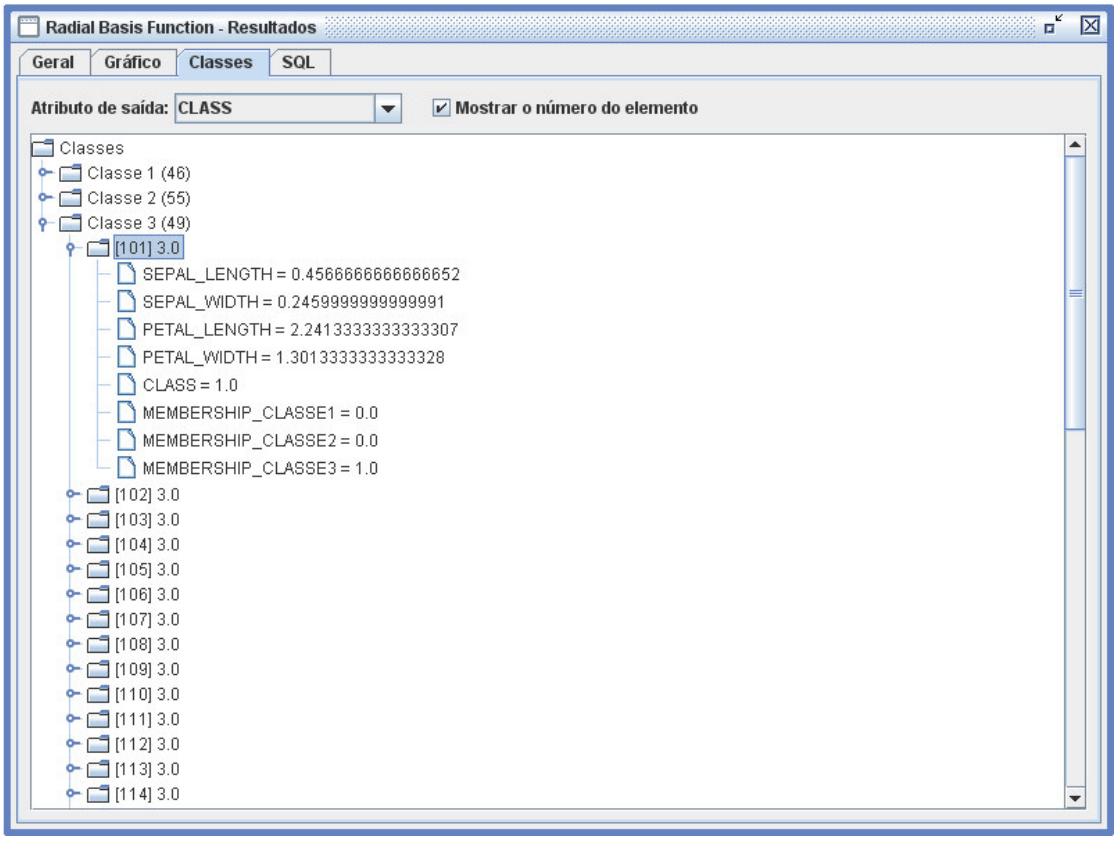

Figura 27. Arvore das classes identificadas pela Rede RBF

Além disso, a ferramenta permite a exportação dos resultados gerados em formato de arquivo SQL (Figura 28), esta funcionalidade permite uma posterior aplicação dos resultados da classificação como entrada para outras tarefas de DM.

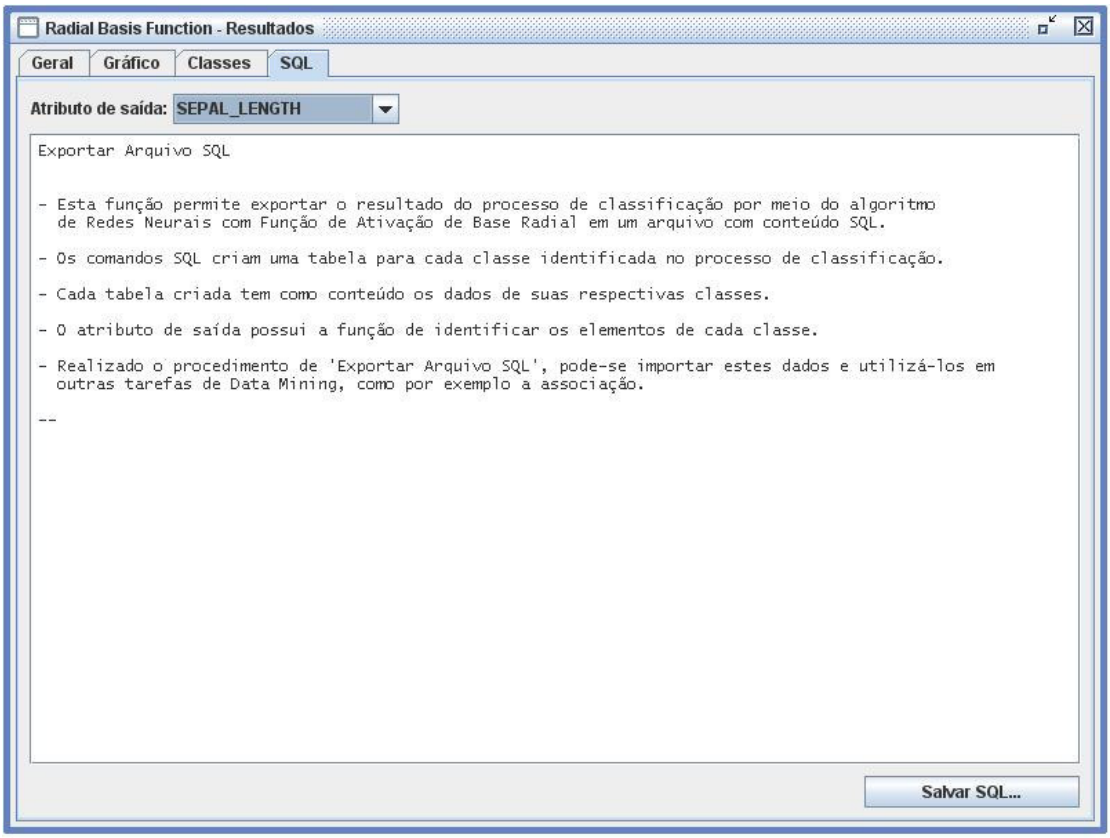

Figura 28. Exportação dos resultados em SQL

A ferramenta também permite que o usuário execute diversas vezes o algoritmo com parâmetros de entrada diferentes, possibilitando a comparação dos resultados encontrados. Além disso, um arquivo de ajuda disponibiliza a documentação necessária para auxiliar o usuário na utilização do classificador RBF e pode ser acessada pelo menu *Ajuda*, submenu *Conteúdo da Ajuda,* tarefa de *Classificação, RBF* (Figura 29).

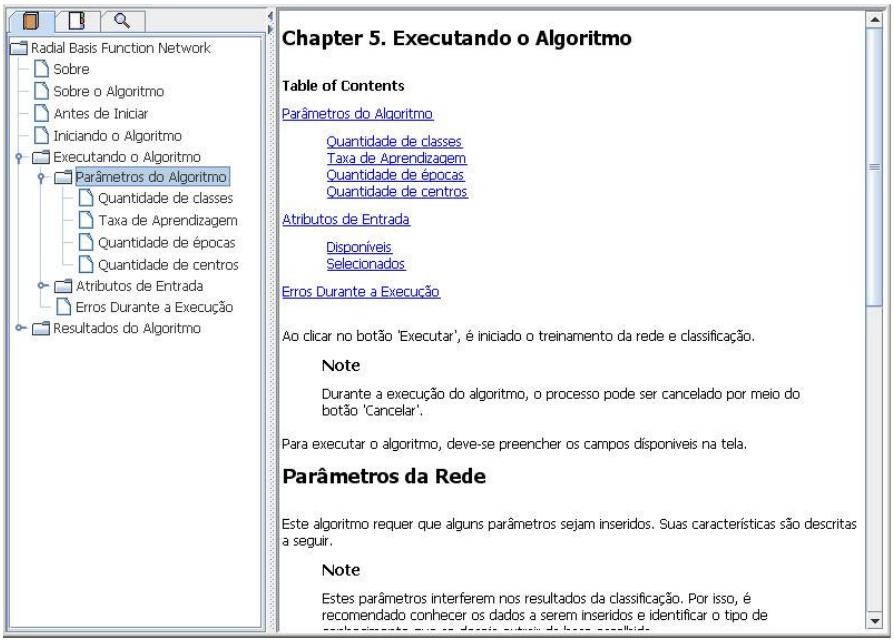

Figura 29. Documentação de ajuda da rede RBF

Finalizada a etapa de implementação da RN – RBF na *Shell* Orion foram realizados alguns testes utilizando a base de dados das iridáceas, a fim de comparar resultados gerados e tempos de processamento do algoritmo.

8.3 RESULTADOS OBTIDOS

Concluída a etapa de implementação, realizou-se no método de redes neurais de base radial para classificação da *Shell* Orion, a análise das classes identificadas, desempenho do algoritmo, usabilidade da aplicação, bem como uma comparação com diferentes ferramentas que apresentam a mesma funcionalidade.

Na realização dos testes do classificador RBF, utilizou-se um microcomputador com sistema operacional Windows XP Professional, processador Core 2 Duo 2.0Ghz e 2Gb de memória RAM.

8.3.1 Classes Identificadas pela Rede RBF

A base de dados das iridáceas foi utilizada para demonstrar a capacidade e desempenho do algoritmo para identificar corretamente a classe de cada registro. Para tal, o algoritmo foi executado com os seguintes parâmetros de entrada:

a) **quantidade de classes:** o problema possui três classes de plantas por isso utilizou-se o valor 3;

b) **quantidade de épocas:** será a quantidade máxima de épocas executadas se o algoritmo não alcançar a convergência pelo erro, optou-se pelo valor 2000;

c) **taxa de aprendizagem da rede:** optou-se pelo valor 0.1 pois não é baixo o suficiente para influenciar o tempo de processamento, nem tão alto que impeça a convergência;

d) **quantidade de centros ou funções de base:** utilizou-se 20 centros que corresponde a um terço da quantidade de dados treinamento, evitando tanto *overfitting* como *underfitting* ;

e) **atributos de entrada:** *sepal_length, sepal_width, petal_length* e *petal_width.*

Este teste foi realizando com o objetivo de classificar os registros de acordo com a espécie de iridáceas, ou seja, o objetivo é identificar quais registros que pertencem às classes: *Íris-Setosa*, *Íris-Versicolor* e *Íris-Vírginica*. Os resultados gerados pelo classificador RBF são descritos na Tabela 8.

Tabela 8. Classes identificadas pela rede RBF na base das iridáceas

Classe	Quantidade de elementos	Porcentagem de elementos	Classe
1 (*íris-setosa*)	49	32,67%	1
2 (*íris-versicolor*)	55	36,67%	1 (1 ocorrência) 2 (50 ocorrências) 3 (4 ocorrência)
3 (*íris-virgínica*)	46	30,67%	3

Os resultados demonstram que o algoritmo obteve desempenho satisfatório, identificando apenas cinco registros em classes incorretas. Na Figura 30 pode-se observar a não-linearidade dos dados da base das iridáceas, demonstrando também a eficiência da rede em classificar este tipo de problema.

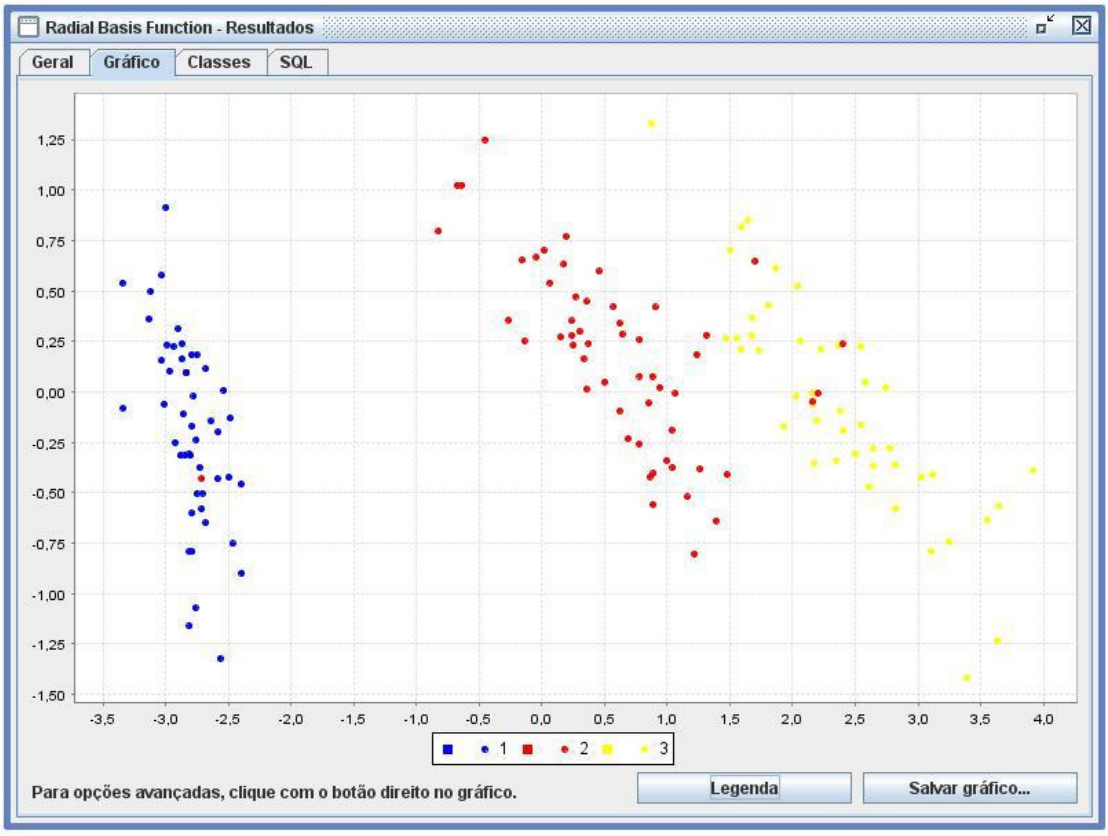

Figura 30. Classificação não-linear da base de dados das iridáceas

Com os resultados obtidos confirmou-se que os parâmetros de entrada selecionados influenciam diretamente nos resultados gerados pelo classificador RBF e devem ser escolhidos de forma criteriosa de acordo com o conhecimento que se deseja obter.

Concluindo que o algoritmo está funcionando de maneira correta, realizou-se uma análise do conhecimento gerado a fim de avaliar o desempenho do classificador RBF.

8.3.2 Análise e Avaliação de Desempenho

Muitas vezes o conhecimento gerado pelo processo de *data mining* pode não ser útil para o usuário. Segundo Han e Kamber (2001) o padrão encontrado é considerado relevante se ele é potencialmente útil, válido e de fácil compreensão.

A análise de desempenho do classificador desenvolvido realizou-se por meio de uma matriz de confusão (Tabela 9). Esta matriz combina o número de classificações corretas (linhas) com o número de classificações preditas pelo classificador (colunas), permitindo uma análise mais precisa do modelo (AMORIM, 2004).

Tabela 9. Matriz de confusão

Classe	**Verdadeira 1**	**Verdadeira 2**	**...**	**Verdadeira c**
Predita 1	(1,1)	(1,2)	...	(1,c)
Predita 2	(2,1)	(2,2)	...	(2,c)
...	...	...	...	...
Predita c	(c,1)	(c,2)	...	(c,c)

Os elementos marcados em cinza demonstram a diagonal principal da matriz de confusão, que representam as concordâncias da classificação. Já os elementos de fora desta diagonal, descrevem as discordâncias da classificação, ou seja, elementos classificados incorretamente.

A partir da matriz de confusão é possível encontrar os valores das seguintes variáveis:

a) **verdadeiros positivos (VP):** corresponde a quantidade de registros classificados corretamente, pode ser obtido pela soma dos elementos da diagonal principal da matriz de confusão;

b) **falsos positivos (FP):** refere-se aos registros classificados incorretamente pelo modelo, é encontrado pela soma dos elementos não pertencentes a diagonal principal;

c) **verdadeiros negativos (VN):** registros que não foram associados à classe considerada e que realmente não pertencem a ela, é calculado separadamente para cada classe e obtido pela soma dos verdadeiros e falsos positivos de todas as outras classes:

$$VN_c = \sum_{i=0}^{C} \left(VP_i + FP_i \right) \quad \forall \quad C \neq c \tag{12}$$

Onde, C é o numero total de classes e c a classe considerada.

d) **falsos negativos (FN):** corresponde ao registros que não foram associados à classe considerada e que pertencem a ela, também é calculado individualmente e é obtido pela soma dos falsos positivos desta classe:

$$FN_c = \sum_{i=0}^{C} \left(FP_{ic} \right) \quad \forall \quad c \neq i \tag{13}$$

Por meio das variáveis descritas é possível calcula índices para análise da qualidade de um modelo classificador, nesta pesquisa foram utilizadas as seguintes métricas de avaliação de desempenho (AMORIM, 2004):

a) **sensibilidade:** capacidade do modelo em identificar os registros que realmente pertencem à classe considerada, corresponde a proporção de verdadeiros positivos e pode ser encontrada pela seguinte equação:

$$S = \frac{VP}{VP + FN} \tag{14}$$

b) **especificidade:** probabilidade de acerto do modelo em identificar os registros que não pertencem à classe considerada, calculado pela proporção de negativos verdadeiros encontrada pela equação 15.

$$E = \frac{VN}{VN + FP} \tag{15}$$

c) **acurácia:** mede a exatidão geral do modelo para identificação das classes por meio da a relação entre o valor estimado e o valor real dada por:

$$A = \frac{VP + VN}{(VP + VN + FP + FN)} \tag{16}$$

d) **erro:** taxa erro de classificação global do modelo, obtido por:

$$e = 1 - A \tag{17}$$

e) **confiabilidade positiva:** precisão do classificador em identificar os verdadeiros positivos, é obtido por meio da seguinte equação:

$$VPP = \frac{VP}{VP + FP} \tag{18}$$

f) **índice kappa:** coeficiente de avaliação da concordância entre dois ou mais métodos de classificação, esta medida é baseada no numero de respostas concordantes entre os classificadores (COLGATON, 1991).

$$k = \frac{\left(\frac{\sum_{i=1}^{c} x_{ii}}{n} \right) - \left(\frac{\sum_{i=1}^{c} \frac{x_{i+} x_{+i}}{n}}{n^2} \right)}{1 - \left(\frac{\sum_{i=1}^{c} z_{ii}}{n^2} \right)} \tag{19}$$

Onde:

a) n representa o numero total de registros;

b) c é a quantidade de classes

c) x_{ii} refere-se aos elementos da diagonal principal

d) x_{i+} é o total da linha i e x_{+i} total da coluna i.

Este índice foi interpretado e classificado por Landis e Koch em 1997 de acordo com a Tabela 10, podendo variar entre zero (nenhuma concordância) e um (total concordância) (SCHWARTSMANN et al, 2006).

Tabela 10. Classificação do coeficiente de Kappa

Índice de Kappa	Interpretação
0	Pobre
0 – 0,2	Ligeira
0,21 – 0,4	Considerável
0,41 – 0,6	Moderada
0,61 – 0,8	Substancial
0,81 – 1	Excelente

Fonte: Adaptado de SCHWARTSMANN, C. et al. (2006)

A fim de analisar o desempenho do classificador, foi construída a matriz confusão para a classificação da base de dados das iridáceas, composta pelos resultados gerados nos testes do modelo desenvolvido conforme mostra a Tabela 11.

Tabela 11. Matriz de confusão para classificação da base das iridáceas

Predita\Verdadeira	Classe 1	Classe 2	Classe3	Total
Classe 1	49	0	0	49
Classe 2	1	50	4	55
Classe 3	0	0	46	46
Total	50	50	50	150

Considerando os valores descritos tabela acima, foi possível extrair as seguintes variáveis da matriz de confusão:

a) **verdadeiros positivos:**

$$VP = (49 + 50 + 46)$$

$$VP = 145$$

b) **falsos positivos:**

$$FP = (1+4)$$

$$FP = 5$$

c) **verdadeiros negativos:**

$$VN_1 = (VP_2 + VP_3 + VP_3 + VP_3)$$

$$VN_1 = (50+46+5+0)$$

$$VN_1 = 101$$

d) **Falsos negativos:**

$$FN_1 = (FP_{21} + FP_{31})$$

$$FN_1 = (1+0) = 1$$

A Tabela 12 demonstra os valores obtidos para as variáveis da matriz de confusão obtidas por meio dos cálculos realizados nos itens a, b, c e d.

Tabela 12. Tabela de falsos negativos e verdadeiros negativos

Classe	FN	VN	VP	FP
Classe 1	1	101	-	-
Classe 2	0	95	-	-
Classe 3	4	100	-	-
Global	5	296	145	5

Utilizando as variáveis extraídas da matriz de confusão, foram calculados os índices de validação do modelo:

a) **sensibilidade:**

$$S = \frac{VP}{VP + FN}$$

$$S = \frac{145}{145 + 5}$$

$$S = 0{,}966$$

b) **especificidade:**

$$E = \frac{VN}{VN + FP}$$

$$E = \frac{296}{296 + 5}$$

$$E = 0{,}9833$$

c) **acurácia:**

$$A = \frac{VP + VN}{(VP + VN + FP + FN)}$$

$$A = \frac{146 + 296}{(146 + 296 + 5 + 5)} = \frac{442}{452}$$

$$A = 0{,}9778$$

d) **erro:**

$$e = 1 - A$$

$$e = 1 - 0{,}977$$

$$e = 0{,}023$$

e) **confiabilidade positiva:**

$$VPP = \frac{VP}{VP + FP}$$

$$VPP = \frac{146}{146 + 5}$$

$$VPP = 0{,}966$$

f) **índice kappa:**

$$k = \frac{\left(\frac{x_{11} + x_{22} + x_{33}}{150}\right) - \left(\frac{\frac{x_{41} \cdot x_{14}}{150} + \frac{x_{42} \cdot x_{24}}{15} + \frac{x_{43} \cdot x_{34}}{150}}{150^2}\right)}{1 - \left(\frac{\frac{x_{41} \cdot x_{14}}{150} + \frac{x_{42} \cdot x_{24}}{15} + \frac{x_{43} \cdot x_{34}}{150}}{150^2}\right)}$$

$$k = \frac{\left(\frac{49 + 50 + 46}{150}\right) - \left(\frac{\frac{50 \cdot 49}{150} + \frac{50 \cdot 55}{15} + \frac{50 \cdot 46}{150}}{150^2}\right)}{1 - \left(\frac{\frac{50 \cdot 49}{150} + \frac{50 \cdot 55}{15} + \frac{50 \cdot 46}{150}}{150^2}\right)}$$

$$k = \frac{0{,}966 - 0{,}002}{1 - 0{,}002}$$

$$k = 0{,}96$$

A Tabela 13 apresenta um resumo dos índices de validação encontrados.

Tabela 13. Resumo dos índices de validação

Índice	Valor
sensibilidade	96,6%
especificidade	98,33%
acurácia	97,78%
erro	2,3%
confiabilidade	96,6%
kappa	0,96

Analisando os valores obtidos para os índices de avaliação, o desempenho do módulo desenvolvido foi considerado muito bom, apresentando índices próxim7os de 100% e erro próximo de zero, também demonstrou maior eficiência na identificação dos verdadeiros

positivos, considerando que os índices de confiabilidade positiva mantiveram-se igual ou superior aos índices de confiabilidade negativa.

Dos valores obtidos para o coeficiente de concordância Kappa, o classificador RBF apresentou índices próximos de um, considerado muito bom. Segundo Colgaton (1991) este coeficiente é satisfatório para avaliar a precisão de um modelo de classificação, pois considera toda a matriz de confusão no seu cálculo, incluindo elementos de fora da diagonal principal.

Finalizada a etapa de análise de desempenho do modelo, realizaram-se alguns testes com diferentes parâmetros de entrada a fim de analisar os tempos de processamento do módulo desenvolvido.

8.3.3 Tempos de Processamento do Classificador RBF

A fim de analisar o desempenho do módulo no que se refere a tempo de processamento, utilizou-se uma base de dados gerada aleatoriamente contendo 6000 registros e 5 atributos com valores aleatórios, sendo que um é referente à classe. Na avaliação dos tempos foram testados diferentes valores para os seguintes parâmetros: quantidade de classes, quantidade de atributos de entrada, quantidade de centros e taxa de aprendizagem da rede. Não foram realizados testes com a quantidade de época, pois o critério de convergência do algoritmo é o erro médio calculado durante o processamento da rede, por este motivo o algoritmo só executa a quantidade máxima de épocas informadas se não atingir a convergência por meio do erro.

Na Tabela 14 pode-se observar os tempos de processamento de acordo com a quantidade de classes e o número de funções de base radial informados.

Tabela 14. Tempos de processamento para os testes com o parâmetro quantidade de centros

Quantidade de classes	Taxa de aprendizagem	Quantidade de centros	Atributos selecionados	Tempo de processamento
6	0.01	100	4	00min:29s.203ms
4	0.01	100	4	00min:15s.578ms
2	0.01	100	4	00min:06s.562ms
6	0.01	75	4	00min:28s.031ms
4	0.01	75	4	00min:13s.781ms
2	0.01	75	4	00min:05s.782ms
6	0.01	50	4	00min:22s.750ms
4	0.01	50	4	00min:12s.266ms
2	0.01	50	4	00min:05s.344ms
6	0.01	20	4	00min:21s.515ms
4	0.01	20	4	00min:11s.421ms
2	0.01	20	4	00min:04s.406ms

Utilizando diferentes valores para a quantidade de classes e centros bem como valores fixos para a quantidade de atributos, foi possível concluir que os parâmetros testados interferem diretamente no tempo de processamento da rede, considerando que os maiores tempos foram registrados para o maior número de centros e classes a serem identificadas. No entanto, no caso de bases com muitos registros, a rede exige maior quantidade de centros para aprender, uma quantidade muito baixa pode inclusive aumentar o tempo de convergência do algoritmo.

De modo a testar a influência dos atributos de entrada, pode-se observar na Tabela 15 os tempos de processamento com diferentes quantidades de atributos e classes.

Tabela 15. Tempos de processamento para os testes com atributos de entrada

Quantidade de classes	Taxa de aprendizagem	Quantidade de centros	Atributos selecionados	Tempo de processamento
6	0.01	50	2	00min:21s.719ms
6	0.01	50	3	00min:22s.734ms
6	0.01	50	4	00min:22s.734ms
4	0.01	50	2	00min:11s.500ms
4	0.01	50	3	00min:12s.125ms
4	0.01	50	4	00min:12s.812ms
2	0.01	50	2	00min:04s.703ms
2	0.01	50	3	00min:04s.703ms
2	0.01	50	4	00min:05s.297ms

A quantidade de atributos de entrada informados também influencia no tempo de processamento da rede, visto que quanto mais atributos selecionados maior a informação que a rede irá processar, portanto é de extrema importância selecionar apenas atributos relevantes para o problema.

O ultimo parâmetro testado foi a taxa de aprendizagem da rede e os resultados podem ser observados na Tabela 16.

Tabela 16. Tempos de processamento para os testes com o parâmetro taxa de aprendizagem

Quantidade de classes	Taxa de aprendizagem	Quantidade de centros	Atributos selecionados	Tempo de processamento
6	0.001	50	4	00min:23s.281ms
4	0.001	50	4	00min:12s.703ms
2	0.001	50	4	00min:05s.063ms
6	0.01	50	4	00min:22s.734ms
4	0.01	50	4	00min:12s.812ms
2	0.01	50	4	00min:05s.297ms
6	0.1	50	4	00min:22s.734ms
4	0.1	50	4	00min:12s.704ms
2	0.1	50	4	00min:05s.547ms
6	0.5	50	4	00min:23s.094ms
4	0.5	50	4	00min:12s.593ms
2	0.5	50	4	00min:04s.937ms

Analisando os resultados obtidos com diferentes taxas de aprendizado, conclui-se que para taxas muito altas, próximas de um, a rede apresentou melhores tempos de

processamento, no entanto mostrou-se pior desempenho na identificação das classes. Considerando que com taxa de aprendizagem 0,1 a rede apresentou classificação satisfatória, optou-se por utilizar esta taxa como valor padrão.

Finalizada a análise dos tempos de processamento do classificador RBF, comparou-se os resultados do modelo desenvolvido com a ferramenta Weka 3.6.2[15] que também disponibiliza a tarefa de classificação por meio de redes RBF.

8.3.4 Comparação com outra Aplicação

A ferramenta Weka é uma ferramenta implementada na linguagem Java e licenciada pela *General Public License* (GPL), de distribuição gratuita e código aberto, desenvolvida pela Universidade de Waikato localizada na cidade de Hamilton, Nova Zelândia (WAIKATO, 2005, tradução nossa).

A Weka disponibiliza funcionalidade nas etapas de pré-processamento e possui diversos métodos para diferentes tarefas de *data mining*, incluindo a tarefa de classificação por meio de redes RBF. Esta ferramenta trabalha com arquivos em formato *arff* (Figura 31) composto pelos dados e por um pequeno cabeçalho que contém informações acerca dos atributos, também disponibiliza acesso à bases de dados SQL utilizando um JDBC. (WAIKATO, 2005, tradução nossa).

[15] Ferramenta gratuita disponível em: http://www.cs.waikato.ac.nz/ml/weka/

```
@RELATION iris

@ATTRIBUTE sepallength  REAL
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  REAL
@ATTRIBUTE petalwidth   REAL
@ATTRIBUTE class  {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
```

Figura 31. Formato de arquivo arff
Fonte: WAIKATO, U. (2005)

A interface de acesso às funcionalidades da Weka pode ser observada na Figura 32, onde é possível analisar os atributos de entrada e aplicar métodos de pré-processamento.

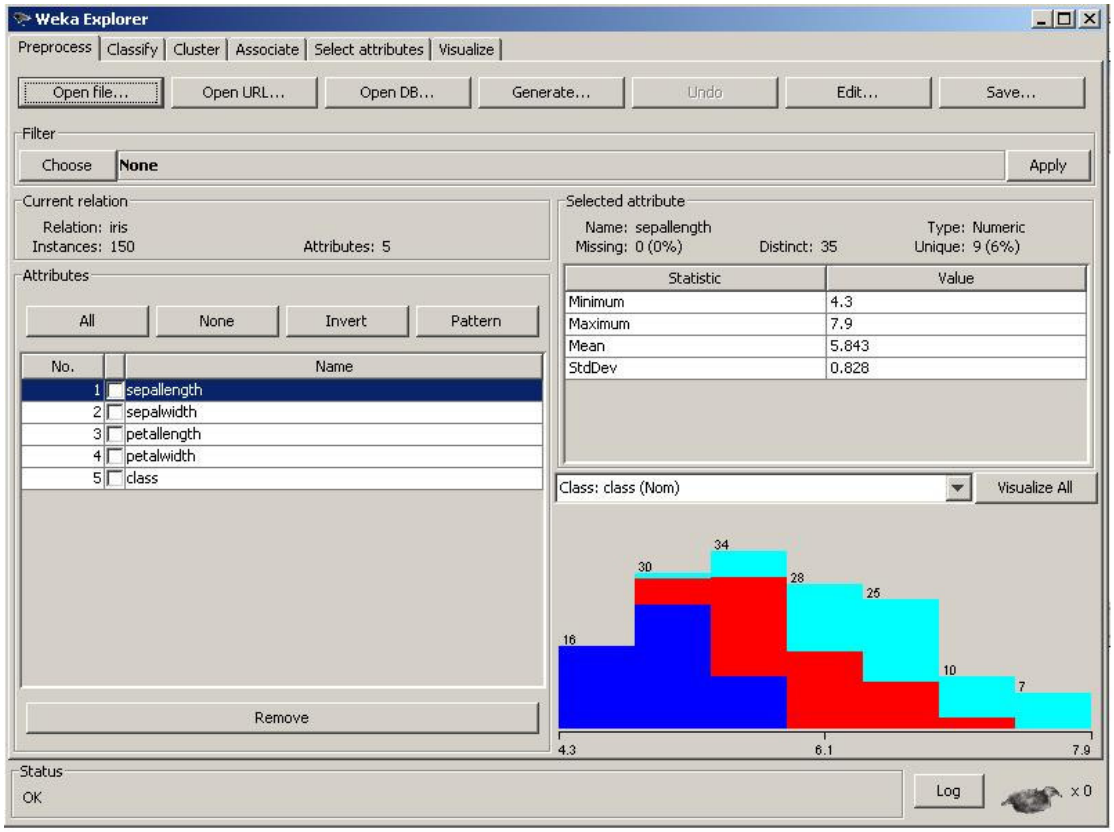

Figura 32. Interface de exploração da ferramenta Weka

A base das iridáceas foi aplicada também na Weka, que já disponibiliza o arquivo *arff* desta base, e os resultados obtidos foram comparados a fim de comparar o desempenho

das duas ferramentas. As Figuras 33 e 34 demonstram o resumo dos resultados obtidos pela *Shell* Orion e Weka respectivamente.

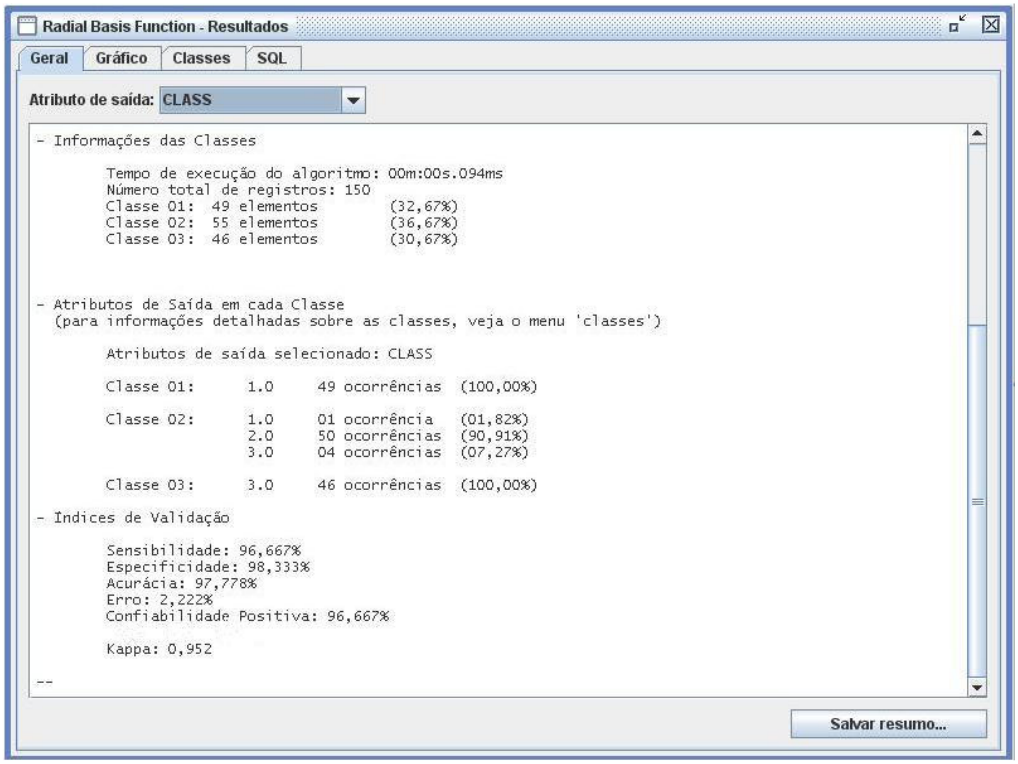

Figura 33. Resultados da *Shell* Orion

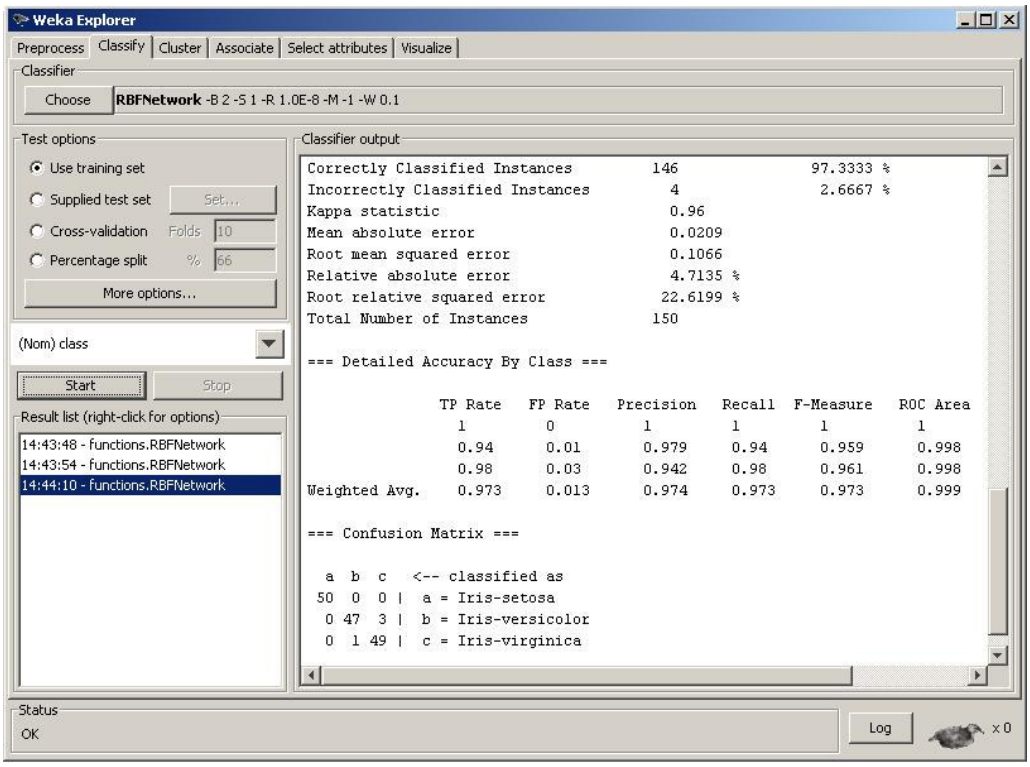

Figura 34. Resultados da Weka

A Weka também permite a visualização dos resultados de forma gráfica assim como a *Shell* Orion. Nas Figura 35 e 36 é possível visualizar o resultado da classificação em ambas as ferramentas.

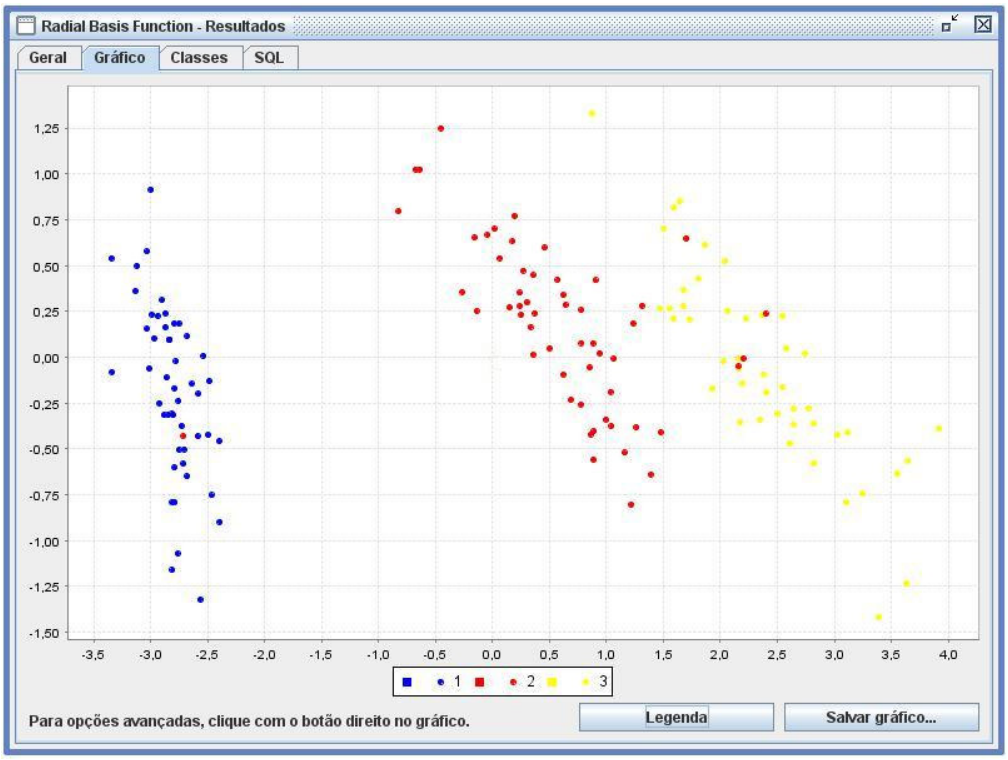

Figura 35. Gráfico gerado pela classificação das iridáceas na *Shell* Orion

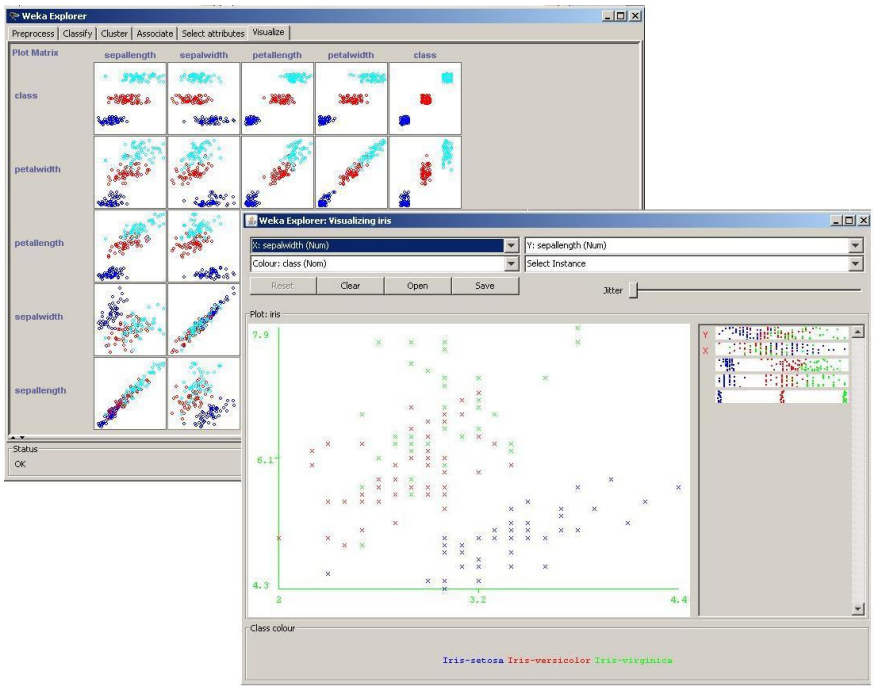

Figura 36. Gráfico gerado pela classificação das iridáceas na Weka

Na Tabela 17 é possível comparar os resultados gerados e os índices de validação das duas ferramentas na classificação da base de dados das iridáceas, utilizando os seguintes atributos de entrada: *sepal_length, sepal_width, petal_length* e *petal_width.*

Tabela 17. Índices de avaliação gerados pela *Shell* Orion e Weka

	***Shell* Orion**	**Weka**
Registros classificados corretamente	145	146
Registros classificados incorretamente	5	4
Sensibilidade	96,6%	97,3%
Especificidade	98,3%	98,6%
Acurácia	97,7%	98,2%
Erro	2,3%	2%
Confiabilidade	96,6%	97,3%
Kappa	0,96	0,96
Tempo de processamento	00min:00s.187ms	00min:00s.140ms

Analisando os resultados obtidos pode-se concluir que o classificador RBF da *Shell* Orion está funcionando de maneira satisfatória, considerando que apresentou 96,6% de registros classificados corretamente e índices de avaliação muito próximos aos da ferramenta Weka que classificou os registros com 97,3% de acerto, porém a Weka teve um desempenho melhor perante os índices de avaliação por utilizar o algoritmo de clusterização *k-means* na definição dos centros das funções de base (PETERMANN, 2006).

Ambas as ferramentas obtiveram valores elevados (entre 0,8 e 1) para o coeficiente kappa, demonstrando alto grau de concordância entre os dois modelos classificadores.

Tratando-se de tempo de processamento, a Weka registrou tempos pouco melhores quando comparados aos da *Shell* Orion. Considerando todos os testes realizados, o algoritmo apresentou resultados satisfatórios que confirmam o correto funcionamento do módulo desenvolvido.

CONCLUSÃO

Esta pesquisa confirmou que o complexo processo de descoberta de conhecimento em bases de dados pode ser simplificado por meio da aplicação de técnicas de *data mining,* auxiliando na tomada de decisão e desenvolvimento de melhores práticas, garantindo vantagem competitiva às organizações, que priorizam cada vez mais o investimento nesta área.

O *data mining* é composto por diversas tarefas, sendo que esta pesquisa aprofundou-se no entendimento da classificação e em sua aplicação por meio do método de redes neurais artificiais com função de base radial. Este método demonstra grande potencial nesta tarefa por utilizar campos receptivos na delimitação das fronteiras de decisão.

Foram descritos inclusive, os cálculos matemáticos realizados pelo algoritmo, o que possibilitou o desenvolvimento do modelo proposto. Entretanto, no decorrer desta pesquisa encontraram-se algumas dificuldades, principalmente no que se refere ao entendimento dos formalismos matemáticos das redes RBF, devido à carência de bibliografia explicativa.

Contudo, conforme demonstraram os testes realizados, o módulo foi desenvolvido com sucesso apresentando funcionamento correto e resultados satisfatórios na classificação e em tempos de processamento, atingindo assim os objetivos propostos nesta pesquisa.

Considerando a vasta gama de tarefas e métodos que compõe o processo de descoberta de conhecimento em bases de dados e a carência de material didático, pesquisas nesta área são muito importantes para a comunidade acadêmica. Por isso, baseando-se no conhecimento adquirido nesta pesquisa são descritas algumas recomendações para trabalhos futuros, a fim de dar continuidade ao projeto da *Shell* Orion *Data Mining Engine*:

a) desenvolver diferentes métodos de treinamento para a rede RBF aplicando algoritmos não-supervisionados na seleção de centros por exemplo, *backpropagation, k-means*[16] *e* Kohonen[17], e outros algoritmos na camada de saída como *Last Mean Square* (LMS[18]) por exemplo;
b) implementar novos algoritmos pelo método de redes neurais artificiais, como por exemplo redes MLP;
c) inclusão de funcionalidades relacionadas a outras etapas do processo de descoberta de conhecimento da *Shell* Orion, como pré-processamento, permitindo a preparação e transformação dos dados aplicados;
d) pesquisar e desenvolver novos algoritmos em tarefas carente de métodos, como por exemplo, associação, regressão e previsão de séries temporais;
e) aplicar os resultados obtidos por meio do classificador RBF em outras tarefas de *data mining*.

[16] Algoritmo de clusterização que atribui cada registro de uma base de dados ao cluster de centro mais próximo (KANTARDZIC, 2003);
[17] Rede neural auto-organizável de aprendizado não-supervisionado e competitivo desenvolvida por Teuvo Kohonen em 1982, aplicada para tarefa de clusterização (HAYKIN, 2001);
[18] Algoritmo de classificação linear de natureza adaptativa desenvolvido por Widrow e Hoff em 1960 (MEHROTRA; MOHAN; RANKA, 1996).

APÊNDICE A

O Método de Redes Neurais com Função de Ativação de Base Radial para Classificação em Data Mining

Ana Paula Scotti[1], Merisandra Côrtes de Matos[2]

[1]Acadêmico do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – UNESC

[2]Professor do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – UNESC

anah.sour@gmail.com, mem@unesc.net

Resumo. *Os avanços computacionais no que se refere ao armazenamento de dados ocasionaram a formação de grandes bases de dados resultando na necessidade de extração do conhecimento, destacando-se o data mining dentre as tecnologias para análise de informações. Deste modo, este artigo demonstra a modelagem e desenvolvimento do algoritmo de redes neurais com função de ativação de base radial para a tarefa de classificação em uma ferramenta gratuita de data mining denominada Shell Orion Data Mining Engine. Esta rede neural tem como objetivo dividir a base de dados não-linear de acordo com o grupo a que cada registro pertence utilizando funções de base radial.*

Palavras-chave*: Data Mining, Classificação, Redes Neurais, Radial Basis Function.*

1. Introdução

Analisar e extrair conhecimento útil de grandes bases de dados tornou-se um problema complexo para as organizações devido ao crescimento do volume de dados armazenados. Para facilitar esta análise são utilizadas ferramentas de *data mining* que são em sua maioria comerciais [Goldschimdt e Passos 2005].

A fim de disponibilizar uma ferramenta gratuita, o Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da UNESC mantém em desenvolvimento a *Shell* Orion *Data Mining Engine*, que já possui diferentes tarefas e métodos implementados.

Dentre as tarefas existentes, a classificação é uma das mais populares e consiste em encontrar propriedades comuns em um conjunto de registros de uma base de dados e relacioná-los a uma classe pré-definida. O método de redes neurais destaca-se nesta tarefa devido a sua capacidade de aprendizagem por experiência e classificação de dados não conhecidos.

Deste modo, neste artigo apresenta-se o desenvolvimento método de redes neurais com função de ativação de base radial para a tarefa de classificação na *Shell* Orion [Han e Kamber 2006].

2. Descoberta de Conhecimento em Bases de Dados

A descoberta do conhecimento em bases de dados auxilia na análise e extração de conhecimento útil de grandes bases. Este processo é dividido em três etapas:

a) **pré-processamento:** consiste na transformação dos dados para tornar possível a aplicação dos algoritmos;

b) **data mining:** refere-se efetivamente à busca por conhecimento e extração de padrões da base de dados, é considerada a etapa mais importante;

c) **pós-processamento:** realiza-se a análise e interpretação dos resultados obtidos com o *data mining,* para facilitar o entendimento do usuário.

2.1. *Data Mining*

Data mining é definido como um processo de reconhecimento de padrões no qual são aplicadas técnicas inteligentes a fim de extrair conhecimento implícito nas bases de dados e auxiliar no processo decisório. O uso desta técnica não é restrito as empresas, oferecendo vantagens também em áreas como medicina, economia, geologia dentre outras, devido à potencialização dos recursos computacionais e no constante aumento do volume de dados [Olson e Delen 2008].

As principais tarefas de *data mining* são:

a) **associação:** busca relações entre os dados que possam identificar uma tendência;

b) **clusterização:** agrupa elementos de uma base com características semelhantes entre si e diferentes de outros grupos;

c) **classificação:** associa cada registro de uma base de dados à uma classe;

d) **previsão:** prevê futuros valores de um índice por meio da análise do comportamento passado.

3. A Tarefa de Classificação no Processo de Data Mining

A classificação é uma tarefa preditiva que realiza o mapeamento dos registros de uma base de dados em uma quantidade finita de conjuntos, atribuindo cada elemento a uma categoria pré-definida [Han e Kamber 2001].

Nesta tarefa o conjunto de dados é dividido em dois grupos: dados de treinamento, composto pelos registros utilizados na fase de aprendizagem, e dados de teste, utilizados na avaliação do modelo gerado Na aprendizagem os dados para os quais as classes são conhecidas são utilizados na criação de um modelo classificador. Posteriormente os dados de teste são utilizados para estimar a capacidade do modelo em classificar dados não conhecidos e a habilidade de atribuir cada registro à classe correta [Russel e Norvig 2004].

4. O Método de Redes Neurais na Tarefa de Classificação

Redes neurais são estruturas nas quais os neurônios estão dispostos em camadas e interligados por conexões conhecidas como pesos sinápticos que representam o conhecimento da rede. Estas estruturas possuem a capacidade de classificar padrões desconhecidos adequando-se à resolução de problemas onde se tem pouco conhecimento das relações entre atributos e classes. São também capazes de adquirir conhecimento por meio de um conjunto reduzido de exemplos e produzir respostas consistentes para dados não conhecidos [Haykin 2001].

O processo de aprendizagem de uma rede neural ocorre por meio do ajuste dos seus pesos sinápticos de acordo com a resposta da rede aos dados de entrada. O modo como é realizado este ajuste é que determina o tipo do aprendizado da rede que pode ser supervisionado quando a rede aprende utilizando exemplos fornecidos por um supervisor externo, ou não-supervisionado, quando utiliza apenas os dados de entrada [Haykin 2001].

As redes neurais de uma só camada são capazes de resolver apenas problemas linearmente separáveis, ou seja, que podem ser satisfeitos por uma reta ou hiperplano como

fronteira de decisão (Figura 1). Já a resolução de problemas de classificação não-lineares necessita de redes neurais com uma ou mais camadas ocultas [Bishop 1995].

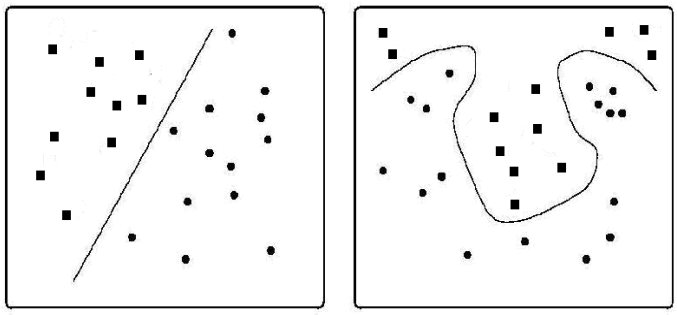

Figura 1. Classificação linear e não-linear

5. O Método de Redes Neurais com Função de Ativação de Base Radial

Uma rede neural com Função de Ativação de Base Radial (RBF) consiste em um modelo neural multicamadas, capaz de aprender padrões complexos e resolver problemas não-linearmente separáveis.

A arquitetura de uma rede RBF está dividida em três camadas (Figura 2): camada de entrada, na qual os padrões são apresentados à rede; camada intermediária ou oculta que realiza o mapeamento não-linear do espaço de entrada utilizando função gaussiana; e camada de saída que fornece a resposta da rede ao padrão apresentado [Theodoridis e Koutroumbas 2006].

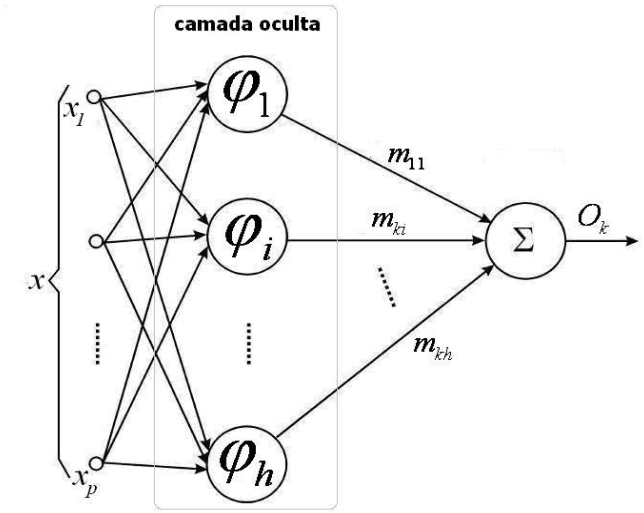

Figura 2. Arquitetura da rede RBF

6. O Método de Redes Neurais com Função de Ativação de Base Radial na Shell Orion Data Mining Engine

A modelagem do módulo de classificação com redes RBF iniciou-se com a construção dos diagramas de caso de uso, atividades e seqüência utilizando os padrões UML. Posteriormente foi desenvolvida a demonstração matemática do funcionamento da rede a fim de facilitar o entendimento e a implementação.

O processo de aprendizado da rede RBF desenvolvida transforma um problema de classificação não-linear em um problema linear, e é dividido nas seguintes etapas:

a) **seleção dos centros (*c*):** um subconjunto dos dados de treinamento é atribuído aos vetores centro das funções de base radial;

b) **definição do raio de abrangência (σ):** calcula-se a área de sensibilidade da função de base em relação ao seu centro utilizando a seguinte equação:

$$\sigma = \frac{dist_{\max}(c_{i,}c_j)}{\sqrt{2H}}, \qquad \forall i \neq j \tag{20}$$

c) **cálculo da ativação dos neurônios ocultos (*u*):** define-se o grau de ativação de cada neurônio da camada oculta utilizando distância euclidiana conforme a equação (2)

$$u_i(t) = \| x(t) - c_i(t) \|, \qquad i = 1,\ldots H \tag{21}$$

d) **mapeamento do espaço não-linear (φ):** na camada oculta da rede, as funções gaussianas definidas pela equação (3) realizam a transformação dos dados de entrada não-lineares;

$$\varphi_i(t) = \exp\left(-\frac{u_i^{\,2}(t)}{2\sigma_i^{\,2}} \right) \tag{22}$$

e) **cálculo das saídas (*O*):** os pesos de saída da rede são atualizados de acordo com a regra do *perceptron* simples e utilizados na próxima iteração.

$$o_k(t) = \left\{ \frac{1, \quad U_k(t) \geq 0}{0, \quad U_k(t) < 0} \right\} \tag{23}$$

Onde $U_k(t)$ é definido pela equação (5):

$$U_k(t) = \sum_{i=1}^{H} m_{ki}(t)\varphi_i(t) \tag{24}$$

f) **cálculo do erro (*e*):** diferença entre a saída desejada e a saída real da rede, onde:

$$e_k(t) = d_k(t) - o_k(t) \tag{25}$$

g) **ajuste das sinapses (*m*):** a atualização dos pesos sinápticos descrita na equação (7) ocorre somente quando o erro for diferente de zero.

$$m_{ki}(t+1) = m_{ki}(t) + \eta e_k \varphi_i(t) \tag{26}$$

h) **condição de parada (*E*):** o algoritmo atinge a convergência quando a rede não apresentar mudanças significantes nas sinapses. Essa condição pode ser verificada por meio da equação (8).

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(e_k(t)\right)^2 \tag{27}$$

A apresentação de todos os vetores de treinamento à rede define uma época de treinamento, nesta fase a condição de parada é testada e se não for satisfeita, o conjunto de treinamento é embaralhado e a rede continua seu processamento iterativamente.

6.1. Implementação e Realização de Testes

A rede RBF foi implementada no módulo de classificação da Shell Orion Data Mining Engine por meio da linguagem de programação Java e ambiente de programação Netbeans 6.8.

A *Shell* Orion possibilita a conexão com *drivers* de diferentes bancos de dados, sendo que nos testes realizados nesta pesquisa optou-se pelo uso do MySQL 5.1, disponível gratuitamente para download em: http://dev.mysql.com/downloads/mysql.

Para executar a tarefa de classificação por meio de redes RBF é necessário definir alguns parâmetros da rede:

a) **quantidade de classes:** número de classes que o algoritmo irá identificar, o valor informado não pode ser maior que a quantidade real de classes;

b) **quantidade de épocas:** quantidade máxima de épocas executadas;

c) **taxa de aprendizagem:** taxa de atualização dos pesos sinápticos que corresponde ao grau de aprendizagem da rede;

d) **quantidade de centros:** quantidade de funções de base radial na camada oculta, este valor não pode ser muito alto para não ocasionar *overfitting*, nem muito baixo que gere *underfitting*;

e) **atributos de entrada:** atributos da base de dados que serão utilizados como valores de entrada da rede neural.

A *Shell* Orion permite que os resultados gerados pelo algoritmo possam ser analisados por meio de resumo, árvore e gráfico. Na Figura 3 observa-se o relatório gerado pelo algoritmo contendo um resumo da classificação

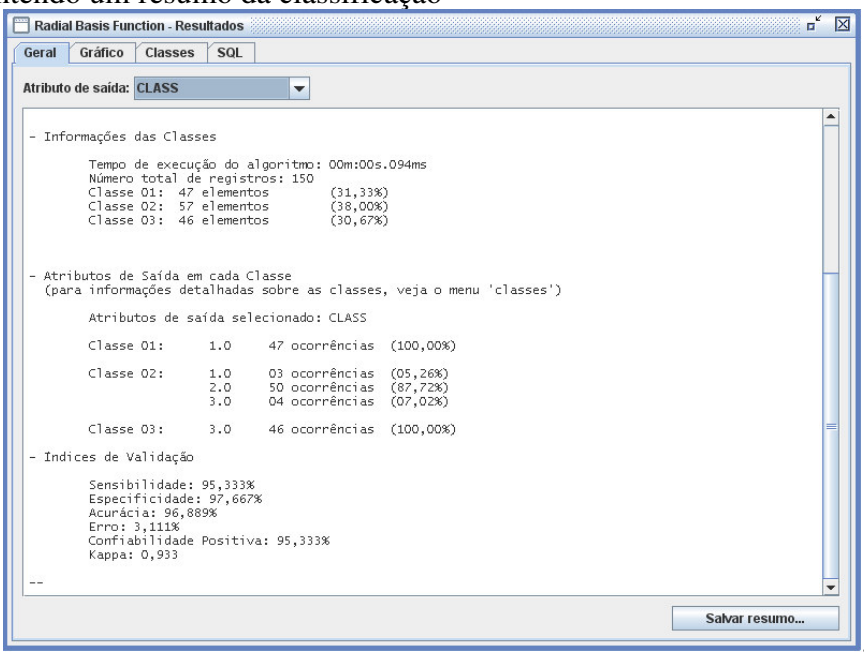

Figura 3. Resumo da classificação por meio da rede RBF

A atribuição dos registros para cada classe pode ser facilmente visualizada também em forma gráfica como mostra a Figura 4, onde as classes identificadas são representadas por meio de *Principal Component Analysis* (PCA). O método PCA transforma uma base de dados de n dimensões em uma matriz de duas dimensões, possibilitando a projeção dos dados graficamente.

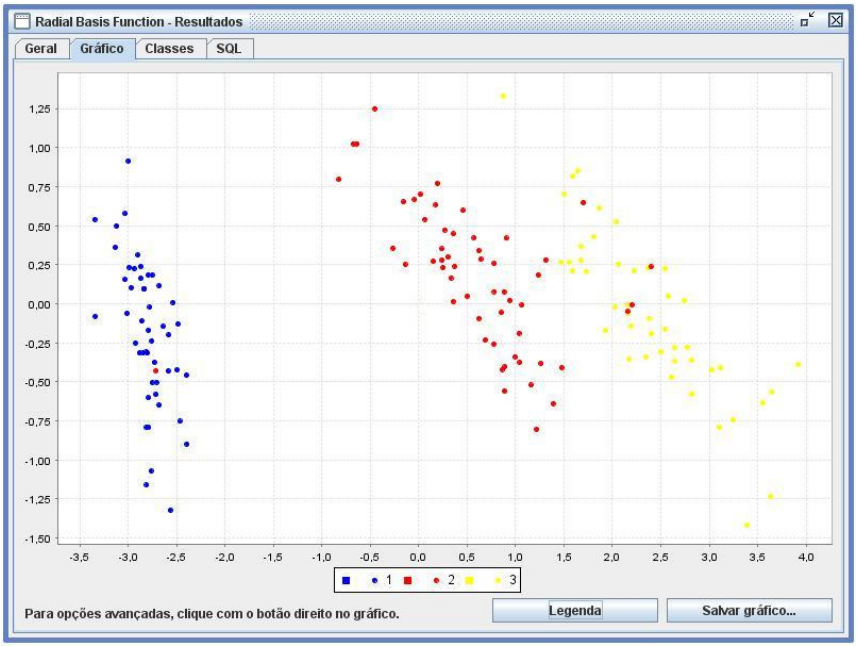

Figura 4. Gráfico gerado pelo classificador RBF

Detalhes dos elementos contidos em cada classe podem ser analisadas individualmente por meio de uma estrutura em árvore (Figura 5).

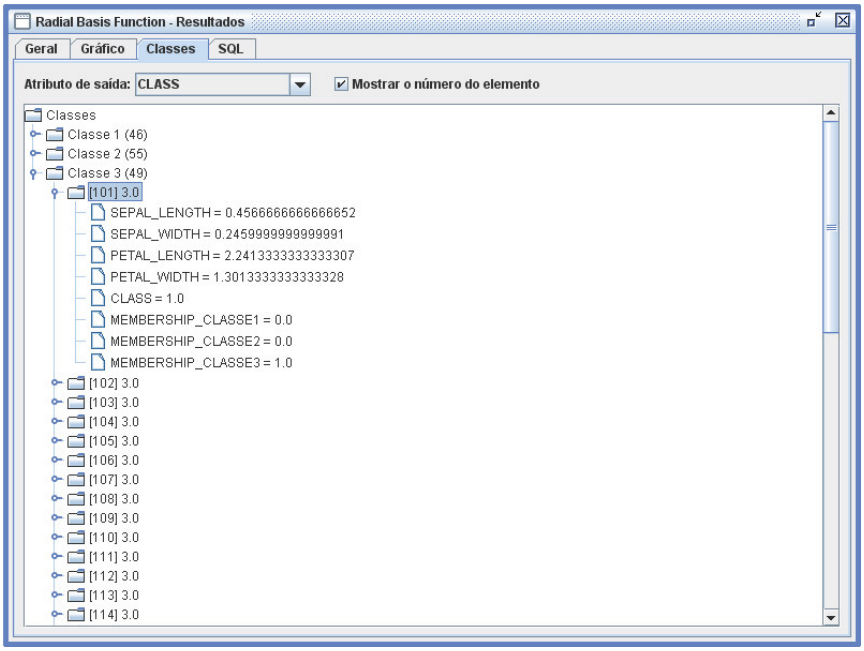

Figura 5. Árvore das classes identificadas pela rede RBF

Além disso, a ferramenta permite a exportação dos resultados gerados em formato de arquivo SQL, esta funcionalidade permite uma posterior aplicação dos resultados da classificação como entrada para outras tarefas de DM. É possível também executar diversas vezes o algoritmo com parâmetros de entrada diferentes, possibilitando a comparação dos resultados encontrados. Além disso, um arquivo de ajuda disponibiliza a documentação necessária para auxiliar o usuário na utilização do classificador RBF.

6.2. Resultados Obtidos

Nos testes realizados no módulo desenvolvido utilizou-se a base de dados das Iridáceas, composta por entradas não-lineares, contendo dados referentes a três tipos de plantas da família das Iridáceas: setosa, versicolor e virgínica, totalizando 150 registros e 4 atributos (*sepal_length, sepal_width, petal_length* e *petal_width*) referentes a largura e comprimento das sépalas e pétalas das plantas.

O algoritmo foi executado com os seguintes parâmetros de entrada: 3 para quantidade de classes; 2000 para quantidade máxima de épocas; taxa de aprendizagem de 0.1 e 20 funções de base radial. Os resultados gerados pelo classificador RBF são descritos na Tabela 1.

Tabela 1. Resultados do classificador RBFpara a base de dados das iridáceas

Classe	Quantidade de elementos	Porcentagem de elementos	Classe
1 (*íris-setosa*)	49	32,67%	1
2 (*íris-versicolor*)	55	36,67%	1 (1 ocorrência) 2 (50 ocorrências) 3 (4 ocorrência)
3 (*íris-virgínica*)	46	30,67%	3

Os resultados demonstram que o algoritmo obteve desempenho satisfatório, identificando apenas cinco registros em classes incorretas e confirmou-se que os parâmentros de entrada selecionados influenciam.

6.3. Avaliação do Desempenho

A análise de desempenho do classificador desenvolvido realizou-se por meio da uma matriz de confusão uma matriz de confusão que combina os valores reais com os valores preditos pelo modelo (Tabela 2).

Tabela 2. Matriz de confusão para a classificação da base das iridáceas

Predita\Verdadeira	Classe 1	Classe 2	Classe3	Total
Classe 1	49	0	0	49
Classe 2	1	50	4	55
Classe 3	0	0	46	46
Total	50	50	50	150

Os elementos marcados em cinza demonstram a diagonal principal da matriz de confusão, que representam as concordâncias da classificação. Já os elementos de fora desta diagonal, descrevem as discordâncias da classificação, ou seja, elementos classificados incorretamente. A partir desta matriz possível calcular os índices de avaliação de desempenho do classificador (Tabela 3).

Tabela 2. Índices de avaliação

Índice	Valor
sensibilidade	96,6%
especificidade	98,33%
acurácia	97,78%
erro	2,3%
confiabilidade	96,6%
kappa	0,95

6.4. Tempos de Processamento

A fim de analisar o desempenho do módulo no que se refere a tempo de processamento, utilizou-se uma base de dados gerada aleatoriamente contendo 6000 registros e 4 atributos.

Nesta avaliação foram testados diferentes valores para os seguintes parâmetros: quantidade de classes, quantidade de atributos de entrada, quantidade de centros e taxa de aprendizagem da rede. Observou-se que quanto maior a quantidade de classes e atributos de entrada, maior é o tempo de processamento da rede. Para taxa de aprendizagem com valores altos a rede apresentou melhores tempos no entanto mostrou pior desempenho na identificação de classes assim como a quantidade de funções de base.

6.4. Comparação com outra Aplicação

Os resultados gerados pelo classificador RBF para a base de dados das iridáceas na *Shell* Orion foi comparados com os resultados obtidos com a aplicação da mesma base de dados no classificador RBF da ferramenta gratuita Weka 3.6.2 disponível em: http://www.cs.waikato.ac.nz/ml/weka/. A Tabela 3 demonstra a comparação entre os índices de avaliação de desempenho de ambas as ferramentas.

Tabela 3. Tempos de processamento

	***Shell* Orion**	**Weka**
Registros classificados corretamente	145	146
Registros classificados incorretamente	5	4
Sensibilidade	96,6%	97,3%
Especificidade	98,3%	98,6%
Acurácia	97,7%	98,2%
Erro	2,3%	2%
Confiabilidade	96,6%	97,3%
Kappa	0,95	0,96
Tempo de processamento	00m:00s.187ms	00m:00s.140ms

Analisando os resultados obtidos pode-se concluir que o classificador RBF da Shell Orion está funcionando de maneira satisfatória, considerando que apresentou 96,6% de registros classificados corretamente e índices de avaliação muito próximos aos da ferramenta

Weka que classificou os registros com 97,3% de acerto, porém a Weka teve um desempenho melhor perante os índices de avaliação e tempo de processamento pouco menor.

Ambas as ferramentas obtiveram valores excelentes (entre 0,8 e 1) para o coeficiente kappa, demonstrando alto grau de concordância entre os dois modelos classificadores.

Considerando todos os testes realizados, o algoritmo apresentou resultados satisfatórios que confirmam o correto funcionamento do módulo desenvolvido.

7. Conclusão

Este artigo apresentou um modelo classificador implementado pelo método de redes neurais com função de ativação de base radial na Shell Orion *Data Mining* Engine, contribuindo com o desenvolvimento da ferramenta.

Diante dos resultados obtidos pode-se confirmar a aplicabilidade de redes RBF para a tarefa de classificação, por utilizarem campos receptivos locais como fronteira de decisão e concluiu-se que o modelo foi desenvolvido com sucesso, pois apresentou funcionamento correto e resultados satisfatórios na classificação e em tempos de processamento.

Referências

Bishop, C. (1995), Neural Networks for Pattern Recognition, Oxford University Press.

Goldschmidt, R., e Passos, E. L. (2005), Data mining: uma guia prático: conceitos, técnicas, ferramentas, orientações e aplicações, Elsevier.

Haykin, S. (2001). Redes neurais: princípios e prática, Bookman, 2. ed.

Russel, S. e Norvig, P. (2004), Inteligência Artificial, Elsevier.

Olson, D. e Delen, D. (2008), Advanced Data Mining Techniques, Springuer.

Kantardzic, M. (2003) Data mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons.

Han, J. e Kamber, M. (2006), Data Mining: Concepts and Techniques, Morgan Kaufmann, 2. ed.

Theodoridis, S. e Koutroumbas, K. (2006), Pattern recognition, Elsevier.

REFERÊNCIAS

AMORIM, Bruno Pereira. **Desenvolvimento de uma Plataforma Híbrida para Descoberta de Conhecimento em Bases de Dados.** 2004. Dissertação de Mestrado. Programa de Pós – Graduação em Ciência da Computação. Universidade Federal de Pernambuco. Recife, 2004.

AZEVEDO, Fernando Mendes de; BRASIL, Lourdes Mattos; OLIVEIRA, Roberto Célio Limão de. **Redes neurais com aplicações em controle e em sistemas especialistas.** Florianópolis: Bookstore, 2000.

BERRY, Michael J.; LINOFF, Gordon. **Data mining techniques:** for marketing, sales, and customer relationship management. Indianapolis: Wiley Publishing, 1997.

BISHOP, Cristian. **Neural Networks for Pattern Recognition.** New York: Oxford University Press, 1995.

BORTOLOTTO, Leandro Sehnem. **O Método de Redes Neurais pelo Algoritmo de Kohonen para Clusterização na Shell Orion Data Mining Engine.** 2007. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

BRAGA, Antônio de Pádua; CARVALHO, André Carlos Ponce de Leon Ferreira de; LUDERMIR, Teresa Bernarda. **Redes Neurais Artificiais:** teoria e aplicações. Rio de Janeiro: LTC, 2000.

CARVALHO, Luís Alfredo Vidal de. **A mineração de dados no marketing, medicina, economia, engenharia e administração.** São Paulo: Érica, 2001.

CASAGRANDE, Diego Paz. **O Módulo da Técnica de Associação pelo Algoritmo Apriori no desenvolvimento da Shell de Data Mining Orion.** Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2005.

CASSETARI JUNIOR, José Márcio. **O Método de Lógica Fuzzy pelo Algoritmo Gustafson-Kessel na Tarefa de Clusterização da Shell Orion Data Mining Engine.** 2008. Trabalho de Conclusão de Curso - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciuma, Santa Catarina, 2008.

CASTRO, Maria Cristina F. **Predição Não-Linear de Séries Temporais Usando Redes Neurais RBF por Decomposição em Componentes Principais.** 2001. Tese de Doutorado. Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas. Campinas, São Paulo, 2001.

COLGATON, R. G. A review of assessing the accuracy of classifications of remotely sensed data. **Remote Sensing of Environment**, v. 49, n. 12, p. 1671-1678, 1991.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gegory; SMYTH, Padhraic. From Data mining to Knowledge Discovery in Databases. **AI Maganize,** v. 17, n. 3, p. 37-54, 1996. Disponível em: <http://www.daedalus.es/fileadmin/daedalus/doc/MineriaDeDatos/fayyad96.pdf>. Acesso em: 10 de ago 2009.

FERNANDES, Marcelo A. C.; DORIA NETO, Adrião D.; BEZERRA, João B. Aplicação das Redes RBF na Detecção Inteligente de Sinais Digitais**. IV Congresso Brasileiro de Redes Neurais,** v. 1, p. 226-230, 1999. Disponível em: <http://www.ele.ita.br/cnrn/artigos4cbrn/ 4cbrn_048.pdf> Acesso em: 4 nov 2009.

FONSECA, José Manuel Matos da. **Indução de Árvores de Decisão**. Tese de Mestrado - Departamento de Informática – Universidade Novas de Lisboa, Lisboa 1994.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel Lopes. **Data Mining:** um guia prático. Rio de Janeiro: Elsevier, 2005.

GUEDES, Gilleanes T. A. **UML**: uma abordagem prática.São Paulo: Novatec, 2008.

GUERRA, Fábio Alessandro. **Análise de métodos de agrupamento para o treinamento de redes neurais de base radial aplicadas à identificação de sistemas.** Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia de Produção e Sistemas. Pontifícia Universidade Católica do Paraná, Curitiba, PR. 2006. 146p.

HAN, Jiawei; KAMBER, Micheline. **Data Mining:** Concepts and Techniques. 2. ed. São Francisco: Morgan Kaufmann, 2006.

HAYKIN, Simon. **Redes neurais:** princípios e práticas. Porto Alegre: Bookman, 2001.

KANTARDIZIC, Mehmed. **Data mining:** Concepts, Models, Methods, and Algorithms. John Wiley & Sons, 2003.

MARTINS, Denis Piazza. **O Algoritmo de Particionamento K-means na Tarefa de Clusterização da Shell Orion Data Mining Engine.** 2007. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

MEHROTRA, Kishan; MOHAN K. Chilukuri; RANKA Sanjay. **Elements of Artificial Neural Networks.** Bradford: Bradford Books, 1996.

MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. **Machine Learning, Neural and Statistical Classification.** Ellis Horwood, 1994.

MONDARDO, Ricardo Lineburger. **O Algoritmo C4.5 na tarefa de Classificação na Shell Orion Data Mining Engine.** 2009. Trabalho de Concluso de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2009.

MOTTA, Custódio G. L. da**. Sistema Inteligente para Avaliação de Riscos em Vias de Transporte Terrestre.** 2004. Programa de Pós-graduação de Engenharia, Universidade Federal do Rio de Janeiro. Disponível em: <http://www.coc.ufrj.br/teses/mestrado/inter/2004/Teses/MOTTA_CGL_04_t_M_int.pdf> Acesso em: 4 set 2009.

NISHIDA, W.; BASTOS, L. Classificação de Imagens de Sensoriamento Remoto Utilizando uma Rede Neural com Função de Base Radial. **Simpósio Brasileiro de Sensoriamento Remoto**. p. 991-100, 1998. Disponível em: < http://marte.dpi.inpe.br/col/sid.inpe.br/deise/1999/02.11.11.58/doc/8_122p.pdf > Acesso em: 13 jan 2010.

OLSON, David L.; DELEN, Dursun. **Advanced Data Mining Techniques.** Lincoln: Springuer, 2008.

ORTEGA, Gustavo Victor C. **Redes Neurais na Identificação de Perdas Comerciais do Setor Elétrico.** 2008. Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia Elétrica da PUC-Rio. Disponível em: <http://www.maxwell.lambda.ele.pucrio.br/Busca_etds.php?strSecao=resultado&nrSeq=13380@1> Acesso em: 30 set 2009.

PAL, Sankar K.; MITRA, Pabitra. **Pattern recognition algorithms for data mining:** scalability, knowledge discovery and soft granular computing. Florida: Chapman & Hall, 2004.

PETERMANN, R. J. **Modelo de Mineração de Dados para Classificação de Clientes em Telecomunicações.** Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia Elétrica. PUC-RS. Disponível em: <http://tede.pucrs.br/tde_busca/arquivo.php?codArquivo=471> Acesso em 05 jul 2010.

PELEGRIN, Diana Colombo. **A Tarefa de Classificação e o Algoritmo ID3 para Indução de Árvores de Decisão na Shell de Data Mining Orion.** Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2005.

PEREGO, Daniel. **O Método de Lógica Fuzzy pelo Algoritmo Gath-Geva na Tarefa de Clusterização da Shell Orion Data Mining Engine.** 2009. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2009.

RAIMUNDO, Lidiane Rosso. **O Algoritmo CART na Tarefa de Classificação da Shell Orion Data Mining Engine.** 2007. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

REZENDE, Solange Oliveira. **Sistemas inteligentes:** fundamentos e aplicações. Barueri: Manole, 2005.

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

SCHWARTSMANN, C. R.; BOSCHIN, L. C.; MOSCHEN, G. M.; GONÇALVES, E. Z.; RAMOS, A. S. N.; GUSMÃO, P. D. F.; JACOBUS, L. S. Classificação das fraturas trocantéricas: avaliação da reprodutibilidade da classificação AO. **Revista Brasileira de Ortopedia.** v. 41, n. 7, p. 264-7, 2006. Disponível em: <http://www.4shared.com/get/64712151/e204d086/Classificao_das_fraturas_troca.html> Acesso em: 4 jun 2010.

SILVA, João Paulo Domingos. **Algoritmos de Classificação baseados em Análise Formal de Conceitos.** 2007. Dissertação de Mestrado. Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais. Disponível em: <http://www.bibliotecadigital.ufmg.br/dspace/bitstream/1843/RVMR-78RHNV/1/jo_opaulodomigossilva.pdf> Acesso em: 11 set 2009.

THEODORIDIS, Sergios; KOUTROUMBAS, Konstantinos. **Pattern recognition.** New York: Elsevier, 2006.

THOMAZ, Carlos E.; FEITOSA, Raul Q.; VEIGA, Álvaro. Design of Radial Basis Function Network as Classifier in Face Recognition Using Eigenfaces. **IEEE Computer Society.** v. 13, n. 3, p. 697-710, 2002. Disponível em: <http://www.dsp.utoronto.ca/juwei/Publication/Juwei_RBF.pdf> Acesso em: 11 de jan de 2010.

TINÓS, Renato. **Detecção e Diagnóstico de Falhas em Robôs Manipuladores Via Redes Neurais Artificiais.** Dissertação de Mestrado. Escola de Engenharia de São Carlos da Universidade de São Paulo. Disponível em: <http://biblioteca.universia.net/ficha.do?id=26825> Acesso em: 4 nov 2009.

TODESCO, José Leomar. **Reconhecimento de Padrões usando Rede Neuronal Artificial com uma Função de Base Radial:** uma aplicação na classificação de cromossomos humanos. Tese de Doutorado. Programa de Pós-Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina. Florianópolis, SC. 1995.

WAIKATO, University of, ***Weka 3***: *Data Mining Software in Java*. Waikato - New Zealand, 2005. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka> Acesso em: 04 de jun de 2009.

WANG, Dianhui; LEE, Nun Kion; DILLON, Tharam S.; HOOGENRAAD, N. J. Protein Sequences Classification Using Radial Basis Function (RBF) Neural Networks. **IEEE Control Systems Magazine.** v. 11, n. 3, p. 31-38, 1991. Disponível em: <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F8534%2F26965%2F01198161.pdf&authDecision=-203>. Acesso em: 10 de jan 2010.

WITTEN, I. H; FRANK, Eibe. **Data mining: practical machine learning tools and techniques with Java implementations.** San Francisco: Morgan Kaufmann, 2000.