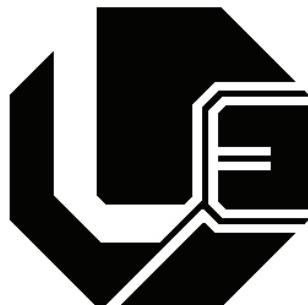


UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



**Sistema de classificação de plantas por meio
de suas folhas usando uma arquitetura
híbrida composta por algoritmos genéticos e
rede neural artificial**

THIAGO HENRIQUE BORGES

ABRIL

2013

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Sistema de classificação de plantas por meio
de suas folhas usando uma arquitetura
híbrida composta por algoritmos genéticos e
rede neural artificial**

THIAGO HENRIQUE BORGES

Texto da dissertação apresentada à Universidade Federal de Uberlândia, perante a banca de examinadores abaixo, como parte dos requisitos necessários para obtenção do título de Mestre em Ciências.

Banca examinadora:

Gilberto Arantes Carrijo, Dr - Orientador (UFU)
Edna Lúcia Flôres, Dra (UFU)
Antônio Cláudio Paschoarelli Veiga, Dr (UFU)
Benedito Alencar de Arruda, Dr (UFMT)

Sistema de classificação de plantas por meio de suas folhas usando uma arquitetura híbrida composta por algoritmos genéticos e rede neural artificial

Thiago Henrique Borges

Texto da dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica na Universidade Federal de Uberlândia, como requisito parcial à obtenção do título de Mestre em Ciências.

Prof. Dr. Gilberto Arantes Carrijo
Orientador

Prof. Dr. Edgard Afonso Lamounier Júnior
Coordenador do curso de Pós-Graduação

*A mente que se abre a uma nova ideia
jamais voltará ao seu tamanho original.*

- Albert Einstein

Agradecimentos

Gostaria de agradecer primeiramente a Deus que me presenteou com a oportunidade de realizar este trabalho.

Quero agradecer também à minha namorada Ariene pelo amor, carinho, compreensão e o apoio incondicional.

Aos meus pais, Carlos e Helena, e aos meus irmãos, Raphael e Bruno, pelos bons exemplos, pela paciência e pelo permanente incentivo.

Aos professores Gilberto e Edna pela confiança, orientação, amizade e motivação.

Aos diversos professores que tive e que contribuíram para que eu conseguisse chegar ao final desta árdua caminhada.

Aos meus colegas do Programa de Pós-graduação, Gustavo, Priscila e Paulo, pelo apoio, pela amizade e pelos incontáveis momentos de descontração.

E a todos aqueles que diretamente ou indiretamente contribuíram para a realização deste trabalho.

Meus sinceros agradecimentos a todos vocês!

Resumo

BORGES, Thiago H. & CARRIJO, Gilberto A. *Sistema de classificação de plantas por meio de suas folhas usando uma arquitetura híbrida composta por algoritmos genéticos e rede neural backpropagation*, Uberlândia, MG. Faculdade de Engenharia Elétrica - UFU, 2013.

O número de plantas com risco de extinção tem aumentado gradativamente. Com a finalidade de diminuir esse risco, faz-se necessário planejar métodos de proteção e identificação das espécies. A grande biodiversidade de plantas existentes no reino vegetal torna os modelos tradicionais de identificação e de taxonomia uma função muito complexa e lenta. A identificação de uma planta pode ser realizada observando várias características, tais com: frutos, sementes, flores, raízes, folhas e caule. A característica mais simples de ser utilizada nessa identificação são as folhas. Este trabalho apresenta um sistema híbrido e automático de identificação de plantas por meio de suas folhas. Esse sistema é composto por Algoritmos Genéticos (AG) e pela Rede Neural Artificial (RNA). O objetivo do AG é realizar uma pré-seleção de plantas formando um grupo de folhas desconhecidas que seriam a resposta mais provável, enquanto que a finalidade da RNA, treinada pelo algoritmo *backpropagation*, é classificar a folha considerando apenas o grupo calculado pelo AG. Vários testes foram realizados e os resultados obtidos mostram que o sistema híbrido atingiu uma taxa de reconhecimento de 93,2 %.

Palavras-chave

Reconhecimento de padrões. Reconhecimento de plantas. Algoritmos genéticos. Redes neurais artificiais.

Abstract

BORGES, Thiago H. & CARRIJO, Gilberto A. *Plants classification system through their leaves using a hybrid architecture composed of genetic algorithms and backpropagation neural network*, Uberlândia, MG. Faculty of Electrical Engineering - UFU, 2013.

The number of plants at risk of extinction has increased gradually. With the purpose of reducing the risk is necessary identify the species for planning protection methods. The biodiversity of species existing in the plant kingdom make the use of traditional models of recognition and taxonomy a process very complex and slow. The identification of a plant can be performed observing his features, such as: fruits, seeds, flowers, roots, leaves and stems. But the simplest feature used are the leaves. This paper presents a hybrid system for identifying plant based on leaf image. This system is composed by Genetic Algorithm (GA) and Artificial Neural Network (ANN). The role played by the GA is to perform a preselection of plants forming a group that the answer of an unknown leaf is more probable and the purpose of ANN, trained by backpropagation algorithm, is to classify the unknown leaf performing the search only in the group calculated by the AG. Several tests were conducted and the results obtained demonstrate that the hybrid system achieved a recognition rate of 93,2%.

Keywords

Patterns recognition. Plants recognition. Genetic algorithms. Artificial neural network.

Publicação

Aqui, é apresentado a publicação relacionada a este trabalho.

Borges, T.; Carrijo, G.; Flôres, E.; Veiga, A. & Pereira, M. **Modelo de taxonomia de plantas usando algoritmos genéticos e rede perceptron multicamadas.** In: Computer on the Beach 2013, Florianópolis, mar 2013.

Sumário

Sumário	ix
Lista de Figuras	xiii
Lista de Tabelas	xvi
1 Introdução	1
1.1 Introdução	1
1.2 Levantamento Bibliográfico	3
1.3 Objetivo e Metodologia desta Dissertação	4
1.4 Motivação deste Trabalho	5
1.5 Estrutura deste Trabalho	6
1.6 Considerações Finais deste Capítulo	7
2 Reconhecimento de Padrões	8
2.1 Introdução	8
2.2 Padrões e Classes de Padrões	9
2.3 Arranjos de Padrões	10
2.4 Fases do Reconhecimento de Padrões	12
2.4.1 Aquisição e pré-processamento	12
2.4.2 Extração de características	13
2.4.3 Treinamento	14
2.4.4 Classificador	14
2.4.5 Testes	20

2.5	Caracterização do Padrão da Folha	20
2.5.1	Introdução	20
2.5.2	Partes das folhas	21
2.6	Considerações Finais deste Capítulo	23
3	Algoritmos Genéticos	24
3.1	Introdução	24
3.2	Componentes do AG	25
3.2.1	Cromossomos	26
3.2.2	População	27
3.2.3	Função de aptidão	28
3.2.4	Seleção de pais	29
3.2.5	Elitismo	33
3.2.6	Reprodução	33
3.2.7	Mutação	36
3.3	Parâmetros de Controle	37
3.4	Execução de um AG Genérico	38
3.5	Considerações Finais deste Capítulo	39
4	Redes Neurais Artificiais	40
4.1	Introdução	40
4.2	Breve Histórico das Redes Neurais	42
4.3	Rede Perceptron Multi-Camadas	43
4.3.1	Algoritmo <i>backpropagation</i>	45
4.3.2	Taxa de aprendizagem (α)	48
4.3.3	<i>Momentum</i> (η)	48
4.3.4	Função de ativação	49
4.3.5	Inicialização dos pesos e dos bias	51
4.3.6	Camada intermediária	52
4.3.7	Treinamento local ou em lote	53
4.3.8	Critério de parada	54

4.4 Considerações Finais deste Capítulo	55
5 Arquitetura Híbrida Proposta	56
5.1 Introdução	56
5.2 Arquitetura Híbrida	57
5.2.1 Captura digital da imagem da folha	57
5.2.2 Processamento da imagem	57
5.2.3 Extração de características	59
5.2.4 Algoritmo genético	63
5.2.5 Redes neurais artificiais	69
5.3 Considerações Finais deste Capítulo	69
6 Resultados Obtidos	71
6.1 Introdução	71
6.2 Banco de Dados de Folhas Utilizado	71
6.3 Interface do Usuário do <i>Software</i> Desenvolvido para Realizar o Reconhecimento das Folhas	74
6.4 Resultados Obtidos	75
6.4.1 Utilização do algoritmo genético como classificador	76
6.4.2 Utilização da RNA com treinamento <i>backpropagation</i> como classificador	79
6.4.3 Utilização do sistema híbrido proposto na classificação de folhas	81
6.5 Comparação dos Resultados	87
6.6 Conclusões	88
7 Conclusões, Contribuições Deste Trabalho e Trabalhos Futuros	89
7.1 Introdução	89
7.2 Conclusões	89
7.3 Contribuições Deste Trabalho	90
7.4 Trabalhos Futuros	90
7.5 Considerações Finais deste Capítulo	91

Referências Bibliográficas	92
Apêndice	98
A Forma das folhas	98
B Formulação das equações do <i>backpropagation</i>	100

Lista de Figuras

2.1	<i>(a) Estrutura em forma de escada; (b) estrutura codificada em termos das primitivas a e b, formando uma representação por string ... abababab ... [18].</i>	11
2.2	<i>Exemplo de arranjo em árvore [18].</i>	12
2.3	<i>Principais etapas do reconhecimento de padrões. Adaptado de [48].</i>	13
2.4	<i>Diagrama de blocos do aprendizado supervisionado [14].</i>	14
2.5	<i>(a) Representação de doze vetores de características em um espaço de características bidimensional; (b) localização dos representantes das classes ω_1 e ω_2 no espaço de características; (c) classificação resultante dos vetores nas classes ω_1 e ω_2 [47].</i>	15
2.6	<i>Função discriminante [48].</i>	16
2.7	<i>Representação das muitas abordagens em reconhecimento estatístico [27].</i>	18
2.8	<i>Esquematização de um sistema sintático [46].</i>	19
2.9	<i>Principais componentes de uma folha. (a) Folha de Pereira com limbo simples; (b) folha de Roseira com limbo composto [2].</i>	21
3.1	<i>Ciclo de execução básico dos AGs. Adaptado de [7].</i>	25
3.2	<i>Exemplo de uma roleta de seleção.</i>	30
3.3	<i>Crossover simples. Adaptado de [31].</i>	34
3.4	<i>Crossover duplo [31].</i>	35
3.5	<i>Crossover uniforme. Adaptado de [31].</i>	35
3.6	<i>Crossover PMX [1].</i>	36
3.7	<i>Crossover cíclico [4].</i>	36

3.8	<i>Mutação simples. Adaptado de [4]</i>	37
4.1	<i>Neurônio biológico [54]</i>	41
4.2	<i>Propagação da sinapse dentro da rede de neurônios [5]</i>	41
4.3	<i>Neurônio proposto por McCulloch & Pitts [14]</i>	43
4.4	<i>Exemplo de arquitetura perceptron multi-camadas com duas camadas inter-mediárias [14]</i>	44
4.5	<i>Fluxo do processamento do algoritmo backpropagation. Adaptado de [37]</i>	45
4.6	<i>Modelo do neurônio da camada intermediária de uma rede MLP</i>	46
4.7	<i>Algumas funções de ativação [13]</i>	49
4.8	<i>(a) Representação da função logarítmica; (b) derivada da função logarítmica</i>	50
5.1	<i>Diagrama de blocos da arquitetura proposta neste trabalho</i>	56
5.2	<i>Histograma RGB [56]</i>	58
5.3	<i>(a) Imagem de uma folha no padrão RGB; (b) imagem convertida para níveis de cinza; (c) imagem binarizada com limiar para conversão de 0,95; (d) resultado da filtragem com um filtro de média de tamanho 3×3; (e) borda extraída da folha utilizando um filtro laplaciano de tamanho 3×3 [56]</i>	59
5.4	<i>Relação entre o comprimento e a largura fisiológica [56]</i>	60
5.5	<i>Elemento estruturante no formato de disco com raio igual a 3 [32]</i>	62
5.6	<i>(a) Imagem da folha no padrão de cinza; (b) imagem processada utilizando um fator estruturante de raio igual a 4; (c) imagem resultante</i>	62
5.7	<i>Região utilizada para gerar os novos valores das características</i>	63
5.8	<i>Classificação de um objeto desconhecido utilizando o algoritmo k-NN</i>	65
5.9	<i>Gráfico da densidade de vizinhos ($p(x)$) versus o espaço de característica (x) traçado segundo o algoritmo 3-NN. Adaptado de [15]</i>	66
5.10	<i>Representação do cromossomo na arquitetura híbrida</i>	68
6.1	<i>Amostra das folhas do banco de dados utilizado neste trabalho</i>	73

6.2	<i>Interface de entrada do software desenvolvido para realizar o reconhecimento das folhas.</i>	74
6.3	<i>Resultado do processamento realizado pelo software implementado.</i>	75
6.4	<i>Gráfico da taxa de acerto obtida utilizando o algoritmo genético como classificador.</i>	77
6.5	<i>Gráfico do tempo de processamento obtido utilizando o algoritmo genético como classificador.</i>	78
6.6	<i>Gráfico da taxa de acerto obtida utilizando a RNA como classificador.</i>	79
6.7	<i>Gráfico do tempo de processamento obtido utilizando a RNA como classificador.</i>	80
6.8	<i>Gráfico da probabilidade da resposta correta estar na população final do AG após a pré-seleção de plantas.</i>	81
6.9	<i>Gráfico do tempo de processamento obtido utilizando o AG como um pré-seletor de plantas com um IC de 95%.</i>	83
6.10	<i>Gráfico da taxa de acerto obtida utilizando o sistema híbrido como classificador considerando um IC de 95%.</i>	84
6.11	<i>Gráfico do tempo de processamento obtido utilizando como classificador o sistema híbrido em um IC de 95%.</i>	86

Lista de Tabelas

2.1	<i>Exemplos de aplicações de reconhecimento de padrões [27].</i>	10
5.1	<i>Parâmetros do AG.</i>	70
5.2	<i>Parâmetros da RNA.</i>	70
6.1	<i>Recursos computacionais utilizados neste trabalho.</i>	71
6.2	<i>Nome científico e quantidade de folhas.</i>	72
6.3	<i>Número de padrões de treinamento e de reconhecimento utilizados em cada ensaio.</i>	76
6.4	<i>Média da taxa de acerto obtida utilizando o algoritmo genético como classificador.</i>	77
6.5	<i>Tempo de processamento médio obtido utilizando o algoritmo genético como classificador.</i>	78
6.6	<i>Média da taxa de acerto obtida utilizando a RNA como classificador.</i>	80
6.7	<i>Tempo de processamento obtido utilizando a RNA como classificador.</i>	81
6.8	<i>Média da probabilidade da resposta correta estar na população final do AG após a pré-seleção das plantas.</i>	82
6.9	<i>Tempo médio de processamento obtido utilizando o AG como pré-seletor de plantas.</i>	82
6.10	<i>Média da taxa de acerto obtida utilizando o sistema híbrido.</i>	84
6.11	<i>Melhor resultado obtido no ensaio em que 10 padrões em cada classe são desconhecidos.</i>	85
6.12	<i>Tempo médio de processamento obtido utilizando o sistema híbrido como classificador.</i>	86

6.13 Comparação entre os algoritmos de classificação referenciados e proposto neste trabalho. 87

Capítulo 1

Introdução

1.1 Introdução

Os sentidos permitem aos seres vivos dotados de órgãos adequados, aprimorarem sua percepção do mundo. A visão é considerada o sentido mais avançado, consequentemente as imagens exercem um importante papel na percepção humana.

As imagens em conjunto com a redução progressiva da relação entre o preço e o desempenho dos computadores, e a expansão das redes de comunicação via Internet criaram oportunidades sem precedentes para o desenvolvimento contínuo do processamento digital de imagens (PDI), não existindo praticamente mais nenhuma área do conhecimento que não seja influenciada de alguma forma pelo PDI.

O processamento digital de imagens é uma subárea do processamento digital de sinais. O PDI consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio do computador. Para simplificar o estudo dessas técnicas dois níveis de abstração são normalmente estabelecidos: processamento de imagens (baixo nível) e análise de imagens (alto nível).

No processamento de imagens ocorrem operações primitivas, tais como, a filtragem, o realce de contraste e o aguçamento das imagens. Nos processos de baixo nível a entrada e a saída são imagens. Na análise de imagens, processos de alto nível, são realizadas funções como a segmentação, a separação de uma imagem em regiões ou objetos, a descrição

desses objetos com a finalidade de reduzi-los a uma forma adequada para o processamento computacional e a classificação (reconhecimento) dos objetos individuais. Portanto, o processamento de alto nível consiste em realizar as funções cognitivas normalmente associadas à visão [18].

A área da informática que realiza o processamento e a análise das imagens, incluindo o aprendizado e a capacidade de fazer inferências e agir com base em informações visuais é conhecida como visão computacional. Essa área representa um ramo da inteligência artificial (IA, do inglês *artificial intelligence*) cujo objetivo é simular a inteligência humana em sistemas automatizados.

Exemplos de sistemas automatizados que utilizam técnicas de PDI e de IA são [47]:

- reconhecimento ótico de caracteres que permite a identificação automática de caracteres escritos em diversos idiomas a partir das imagens de documentos;
- identificação de impressões digitais é uma atividade que possibilita identificar uma pessoa utilizando suas impressões digitais;
- identificação de faces permite a distinção de indivíduos a partir de imagens ou sequências de vídeo, auxiliando o reconhecimento de pessoas em fichas criminais, a criação de retratos falados e a monitoração em sistemas de vigilância;
- reconhecimento de assinaturas possibilita a verificação da autenticidade de assinaturas em cheques e outros documentos;
- identificação de vasos sanguíneos de imagens da retina pode auxiliar os médicos na detecção de doenças como a retinopatia diabética, causa comum de cegueira em adultos; e
- reconhecimento automático de placas de veículos possibilita o computador localizar e interpretar o conteúdo da placa de um veículo para uso em medição e planejamento do fluxo de tráfego, reconhecimento de veículos em situação irregular, controle de pedágios e estacionamentos .

O objetivo desta dissertação é realizar a identificação de plantas por meio de suas folhas. A identificação de espécies vegetais é uma importante, porém difícil tarefa, prin-

cipalmente devido à biodiversidade das espécies. Por isso, exige-se, desde o processo de aquisição dos exemplares até a lenta etapa de comparação com as amostras já catalogadas, um grande esforço por parte dos taxonomistas e/ou profissionais de áreas correlatas. Em face disso, este trabalho desenvolve um sistema de reconhecimento de plantas eficaz e automático utilizando uma arquitetura híbrida composta por Algoritmos Genéticos (AG) e Rede Neural Artificial (RNA).

Este capítulo apresenta o levantamento bibliográfico, o objetivo e a metodologia utilizada neste trabalho, a motivação e a estrutura desta dissertação.

1.2 Levantamento Bibliográfico

Métodos de visão computacional associados à técnicas de reconhecimento de padrões vêm sendo utilizados em procedimentos automatizados de reconhecimento de plantas. Alguns dos trabalhos correlatos a esta dissertação existentes na literatura são:

- Wu *et al.* [56] propuseram um algoritmo de extração de características baseado nas relações físicas das folhas, tais como, a razão entre o comprimento e a largura da folha. Eles utilizaram a rede neural probabilística (PNN) para classificar 32 tipos de folhas. A taxa de reconhecimento obtida utilizando esse sistema foi de 90,3%;
- Wang *et al.* [53] desenvolveram um sistema de reconhecimento que continha um extrator de características que analisava a geometria das folhas e um classificador baseado em hiperesfera para identificar 20 tipos de folhas. Nesse trabalho a taxa de reconhecimento obtida por eles foi de 92%;
- Uluturk *et al.* [51] desenvolveram um novo método de extração de características e utilizaram a rede PNN para classificar as mesmas folhas do banco de dados de Wu *et al.* O resultado obtido por eles foi de 92,5%;
- Na identificação de 20 espécies de plantas Gu *et al.* [20] combinaram a transformada wavelet e a interpolação gaussiana para extrair a borda e as nervuras das folhas. E posteriormente, para classificá-las eles utilizaram os métodos: vizinho mais próximo,

do inglês *1-nearest neighbor* (1-NN); k-vizinhos mais próximos (k-NN); e a rede PNN. As taxas de acerto obtidas por eles usando esses métodos foram de 93%, 85% e 91%, respectivamente;

- Kadir *et al.* [28] extraíram das folhas a forma, a cor e a textura. Como classificador eles utilizaram a rede PNN e obtiveram uma taxa de acerto de 94% na classificação das folhas do banco de dados de [56]; e
- Valliammal *et al.* [52] desenvolveram um sistema de reconhecimento chamado *Plant Leaf Image Retrieval* (PLIR) que é composto por um extrator de características que analisa o formato, a cor, a textura e as bordas das folhas. Para reduzir o tamanho do vetor de características obtido durante a extração, eles utilizaram o algoritmo genético (AG) que seleciona em cada execução as características mais relevantes, e, por último, o classificador adotado no trabalho foi a máquina de vetores suporte (SVM). A taxa de acerto obtida por eles foi de 94%.

1.3 Objetivo e Metodologia desta Dissertação

O objetivo desta dissertação é desenvolver um sistema de reconhecimento de plantas utilizando uma arquitetura híbrida composta por algoritmos genéticos (AG) e pela rede neural artificial (RNA) com treinamento *backpropagation*. A finalidade do AG neste trabalho é realizar uma pré-seleção de plantas que possuem características de folhas semelhantes a da folha a ser reconhecida, enquanto que o objetivo da RNA é classificar a folha realizando a busca somente dentro do grupo pré-selecionado pelo AG.

Para atingir o objetivo citado no parágrafo anterior um estudo a respeito das teorias de algoritmos genéticos e redes neurais artificiais é desenvolvido com a finalidade de mostrar as vantagens e as desvantagens de cada método quando utilizados para reconhecer as folhas das plantas.

Em seguida, é apresentado os fundamentos do sistema híbrido e os detalhes de sua implementação. São mostrados também todos os testes e as simulações necessárias para avaliar a eficiência do sistema híbrido.

1.4 Motivação deste Trabalho

Além das plantas estarem presentes em todos os lugares do planeta elas desenvolvem uma importante função no meio ambiente, pois não seria possível a presença de vida na Terra se as plantas não existissem ou se deixassem de existir. Algumas das funções das plantas são:

- realização da fotossíntese;
- atenuação do efeito estufa; e
- manutenção dos solos com a finalidade de facilitar a penetração da água e evitar a erosão prematura.

As plantas também são utilizadas como matérias-primas para a fabricação de medicamentos, tecidos e alimentos. Um exemplo do uso inadequado das plantas é a produção de drogas alucinógenas.

Desta forma, alguns motivos que justificam o desenvolvimento de pesquisas a respeito da identificação e da classificação de plantas são:

- evitar o risco iminente de extinção de algumas espécies conhecidas e outras que ainda são desconhecidas pelos pesquisadores e que poderiam servir de matéria-prima em diversas finalidades; e
- combater plantações ilícitas por meio de sensoriamento remoto utilizando câmeras de alta resolução embarcadas em aeronaves não-tripuladas (VANTs). Assim, pode-se evitar a necessidade de fixar bases e pessoas em lugares inóspitos.

Portanto, com a finalidade de catalogar e proteger a flora terrestre bancos de dados de plantas estão sendo construídos. Para lidar com esse volume de informações o desenvolvimento de métodos de classificação rápidos e eficientes tornou-se uma importante área de pesquisa.

O *software* desenvolvido neste trabalho pode ser utilizado em pesquisas e também pela sociedade em geral devido à popularização da microinformática que facilitou a acessibilidade aos novos sistemas de computação, tais como: *smartphones*, *notebooks*, *tablets*,

etc. Pela alta capacidade de processamento desses sistemas é possível que qualquer pessoa em qualquer lugar utilize o *software* desenvolvido.

Softwares de identificação de plantas que utilizam esses aparelhos vem surgindo no mercado, como por exemplo o Leafsnap. Este software é disponível gratuitamente para aparelhos que utilizam a plataforma iOS. Na sua versão atual ele foi desenvolvido para reconhecer plantas da região Nordeste dos Estados Unidos, mas já existem planos de expansão para reconhecer plantas de todo esse país [29].

1.5 Estrutura deste Trabalho

Esta dissertação foi estruturada da seguinte forma:

- Capítulo 1: apresenta o levantamento bibliográfico, o objetivo e a metodologia utilizada nesta dissertação, a motivação e a estrutura deste trabalho;
- Capítulo 2: define as principais arquiteturas existentes para realizar o reconhecimento de padrões e apresenta o padrão folha que é utilizado para reconhecer as plantas neste trabalho.
- Capítulo 3: mostra os fundamentos dos algoritmos genéticos, abordando sua inspiração biológica, conceitos fundamentais e seus principais componentes;
- Capítulo 4: descreve os fundamentos das redes neurais artificiais, as principais arquiteturas e os métodos de treinamento, bem como uma descrição mais detalhada do algoritmo de treinamento utilizado neste trabalho;
- Capítulo 5: apresenta o modelo proposto neste trabalho com seus componentes e parâmetros;
- Capítulo 6: apresenta o banco de dados de folhas utilizado neste trabalho, a interface do *software* desenvolvido para realizar o reconhecimento das plantas por meio de suas folhas, os tempos de processamento e os resultados obtidos nas simulações, comparando esses resultados com os resultados existentes na literatura; e

- Capítulo 7: apresenta as conclusões, as contribuições e os trabalhos futuros que poderão ser realizados a partir desta dissertação.

1.6 Considerações Finais deste Capítulo

Este capítulo apresentou o levantamento bibliográfico, o objetivo e a metodologia utilizada nesta dissertação, a motivação e a estrutura deste trabalho.

O próximo capítulo define as principais arquiteturas existentes para realizar o reconhecimento de padrões e apresenta o padrão folha que é utilizado para reconhecer as plantas neste trabalho.

Capítulo 2

Reconhecimento de Padrões

2.1 Introdução

Padrões são os meios pelos quais o mundo é interpretado pelos homens e pelos animais e, a partir dessa interpretação são elaboradas atitudes e decisões. O ser humano é hábil em reconhecer rostos, vozes, caligrafias e, até mesmo, estados de humor de pessoas conhecidas. Alguns animais também têm essa característica desenvolvida, tais como os cães farejadores, que vasculham bagagens em terminais de desembarque em busca de drogas. O grau de refinamento do reconhecimento de padrões, por parte do ser humano, pode chegar ao ponto de distinguir uma pintura de um grande mestre daquela realizada por um exímio falsário ou, ainda mais, ele pode estabelecer uma tomada de decisão por parte de um operador em um dia de grande movimentação em uma bolsa de valores.

O aprendizado de padrões nos seres vivos está diretamente relacionado aos estímulos pelos quais os indivíduos foram expostos e esse aprendizado evolui com a experiência. O atual desafio da ciência é desenvolver máquinas que reproduzem tais comportamentos. Pesquisas em reconhecimento de voz, de face, de caligrafias, de textos impressos, etc., estão em fase avançada de desenvolvimento, mas muito ainda precisa ser realizado para que seu desempenho se assemelhe ao do ser humano [11].

O termo reconhecimento de padrões (RP), do inglês *pattern recognition*, surgiu no início da década de 60, para denominar a promissora área de desenvolvimento de algorit-

mos de reconhecimento de formas geométricas.

Existem muitas definições sobre reconhecimento de padrões. Duda e Hart [15], caracterizaram RP como sendo “o campo que consiste no reconhecimento de regularidades significativas em meios ruidosos e complexos, realizados por máquinas”, ou ainda, “a procura por uma estrutura em dados”. Ross [45] definiu o reconhecimento de padrões como sendo “um processo de identificação de estruturas nos dados por comparações com estruturas conhecidas”.

Em linhas gerais, o principal objetivo da teoria de reconhecimento de padrões é a classificação de padrões. Existem diversas abordagens para o RP realizar a classificação, no entanto, duas destas abordagens estão presentes em praticamente toda a literatura que apresenta o assunto, podendo ser consideradas como abordagens clássicas de RP. Essas abordagens são o RP estatístico e o RP sintático ou estrutural. Exemplos de abordagens mais recentes são as redes neurais e a abordagem que utiliza algoritmos evolutivos.

Este capítulo apresenta as principais definições, características e fases de um sistema de reconhecimento de padrões, com ênfase nos classificadores clássicos. Além disso é realizada uma breve descrição a respeito do padrão folha que é utilizado para reconhecer as plantas neste trabalho.

2.2 Padrões e Classes de Padrões

O termo característica é usado frequentemente na literatura de reconhecimento de padrões para denominar um descritor que pode ser quantitativo ou qualitativo.

Todo objeto pode ser identificado e representado por meio de suas características. E o conjunto de todas as características ou o arranjo das características de um mesmo objeto recebe o nome de padrão.

Uma classe de padrões é uma família de padrões que compartilham propriedades comuns. Normalmente no meio científico as classes de padrões são indicadas por ω_1 , $\omega_2, \dots, \omega_W$ onde W é o número de classes [18].

A Tabela 2.1 apresenta algumas das principais aplicações que utilizam reconhecimento de padrões e seus respectivos padrões de entrada e classes de padrões.

Tabela 2.1: Exemplos de aplicações de reconhecimento de padrões [27].

Domínio do Problema	Aplicação	Padrão de Entrada	Classes de Padrões
Bioinformática	Análise de sequências	DNA/Sequência de proteína	Tipos de genes conhecidos/padrões
Mineração de dados	Pesquisa por padrões significantes	Pontos em espaços multidimensional	Agrupamentos compacto e bem separados
Classificação de documentos	Pesquisa em Internet	Documentos de texto	Categorias de documentos (Negócios, Esportes, etc.)
Reconhecimento de caracteres	Leitura de máquinas para cegos	Imagem do documento	Caracteres alfanuméricos, palavras.
Recuperação em banco de dados multimídia	Pesquisa em Internet	Vídeo clipe	Gêneros de filme
Reconhecimento biométrico	Identificação de pessoas	Faces, íris, impressões digitais	Controle de acesso à usuários autorizados
Sensoriamento remoto	Prognóstico de rendimento de colheita	Imagem multiespectral	Categoria de solo, padrões de crescimento de colheita
Reconhecimento de voz	Atendimento a lista sem ajuda do operador	Onda sonora	Padrão de voz, palavra falada

2.3 Arranjos de Padrões

Três arranjos de padrões são normalmente utilizados na prática. Esses arranjos são: os vetores (usados em descrições quantitativas), as *strings* e as árvores (utilizados em descrições estruturais).

Os vetores de características geralmente são representados por letras minúsculas em negrito, como por exemplo: **x**, **y** e **z**. E assumem a forma de vetor coluna, como mostrado na Equação (2.1).

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.1)$$

onde:

- x_i i -ésimo descriptor; e

- n número de descritores associados ao padrão.

Os descritores presentes nos vetores de características correspondem às medidas físicas dos padrões a que eles se referem, tais como: área, perímetro e largura. Portanto, eles podem assumir qualquer valor real maior do que zero.

A utilização de abordagens estruturais são mais indicadas para o grupo de problemas de reconhecimento, em que não somente as medidas quantitativas são importantes para identificar as classes, mas também as relações espaciais dos objetos (fronteiras, forma, regiões, etc.).

A Figura 2.1(a) mostra um padrão simples em forma de escada. Esse padrão pode ser amostrado e expresso em termos de um vetor de características. No entanto, a estrutura básica composta das repetições de dois elementos primitivos simples é perdida nesse método de descrição. Uma descrição mais significativa pode ser realizada por meio da definição dos elementos a e b . Dessa maneira o padrão assume a forma de uma *string* de símbolos finitos, $\omega = \dots abababab\dots$, como ilustrado na Figura 2.1(b). A estrutura dessa classe particular de padrões é obtida a partir da conectividade do tipo “cabeça-à-cauda”, além de permitir apenas dois tipos de símbolos, fator esse que facilita a análise [18].

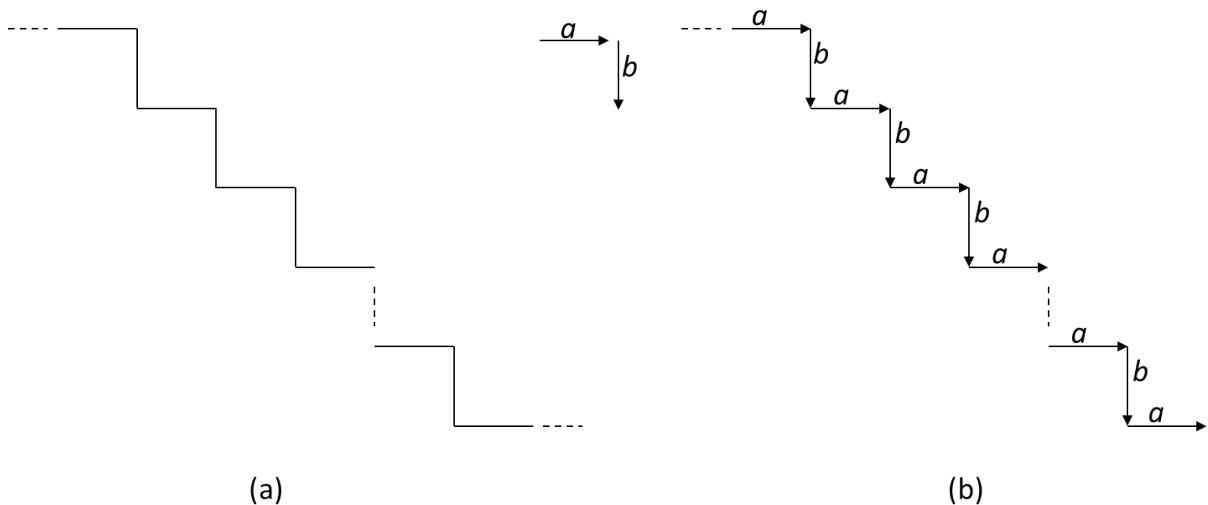


Figura 2.1: (a)Estrutura em forma de escada; (b)estrutura codificada em termos das primitivas a e b , formando uma representação por string $\dots abababab\dots$ [18].

Representações por *strings* geram de forma adequada padrões de objetos e outras entidades cujas estruturas baseiam-se em conectividades relativamente simples, geralmente associadas à fronteira da forma. Uma abordagem mais poderosa e adequada à muitas

aplicações é o uso de descrições por árvores. Basicamente, a maioria dos esquemas de organização hierárquica resultam em estruturas do tipo árvore. A Figura 2.2 mostra um exemplo de árvore do tipo “cabeça para baixo” que pode ser usada para representar uma cidade por meio da relação estrutural “composto de”.

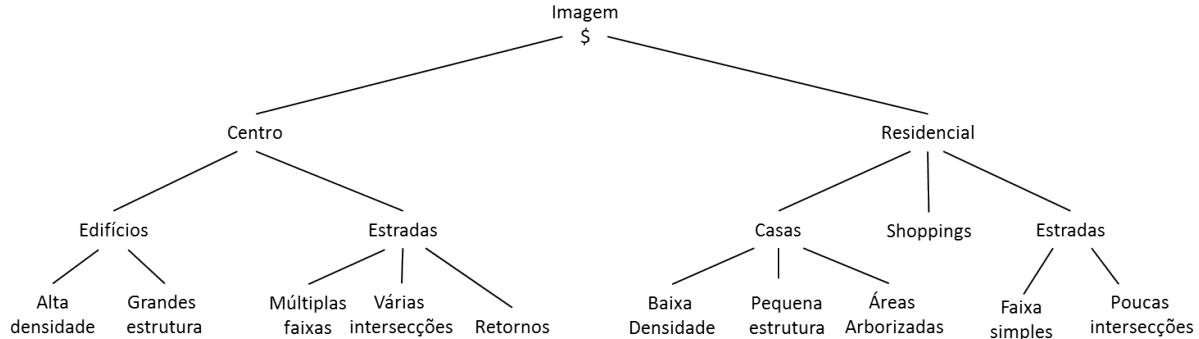


Figura 2.2: Exemplo de arranjo em árvore [18].

Os arranjos do tipo árvore começam a ser analisados a partir da raiz (representada pelo símbolo \$). Na Figura 2.2, a raiz representa a imagem de uma cidade. O próximo nível indica que a imagem é composta por um centro e uma zona residencial, que por sua vez, é composta por moradias, rodovias e shoppings. O nível seguinte descreve ainda mais as moradias e as rodovias. Esse tipo de subdivisão pode continuar até atingir o limite da capacidade de definição das diferentes regiões de uma cidade.

O conjunto de todos os padrões possíveis, independente do arranjo que eles utilizam forma o espaço do padrão denominado também de espaço de características [18].

2.4 Fases do Reconhecimento de Padrões

As principais fases de um sistema de reconhecimento de padrões são: aquisição/pré-processamento, extração de características, treinamento, módulo de classificação e testes. A Figura 2.3 apresenta um esquema desse sistema.

2.4.1 Aquisição e pré-processamento

O processo de reconhecimento inicia utilizando um sensor, tal como, uma câmera fotográfica digital ou *scanner* para capturar o padrão a ser reconhecido. A finalidade da

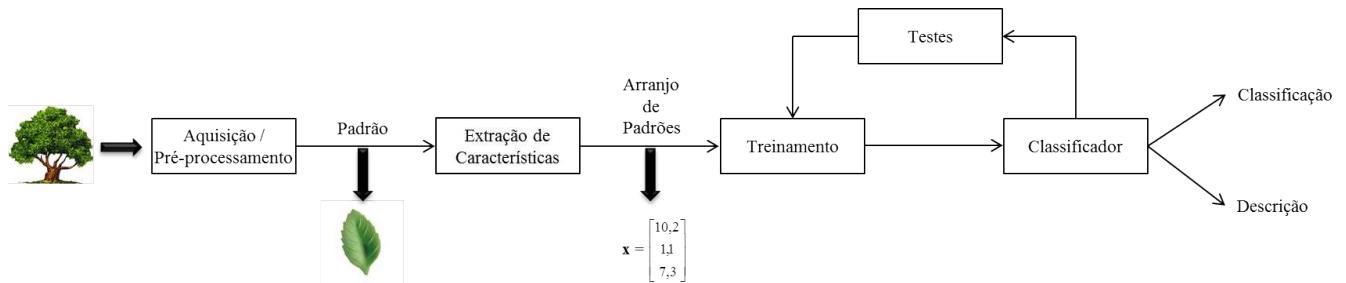


Figura 2.3: *Principais etapas do reconhecimento de padrões. Adaptado de [48].*

etapa de pré-processamento é segmentar (separar) o padrão de interesse do fundo, remover ruídos, normalizar o padrão e qualquer outra operação que venha a contribuir na definição de uma representação concisa do padrão [27].

2.4.2 Extração de características

A extração de características é o processo pelo qual um problema é simplificado de forma a ser possível a sua implementação computacional. Na prática, o volume de informações associado a um padrão é muito grande, não raramente, esse volume tende ao infinito. Portanto, reduzir o volume de informações torna-se essencial. Esse processo de redução do volume de características é realizado por meio de cálculos estatísticos (médias, desvios-padrões, etc.) e pela escolha das características mais relevantes por meio de transformações matemáticas (análise da componente principal).

Frequentemente, ocorre no senso comum um equívoco relacionado à extração de características. Isso é porque existe uma tendência de supor que se for utilizado o maior número possível de informações disponíveis em um determinado problema, a melhor solução sempre é encontrada. Porém, um volume muito grande de dados implica em sobrecarga dos sistemas tornando-os excessivamente restritivos e lentos. Por exemplo: se uma pessoa tomasse a cada momento todas as informações que observasse em um objeto, jamais poderia reconhecê-lo, porque esse objeto se transforma a cada instante, quer seja pela sua mudança de estado ou pela mudança no plano de observação adotado.

Por outro lado, a falta de informações leva à classificação falsa, uma vez que se pode considerar objetos semelhantes de classes diferentes como sendo o mesmo objeto. Para obter uma boa extração de características, deve-se procurar encontrar o conjunto

resumido de características de forma adequada ao problema em tratamento [42].

2.4.3 Treinamento

Uma vez extraídas as características dos padrões é necessário fornecer ao módulo de treinamento o arranjo de padrões.

A etapa do treinamento consiste na aprendizagem do sistema para que o mesmo seja capaz de identificar um padrão desconhecido como sendo pertencente a uma classe.

Se o treinamento do classificador exige amplo conhecimento *a priori* da estrutura estatística dos padrões a serem analisados e o padrão de entrada é identificado como membro de uma classe pré-definida pelos padrões de treinamento, o classificador é chamado de classificador paramétrico e a classificação é da forma supervisionada.

No aprendizado supervisionado a entrada e a saída desejada são fornecidas por um supervisor (professor) externo que indica um comportamento para o sistema. Esse supervisor direciona o processo e ajusta os parâmetros do sistema a cada etapa do treinamento com o objetivo de encontrar uma solução. A Figura 2.4 mostra o diagrama de blocos do aprendizado supervisionado.

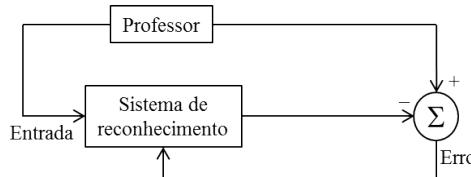


Figura 2.4: Diagrama de blocos do aprendizado supervisionado [14].

Por outro lado, se o classificador utiliza determinado modelo estatístico, ajustando-se mediante os processos adaptativos e a associação entre os padrões acontece com base nas similaridades entre os padrões de treinamento, o classificador é chamado de classificador não-paramétrico e a classificação é da forma não supervisionada.

2.4.4 Classificador

Conforme mostrado anteriormente neste capítulo, pode-se dividir as abordagens de reconhecimento de padrões em duas categorias: a clássica composta pelos métodos

estatístico e sintático (ou estrutural) e as novas abordagens compostas pelos métodos de algoritmos evolutivos e pelas redes neurais.

Este capítulo apresenta apenas os métodos da abordagem clássica. As demais abordagens são apresentadas com mais detalhes nos próximos capítulos.

Reconhecimento de padrões estatístico

De forma geral, no reconhecimento estatístico de padrões um padrão é representado por um conjunto de n características, ou atributos, arranjados em um vetor de características n -dimensional. São adicionados a esta abordagem, conceitos estatísticos bem conhecidos e fundamentados da teoria de decisão para estabelecerem fronteiras de separação entre as classes.

Um algoritmo simples de estabelecimento de fronteiras é aquele que atribui a um padrão a classe que possui a menor distância em relação ao vetor de características representativo, como ilustrado na Figura 2.5.

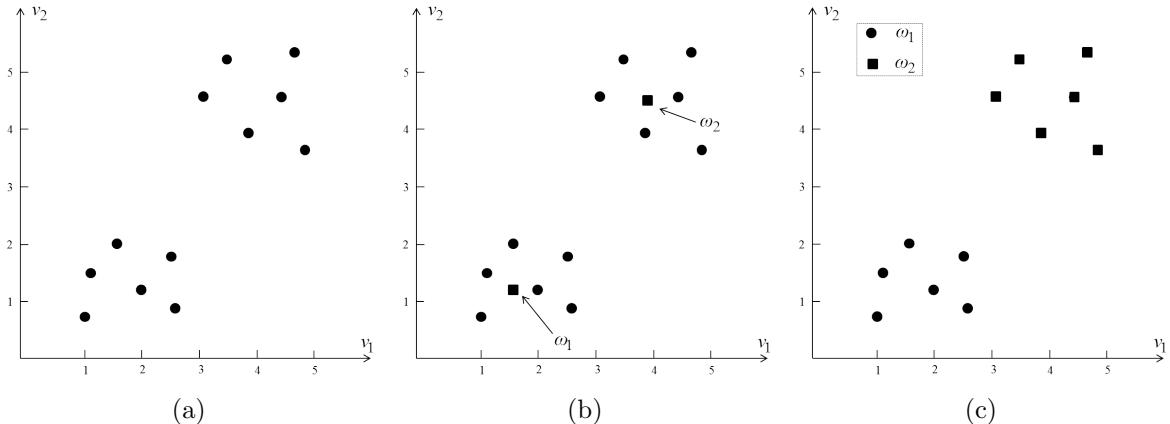


Figura 2.5: (a) Representação de doze vetores de características em um espaço de características bidimensional; (b) localização dos representantes das classes ω_1 e ω_2 no espaço de características; (c) classificação resultante dos vetores nas classes ω_1 e ω_2 [47].

Na Figura 2.5(a) doze padrões são apresentados em um plano cartesiano bidimensional. Após a determinação dos vetores característicos mais representativos de cada classe, ω_1 e ω_2 , na Figura 2.5(b), um algoritmo de classificação, que considera a distância Euclidiana como medida de dissimilaridade entre os representantes de cada classe e os padrões, é utilizado para obter a classificação final mostrada na Figura 2.5(c), em que os

padrões marcados com um círculo pertencem à classe ω_1 e os marcados com um quadrado pertencem à classe ω_2 [47].

Uma variante desse modelo de classificação é o modelo que em vez de utilizar a medida de dissimilaridade diretamente para determinar qual classe um determinado padrão deve ser atribuído utiliza uma superfície de decisão.

O modelo de superfície de decisão analisa a similaridade entre os padrões de cada classe destacando a aproximação entre os objetos no espaço de características. Dessa forma, as classes formam agrupamentos que podem ser separados por uma curva discriminante ou uma hipersuperfície. E a classificação de um padrão desconhecido acontece de acordo com o posicionamento do seu vetor de características nas regiões determinadas.

A Figura 2.6 exemplifica esta relação.

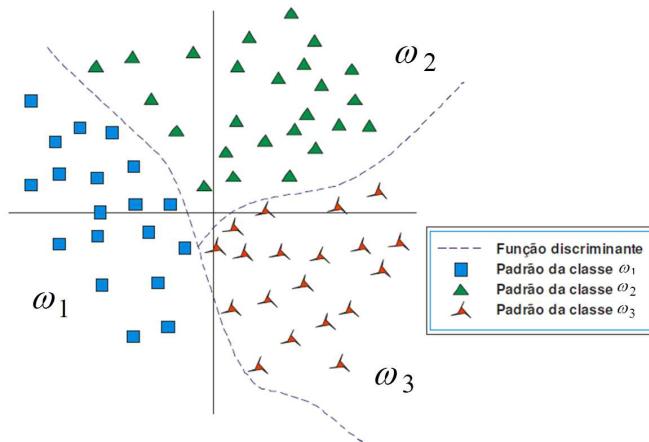


Figura 2.6: *Função discriminante* [48].

Esses algoritmos para classificação podem ser separados em dois grupos conforme a superfície de decisão: lineares e não-lineares. Os classificadores lineares são aqueles em que a superfície de decisão é uma reta ou um hiperplano quando o vetor de características contém mais de duas medidas. Por outro lado, quando a superfície de decisão é mais complexa que uma reta ou um hiperplano o classificador é denominado não-linear.

Outros modelos estatísticos mais complexos, tais como: a regra de decisão de Bayes, a máxima verossimilhança (que pode ser vista como um caso particular da regra de Bayes) e a Neyman-Pearson podem ser utilizadas para definir a fronteira de decisão. Mas, esses modelos necessitam de um conhecimento estatístico mais aprofundado dos vetores de características.

De acordo com Jain et al. [27] muitos métodos são utilizados para projetar um classificador estatístico de padrões. Esses métodos são traçados de acordo com o tipo de informação disponível sobre a densidade condicional das classes. Se todas as densidades condicionais das classes forem inteiramente especificadas, então a regra de decisão de Bayes ótima pode ser utilizada no classificador. Entretanto, as densidades condicionais das classes são geralmente desconhecidas e devem ser determinadas (aprendidas) a partir dos padrões de treinamento disponíveis.

Se a forma da densidade condicional da classe é conhecida (por exemplo, Gaussiana multivariada), mas alguns dos parâmetros da densidade, tais como: média dos vetores e matrizes covariância, são desconhecidos, então se tem um problema paramétrico de decisão. Uma estratégia comum para este tipo de problema é substituir os parâmetros incógnitos nas funções de densidade por seus valores estimados, resultando no chamado classificador de Bayes conectado (“*plug-in*”). A regra bayesiana nesta situação requer informações adicionais a respeito da forma de uma distribuição *a priori* dos parâmetros incógnitos.

Se a forma da densidade condicional da classe é incógnita, então se opera em um modo não-paramétrico. E neste caso ou se estima a função densidade ou se constrói diretamente a fronteira de decisão baseado nos dados de treinamento (por exemplo, a regra dos k vizinhos mais próximos).

As várias dicotomias que aparecem no reconhecimento estatístico são apresentadas por Jain et al. [27] e são mostradas na Figura 2.7. Percorrendo essa figura do topo para baixo e da esquerda para a direita, a dificuldade do problema de classificação aumenta, pois menos informações são disponíveis para a obtenção da solução. Isso justifica a preferência dos projetistas pela implementação usando a regra de decisão de Bayes.

A área da análise de agrupamento (*clustering analysis*) trabalha essencialmente com problemas de tomada de decisão entre as formas de aprendizagem não-paramétrica e não supervisionada. Por conseguinte, em algumas situações, o número de categorias ou grupos pode às vezes não ser especificado levando a uma tarefa de encontrar uma categorização razoável para os dados (se ela existir). Os algoritmos de análise de agrupamentos ligados a várias técnicas de visualização e projeção de dados multidimensionais são também

referenciados como métodos de análise exploratória de dados.

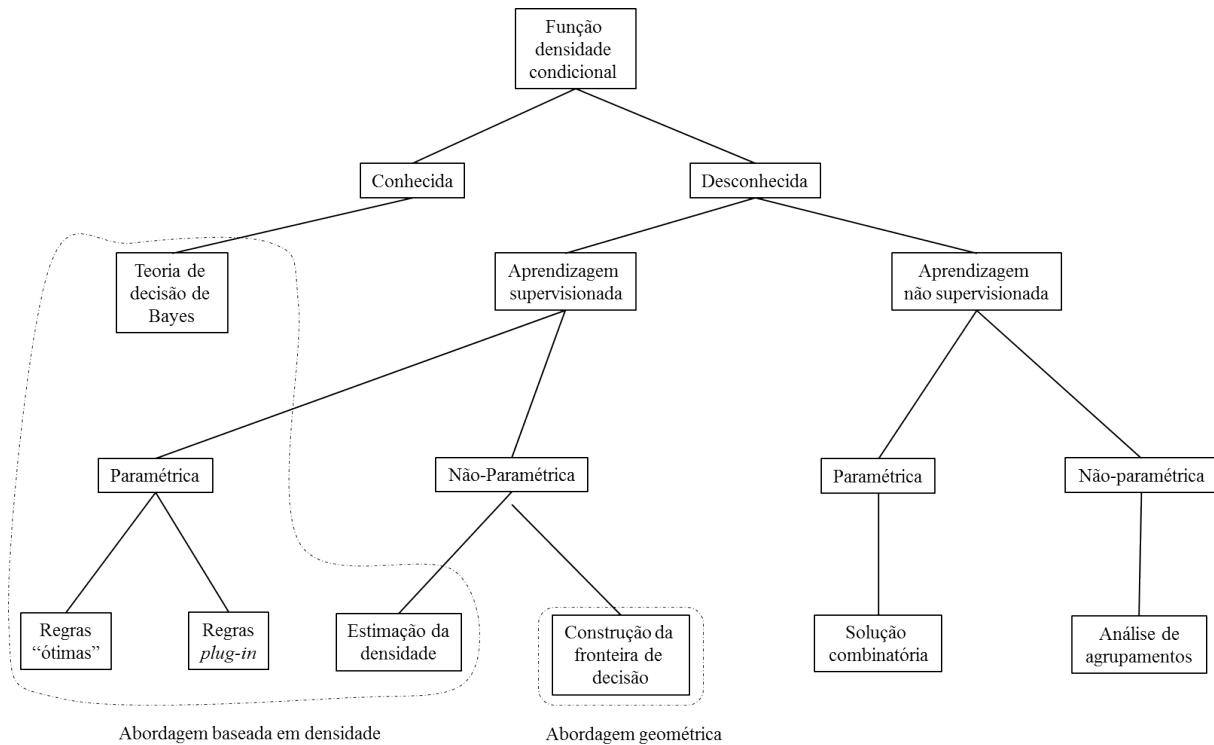


Figura 2.7: *Representação das muitas abordagens em reconhecimento estatístico [27].*

Outra dicotomia em reconhecimento estatístico de padrões pode ser baseada nas fronteiras de decisão obtidas diretamente (abordagem geométrica) ou indiretamente (abordagem probabilística baseada na densidade) como mostrado na Figura 2.7. A abordagem probabilística requer que primeiro seja estimada a função densidade de probabilidade e depois seja construída as funções discriminantes que especificam as fronteiras de decisão. Por outro lado, a abordagem geométrica geralmente conduz diretamente à fronteira de decisão a partir da otimização de algumas funções de risco. É possível mostrar que sob certas suposições em funções discriminantes as duas abordagens são equivalentes [27].

Reconhecimento de padrões sintático

Muitas vezes, as informações significativas em um padrão não se resumem apenas aos valores numéricos de um conjunto de características. A inter-relação e a interconexão espaciais dos padrões produzem informações estruturais importantes que facilitam sua descrição e classificação. Portanto, torna-se necessário utilizar extratores de características sintáticos para quantificar as informações estruturais importantes e poder avaliar a

similaridade estrutural entre os padrões.

Os elementos que compõem um sistema sintático são mostrados na Figura 2.8.

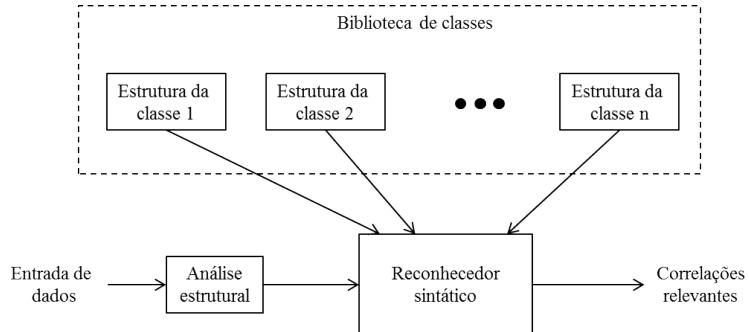


Figura 2.8: *Esquematização de um sistema sintático [46]*.

As estruturas das classes de padrões devem ser inferidas a partir dos padrões de treinamento (Figura 2.8). Uma vantagem do reconhecimento sintático é que além de fazer a classificação, essa abordagem também provê uma descrição de como os padrões são gerados a partir dela. Esse paradigma pode ser aplicado em padrões com estrutura definida, como imagens com textura e análise de formas de contornos.

O reconhecimento sintático pode ser realizado de duas formas: abordagem hierárquica e abordagem relacional [50].

Na abordagem hierárquica os sistemas de reconhecimento sintáticos são baseados na geração e na análise de padrões complexos por uma decomposição hierárquica em padrões mais simples. Um exemplo dessa abordagem é a linguagem escrita, na qual um parágrafo pode ser decomposto em sentenças, que por sua vez, podem ser decompostas em palavras, estas em letras, que ainda podem ser decompostas em traços. Os subpadrões elementares (mais simples) são denominados de primitivas, e os padrões mais complexos são representados por combinações das primitivas [27].

Na abordagem relacional as propriedades e as relações das características observadas são usadas para construir modelos baseados em estruturas mais complexas. Geralmente, a informação relacional nestes modelos é a única informação significativa. Desta forma, muitas outras classes compartilham a mesma característica de baixo-nível [46].

2.4.5 Testes

Um sistema de reconhecimento de padrões deve ser treinado usando as amostras de treinamento disponíveis não importando qual regra de classificação ou decisão é usada. Assim, o desempenho de um classificador depende tanto do número de amostras de treinamento disponíveis quanto dos valores específicos das amostras. Ao mesmo tempo, o objetivo de se projetar um sistema de reconhecimento é classificar futuras amostras de testes que serão provavelmente diferentes das amostras de treinamento. Então, ao se otimizar um classificador para maximizar o seu desempenho no conjunto de treinamento, nem sempre pode resultar no desempenho desejado em um grupo de teste [27].

A habilidade de generalização de um classificador está relacionada ao seu desempenho na classificação de padrões de teste que não foram utilizados no processo de treinamento. Uma habilidade de generalização pobre de um classificador pode ser atribuída a qualquer um dos seguintes fatores:

1. o número de atributos e/ou características é relativamente tão grande quanto ao número de amostras de treinamento;
2. o número de parâmetros desconhecidos associados ao classificador é grande (exemplo, classificadores polinomiais ou uma rede neural muito grande); e
3. o classificador é muito otimizado para o conjunto de treinamento (supertreinamento).

2.5 Caracterização do Padrão da Folha

2.5.1 Introdução

A identificação de uma planta pode ser realizada observando características tais como: raízes, flores, caule, frutos e sementes. Porém, devido à complexidade geométrica dessas partes, a característica mais simples de ser analisada é a folha.

A folha é um apêndice caulinar presente em quase todos os vegetais superiores, salvo

exceções, como por exemplo, em algumas espécies das famílias *Euphorbiaceae* e *Cactaceae*, em que na última é comum a transformação das folhas em espinhos.

A folha é o órgão vegetativo que apresenta grande polimorfismo e adaptações a diferentes ambientes e funções. Em geral, as folhas são estruturas planas (finas e amplas) para melhor captar a luz solar com várias microaberturas (estômatos) que são responsáveis pela absorção do gás carbônico na atmosfera.

As folhas são também a principal fonte de perda de água das plantas, chegando ao ponto de em uma transpiração excessiva levar a planta a um quadro de desidratação, que pode causar até mesmo sua morte. Em um sentido amplo, a forma e a anatomia da folha devem possibilitar uma relação que permita a captura da luz e a absorção de gás carbônico, com a finalidade de evitar a perda excessiva de água.

Além da fotossíntese, outros processos fisiológicos importantes para as plantas têm origem nas folhas. Esses processos são: respiração, transpiração e reserva de nutrientes. Para os seres vivos as folhas são importantes na alimentação, sendo amplamente utilizadas pelo homem na indústria e no comércio [12].

2.5.2 Partes das folhas

Uma folha típica é composta das seguintes partes: limbo (lâmina), pecíolo, bainha, e um par de apêndices foliares chamados estípulas localizadas na base do pecíolo. As Figuras 2.9 (a) e (b) mostram essas partes em uma folha de Pereira e em uma folha de Roseira, respectivamente.

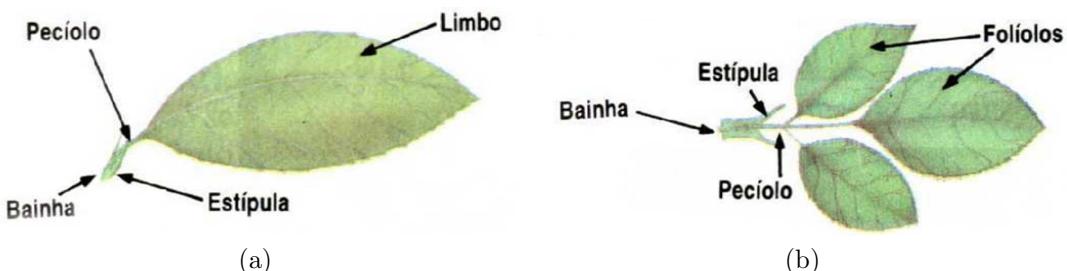


Figura 2.9: Principais componentes de uma folha. (a) Folha de Pereira com limbo simples; (b) folha de Roseira com limbo composto [2].

O pecíolo é o eixo que sustenta a folha e serve para unir a lâmina foliar ao caule.

Geralmente ele é arredondado na porção inferior e achatado ou côncavo na porção superior. O pecíolo auxilia na sustentação da lâmina, mas ao mesmo tempo ele é flexível. O pecíolo desempenha um importante papel na exposição da lâmina foliar à luz (fototropismo), podendo ainda estar unido à base da lâmina foliar, como ocorre na maioria das plantas, ou preso no meio da lâmina foliar.

A bainha é a porção terminal do pecíolo que adere ao caule. Ela é geralmente bem desenvolvida como no caso das *Poaceae* e na família das *Apiaceae* a função da bainha é proteger às gemas axilares.

As estípulas são estruturas laminares e geralmente aparecem aos pares. Elas estão presentes na base das folhas e variam muito em forma e tamanho, podendo ser livres ou não. As estípulas podem ser axilares (ocupando a posição axilar) ou interpeciolares (localizadas entre os pecíolos das folhas diferentes). Em alguns casos as estípulas são bastante desenvolvidas e auxiliam na fotossíntese, compensando assim a redução do limbo de algumas folhas, como acontece, por exemplo, na ervilha. E ainda elas podem estar transformadas em espinhos, como ocorre na coroa-de-cristo. Em algumas espécies as estípulas estão presentes apenas quando a folha é bem jovem, caindo em seguida. Nessas espécies as estípulas têm o papel de proteger as gemas, do mesmo modo que as escamas de gemas.

O limbo é uma lâmina verde sustentada pelas nervuras, sua estrutura planar possibilita uma ampla área para a captação da luz solar e do gás carbônico. A forma da folha é definida pela forma geral do limbo e apresenta grande variedade (o Apêndice A deste trabalho mostra as principais formas que o limbo pode assumir). As folhas geralmente são classificadas quanto à divisão do limbo, podendo ser simples quando a lâmina é única (Figura 2.9(a)), ou composta (Figura 2.9(b)), quando a lâmina é dividida em folíolos (subdivisões das folhas das plantas vasculares). O limbo foliar tem grande importância nos trabalhos de taxonomia, filogênese e identificação de plantas. Uma enorme gama de conceitos foi desenvolvida para melhor definir as diversas características do limbo. Assim, eles podem ser classificados quanto a forma, margem, base, ápice, ausência ou presença de tricomas e consistência [12].

2.6 Considerações Finais deste Capítulo

Neste capítulo foram apresentadas as principais definições, características e fases de um sistema de reconhecimento de padrões, com ênfase nos classificadores clássicos (estatístico e sintático).

Além disso, este capítulo apresentou o padrão folha que é utilizado para reconhecer as plantas neste trabalho.

O próximo capítulo apresenta a teoria de algoritmos genéticos, apresentando seu histórico, conceitos e fundamentos, bem como as principais metodologias existentes em um sistema que utiliza essa técnica.

Capítulo 3

Algoritmos Genéticos

3.1 Introdução

Os algoritmos genéticos (AGs) foram criados em 1975 por John Holland [25]. AGs são métodos computacionais de busca, otimização e aprendizado da máquina.

Para a criação dos AGs Holland se inspirou nos mecanismos de evolução natural propostos por Charles Darwin em 1859. Em seu trabalho, “A origem das espécies” [10], Darwin estabeleceu que no processo biológico depois de muitas gerações a população natural evolui de acordo com os princípios da seleção natural.

Cada indivíduo na natureza possui seu próprio código genético que representa determinadas características, tais como: a cor dos olhos e a altura. Quando dois indivíduos se acasalam, eles combinam seus códigos genéticos por um processo de cruzamento, originando descendentes com novos códigos genéticos, que podem ser tanto mais quanto menos adaptados ao meio do que seus pais.

A teoria da evolução estabelece que se um indivíduo é menos adaptado ao meio ele tem poucas chances de transmitir seu código genético para as próximas gerações uma vez que dificilmente ele consegue acasalar-se. Assim, depois de muitas gerações, somente as características genéticas dos indivíduos mais adaptados continuam a existir.

Eventualmente dentro de uma população podem ocorrer mutações. Essas mutações causam alterações no código genético dos indivíduos e produzem novas características que

são improváveis de aparecer pela simples recombinação dos genes existentes. Se essas novas características são ruins, isto é, não melhoram a adaptação do indivíduo, elas são extintas da população. Por outro lado, se elas melhoram a adaptação do indivíduo, elas são transmitidas às futuras gerações e continuam existindo [7].

Os mecanismos de cruzamento, mutação e seleção natural são alguns dos conceitos que os algoritmos genéticos tentam reproduzir da teoria da evolução. Para Holland [25], esses conceitos poderiam ser implementados em máquinas desde que um modelo matemático que o suportasse fosse desenvolvido.

Este capítulo descreve em detalhes os principais componentes para o desenvolvimento de um sistema baseado em algoritmos genéticos.

3.2 Componentes do AG

Genericamente, todo sistema que utiliza algoritmo genético segue o fluxograma apresentado mostrado na Figura 3.1.

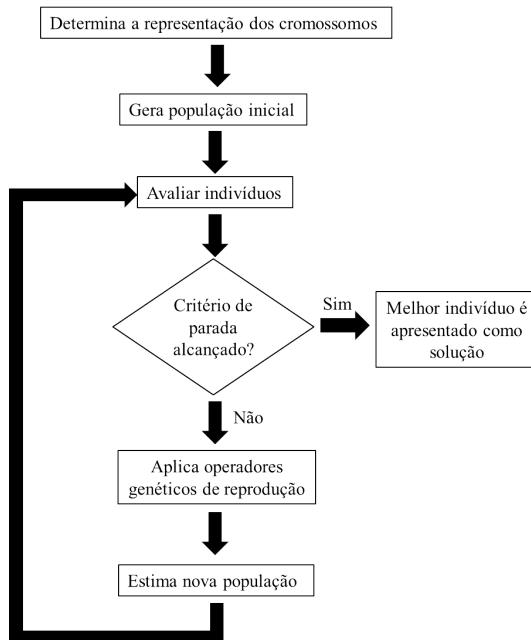


Figura 3.1: *Ciclo de execução básico dos AGs. Adaptado de [7].*

A análise do fluxograma da Figura 3.1 começa com a definição de uma representação adequada (*string* de caracteres, números inteiros, etc.) para as possíveis soluções. Em seguida, é gerada uma população inicial de forma aleatória contendo n indivíduos.

Cada indivíduo dessa população é descrito por um cromossomo que é constituído de um código que representa uma possível solução para o problema. Todos os indivíduos gerados devem ser avaliados e um valor de aptidão é associado a cada um deles de forma a representar o quanto cada solução é adequada ao problema em questão ou em outras palavras, o quanto os indivíduos estão adaptados ao meio. Caso algum desses indivíduos atinja um nível satisfatório preestabelecido de aptidão o algoritmo é interrompido e o melhor indivíduo é eleito como sendo a solução para o problema. Caso contrário, integrantes da população são selecionados para a realização das operações genéticas de cruzamento e mutação, concebendo novos indivíduos.

Todo o processo do AG descrito nessa seção é repetido até que algum critério de parada seja satisfeito. Exemplos de critério de parada são: o número de gerações, o tempo de execução ou um valor aceitável de aptidão.

3.2.1 Cromossomos

Basicamente, todo indivíduo seja ele animal, vegetal ou mesmo organismos inferiores como vírus e bactérias são formados por uma ou mais células e dentro de cada célula existe uma cadeia de cromossomos que descrevem todas as características das espécies.

Os cromossomos sempre aparecem em pares e o número de pares varia de espécie para espécie. Os seres humanos, por exemplo, têm 23 pares de cromossomos por célula, os burros têm 31 pares e as carpas, 52.

Um cromossomo é constituído por genes, que são blocos de sequências de ácido desoxirribonucleico (DNA) com uma posição bem definida chamada *locus*. Esses genes controlam uma característica hereditária específica, como a cor do cabelo, a altura, a cor dos olhos entre outras [31].

Em um AG os cromossomos são as representações matemáticas finitas das possíveis soluções. O conjunto de cromossomos de uma mesma geração é chamado de população. A população é a região que comprehende as soluções possíveis ou viáveis do problema.

As primeiras representações dos cromossomos utilizadas em algoritmos genéticos comprehendiam o alfabeto binário $\{0,1\}$, contudo, as representações baseadas em caracte-

res, números inteiros ou reais também estão sendo utilizadas amplamente.

Apesar de muitos trabalhos terem sido desenvolvidos nesse campo de estudo, não existe uma fórmula para uma boa codificação ou representação de um cromossomo. Para que bons resultados práticos sejam obtidos os AGs devem ser desenvolvidos sob medida para o problema em questão e experimentos devem ser conduzidos de forma a obter o melhor desempenho. É importante apenas que algumas regras gerais sejam seguidas [31]:

1. a representação deve ser a mais simples possível;
2. se ocorrerem soluções proibidas ao problema, elas não devem ter uma representação; e
3. se o problema impuser condições de algum tipo, estas devem estar implícitas dentro das representações.

Portanto, a melhor estratégia a ser empregada para encontrar uma codificação cromossômica adequada é a utilização de conhecimentos específicos do domínio do problema em questão.

3.2.2 População

A população é o conjunto de todos os indivíduos de uma mesma geração. A formação de uma população inicial é muito importante para a execução do algoritmo, pois é a partir dela que todo o processo acontece.

Uma população inicial considerada boa deve abranger todo o espaço de busca. Quanto mais “espalhados” os indivíduos da população inicial estiverem em relação ao espaço de busca, melhores são as possibilidades de que o algoritmo genético evolua para uma solução considerada ótima. Caso os cromossomos sejam posicionados muito próximos entre si, geralmente observa-se uma evolução à pontos ótimos locais, causando uma repentina cessão na evolução da aptidão de forma que os indivíduos tendem a se tornarem idênticos após as consecutivas gerações.

3.2.3 Função de aptidão

A função de aptidão, também chamada de função de custo, é a maneira utilizada pelos AGs para determinar a qualidade de um individuo como solução do problema. Ela calcula um valor numérico que reflete o quanto bons são os parâmetros representados no cromossomo. Pode-se entender mais facilmente esse conceito analisando a função de aptidão como sendo uma nota atribuída ao cromossomo na resolução do problema. Esta nota é utilizada para diferenciar as boas das más soluções.

No cálculo da função de aptidão são utilizados todos os valores armazenados no cromossomo (os parâmetros) e um valor numérico é retornado. Esse valor representa a qualidade da solução obtida usando aqueles parâmetros. Como os AGs são técnicas de maximização, a função de avaliação deve ser tal que se o cromossomo c_1 representa uma solução melhor do que o cromossomo c_2 , então a avaliação de c_1 deve ser maior do que a avaliação de c_2 [31].

Em muitos casos, a função de aptidão é a única ligação verdadeira do programa com o problema real. Para certos tipos de problemas, como por exemplo, a maximização de uma função, o valor de aptidão é o próprio cálculo da função para um determinado parâmetro. Isto leva à conclusão de que um mesmo AG pode ser utilizado para descobrir o máximo de toda e qualquer função de n variáveis sem nenhuma alteração das estruturas de dados e procedimentos adotados, alterando-se apenas a função de avaliação (que neste caso específico seria exatamente a função a ser maximizada).

Portanto, a função de aptidão deve ser escolhida com bastante cuidado. Ela deve considerar todo o conhecimento do problema a ser resolvido, abrangendo tanto as suas restrições quanto seus objetivos de qualidade.

Segundo Michalewicz [34], alguns fatores devem ser considerados na elaboração de uma função de aptidão:

- Determinar se o problema é relacionado à maximização ou a minimização;
- Definir se os ambientes são determinísticos ou não;
- A possibilidade da função de aptidão ser alterada dinamicamente durante a execução

do algoritmo;

- A avaliação da aptidão deve consumir pouco tempo de processamento de forma a viabilizar a simplificação da medida de aptidão em alguns casos;
- A função de aptidão deve alocar diferentes valores de aptidão para os indivíduos distintos, permitindo uma melhor eficiência na execução dos operadores de seleção;
- Deve contemplar e considerar as restrições do problema ao máximo; e
- Pode incorporar múltiplos objetos.

Geralmente o maior custo computacional de um algoritmo genético ocorre a partir do cálculo da função de aptidão, uma vez que esta avalia todos os indivíduos geração a geração. Com a finalidade de minimizar esta carga computacional, alguns cuidados especiais foram propostos por Haupt *et al.* [21] como:

- Não gerar indivíduos idênticos na população inicial;
- Garantir que a reprodução não gere indivíduos idênticos; e
- Criar uma memória para os AGs desconsiderando os indivíduos gerados anteriormente e considerados inaptos ou “menos aptos”.

Na prática, observa-se que desses critérios, apenas o primeiro é levado em conta na elaboração dos algoritmos genéticos.

3.2.4 Seleção de pais

O método de seleção de pais deve simular o mecanismo de seleção natural que atua sobre as espécies biológicas, em que os pais mais capazes geram mais filhos quando comparados com os pais menos aptos.

Consequentemente, no AG deve-se privilegiar os indivíduos com função de aptidão alta, sem desprezar completamente aqueles indivíduos com função de aptidão extremamente baixa, pois, os indivíduos com péssima avaliação podem possuir características

genéticas únicas, que não estão presentes em nenhum outro cromossomo da população, mas que quando cruzado com outro indivíduo geram um filho que pode ser a melhor solução para o problema em estudo. Por outro lado, se apenas os melhores indivíduos reproduzirem, a população tende a ser composta de indivíduos cada vez mais semelhantes e nas futuras gerações vai faltar diversidade genética a população para que a evolução possa prosseguir de forma satisfatória.

Existem várias formas de realizar a seleção de pais. Alguns operadores de seleção são apresentados a seguir.

Seleção estocástica com reposição

O método de seleção estocástica com reposição, proposto inicialmente por Holland [25], também conhecido como método de seleção por roleta é o mais utilizado em implementações de algoritmos genéticos. A cada indivíduo da população corrente é atribuído uma fatia de uma roleta imaginária, cujo tamanho dessa fatia é proporcional à aptidão do indivíduo, como ilustrado na Figura 3.2.

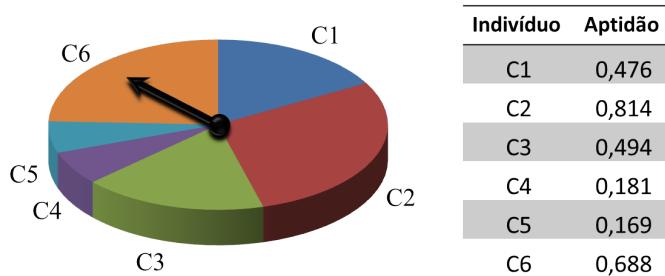


Figura 3.2: Exemplo de uma roleta de seleção.

Para selecionar um indivíduo é simulado um giro de um ponteiro virtual sobre a roleta. A posição em que o ponteiro parar indica qual indivíduo foi selecionado. A probabilidade p_j de qualquer indivíduo ser selecionado é calculada pela Equação (3.1).

$$p_j = \frac{f_j}{\sum_{k=1}^{TamanhoPop} f_k} \quad (3.1)$$

onde:

- f_j - aptidão do cromossomo j na população corrente;

- $TamanhoPop$ - número de indivíduos na população; e
- f_k - aptidão do k -ésimo indivíduo da população.

Seleção por torneio simples

Na seleção por torneio simples ocorre a promoção de um torneio entre um grupo de N indivíduos tomados aleatoriamente na população. O indivíduo de maior aptidão do grupo é selecionado, enquanto os demais são desconsiderados. Nesse método a pressão seletiva pode ser controlada pelo tamanho dos grupos do torneio.

Seleção por torneio estocástico

A seleção por torneio estocástico é similar ao torneio simples se diferenciando apenas no modo de como os indivíduos dos grupos são selecionados. Por esse método a escolha dos indivíduos é realizada pelo método da roleta ao invés de serem tomados da população corrente aleatoriamente.

Seleção por truncamento

Na seleção por truncamento é construído um subconjunto contendo os melhores indivíduos da população. E a escolha dos pais que irão participar do cruzamento é realizada de forma aleatória, de modo que a probabilidade de escolha de qualquer indivíduo seja a mesma.

Blickle [6] mostrou que este método é o que causa maior perda de diversidade e por isto é o que tende a fazer com que o AG apresente o pior desempenho.

Seleção por ordenação

A seleção por ordenação ou seleção por *ranking*, descrita por Mitchell [36], é um método de seleção que evita a convergência prematura e a dominância de um superindivíduo. O princípio dessa seleção consiste na substituição da função de aptidão por um *ranking*.

Ao utilizar o *ranking* em vez da função de aptidão a pressão seletiva sobre a população se mantém no mesmo nível, desde a primeira até a última geração, não importando

o grau de convergência genética ocorrido na população durante o processamento do AG.

Para aplicar este método primeiro é necessário ordenar os indivíduos para depois estabelecer o *ranking*. Este passo é extremamente oneroso em termos de custo computacional com uma complexidade de $O(n \log n)$ operações. Após estabelecer o *ranking* é realizado o seu mapeamento para uma nova função de aptidão.

Uma maneira de realizar esse mapeamento é utilizar o método linear [31], no qual cada indivíduo recebe uma avaliação que é calculada pela Equação (3.2).

$$E(i, t) = Mn + (Mx - Mn) \cdot \frac{Rank(i, t) - 1}{N - 1} \quad (3.2)$$

onde

- Mn - valor da avaliação que é atribuída ao pior indivíduo colocado no *ranking*;
- Mx - valor da avaliação que é atribuído ao melhor indivíduo colocado no *ranking*;
- N - número de indivíduos na população; e
- $Rank(i, t)$ - *ranking* do indivíduo i na população mantida pelo AG na geração t .

Uma vez definidos os novos valores de avaliação destes indivíduos um método tradicional, tal como o da roleta, pode ser utilizado para a escolha dos pais que serão submetidos aos operadores genéticos.

Pode-se verificar que com o método linear a avaliação do indivíduo de *ranking* $N/2$ localizado exatamente no meio do caminho entre a avaliação do melhor e do pior indivíduos é igual à média das avaliações, o que lhe garante ao menos uma chance de ser selecionado a cada geração.

O problema dos métodos que reduzem a pressão seletiva sobre a população é que eles tornam o tempo de convergência do AG um pouco maior quando comparado com os demais métodos. Entretanto, a manutenção da diversidade na população garante que o AG varra um pedaço maior do espaço de soluções, ficando menos suscetível à captura de máximos locais [31].

3.2.5 Elitismo

O elitismo é utilizado para aumentar a convergência do AG na direção da solução ótima. Este recurso consiste em preservar os n melhores indivíduos de cada geração. Estes bons indivíduos poderiam ser perdidos se eles não fossem selecionados para a reprodução ou se eles fossem destruídos pelos operadores de cruzamento e mutação. Portanto, os n melhores indivíduos são conduzidos diretamente da população atual para a população da próxima geração [36].

3.2.6 Reprodução

A reprodução no AG também chamada de cruzamento ou *crossover* é um processo inspirado na recombinação biológica dos genes, ou seja, na troca do material genético entre os pais para a geração dos filhos.

Após os pais serem escolhidos a partir de um dos métodos de seleção apresentados na subseção 3.2.4 deste capítulo, os indivíduos são organizados em pares para que o cruzamento seja efetivado originando os filhos. Para cada par de pais selecionados para o *crossover* é originado um par de filhos.

A frequência de recombinações é controlada pela probabilidade de cruzamento ou taxa de *crossover*, p_c , que representa o percentual da população que é submetida ao processo de reprodução a cada geração.

O modo que a operação de reprodução é realizada depende do domínio e das restrições do problema em questão. Alguns dos tipos mais conhecidos de cruzamento são: o simples, o duplo e o uniforme.

Para alguns domínios de problema cujo processo de reprodução não pode gerar genes repetidos, como é o caso do problema do caixeiro viajante, em que são conhecidas as cidades a serem visitadas por ele, busca-se um trajeto que passe por todas elas uma única vez de maneira a formar o menor percurso possível.

Desse modo, uma possível representação cromossômica para esse problema é construir um cromossomo de tamanho igual ao número de cidades a serem visitadas, contendo algarismos representativos para cada uma das cidades como conteúdo genético, onde a

ordem no cromossomo representa a ordem de visitação.

Como a solução do problema requer uma única visita a cada cidade, necessita-se de uma operação de cruzamento que não gere genes repetidos. E para isso é encontrado na literatura operadores de *crossover* que consideram alguns desses requisitos de cruzamento. Dois dos mais utilizados são o cruzamento PMX e o cruzamento cíclico.

Crossover simples

No *crossover* simples um ponto no cromossomo de cada um dos pais é selecionado aleatoriamente para marcar a posição onde ambos são cortados. Desse modo, a primeira parte do pai 1 é combinada com a segunda parte do pai 2, formando um dos filhos. O mesmo ocorre na formação do segundo filho, que recebe a primeira parte do pai 2 combinada com a segunda parte do pai 1. A Figura 3.3 apresenta um exemplo dessa operação genética.

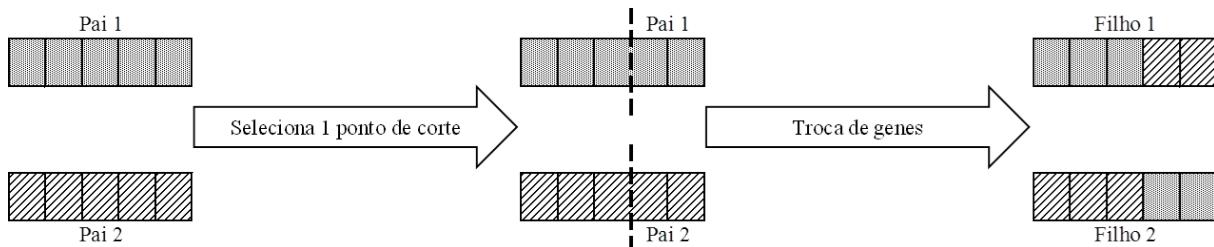


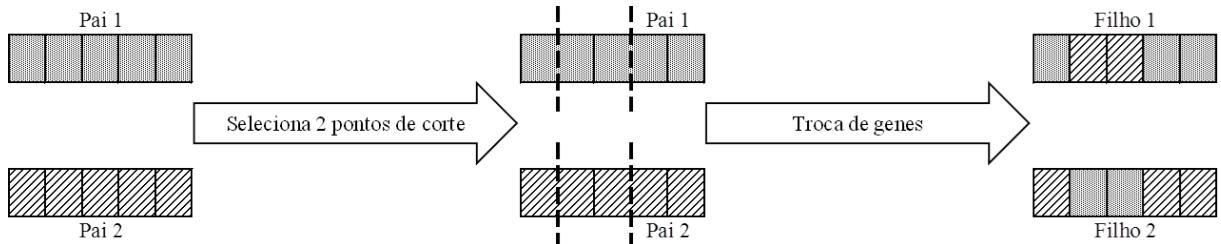
Figura 3.3: *Crossover simples*. Adptado de [31].

Crossover duplo

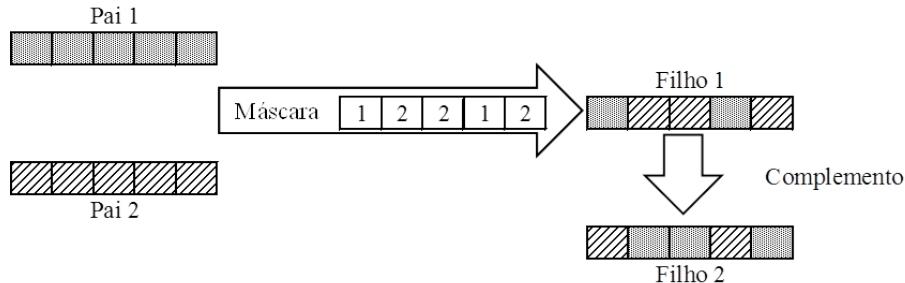
Similar ao *crossover* simples este cruzamento escolhe dois pontos de corte no cromossomo de cada pai. Assim, o primeiro filho é formado pela parte do primeiro pai fora dos pontos de corte e pela parte do segundo pai entre os pontos de corte. E o segundo filho é formado pelas partes restantes. A Figura 3.4 ilustra o funcionamento deste cruzamento.

Crossover uniforme

Esse operador genético de reprodução é caracterizado por realizar o cruzamento a partir de uma máscara do mesmo tamanho do cromossomo contendo informações de qual

Figura 3.4: *Crossover duplo* [31].

pai fornece cada gene ao primeiro filho. O segundo filho é gerado pelo complemento dessa máscara. A Figura 3.5 ilustra um cruzamento uniforme.

Figura 3.5: *Crossover uniforme*. Adaptado de [31].

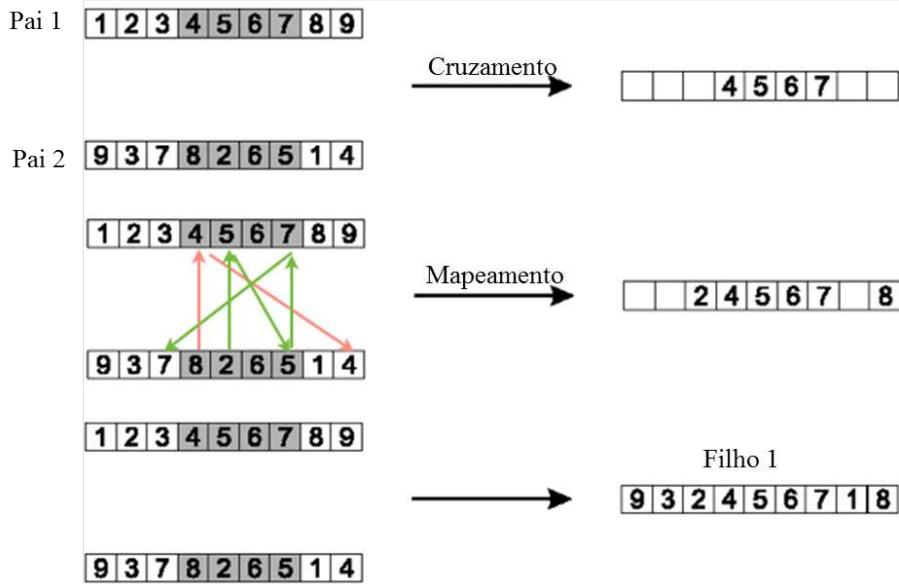
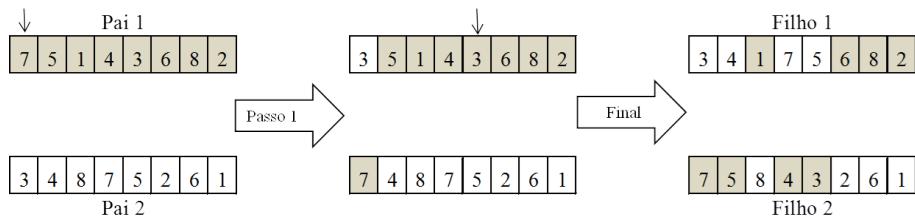
Crossover PMX

Conhecido também como cruzamento parcial o *crossover Partially Matched Crossover* (PMX) realiza uma seleção aleatória de dois pontos de cruzamento e promove o intercâmbio genético entre os pais nesses pontos, tal como acontece no cruzamento duplo.

Em seguida, é realizada uma análise nos genes externos aos pontos de corte dos cromossomos dos filhos gerados para verificar se existem repetições de genes. Se ocorrerem repetições é realizado um mapeamento para descobrir qual é seu correspondente para que a sua substituição seja efetuada. A Figura 3.6 ilustra o funcionamento desse operador.

Crossover cíclico

O cruzamento cíclico é realizado a partir de uma única posição de forma que os genes são trocados entre os parceiros até formar um ciclo. Os demais genes fora do ciclo não são trocados. A Figura 3.7 exemplifica essa operação. Começando da primeira posição de cromossomos, troca-se 7 por 3, 3 por 5, 5 por 4 e 4 por 7, completando o ciclo 7-3-5-4-7.

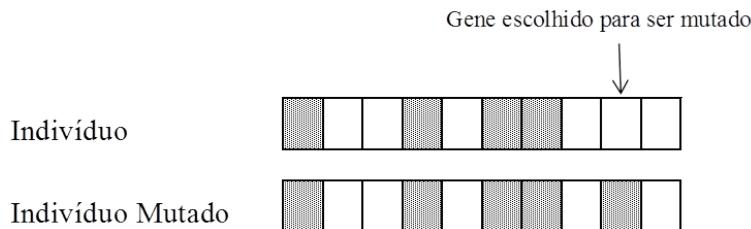
Figura 3.6: *Crossover PMX [1]*.Figura 3.7: *Crossover cíclico [4]*.

3.2.7 Mutação

A mutação é uma operação que simplesmente modifica aleatoriamente alguma característica genética de um indivíduo. Esse operador é importante pois permite a criação de novas características que não existem ou que aparecem de forma irrelevante na população.

A mutação introduz diversidade genética à população de forma a assegurar que a probabilidade de atingir qualquer ponto do espaço de busca não seja nula. Contudo, sua utilização demasiada pode transfigurar as características correntes dos indivíduos na geração, tornando o AG puramente probabilístico. Por isso, geralmente, o operador de mutação é aplicado aos indivíduos com uma taxa de mutação (p_m) pequena.

A mutação bem como a reprodução devem ser realizadas de forma a não desrespeitar as restrições do problema. A Figura 3.8 ilustra o funcionamento da mutação simples, no qual, um gene escolhido aleatoriamente tem seu valor invertido.

Figura 3.8: *Mutação simples.* Adaptado de [4]

3.3 Parâmetros de Controle

O funcionamento dos algoritmos genéticos é baseado em uma série de parâmetros que exercem grande influência sobre seu comportamento e seu desempenho [4]. Esses parâmetros são:

- tamanho da população (T_p);
- método de seleção;
- probabilidade de cruzamento;
- probabilidade de mutação;
- elitismo; e
- critério de parada.

A intuição e o bom senso são utilizados normalmente na escolha dos valores desses parâmetros. Alguns autores aconselham que estes valores devem ser determinados com base em experimentações enquanto outros, motivados pela análise dos problemas de convergência, acreditam que a variação dinâmica desses parâmetros faz com que os AGs apresentem melhor desempenho. De toda maneira é de comum acordo que:

- uma população muito pequena implica em pouca cobertura do espaço de busca e consequentemente maiores probabilidades de convergência prematura. Em contrapartida, uma população muito grande possibilita uma ampla cobertura do espaço de busca causando elevada carga computacional e possível lentidão;

- quanto maior é a probabilidade de cruzamento mais rapidamente novas estruturas são introduzidas na população. Dessa forma cromossomos com boas aptidões podem ser deformados ou até mesmo perdidos mais rapidamente do que a capacidade do processo de seleção tem de mantê-los. Para combater essa anomalia o operador elitismo geralmente é inserido às aplicações como uma forma de garantir que os melhores indivíduos não sejam perdidos pelas operações de *crossover*; e
- mutações previnem que a busca não fique parada em regiões pouco promissoras do espaço (máximos ou mínimos locais), possibilitando que qualquer ponto do espaço tenha uma probabilidade não nula de ser atingido. Por outro lado, uma taxa muito alta de mutação torna o AG uma busca essencialmente aleatória.

3.4 Execução de um AG Genérico

O pseudo-código 3.4.1 contém os principais passos para a implementação computacional dos algoritmos genéticos. Primeiramente, na linha 3, uma população contendo “*TamanhoPop*” indivíduos é gerada aleatoriamente e armazenada na variável *Pop*. Em seguida, a função de aptidão da linha 5, “*CalculaAptidao*”, avalia todos os indivíduos gerados para se conhecer o quanto cada cromossomo está adaptado ao meio.

As linhas 6 a 19 mostram os passos para a criação da população da próxima geração. O procedimento é iniciado com a implementação do “*Elitismo*” utilizado para aumentar a convergência do AG, em que os n melhores indivíduos da população atual são conduzidos diretamente para a população da próxima geração.

Em seguida, nas linhas 9 e 10 são selecionados os indivíduos que irão participar do processo de cruzamento. A seleção dos indivíduos pode ser efetuada por qualquer um dos processos apresentados na seção 3.2.4 deste capítulo.

Uma vez que dois indivíduos são selecionados, existe uma probabilidade “*ProbCruzamento*” previamente definida de seus genes serem recombinados por algum operador de cruzamento apresentado na seção 3.2.6 deste capítulo.

Para que o cruzamento aconteça com a probabilidade desejada o seguinte procedi-

mento é executado: é gerado um número aleatório “*NoAleatorio1*” entre 0 e 1. Se esse número é menor do que “*ProbCruzamento*”, o cruzamento é realizado e os dois novos indivíduos criados são adicionados à nova população; caso contrário, os dois indivíduos selecionados vão diretamente para a nova população sem nenhuma modificação (linhas 11 e 12).

O último operador genético é a mutação que é aplicada com uma pequena probabilidade “*ProbMutacao*” (linhas 15 a 17). Geralmente, esse operador é introduzido depois do cruzamento sobre cada um dos descendentes. Ele atua alterando o valor de um ou mais genes do cromossomo que são escolhidos aleatoriamente.

Quando a nova população está completa todo o processo descrito é repetido até que algum critério de parada seja satisfeito.

```

1  // Procedimento AG
2
3  Pop = GeraPopulacaoInicial(TamanhoPop)
4  REPETIR
5      CalculaAptidao(Pop);
6      Elitismo(n, Pop, NovaPop);
7      i = n+1;
8      REPETIR
9          Ind1 = Selecao(Pop);
10         Ind2 = Selecao(Pop);
11         NovaPop(i) = Ind1;
12         NovaPop(i+1) = Ind2;
13         Se NoAleatorio1 < ProbCruzamento
14             [NovaPop(i),NovaPop(i+1)] = Cruzamento(Ind1,Ind2);
15         Se NoAleatorio2 < ProbMutacao
16             NovaPop(i) = Mutacao(NovaPop(i));
17             NovaPop(i+1) = Mutacao(NovaPop(i+1));
18             i = i+2;
19         ATÉ que NovaPop esteja completa
20         Pop = NovaPop;
21     ATÉ que o critério de parada seja satisfeito

```

Código 3.4.1: *Procedimento do algoritmo genético [7]*.

3.5 Considerações Finais deste Capítulo

Além de introduzir os conceitos e fundamentos dos AGs este capítulo descreveu em detalhes os principais componentes para o seu desenvolvimento.

O próximo capítulo apresenta a teoria e as aplicações das rede artificiais neurais.

Capítulo 4

Redes Neurais Artificiais

4.1 Introdução

Redes neurais artificiais (RNAs) são sistemas paralelos e distribuídos compostos por unidades de processamento simples (nodos) que calculam determinadas funções matemáticas (normalmente não-lineares). Essas unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões. Na maioria dos modelos estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. O funcionamento dessas redes é inspirado no cérebro humano [14].

O cérebro humano contém em torno de 100 bilhões de neurônios. Os neurônios são células altamente especializadas que se comunicam com milhares de outros neurônios continuamente e em paralelo por meio dos impulsos nervosos. As principais partes de um neurônio são: o corpo da célula, os dendritos e o axônio.

O corpo celular contém um núcleo grande e arredondado de onde originam dois prolongamentos: os dendritos (do grego *déndron*: árvore) que são as ramificações que têm a função de captar estímulos, e os axônio (do grego *áxon*: eixo) o maior dos prolongamentos que tem a função de transmitir os impulsos nervosos para as outras células. A Figura 4.1 ilustra de forma simplificada os componentes do neurônio.

Quando um neurônio biológico recebe os sinais de entrada por meio dos dendritos ele os processa no corpo celular e gera os sinais de saída que são transmitidos para outros neurônios por meio dos axônios. A Figura 4.2 mostra a propagação da sinapse dentro de uma rede de neurônios.

O ponto de contato entre a terminação axônica de um neurônio e o dendrito de outro

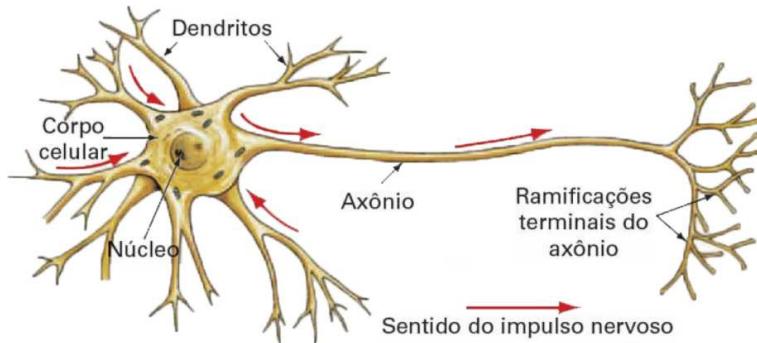


Figura 4.1: *Neurônio biológico* [54].

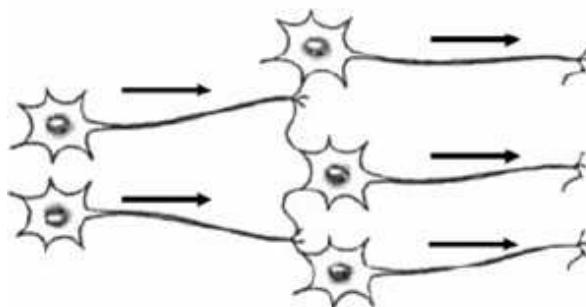


Figura 4.2: *Propagação da sinapse dentro da rede de neurônios* [5].

é chamado de sinapse. É pelas sinapses que os nodos se unem funcionalmente formando as redes neurais. As sinapses funcionam como válvulas controlando a transmissão dos impulsos ao longo da rede de neurônios.

A comunicação entre os neurônios pode ser inibida ou potencializada pela concentração de neurotransmissores.

Neurotransmissores são substâncias químicas produzidas pelos neurônios. Eles podem estimular a continuidade de um impulso ou efetuar a reação final no órgão ou músculo alvo.

A estrutura individual dos nodos, a topologia das conexões e as sinapses formam a base do estudo das RNAs.

Várias aplicações destes sistemas são relatadas na literatura, podendo, dependendo dos objetivos a que se destinam, serem summarizados em seis categorias [22]: associação e reconhecimento de padrões, aproximação de funções, controle, filtragem e previsão não-linear. Para essas aplicações diferentes topologias podem ser utilizadas, como por exemplo:

- as redes *feedforward* possuindo uma ou mais camadas de processadores, cujo fluxo

de dados é sempre em uma única direção, isto é, não existe realimentação; ou,

- as redes recorrentes, que são redes com conexões entre os processadores da mesma camada e/ou com os processadores das camadas anteriores (realimentação).

Além disso, diferentes tipos de aprendizado são relatados, sendo usualmente classificados em três paradigmas principais: aprendizados supervisionados (perceptron, perceptron multi-camadas e *radial basis function*), aprendizado não supervisionado (redes neurais alto-organizáveis, teoria da ressonância adaptativa e *Hopfield*) e aprendizado por reforço (programação dinâmica).

Este capítulo apresenta a teoria relacionada às redes neurais artificiais, bem como seu histórico, etapas de funcionamento e o algoritmo de treinamento utilizado neste trabalho (algoritmo *backpropagation*).

4.2 Breve Histórico das Redes Neurais

O primeiro modelo artificial de um neurônio biológico foi proposto em 1943 por McCulloch & Pitts [33]. Ele era uma simplificação do que se conhecia até então a respeito do neurônio biológico. Sua descrição matemática resultou em um modelo com n terminais de entrada x_1, x_2, \dots, x_n que representam os dendritos e apenas um terminal de saída y que representa o axônio.

Nesse modelo para simular o comportamento das sinapses os terminais de entrada do neurônio possuem pesos w_1, w_2, \dots, w_n cujos valores podem ser positivos ou negativos, dependendo das sinapses correspondentes serem inibitórias ou excitatórias. Os pesos determinam em que grau o neurônio deve considerar os sinais de disparo que ocorrem naquela conexão. A operação $x_i w_i$ é o resultado de uma sinapse i no neurônio pós-sináptico. A Figura 4.3 ilustra uma descrição deste modelo.

Em 1969, Marvin Minsky e Seymour Papert [35] analisaram matematicamente o perceptron e demonstraram que as redes de uma camada não são capazes de solucionar problemas simples como a operação lógica XOR (ou exclusivo). Como eles não acreditavam na possibilidade de construir um método de treinamento de redes com mais de uma camada, eles concluíram que as redes neurais seriam sempre suscetíveis a essa limitação.

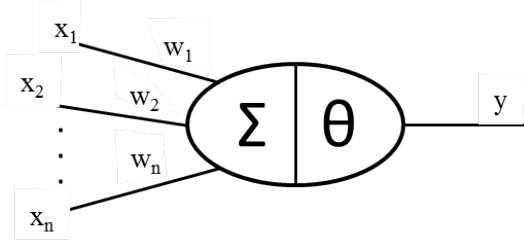


Figura 4.3: *Neurônio proposto por McCulloch & Pitts* [14].

Este trabalho desinteressou os pesquisadores e quase provocou a extinção desse ramo de pesquisa em redes neurais.

Somente no início da década de 80 após a obtenção de alguns importantes resultados teóricos destacando, dentre eles, a descoberta da retropropagação do erro, a implementação de redes multi-camadas e o aumento da capacidade de processamento dos computadores que ressurgiu o interesse pelas redes neurais.

4.3 Rede Perceptron Multi-Camadas

O trabalho original de McCulloch e Pitts [33] modelou um neurônio biológico e sua capacidade computacional com a apresentação de vários exemplos de topologias de rede com a capacidade de cálculo de funções booleanas. Esse modelo de neurônio em conjunto com o surgimento do computador digital criou na época a expectativa de que era possível construir uma máquina inteligente por meio de operadores lógicos básicos. Somente com a divulgação do perceptron de Frank Rosenblatt em 1957 [44] que o conceito de aprendizado foi introduzido nas RNAs.

A rede neural perceptron multi-camadas, do inglês *MultiLayer Perceptron* (MLP), representa uma generalização do perceptron proposto por Rosemblatt e é uma das mais conhecidas e aplicadas arquiteturas de redes neurais.

A arquitetura MLP consiste de uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída. Os neurônios pertencentes à camada de entrada são denominados unidades de entrada e são ligados à camada intermediária por meio de conexões. A camada intermediária, cujos neurônios são chamados unidades ocultas, transmite as informações recebidas ou para as demais camadas intermediárias, se elas existirem, ou para a camada de saída. Finalmente, a camada de saída, cujos neurônios são

denominados unidades de saída, fornece a resposta da rede neural para a entrada aplicada.

A Figura 4.4 mostra a arquitetura de uma MLP com duas camadas intermediárias.

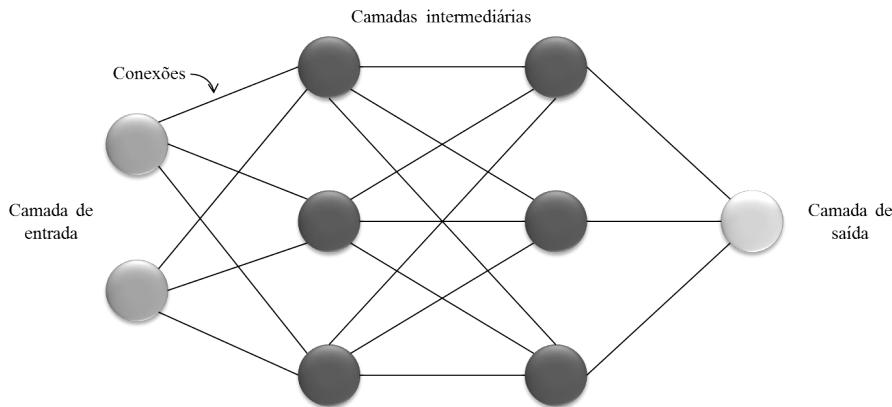


Figura 4.4: *Exemplo de arquitetura perceptron multi-camadas com duas camadas intermediárias [14].*

Em uma rede MLP a comunicação entre os neurônios é realizada por meio das arestas e a cada aresta está associado um peso. Porém, a comunicação ocorre somente entre as unidades das camadas distintas, portanto, não existe comunicação entre as unidades de uma mesma camada.

Cada neurônio da rede tem uma função diferenciável conhecida como função de ativação. Alguns exemplos de função de ativação utilizados na prática são: função linear, função sigmóide, função logarítmica, função tangente hiperbólica, etc.

Assim, a arquitetura da rede neural MLP apresenta três características que podem variar de projeto para projeto. Essas características são: variações quanto ao número de camadas e neurônios intermediários da rede, tipo de conexão determinada pelas sinapses e função de ativação.

O treinamento de uma rede MLP é do tipo supervisionado, ou seja, é necessário um “professor” para indicar a resposta desejada para um padrão de entrada apresentado à rede durante a fase de aprendizagem. Um erro é calculado pela diferença entre as respostas desejada e calculada. Os parâmetros da rede (pesos e limiares) são ajustados de acordo com esse erro. Assim, a rede é adaptada ou treinada por um processo iterativo de ajustes sucessivos dos pesos.

O algoritmo de aprendizado mais conhecido para o treinamento das redes MLP é o *backpropagation*. Este algoritmo é baseado na regra delta proposta por Widrow & Hoff

[55]. Esta regra estabelece uma forma de definir o erro na saída da rede possibilitando o ajuste de seus pesos.

O treinamento da MLP utilizando o algoritmo *backpropagation* é realizado em duas fases onde cada uma delas percorre a rede em um sentido. Estas fases são denominadas *forward* e *backward*. A *forward* é usada para definir a saída da rede para um determinado padrão de entrada. A *backward* utiliza as saídas desejada e calculada pela rede para atualizar os pesos de suas conexões [37]. A Figura 4.5 ilustra estas duas fases.

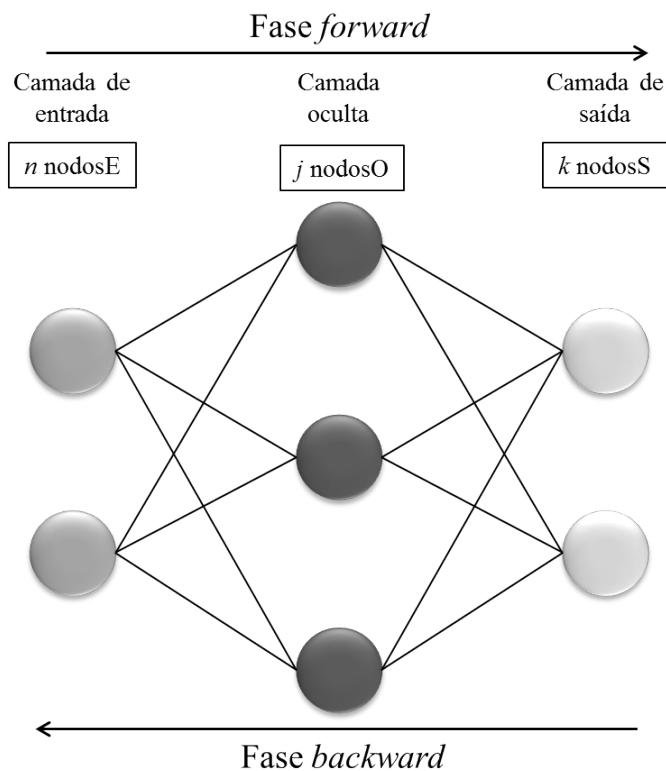


Figura 4.5: *Fluxo do processamento do algoritmo backpropagation*. Adaptado de [37].

4.3.1 Algoritmo *backpropagation*

O algoritmo de aprendizagem *backpropagation* atua alterando os pesos da rede com a finalidade de minimizar a diferença entre as saídas desejada e calculada pelo modelo. Se o erro na camada de saída não é menor do que uma determinada tolerância, então o erro é retropropagado para a rede.

O algoritmo *backpropagation* juntamente com a apresentação das equações aplicadas a uma rede MLP com apenas uma camada intermediária (arquitetura apresentada na Figura 4.5) é apresentado a seguir neste capítulo.

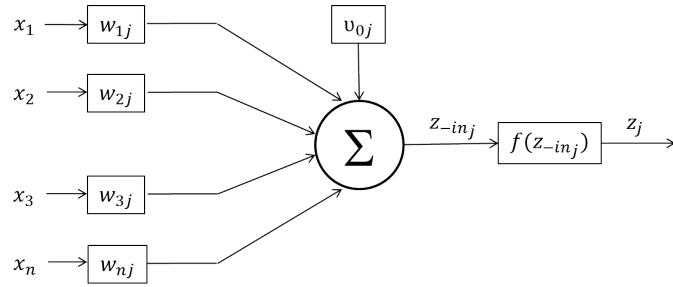


Figura 4.6: *Modelo do neurônio da camada intermediária de uma rede MLP.*

Durante a fase *forward* cada unidade de entrada recebe o sinal e o espalha para cada um dos neurônios da camada oculta (neurônio representado na Figura 4.6). Em cada neurônio dessa camada é realizada a soma do bias com cada uma das sinapses que chegam nele. A sinapse é a operação de multiplicação entre o sinal de entrada (x_i) e o peso correspondente entre as camadas de entrada e oculta (w_{ij}). E o bias (v_{0j}) é um limiar utilizado para aumentar o grau de liberdade da função de ativação melhorando a capacidade de aproximação da rede em direção à resposta. A Equação (4.1) mostra o cálculo da sinapse de entrada (z_{-in_j}) na camada oculta.

$$z_{-in_j} = v_{0j} + \sum_{i=1}^{nodosE} x_i w_{ij} \quad (4.1)$$

onde:

- $nodosE$ - número de unidades de entrada presentes na rede.

Cada neurônio da camada oculta calcula a sua função de ativação ($f(z_{-in_j})$) e envia o resultado para cada neurônio da saída, como mostrado na Equação (4.2).

$$z_j = f(z_{-in_j}) \quad (4.2)$$

O valor da sinapse de entrada dos neurônios da camada de saída (y_{-in_k}) é obtido pela Equação (4.3).

$$y_{-in_k} = \omega_{0k} + \sum_{j=1}^{nodosO} z_j w_{jk} \quad (4.3)$$

onde:

- z_j - saída da camada intermediária;

- w_{jk} - peso entre as camadas intermediária e de saída;
- ω_{0k} - bias da unidade de saída; e
- $nodosE$ - número de unidades ocultas presentes na rede.

A resposta da rede para o padrão de entrada fornecido é encontrada pela função de ativação, calculada conforme mostrado na Equação (4.4).

$$y_k = f(y_{-in_k}) \quad (4.4)$$

Com a finalidade de determinar o erro associado, na fase *backward* a unidade de saída compara o resultado da função de ativação calculado pela Equação (4.4) com o valor da resposta previamente conhecida (t_k). Baseado nesse erro, o fator δ_k é calculado pela Equação (4.5).

$$\delta_k = f(t_k - y_k) f'(y_{-in_k}) \quad (4.5)$$

onde:

- $f'(y_{-in_k})$ - derivada da função de ativação dos neurônios da camada de saída.

O fator δ_k é utilizado para distribuir o erro da unidade de saída de volta às camadas anteriores e também para atualizar os pesos entre as camadas de saída e oculta, como mostrado na Equação (4.6).

$$w_{jk}(\text{novo}) = w_{jk}(\text{antigo}) + \alpha \delta_k z_j \quad (4.6)$$

onde:

- α - taxa de aprendizagem.

De maneira similar, o fator δ_j é encontrado usando a Equação (4.7).

$$\delta_j = f'(z_{-in_j}) \sum_{k=1}^{nodosS} \delta_k w_{jk} \quad (4.7)$$

onde:

- $nodosS$ - número de unidades de saída presentes na rede.

O fator δ_j é usado para atualizar os pesos entre as camadas oculta e de entrada, como mostrado na Equação (4.8).

$$w_{ij} (\text{novo}) = w_{ij} (\text{antigo}) + \alpha \delta_j x_i \quad (4.8)$$

O processo de treinamento da rede é repetido até que algum critério de parada seja satisfeito.

A formulação completa dessas equações pode ser encontrada no Apêndice B desta dissertação.

4.3.2 Taxa de aprendizagem (α)

A taxa de aprendizagem (α) é uma constante de proporcionalidade que assume valores dentro do intervalo de 0 a 1. Essa taxa determina a natureza da adaptação dos pesos.

Um alto valor de α produz uma rápida aprendizagem resultando em grandes alterações nos pesos. Entretanto, aumenta o risco da rede se tornar instável, podendo oscilar ao redor da solução e provocar a saturação dos neurônios.

Por outro lado, se a taxa de aprendizagem assume valores pequenos, as mudanças nos pesos sinápticos são menores de uma iteração para outra e as trajetórias no espaço definido pelos pesos são mais suaves. Este benefício é alcançado com um alto custo no processo de treinamento, pois esse torna-se mais lento por requerer muito mais iterações até a convergência [38].

4.3.3 *Momentum* (η)

O principal problema de uma rede *backpropagation* é o seu elevado tempo de treinamento. Uma forma de minimizar este problema é considerar os efeitos de segunda ordem para o gradiente descendente. Outro problema indesejável nessa rede e que ocorre frequentemente é a convergência para mínimos locais. Estes mínimos são pontos na superfície de erro que apresentam uma solução estável, embora não seja a saída mais adequada para a RNA [37]. Uma forma de atenuar esses problemas é aplicando o termo *momentum* ao algoritmo do *backpropagation*.

No *backpropagation* com *momentum* a alteração dos pesos é baseada em uma combinação do gradiente descendente atual (t) com o gradiente descendente anterior ($t - 1$). Esta modificação acelera o treinamento da rede em regiões muito planas da superfície do erro, e suprime a oscilação dos pesos em vales e ravinas nas situações em que existem padrões de treinamento cujas características são muito diferentes das características da maioria dos outros padrões fazendo com que o gradiente descendente tome direções indesejáveis [16].

O termo *momentum* é adicionado à Equação (4.8), obtendo-se a Equação (4.9).

$$w_{ij}(t + 1) = w_{ij}(t) + \eta \delta_j(t) x_i(t) + \alpha(w_{ij}(t) - w_{ij}(t - 1)) \quad (4.9)$$

onde:

- η - termo *momentum* que deve assumir valores entre 0 e 1.

4.3.4 Função de ativação

A função de ativação é muito importante para o comportamento de uma RNA, porque é ela quem define a saída do neurônio artificial, e, portanto o caminho pelo qual a informação é conduzida. É desejável que a função de ativação seja contínua, diferenciável e, de preferência, não-decrescente. Com o objetivo de obter uma melhor eficiência computacional a derivada da função de ativação deve ser fácil de ser calculada [16]. A Figura 4.7 mostra as funções de ativação mais comuns.

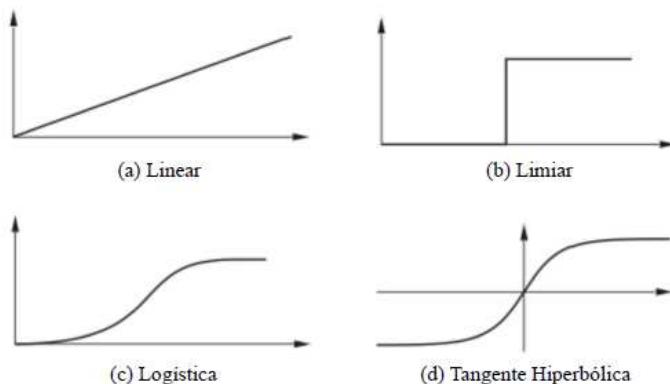


Figura 4.7: Algumas funções de ativação [13].

Uma das funções mais utilizadas na literatura é a função logística (sigmóide binária) mostrada na Figura 4.7(c). Essa função assume valores que pode variar entre 0 e 1 e

é definida pela Equação (4.10).

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (4.10)$$

E sua derivada é mostrada na Equação (4.11).

$$f'_1(x) = f_1(x) [1 - f_1(x)] \quad (4.11)$$

Hecht-Nielsen [24] demonstrou que as funções de ativação sem saturação também podem ser utilizadas no treinamento das redes MLP. Elas devem ser empregadas em aplicações em que a saturação não é benéfica, como nos casos em que a resposta da rede assume valores contínuos e não somente binários ou bipolares.

Um exemplo de função de ativação sem saturação é a função logarítmica definida pela Equação (4.12).

$$f(x) = \begin{cases} \log(1+x) \text{ para } x > 0 \\ -\log(1-x) \text{ para } x < 0 \end{cases} \quad (4.12)$$

A derivada da Equação (4.12) é mostrada na Equação (4.13).

$$f'(x) = \begin{cases} \frac{1}{1+x} \text{ para } x > 0 \\ \frac{1}{1-x} \text{ para } x < 0 \end{cases} \quad (4.13)$$

A Figura 4.8 mostra a plotagem dos gráficos da função logarítmica e de sua derivada.

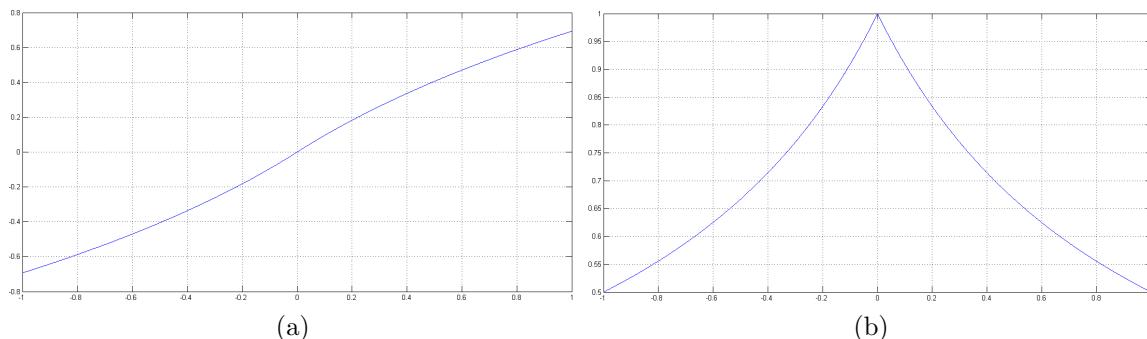


Figura 4.8: (a) Representação da função logarítmica; (b) derivada da função logarítmica.

4.3.5 Inicialização dos pesos e dos bias

A primeira etapa do treinamento de um algoritmo *backpropagation* é a inicialização dos pesos e dos bias da rede. Uma boa escolha dos parâmetros iniciais é fundamental para um bom desempenho desse algoritmo. Caso a inicialização dos pesos seja realizada inadequadamente pode ocorrer um fenômeno chamado “saturação prematura” [30].

Desse modo, os pesos iniciais e os bias não devem assumir valores que tornem as funções de ativação ou suas derivadas iguais à zero. Consequentemente, os valores iniciais não devem ser muito grandes com a finalidade de evitar que a derivada da função de ativação entre na região de saturação e assuma valor muito pequeno, e nem que os pesos iniciais sejam muito pequenos de forma que o somatório da entrada das camadas assumam valores próximos de zero promovendo um aprendizado muito lento. Portanto, a escolha correta dos pesos iniciais e dos bias contribuem para a rápida convergência da resposta da rede.

O critério de Nguyen-Widron de inicialização das variáveis [39] propõe inicializar os pesos que ligam a camada oculta à camada de saída com valores aleatórios entre $-0,5$ e $0,5$ e os pesos que ligam a camada de entrada à camada oculta com os valores que estimulam um aprendizado mais rápido dos neurônios dessa camada. Este estímulo é definido primeiramente calculando o fator de escala β como mostrado na Equação (4.14).

$$\beta = 0,7 \sqrt[3]{p} \quad (4.14)$$

onde:

- n - número de entradas; e
- p - número de neurônios da camada oculta.

Após calcular o fator de escala β cada peso que chega à camada oculta ($w_{ij}(0)$) deve ser inicializado com valores aleatórios entre $-0,5$ e $0,5$, e em seguida, os pesos dessa camada são corrigidos (inicializados), conforme mostra a Equação (4.15).

$$w_{ij} = \frac{\beta w_{ij}(0)}{|w_{ij}(0)|} \quad (4.15)$$

onde:

- $|w_{ij}(0)|$ - módulo do vetor $w_{ij}(0)$.

Os bias dos neurônios das camadas oculta (v_{0j}) e de saída (ω_{0k}) são inicializados com valores aleatórios entre $-\beta$ e β e entre $-0,5$ e $0,5$, respectivamente.

4.3.6 Camada intermediária

As unidades intermediárias de uma rede MLP funcionam como detectores de características. Elas geram uma codificação interna dos padrões de entrada, que é então utilizada para a definição da saída da rede. Devido a um número suficientemente grande de unidades intermediárias, é possível formar representações internas para qualquer conjunto de padrões de entrada.

Cybenko [9] investigou o número de camadas intermediárias necessárias para a implantação de classes e funções em uma RNA. Os resultados obtidos indicam que:

- uma camada intermediária é suficiente para aproximar qualquer função contínua; e
- duas camadas intermediárias são suficientes para aproximar qualquer função matemática.

Deve ser observado, contudo, que em alguns casos a utilização de duas ou mais camadas intermediárias pode facilitar o treinamento da rede. A utilização de um grande número de camadas intermediárias não é recomendada, no entanto, pois, cada vez que o erro medido durante o treinamento é propagado para a camada anterior, ele se torna menos útil ou preciso. A única camada que tem uma noção precisa do erro cometido pela rede é a camada de saída. A última camada intermediária recebe apenas uma estimativa sobre o erro. A penúltima camada intermediária, uma estimativa da estimativa, e assim por diante.

Com relação ao número de nodos nas camadas intermediárias, este é em geral definido empiricamente. Este número depende fortemente da distribuição dos padrões de treinamento e da validação da rede. No entanto, alguns métodos têm sido propostos. Os mais utilizados são:

- definir o número de unidades em função do número de entradas e saídas; e
- utilizar um número de conexões dez vezes menor do que o número de padrões de treinamento. Este método apenas reduz a incidência de *overfitting*. Se o número de padrões de treinamento é muito maior do que o número de conexões o *overfitting* é improvável, mas pode ocorrer *underfitting* (a rede não converge durante o seu treinamento).

O número adequado de nodos na camada intermediária depende de vários fatores, como:

- número de padrões de treinamento;
- quantidade de ruído presente nos exemplos;
- complexidade da função a ser aprendida; e
- distribuição estatística dos dados de treinamento.

Portanto, deve-se ter cuidado para não utilizar nem unidades demais, o que pode levar a rede a memorizar os padrões de treinamento, em vez de extrair as características gerais que permitirão a generalização ou o reconhecimento dos padrões não vistos durante o treinamento (*overfitting*), nem um número muito pequeno, que pode forçar a rede a gastar tempo em excesso tentando encontrar uma representação ótima (as unidades utilizadas podem ficar sobrecarregadas, tendo que lidar com um elevado número de restrições) [14].

4.3.7 Treinamento local ou em lote

O aprendizado em uma rede *backpropagation* acontece com a apresentação dos padrões pertencentes ao conjunto de treinamento repetidas vezes. Cada apresentação de todo o conjunto de treinamento é chamada de época.

O processo de aprendizagem é repetido até que os pesos e os bias estejam estáveis e o erro médio quadrático (MSE) atinja um valor mínimo aceitável preestabelecido.

Segundo Ridmiller [43] existem basicamente dois possíveis métodos para se calcular e executar a atualização dos pesos durante o processo de aprendizagem: treinamento local e treinamento em lote.

Treinamento local

No treinamento local a atualização dos pesos é realizada imediatamente após a obtenção do gradiente decrescente referente a um único padrão apresentado à rede. Assim, para um conjunto contendo N padrões de treinamento são realizadas N correções de pesos em uma mesma época.

Este tipo de método também é conhecido como aprendizagem *on-line*, ou aprendizagem padrão a padrão. Os melhores resultados são produzidos quando os conjuntos de amostras são apresentados aleatoriamente a cada época, pois a alteração da ordem com que os padrões são apresentados ajuda a evitar o *overfitting*.

Treinamento em lote

O treinamento em lote é também conhecido como treinamento *off-line* ou *batch*. Este treinamento atualiza os pesos somente depois que todas as amostras de treinamento são apresentadas à rede, ou seja, é efetuada apenas uma atualização por época.

O modo de aprendizagem local é mais eficiente quando usado em grandes conjuntos de treinamento que contenham quantidades significativas de informação redundantes. Como as amostras são apresentadas aleatoriamente, o uso desse método transforma a busca pelos pesos em uma busca estocástica. A busca estocástica reduz a possibilidade do algoritmo *backpropagation* ficar preso em um mínimo local. Por outro lado, a utilização do método de aprendizagem em lotes fornece informações mais seguras com relação ao vetor gradiente [22].

4.3.8 Critério de parada

Um método de critério do algoritmo consiste em terminar o processamento quando um determinado número de ciclos é realizado ou quando a soma dos erros quadráticos de todos os padrões apresentados é menor do que um determinado valor. O erro quadrático médio total (MSE) é definido pela Equação(4.16).

$$E = \frac{1}{2} \sum_p \sum_{k=1}^{k=nodosS} (t_k - y_k)^2 \quad (4.16)$$

onde:

- E - medida do erro total ou função de custo a ser minimizada;
- p - número de padrões apresentados;
- $nodosS$ - número de unidade de saída;
- t_k - resposta previamente conhecida; e
- y_k - resposta calculada pela rede.

Porém, o critério de parada baseado em um valor mínimo de MSE não garante a convergência do resultado, pois o algoritmo pode interromper a execução prematuramente.

A parada baseada em um número fixo de iterações, também não é recomendada porque ela não leva em consideração o estado atual do processo de aprendizagem.

Um bom critério de parada deve considerar a existência dos mínimos locais.

Um dos critérios apresentados por Haykin [23] considera essa propriedade baseada no fato de que a função custo é estacionária no ponto $w = \mathbf{w}^*$, onde \mathbf{w}^* é um vetor peso que representa um mínimo, podendo ser local ou global. Este critério de parada pode ser definido da seguinte forma: o algoritmo *backpropagation* converge se a variação do erro quadrático médio de uma época para outra assume um valor suficientemente pequeno.

Um outro critério de parada existente na literatura é o teste de generalização do algoritmo. Esse método divide o conjunto de treinamento em dois, no qual, um dos conjuntos é responsável por treinar e o outro é responsável por testar a rede após cada época. Assim, se quando testada a rede atingir um percentual de acerto preestabelecido o treinamento é encerrado.

4.4 Considerações Finais deste Capítulo

Este capítulo apresentou uma breve introdução dos conceitos da teoria de redes neurais, o modelo de rede neural perceptron multi-camadas, suas etapas de processamento e uma descrição do método de treinamento *backpropagation*.

O próximo capítulo apresenta a arquitetura híbrida proposta por este trabalho, destacando suas etapas de funcionamento, além da descrição das características extraídas das folhas e da apresentação dos padrões usados.

Capítulo 5

Arquitetura Híbrida Proposta

5.1 Introdução

As redes neurais podem ser utilizadas em reconhecimento de padrões, como por exemplo, no reconhecimento de voz [40], caracteres [26], imagens [57], etc. Os AGs podem ser usados em problemas de otimização, tais como, roteamento [17], escalonamento de datas e horários [41], etc. Neste trabalho foi desenvolvida uma arquitetura híbrida composta de AG e rede neural *backpropagation* com o objetivo de identificar plantas por meio de suas folhas. A Figura 5.1 mostra o diagrama de blocos dessa arquitetura.

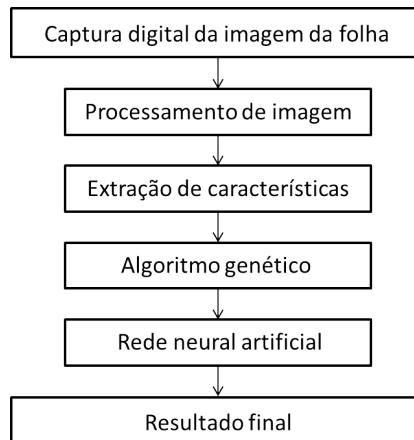


Figura 5.1: *Diagrama de blocos da arquitetura proposta neste trabalho.*

O processo de reconhecimento da arquitetura híbrida mostrado na Figura 5.1 inicia-se com a captura digital da imagem da folha, seguida pelas etapas de processamento e extração de características do padrão a ser reconhecido. Em seguida, o AG seleciona um grupo de classes cuja resposta do padrão a ser reconhecido é a mais provável e a RNA classifica o padrão, produzindo o resultado final.

Para classificar uma folha desconhecida como pertencente a uma determinada espécie é necessário conhecer *a priori* pelo menos uma folha de cada classe. No entanto, quanto maior for o número de folhas conhecidas mais preciso e generalista será o sistema de reconhecimento. Nesta dissertação, as folhas conhecidas são denominadas padrões de treinamento e as demais padrões de reconhecimento.

Neste capítulo são descritos todos os componentes do sistema híbrido proposto, mostrado no diagrama de blocos da Figura 5.1.

5.2 Arquitetura Híbrida

5.2.1 Captura digital da imagem da folha

As imagens das folhas usadas no sistema de reconhecimento ou mesmo as imagens das folhas já conhecidas e que são utilizadas para treinar esse sistema para reconhecer os padrões, devem ser adquiridas do meio ambiente por meio de *scanners* ou de câmeras digitais.

5.2.2 Processamento da imagem

Qualquer imagem que entra no sistema deve passar por um estágio de processamento de imagem. Para facilitar a compreensão esse estágio é dividido em duas etapas: pré-processamento da imagem e análise de imagem [56].

Pré-processamento da imagem

Após serem adquiridas as imagens no padrão RGB, primeiramente, são convertidas para o padrão de imagem em níveis de cinza. A Equação (5.1) é utilizada para realizar essa conversão *pixel a pixel*.

$$gray = 0,2989R + 0,5870G + 0,1140B \quad (5.1)$$

onde R, G e B correspondem às matrizes de cor: vermelho, verde e azul da imagem de entrada, respectivamente.

Depois dessa conversão é necessário transformar a imagem resultante em uma imagem binária. Para realizar essa nova conversão é preciso encontrar um limiar de trans-

formação. Esse limiar é encontrado nos histogramas das imagens RGB, por meio da identificação do ponto de mínimo entre os objetos existentes na figura. Com a finalidade de obter um limiar que satisfaça o maior número de imagens deste trabalho foi utilizado o estudo de Wu *et al.* [56]. Neste estudo Wu *et al.* realizaram a média dos histogramas de 3000 imagens de folhas no formato RGB e obtiveram os histogramas mostrados na Figura 5.2.

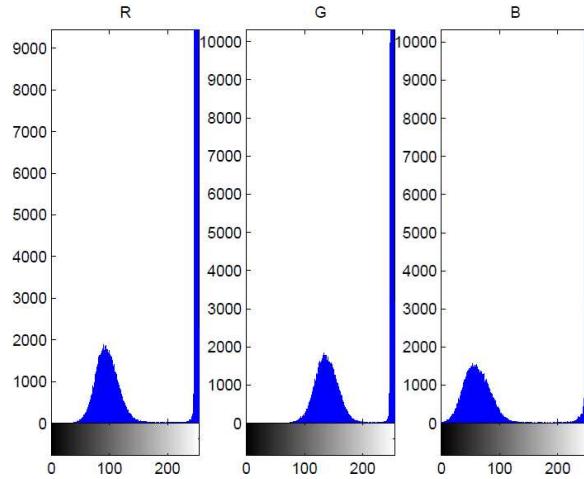


Figura 5.2: *Histograma RGB* [56].

Pode-se observar na Figura 5.2 que existem dois picos em cada histograma. Os picos localizados à esquerda são os *pixels* que formam a imagem da folha e os picos localizados à direita são os *pixels* que constituem o fundo. O ponto de mínimo entre os picos representa o limiar de separação entre a imagem da folha e o fundo. O trabalho de Wu *et al.* concluiu que o ponto de separação que melhor divide os objetos constituintes da imagem é o que tem valor 242, portanto, o valor do limiar de transformação utilizado para binarizar as imagens é o 0,95 (242/255).

Com esse limiar a imagem resultante é formada substituindo todos os *pixels* da imagem de entrada que possuem luminância maior do que o limiar calculado pelo valor 1, os demais são substituídos pelo valor 0.

Com a finalidade de remover os ruídos da imagem binarizada é realizada uma convolução da imagem com um filtro de média de tamanho 3×3 .

Análise de imagem

Na análise de imagem as bordas das folhas são obtidas convoluindo a imagem binarizada com o filtro laplaciano de tamanho 3×3 [18] mostrado na Equação (5.2) [18].

$$\begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix} \quad (5.2)$$

A Figura 5.3 mostra a imagem de uma folha passando por todas as etapas descritas nesta subseção.

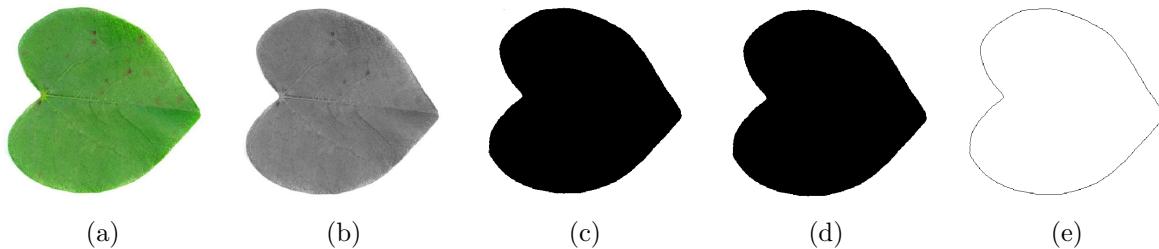


Figura 5.3: (a) Imagem de uma folha no padrão RGB; (b) imagem convertida para níveis de cinza; (c) imagem binarizada com limiar para conversão de 0,95; (d) resultado da filtragem com um filtro de média de tamanho 3×3 ; (e) borda extraída da folha utilizando um filtro laplaciano de tamanho 3×3 [56].

5.2.3 Extração de características

Neste trabalho quase todas as características extraídas das folhas são obtidas automaticamente com exceção do comprimento fisiológico que necessita da intervenção humana para ser adquirido.

As 12 características extraídas são obtidas a partir de 5 características geométricas básicas das folhas [56].

Características geométricas básicas

As 5 características geométricas básicas extraídas das folhas são:

1. Diâmetro (D): definido pela maior distância percorrida entre dois pontos opostos localizados na margem da folha;

2. Comprimento fisiológico (L_p): é a única característica que necessita da interferência do usuário para ser adquirida, pois é necessário apontar com o auxílio do mouse as extremidades da nervura principal da folha;
3. Largura fisiológica (W_p): infinitas linhas perpendiculares à nervura principal podem ser traçadas, a maior dessas linhas é considerada a largura fisiológica;

Como as coordenadas dos *pixels* são discretas as linhas que cortam perpendicularmente a nervura principal da folha não são completamente retilíneas, e, portanto, podem variar alguns graus. Em razão disso, para este trabalho são consideradas linhas perpendiculares todas aquelas linhas que cortam a nervura principal da folha e estão dentro do intervalo $90^\circ \pm 0,5^\circ$.

A relação entre o comprimento e a largura fisiológica é mostrada na Figura 5.4.

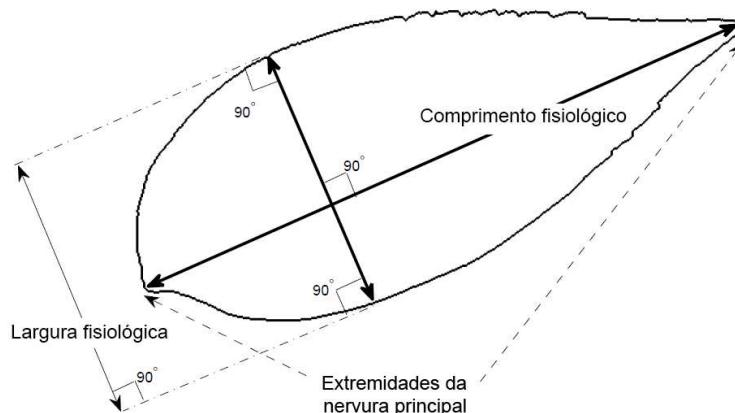


Figura 5.4: *Relação entre o comprimento e a largura fisiológica [56]*.

4. Área da folha (A): a área da folha é calculada somando todos os *pixels* que formam a folha; e
5. Perímetro da folha (P): o perímetro é calculado contando o número de *pixels* que formam a borda da folha.

As 12 características morfológicas digitais extraídas das folhas

Considerando as 5 características geométricas básicas extraídas da folha, pode-se definir as 12 características digitais utilizadas no sistema de reconhecimento.

1. Fator de suavização: é a relação entre as áreas filtradas pelos filtros de médias 5×5 e 2×2 ;
2. Relação de aspecto: é a relação entre o comprimento fisiológico (L_p) pela largura fisiológica (W_p);
3. Fator de forma (FF): essa característica é usada para descrever o grau de proximidade da folha com uma circunferência. Este fator é obtido pela Equação (5.3).

$$FF = \frac{4\pi A}{P^2} \quad (5.3)$$

onde:

- A - área da folha; e
 - P - perímetro da folha.
4. Retangularidade (R): essa característica é usada para medir a similaridade de uma folha com um retângulo. A retangularidade é obtida utilizando a Equação (5.4).

$$R = \frac{L_p W_p}{A} \quad (5.4)$$

onde:

- L_p - comprimento da folha;
 - W_p - largura da folha; e
 - A - área da folha.
5. Fator de estreitamento: esse fator é definido como sendo a relação entre o diâmetro (D) e o comprimento fisiológico da folha (L_p);
 6. Relação perímetro pelo diâmetro: é a relação entre o perímetro da folha (P) e o diâmetro da folha D ;
 7. Relação entre o perímetro e a soma da largura com o comprimento da folha: é a relação entre o perímetro da folha (P) e a soma da largura L_p com o comprimento da folha W_p .

As cinco características restantes são encontradas analisando as formas das folhas. Essa análise é realizada aplicando a operação morfológica de abertura [47] nas imagens em níveis de cinza Figura 5.6 (a). Quatro elementos estruturantes de raios 1, 2, 3 e 4 são usados nessa operação. A Figura 5.5 mostra um elemento estruturante no formato de disco com raio igual a 3.

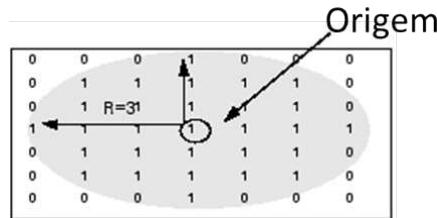


Figura 5.5: *Elemento estruturante no formato de disco com raio igual a 3 [32].*

Após aplicar a operação de abertura as formas das folhas são realçadas Figura 5.6 (b), executando a operação de subtração entre a imagem da folha obtida com a imagem da folha em tons de cinza e em seguida binarizando essa nova imagem é obtida a imagem da Figura 5.6 (c). Fazendo a contagem dos *pixels* de cada operação realizada obtém-se as áreas A_{v1} , A_{v2} , A_{v3} e A_{v4} .

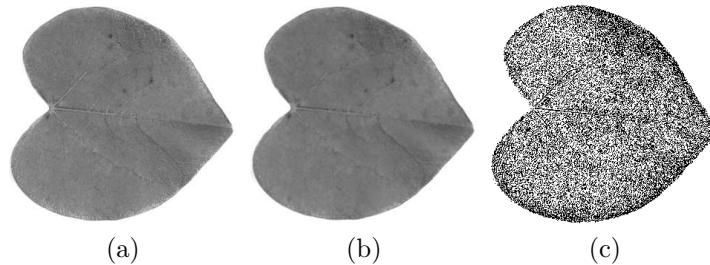


Figura 5.6: (a) *Imagen da folha no padrão de cinza;* (b) *imagem processada utilizando um fator estruturante de raio igual a 4;* (c) *imagem resultante.*

Assim, as demais características são encontradas utilizando as Equações (5.5) a (5.9).

$$8. \frac{A_{v1}}{A} \quad (5.5)$$

$$9. \frac{A_{v2}}{A} \quad (5.6)$$

$$10. \frac{A_{v3}}{A} \quad (5.7)$$

$$11. \frac{A_{v4}}{A} \quad (5.8)$$

$$12. \frac{A_{v4}}{A_{v1}} \quad (5.9)$$

onde: A - área da folha.

5.2.4 Algoritmo genético

A escolha da utilização do algoritmo genético para atuar como pré-seletor foi motivada após a observação dos resultados dos vários testes realizados usando somente o AG como classificador. Nessa classificação, verificou-se que a classe correta de uma folha a ser reconhecida estava frequentemente presente na população final do AG, porém, ela não era apresentada como a melhor solução.

A identificação de um padrão a ser reconhecido utilizando o AG inicia com a extração de suas características. Essas características são a base da criação da população inicial. O primeiro elemento desta população possui exatamente as 12 características extraídas da folha e os demais indivíduos são derivados desse elemento. Esta derivação é realizada da seguinte forma: 5 características são escolhidas aleatoriamente a partir do primeiro indivíduo para serem mantidas e as outras 7 são geradas, obedecendo uma distribuição normal de média igual ao valor da característica original e desvio padrão igual a 5% do valor da média. A Figura 5.7 mostra a região dos valores que as características podem assumir durante o processo de criação da população. Este processo é repetido 59 vezes, gerando, assim, uma população inicial de 60 indivíduos.

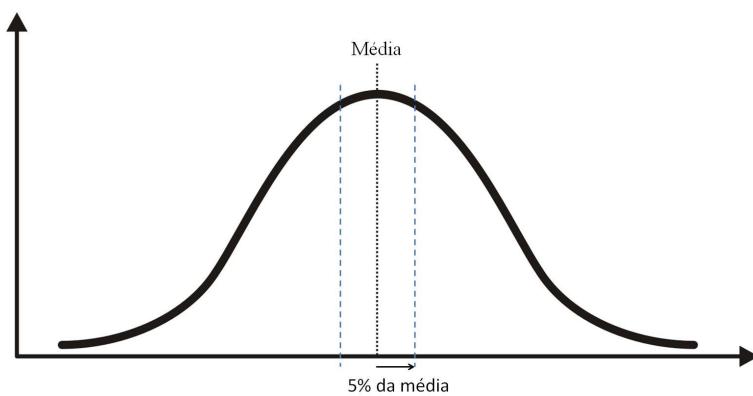


Figura 5.7: Região utilizada para gerar os novos valores das características.

A população inicial é avaliada pela função aptidão. A análise desta função utiliza duas tabelas. A primeira armazena os valores máximos e mínimos de cada uma das características dos padrões de treinamento e a segunda tabela armazena o valor mais

representativo de cada uma das características das plantas conhecidas, baseada nos k vizinhos mais próximos [15].

O método dos k vizinhos mais próximos, do inglês *k-nearest neighbor* (*k*-NN), é um dos métodos de classificação supervisionada mais simples existente na literatura. Esse classificador identifica os objetos desconhecidos com base nos padrões de treinamento que estão mais próximos deles no espaço de características.

Para utilizar o *k*-NN é necessário:

1. Um conjunto de padrões de treinamento;
2. Definir uma métrica para calcular a distância entre os padrões de treinamento e o padrão desconhecido; e
3. Definir o valor do número de vizinhos mais próximos que serão considerados pelo algoritmo (k).

Existem diversas métricas que podem ser utilizadas para calcular a distância entre os padrões de treinamento e o padrão a ser reconhecido. As mais conhecidas são:

- Distância Euclidiana: obtida pela Equação (5.10).

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (5.10)$$

onde:

– p e q - pontos em um espaço euclidiano n -dimensional;

- Distância de Mahalanobis: encontrada utilizando a Equação (5.11).

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad (5.11)$$

onde:

– x - vetor de entrada;

– μ - vetor média; e

– S a matriz de covariância.

Neste trabalho foi utilizada a distância euclidiana para calcular a distância entre os padrões de treinamento e o padrão a ser reconhecido.

Conhecida a distância entre o padrão a ser reconhecido e os padrões de treinamento a classificação é realizada considerando a maioria simples de padrões de treinamento em um universo de k padrões que tem a menor distância em relação ao padrão a ser reconhecido.

O valor de k preferencialmente deve ser um valor ímpar com o objetivo de evitar empates no momento da contagem dos vizinhos, e, também, não deve ser nem muito pequeno e nem muito grande com a finalidade de evitar que a classificação fique sensível a pontos de ruído, e que a vizinhança inclua os elementos de outras classes, respectivamente. A Figura 5.8 exemplifica a classificação de um objeto desconhecido de acordo com o valor de k . Nessa figura o objeto a ser reconhecido no centro pode ser classificado como: pertencente à classe dos quadrados se $k = 1$; pertencente à classe dos triângulos se $k = 3$; e pertencente à classe dos quadrados se $k = 7$.

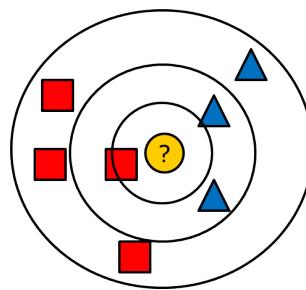


Figura 5.8: *Classificação de um objeto desconhecido utilizando o algoritmo k-NN.*

Para k igual a 3 o algoritmo k -NN foi adaptado para extrair o valor mais representativo de cada uma das 12 características de uma classe de folhas, baseado nos padrões de treinamento disponíveis, ou seja, o algoritmo 3-NN verifica cada espaço de característica de forma individual e fornece um valor que melhor representa aquela característica analisada. A Figura 5.9 ilustra o gráfico da densidade de vizinhos ($p(x)$) para um espaço de característica formado por 8 padrões de treinamento, onde o maior pico representa o valor mais representativo dessa característica de acordo com o algoritmo 3-NN [15].

De posse da tabela que armazena os valores máximos e mínimos de cada característica e da tabela que armazena o valor mais representativo baseado no algoritmo 3-NN, a função aptidão do AG é capaz de calcular o *fitness* das soluções contidas na população.

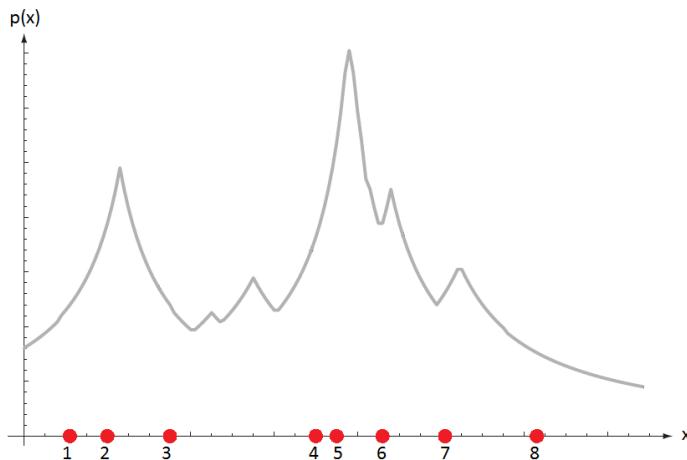


Figura 5.9: Gráfico da densidade de vizinhos ($p(x)$) versus o espaço de característica (x) traçado segundo o algoritmo 3-NN. Adaptado de [15].

A função aptidão verifica se a característica que está sendo analisada encontra-se entre os valores máximo e mínimo da correspondente característica de cada uma das classes. Se ela não está entre esses valores, sua avaliação é zero. Se ela encontra-se entre os valores um estudo é realizado para conhecer a distância da característica até o valor mais representativo da classe.

O procedimento para obter a distância em que deve ser considerada a característica que está sendo analisada como sendo próxima ou não do valor mais representativo da respectiva característica da folha na tabela construída funciona da seguinte forma: a partir de uma distribuição igualitária de distâncias faz-se variar a distância inicial de cada característica em um processo iterativo, e verifica-se o número de acertos de reconhecimento tomando como base a simples comparação entre os padrões de treinamento e a tabela que armazena o valor mais representativo de cada folha. O procedimento é encerrado quando a taxa de acerto parar de evoluir. Esse cálculo é executado apenas uma vez, pois depois que as distâncias são encontradas não é mais necessário recalcular-as a cada execução. Esse procedimento deve ser realizado, pois existem características mais vulneráveis à proximidade do valor mais representativo do que outras.

Os valores encontrados pelo algoritmo são apresentados na Equação (5.12).

$$\text{distancia}(\%) = [0,35 \ 0,63 \ 0,1 \ 0,04 \ 0,11 \ 0,1 \ 0,09 \ 0,53 \ 0,19 \ 0,17 \ 0,55 \ 1,0] \quad (5.12)$$

Agora, para avaliar as características foi desenvolvido um procedimento que encontra o valor do peso atribuído a cada posição do vetor de características. O processamento desse algoritmo é parecido com o procedimento do algoritmo que encontra o valor das distâncias. Desse modo, são mostrados na Equação (5.13) os pesos encontrados para fazer a avaliação de cada uma das 12 características, obedecendo a sequência apresentada na subseção 5.2.3 deste capítulo.

$$\text{pesos} = [14 \ 25 \ 23 \ 24 \ 17 \ 5 \ 4 \ 8 \ 0 \ 25 \ 22 \ 16] \quad (5.13)$$

Quanto mais próximo a característica analisada está do valor mais representativo, melhor é a sua avaliação.

Assim, se o valor da característica que está sendo analisada encontra-se dentro do intervalo de distância mostrado na Equação (5.12), o peso dessa característica é acrescido de 2 pontos. Por exemplo, se a primeira característica do padrão desconhecido está a uma distância de $\pm 35\%$ do seu valor mais representativo, o valor do peso dessa característica passa a valer 16 ($14+2$), se a segunda característica está a uma distância de $\pm 63\%$ do seu valor mais representativo, o valor do peso dessa característica passa a valer 27 ($25+2$), se a terceira característica não está a uma distância de $\pm 10\%$ do seu valor mais representativo, o valor do peso dessa característica é obtido diretamente da Equação (5.13), ou seja, vale 23, e essa avaliação é repetida até a décima segunda característica.

O valor total da aptidão do indivíduo é calculado pela soma da avaliação de cada uma das 12 características morfológicas das folhas. Esse processo de comparação é repetido entre todos os elementos do banco de dados, composto de 32 plantas. No final, o indivíduo que possui a maior aptidão é separado e rotulado com o valor da classe que ele mais se aproxima.

A próxima etapa do algoritmo AG é selecionar os pais que serão utilizados no cruzamento. O método de seleção dos pais usado neste trabalho é o *ranking* que foi apresentado na subseção 3.2.4 do Capítulo 3 desta dissertação. Os valores M_n e M_x utilizados para calcular o *ranking* usando a Equação (3.2) são 0,9 e 1,1, respectivamente. A escolha desses valores garante que o somatório das avaliações atribuídas aos indivíduos da população em

uma determinada geração seja sempre igual ao número de indivíduos da população.

As 12 características presentes nos indivíduos que formam a população do AG podem assumir qualquer valor real maior do que zero. Assim, para aplicar os operadores genéticos, cada uma das características dos indivíduos selecionados para o cruzamento são binarizadas em 25 bits. Dessa forma, os cromossomos dessa população depois de codificados passam a ser compostos por 300 genes (12×25). A Figura 5.10 ilustra a formação do cromossomo na arquitetura híbrida.

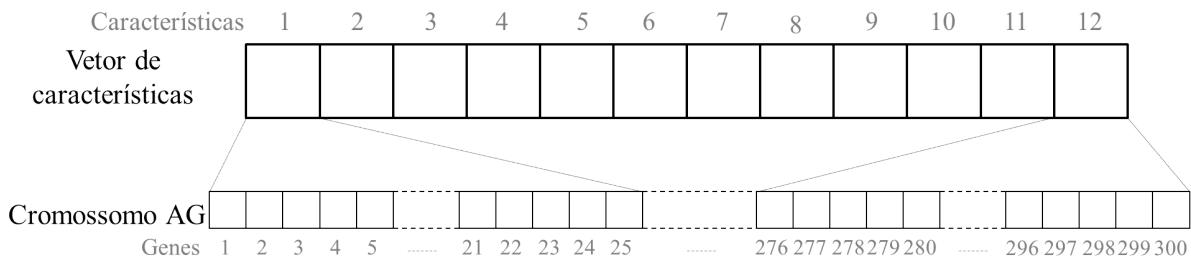


Figura 5.10: *Representação do cromossomo na arquitetura híbrida.*

O operador de cruzamento implementado neste trabalho é o *crossover* de dois pontos com a probabilidade de cruzamento de 90%. Em cada um dos pais esse cruzamento escolhe dois pontos de corte, sendo que esses pontos devem ser múltiplos de 25 para não separar os genes de uma mesma característica. Assim, o primeiro filho é formado pela parte do primeiro pai fora dos pontos de corte e pela parte do segundo pai entre os pontos de corte. O segundo filho é formado pelas partes restantes.

Com a finalidade de garantir a continuidade da existência de diversidade genética na população o operador de mutação foi utilizado com a probabilidade de ocorrência de 10%. O funcionamento desse operador baseia-se na troca de bits do cromossomo. Nesta troca, 3 bits da cadeia do cromossomo são escolhidos aleatoriamente para serem invertidos.

O ciclo do AG termina na conversão de binário para decimal de cada um dos genes e na reavaliação de todos os novos indivíduos pela função aptidão. Com o objetivo de estimular o crescimento do AG na direção da resposta foi empregado a técnica do elitismo. Esta técnica substitui o pior indivíduo da população corrente pelo melhor indivíduo da população anterior.

A adaptação do AG para funcionar como pré-seletor consistiu em ajustar o critério de parada do algoritmo, para satisfazer a situação em que a população inicial é homogênea e

necessita ser diversificada por meio dos cruzamentos e das mutações do AG, como também para a situação contrária em que a população inicial é muito heterogênea e é necessário eliminar as piores classes de plantas. Desse modo, o critério de parada do AG foi definido como sendo o momento em que o número de indivíduos da população se torna menor ou igual a 5 ou o número de ciclos do algoritmo é maior do que 5.

5.2.5 Redes neurais artificiais

A rede neural *backpropagation* é treinada para classificar a folha desconhecida considerando apenas as classes obtidas pelo AG com a vantagem de que a procura pela resposta final ocorre em um universo de busca menor, que antes era de 32 classes de plantas e após a pré-seleção passou a ser de no máximo 5 classes.

Com o objetivo de reduzir o tempo de treinamento da RNA considerando a diminuição do volume da informação a ser processada, os valores da taxa de aprendizagem e do *momentum* que forneceram os melhores resultados após vários testes foram 0,1 e 0,5, respectivamente.

Em resumo, para cada folha a ser reconhecida pelo sistema proposto neste trabalho é necessário que o AG realize a pré-seleção das folhas tomando como base os padrões disponíveis para o treinamento, para, em seguida, a RNA ser treinada com as classes resultantes da etapa anterior. Como não existe a garantia de que a pré-seleção executada pelo AG forneça sempre a mesma resposta para um mesmo padrão desconhecido, não é possível criar RNAs especializadas em conjuntos de classes, dessa forma é necessário treinar a rede neural para cada padrão que se deseja reconhecer.

As Tabelas 5.1 e 5.2 apresentam os parâmetros utilizados para construir o AG e a RNA da arquitetura híbrida proposta neste trabalho.

5.3 Considerações Finais deste Capítulo

Neste capítulo foram descritos todos os componentes que fazem parte do sistema híbrido.

No próximo capítulo são apresentados a interface de usuário do *software* desenvolvido para realizar o reconhecimento das folhas, o banco de dados utilizado e os resultados

Tabela 5.1: *Parâmetros do AG.*

Parâmetros do AG	Valores
Número de ciclos	5
Número de genes	12
Número de bits na binarização	25
Tamanho da população	60
Probabilidade de cruzamento	90%
Elitismo	1
Probabilidade de mutação	10%

Tabela 5.2: *Parâmetros da RNA.*

Parâmetros da RNA	Valores
Termo <i>momentum</i> (η)	0,5
Taxa de aprendizagem (α)	0,1
Número de neurônios na camada oculta	60
Erro máximo acumulado	0,05

obtidos nos testes realizados utilizando as redes neurais, o algoritmo genético e o sistema híbrido. É também realizada uma comparação dos resultados obtidos neste trabalho com os resultados existentes na literatura.

Capítulo 6

Resultados Obtidos

6.1 Introdução

Este capítulo apresenta o banco de dados de folhas utilizado neste trabalho, a interface do *software* desenvolvido para realizar o reconhecimento das plantas por meio de suas folhas, os tempos de processamento e os resultados obtidos nas simulações utilizando como classificador o algorítimo genético, a rede neural artificial e a arquitetura híbrida proposta. Em seguida, as taxas de acerto desses três métodos são comparados com as taxas de acerto dos métodos existentes na literatura.

Tanto para o desenvolvimento dos sistemas de reconhecimento quanto para a execução das simulações foram utilizados os recursos computacionais mostrados na Tabela 6.1.

Tabela 6.1: *Recursos computacionais utilizados neste trabalho.*

Processador	Intel Core 2 Quad 2,4 GHz
Memória RAM	4 Gb
Sistema operacional	Windows 7
Disco rígido	180 Gb
Ferramenta de programação	Matlab R2009a

6.2 Banco de Dados de Folhas Utilizado

O banco de dados de folhas utilizado nesta dissertação é constituído por folhas de plantas da região de Shanghai, na China. Esse banco é composto por 32 grupos de plantas totalizando 1907 folhas com resolução de 800×600 *pixels*, sem restrições em relação à translação, a rotação e ao escalonamento da folha. O banco de dados faz parte de um projeto de reconhecimento de plantas chamado FLAVIA e vem sendo amplamente utilizado em pesquisas relacionados ao tema. As pesquisas [28, 51, 56] são

exemplo de trabalhos que utilizaram esse banco. O arquivo com todos os exemplares pode ser encontrado para download em [3].

A Tabela 6.2 mostra o nome científico das plantas e a quantidade de folhas presentes em cada classe do banco e a Figura 6.1 ilustra uma folha de cada planta descrita nessa tabela.

Tabela 6.2: *Nome científico e quantidade de folhas.*

Nome científico	Quantidade de folhas
<i>Phyllostachys edulis</i>	59
<i>Aesculus chinensis</i>	63
<i>Berberis anhweiensis</i>	65
<i>Cercis chiensis</i>	72
<i>Indigofera tinctoria</i>	73
<i>Acer dalmatum</i>	56
<i>Phoebe zhennan</i>	62
<i>Kalopanax septemlobus</i>	52
<i>Cinnamomum japonicum</i>	55
<i>Koelreuteria paniculata</i>	59
<i>Ilex macrocarpa</i>	50
<i>Pittosporum tobira</i>	63
<i>Chimonanthus praecox</i>	52
<i>Cinnamomum camphora</i>	65
<i>Viburnum awabuki</i>	60
<i>Osmanthus fragrans</i>	56
<i>Cedrus deodara</i>	77
<i>Ginkgo biloba</i>	62
<i>Lagerstroemia indica</i>	61
<i>Nerium oleander</i>	66
<i>Podocarpus macrophyllus</i>	60
<i>Prunus yedoensis</i>	55
<i>Ligustrum lucidum</i>	55
<i>Tonna sinensis</i>	65
<i>Prunus pérsica</i>	54
<i>Manglietia fordiana</i>	52
<i>Acer buergerianum</i>	53
<i>Mahonia bealei</i>	55
<i>Magnolia grandiflora</i>	57
<i>Populus × canadensis</i>	64
<i>Liriodendron chinense</i>	53
<i>Citrus reticulata</i>	56



Figura 6.1: Amostra das folhas do banco de dados utilizado neste trabalho.

6.3 Interface do Usuário do *Software* Desenvolvido para Realizar o Reconhecimento das Folhas

A Figura 6.2 apresenta a interface do usuário do *software* desenvolvido neste trabalho para realizar o reconhecimento das plantas por meio de suas folhas.

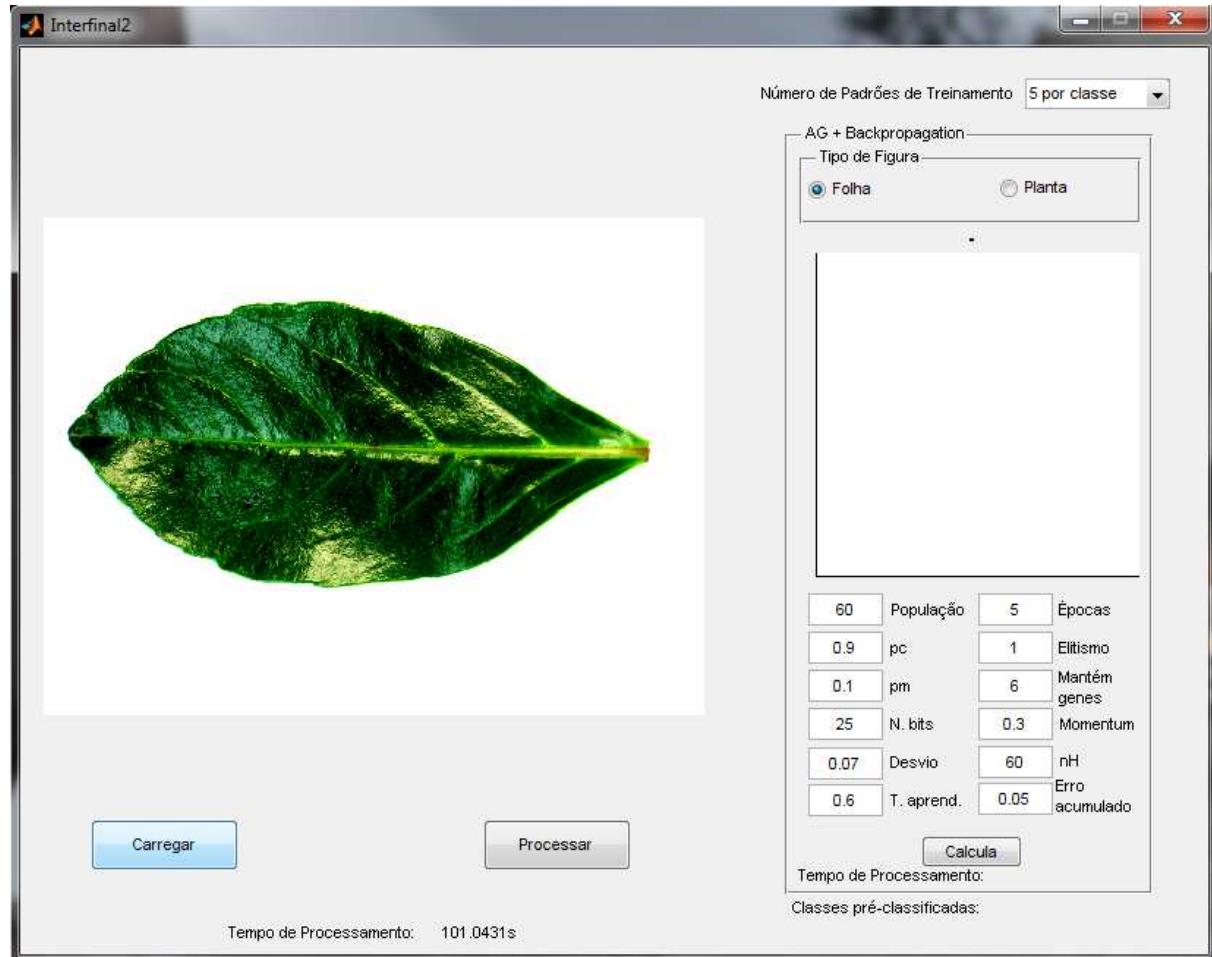


Figura 6.2: Interface de entrada do *software* desenvolvido para realizar o reconhecimento das folhas.

À esquerda da janela da Figura 6.2 é mostrada a imagem da folha a ser reconhecida pelo sistema, abaixo dessa imagem é apresentado o tempo necessário para que o extrator de características realize o pré-processamento dessa folha. À direita, estão os parâmetros do sistema híbrido que podem ser alterados pelo usuário e na parte superior da janela encontra-se o campo em que é possível configurar o número de padrões de treinamento que o sistema utiliza para reconhecer as folhas.

A Figura 6.3 mostra a janela do resultado obtido utilizando o sistema híbrido proposto nesta dissertação. Pode-se observar nesta figura que o sistema reconheceu correta-

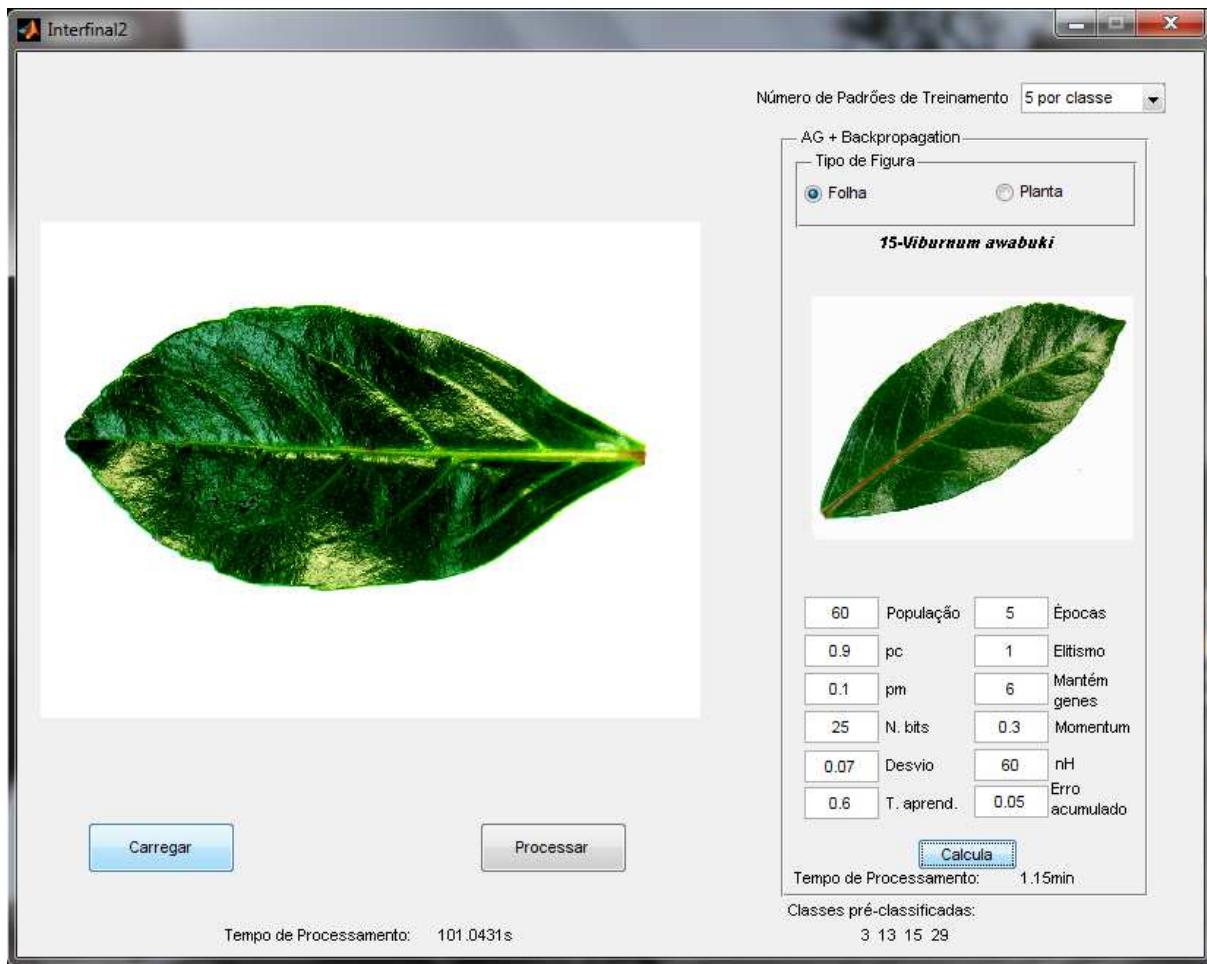


Figura 6.3: *Resultado do processamento realizado pelo software implementado.*

mente a folha a ser reconhecida apresentando o nome científico da planta e a imagem de uma outra folha da mesma classe. É mostrado também a soma dos tempos de processamento do AG e da RNA necessários para realizar somente o reconhecimento da folha. E na parte inferior dessa janela é apresentado as classes que o AG pré-selecionou nesta execução.

6.4 Resultados Obtidos

Pode-se observar na Tabela 6.2 que os grupos de plantas presentes no banco de dados utilizado não possuem a mesma quantidade de folhas. Devido a isso, as simulações que mostram a taxa de acerto e o tempo de execução para cada um dos sistemas de reconhecimento são divididas em nove ensaios, em que, os padrões de treinamento por classe para os oito primeiros ensaios são variados de 5 em 5 até alcançar o número 40, e na última seção de ensaios todas as classes possuem 10 padrões desconhecidos. Com

10 padrões a serem reconhecidos é possível comparar diretamente os resultados obtidos neste trabalho com os resultados obtidos por Wu *et al.* [56] e Kadir *et al.* [28], que reconheceram esse mesmo número de padrões por classe.

O número de padrões a serem reconhecidos que passam um a um pelo sistema de reconhecimento é o complemento do número de padrões de treinamento. Por exemplo, se a classe possui 60 folhas e 5 estão sendo utilizadas como padrões de treinamento, 55 são os padrões desconhecidos.

Em todos os testes realizados as folhas utilizadas como padrões de treinamento em cada classe são escolhidas aleatoriamente. A Tabela 6.3 mostra o número de padrões de treinamento e o número de padrões a ser reconhecidos utilizados em cada ensaio.

Tabela 6.3: *Número de padrões de treinamento e de reconhecimento utilizados em cada ensaio.*

Ensaio	Padrões de treinamento	Padrões de reconhecimento
1	5 por classe = 160	1747
2	10 por classe = 320	1587
3	15 por classe = 480	1427
4	20 por classe = 640	1267
5	25 por classe = 800	1107
6	30 por classe = 960	947
7	35 por classe = 1120	787
8	40 por classe = 1280	627
9	1587	320

Em vez de estimar os parâmetros, taxa de acerto e tempo de processamento, por um único valor os resultados apresentados graficamente neste capítulo são mostrados em um intervalo de confiança (IC). O IC é uma representação estatística que indica o quanto prováveis de acontecer são estas estimativas com base em um conjunto de amostras.

O intervalo de confiança escolhido para representar os resultados obtidos neste trabalho é o de 95%. Assim, os resultados podem ser interpretados da seguinte forma: em 95% dos casos o resultado esperado está entre os limites superior e inferior indicados no gráfico.

6.4.1 Utilização do algoritmo genético como classificador

A Figura 6.4 mostra os resultados obtidos no reconhecimento das folhas das plantas utilizando somente o AG. Cada ensaio foi repetido 5 vezes com a finalidade de encontrar

o intervalo de confiança de 95%. A Tabela 6.4 apresenta o valor médio das observações realizadas.

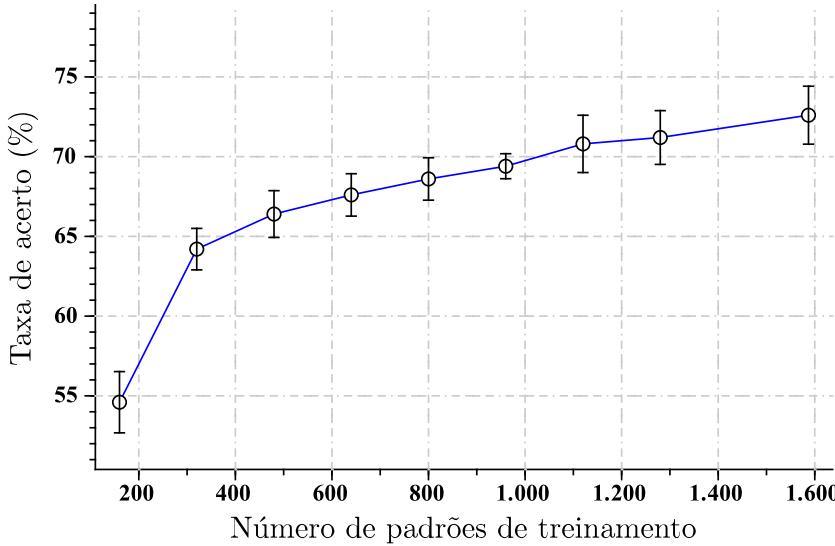


Figura 6.4: *Gráfico da taxa de acerto obtida utilizando o algoritmo genético como classificador.*

Tabela 6.4: *Média da taxa de acerto obtida utilizando o algoritmo genético como classificador.*

Padrões de treinamento	Padrões de reconhecimento	Taxa de acerto média (%)
160	1747	55
320	1587	64
480	1427	66
640	1267	68
800	1107	69
960	947	70
1120	787	71
1280	627	71
1587	320	73

Pode-se verificar na Tabela 6.4 e na Figura 6.4 que à medida que aumenta o número de padrões de treinamento a taxa de reconhecimento também aumenta. O motivo desse aumento é que o AG se torna mais generalista, e, consequentemente, melhora sua capacidade de reconhecimento. Mas, se comparado com a taxa de acerto das arquiteturas presentes na literatura (*cf.* Tabela 6.13) esse classificador apresenta desempenho muito inferior.

A Figura 6.5 mostra o tempo de processamento do algoritmo genético em um IC 95% para gerar os resultados da Tabela 6.4 e da Figura 6.4. A Tabela 6.5 apresenta o tempo médio desses processamentos.

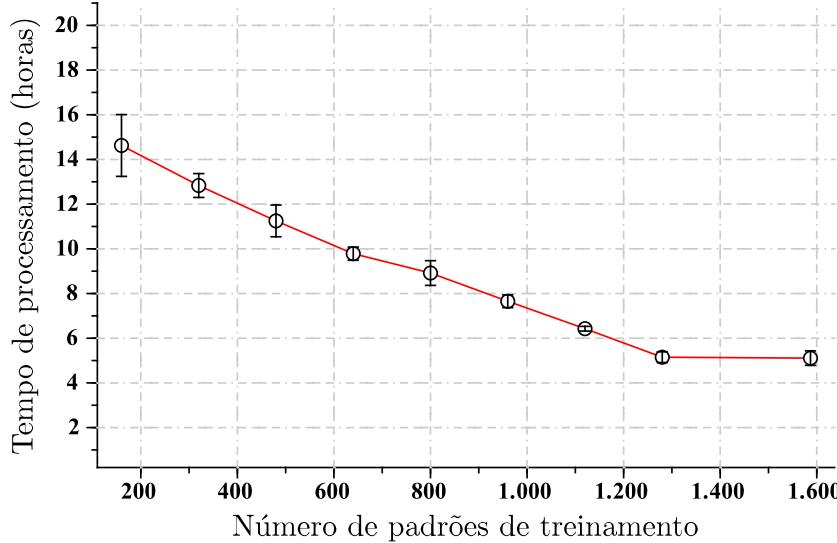


Figura 6.5: *Gráfico do tempo de processamento obtido utilizando o algoritmo genético como classificador.*

Tabela 6.5: *Tempo de processamento médio obtido utilizando o algoritmo genético como classificador.*

Padrões de treinamento	Padrões de reconhecimento	Tempo de processamento (horas)
160	1747	14,6
320	1587	12,8
480	1427	11,2
640	1267	9,8
800	1107	8,9
960	947	7,6
1120	787	6,4
1280	627	5,1
1587	320	5,1

Pode-se observar na Tabela 6.5 e na Figura 6.5 que à medida que o número de padrões de treinamento aumenta o tempo para reconhecer todas as folhas desconhecidas diminui. A diminuição do tempo ocorre porque no último ensaio o algoritmo precisa reconhecer menos folhas (320 no total como mostrado na Tabela 6.3) e leva em média 0,95 minutos em cada uma, $\left(\frac{(5,1 \times 60)}{320}\right)$. Mas se for analisar o tempo para reconhecer

cada padrão individualmente, este aumenta se comparado com o tempo dos primeiros ensaios. Por exemplo, no primeiro ensaio o tempo de processamento para reconhecer cada padrão desconhecido é de apenas 0,5 minutos, $\left(\frac{(14,6 \times 60)}{1747}\right)$. O aumento do tempo para processar cada padrão é justificado, pois existem mais operações nos cálculos quando mais padrões de treinamento são inseridos.

6.4.2 Utilização da RNA com treinamento *backpropagation* como classificador

A Figura 6.6 e a Tabela 6.6 mostram os resultados obtidos no reconhecimento das folhas das plantas utilizando somente a RNA treinada pelo algoritmo *backpropagation*. Para esse classificador cada ensaio foi realizado apenas uma vez devido ao alto tempo de processamento requerido.

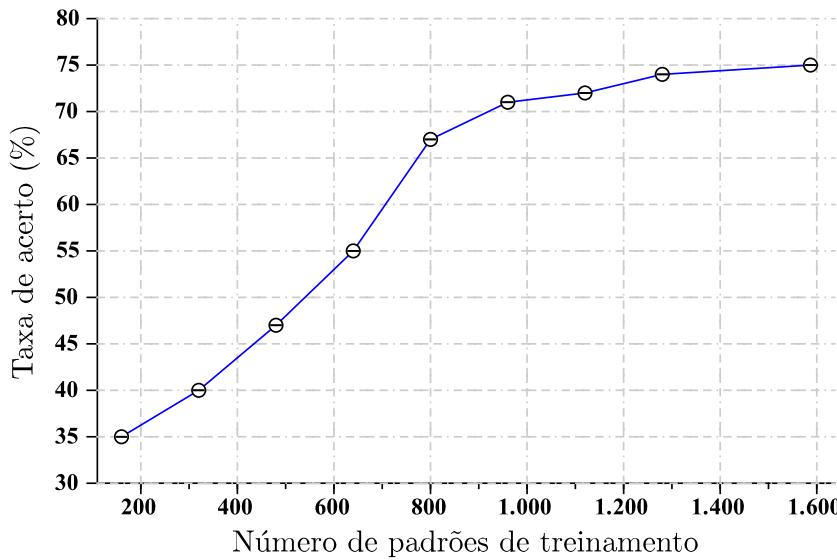


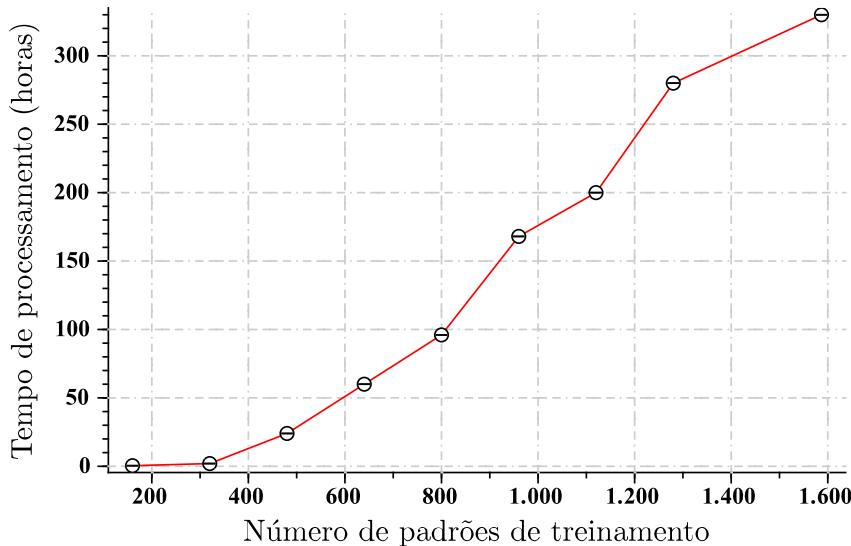
Figura 6.6: Gráfico da taxa de acerto obtida utilizando a RNA como classificador.

Pode-se verificar na Tabela 6.6 e na Figura 6.6, assim, como aconteceu no reconhecimento utilizando o AG que a taxa de acerto obtida com a RNA aumenta quando o número de padrões de treinamento também aumenta. Novamente, o motivo desse aumento é que a RNA se torna mais generalista, e, consequentemente, melhora sua capacidade de reconhecimento. Contudo, apesar da taxa de acerto da RNA nos últimos ensaios ser um pouco superior à taxa de acerto do AG, ela ainda apresenta desempenho bem inferior às taxas obtidas com as arquiteturas presentes na literatura.

Tabela 6.6: *Média da taxa de acerto obtida utilizando a RNA como classificador.*

Padrões de treinamento	Padrões de reconhecimento	Taxa de acerto (%)
160	1747	35
320	1587	40
480	1427	47
640	1267	55
800	1107	67
960	947	71
1120	787	72
1280	627	74
1587	320	75

A Figura 6.7 e a Tabela 6.7 mostram o tempo de processamento da RNA para encontrar os resultados obtidos na Tabela 6.6 e na Figura 6.6.

Figura 6.7: *Gráfico do tempo de processamento obtido utilizando a RNA como classificador.*

Pode-se observar na Tabela 6.7 e na Figura 6.7 que com o aumento do número de padrões de treinamento o tempo necessário para treinar a rede RNA também aumenta. Esse aumento é justificado pelo crescimento do volume de informações que a rede necessita processar. Cada padrão inserido representa 12 características a mais que devem ser levadas em conta nos cálculos da correção dos pesos da rede. O aumento do tempo de processamento foi tão expressivo que chegou ao ponto de inviabilizar a execução de novos testes.

Tabela 6.7: *Tempo de processamento obtido utilizando a RNA como classificador.*

Padrões de treinamento	Padrões de reconhecimento	Tempo de processamento (horas)
160	1747	0.45
320	1587	2
480	1427	24
640	1267	60
800	1107	96
960	947	168
1120	787	200
1280	627	280
1587	320	330

6.4.3 Utilização do sistema híbrido proposto na classificação de folhas

Utilização do AG como pré-seletor

A justificativa apresentada na subseção 5.2.4 desta dissertação para utilizar o algoritmo genético como pré-seletor foi de que após a observação dos resultados obtidos nos testes realizados usando somente o AG como classificador, verificou-se que a classe correta de uma folha a ser reconhecida estava frequentemente presente na população final desse algoritmo. O gráfico da Figura 6.8 e a Tabela 6.8 mostram a eficácia da utilização do AG como pré-seletor. Esse gráfico apresenta os resultados obtidos em um IC de 95% traçado com base em 5 testes de cada ensaio e essa tabela mostra o valor médio das amostras.

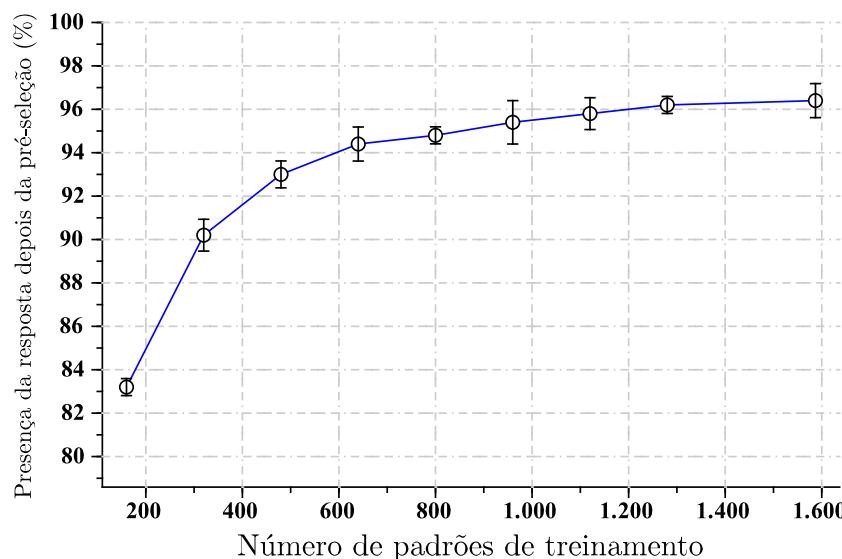
Figura 6.8: *Gráfico da probabilidade da resposta correta estar na população final do AG após a pré-seleção de plantas.*

Tabela 6.8: *Média da probabilidade da resposta correta estar na população final do AG após a pré-seleção das plantas.*

Padrões de treinamento	Padrões de reconhecimento	Probabilidade da resposta estar presente na população final do AG (%)
160	1747	83
320	1587	90
480	1427	93
640	1267	94
800	1107	95
960	947	95
1120	787	96
1280	627	96
1587	320	96

Pode-se observar na Tabela 6.8 e na Figura 6.8 que à medida que cresce o número de padrões de treinamento, aumenta a probabilidade da classe da folha a ser reconhecida estar dentro da população final do AG, como, por exemplo, para o treinamento com 5 padrões por classe (ensaio 1) essa probabilidade é de 83% em média; no entanto para o treinamento com 1587 padrões (ensaio 9) essa probabilidade chegou a ser maior do que 97% quando considerado o limite superior do IC de 95%.

A Figura 6.9 e a Tabela 6.9 mostram o tempo necessário para o algoritmo genético pré-selecionar as classes de folhas que possuem maior aptidão com a folha desconhecida. O tempo mostrado nesta tabela e nessa figura são dos testes que geraram os resultados da Tabela 6.8 e da Figura 6.8.

Tabela 6.9: *Tempo médio de processamento obtido utilizando o AG como pré-seletor de plantas.*

Padrões de treinamento	Padrões de reconhecimento	Tempo de processamento (minutos)
160	1747	51
320	1587	49
480	1427	45
640	1267	42
800	1107	36
960	947	31
1120	787	25
1280	627	20
1587	320	20

De acordo com os resultados apresentados na Tabela 6.9, pode-se observar uma

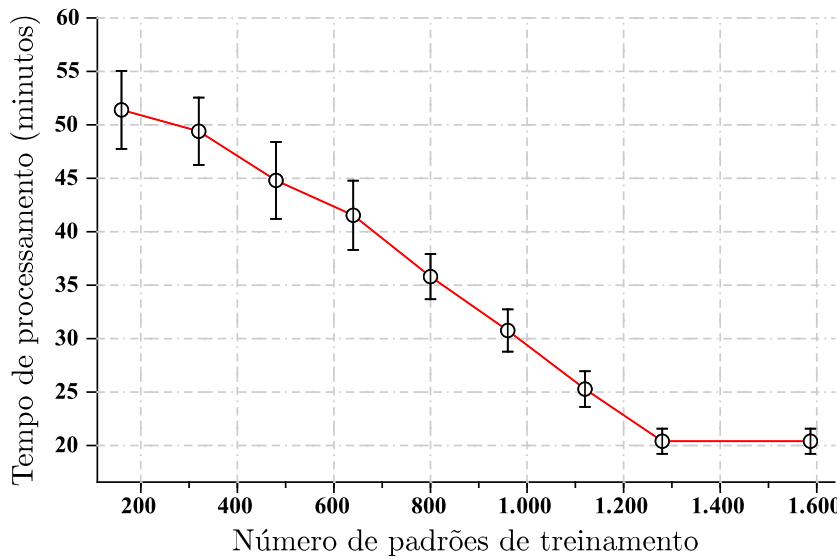


Figura 6.9: *Gráfico do tempo de processamento obtido utilizando o AG como um pré-seletor de plantas com um IC de 95%.*

diminuição do tempo médio de processamento do algoritmo genético para pré-selecionar as classes de maior aptidão com a imagem da folha a ser reconhecida. Assim, como aconteceu na simulação utilizando o AG como classificador, a diminuição do tempo de processamento é justificada, pois menos folhas foram utilizadas no último ensaio quando comparado com os ensaios anteriores, mas se é analisado o tempo para realizar a pré-seleção de cada padrão individualmente, este aumenta quando o número de padrões de treinamento também aumenta. Por exemplo, no primeiro ensaio o tempo médio necessário para pré-selecionar as classes é 1,75 segundos, $\left(\frac{(51 \times 60)}{1747}\right)$, e para o último ensaio esse tempo é de 3,75 segundos, $\left(\frac{(20 \times 60)}{320}\right)$. O aumento do tempo para pré-selecionar cada padrão individualmente no último ensaio é justificado, pois quanto mais padrões de treinamento são utilizados mais operações são realizadas.

Utilização do sistema híbrido proposto na classificação

A Figura 6.10 mostra os resultados obtidos no reconhecimento das folhas das plantas utilizando o sistema híbrido proposto neste trabalho considerando um IC de 95%. A Tabela 6.10 apresenta o valor médio das observações realizadas.

Pode-se verificar na Tabela 6.10 e na Figura 6.10 que à medida que cresce o número de padrões de treinamento, a taxa de reconhecimento aumenta. O motivo desse aumento é

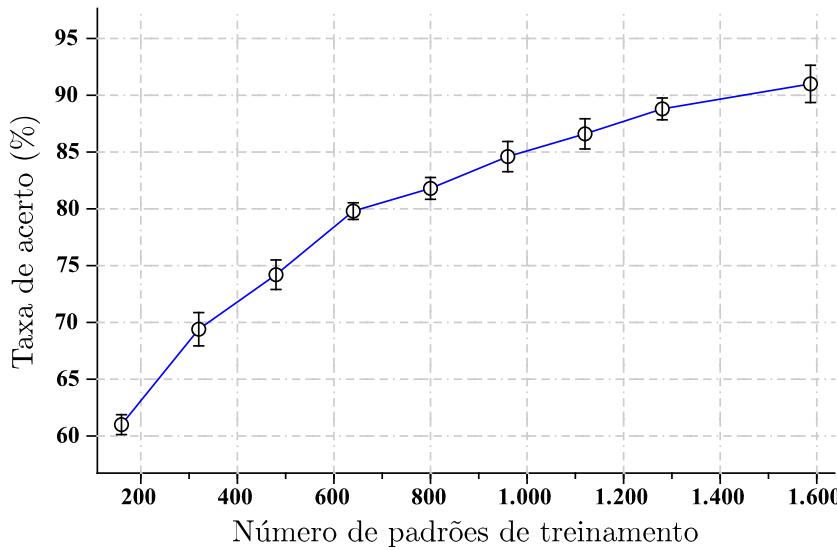


Figura 6.10: Gráfico da taxa de acerto obtida utilizando o sistema híbrido como classificador considerando um IC de 95%.

Tabela 6.10: Média da taxa de acerto obtida utilizando o sistema híbrido.

Padrões de treinamento	Padrões de reconhecimento	Taxa de acerto (%)
160	1747	61
320	1587	69
480	1427	74
640	1267	80
800	1107	82
960	947	85
1120	787	87
1280	627	89
1587	320	91

que o sistema híbrido ficou mais generalista e melhorou sua capacidade de reconhecimento.

Com uma taxa média de acerto de 91% chegando a ultrapassar 93% quando considerado o limite superior do intervalo de confiança de 95%, pode-se observar na Tabela 6.13 que o sistema híbrido proposto nesta dissertação está entre as melhores taxas de reconhecimento existentes na literatura.

A Tabela 6.11 mostra o resultado da simulação do nono ensaio em que foram reconhecidos 10 padrões por classe e obteve-se a maior taxa de reconhecimento (93,2%). Pode-se observar nessa tabela em quais classes ocorreram os erros de pré-seleção e de classificação.

Tabela 6.11: Melhor resultado obtido no ensaio em que 10 padrões em cada classe são desconhecidos.

Nome Científico	Quant. de Folhas	Padrões treinados	Folhas pré-selecionadas incorretamente	Folhas classificadas incorretamente
<i>Phyllostachys edulis</i>	59	49	0	1
<i>Aesculus chinensis</i>	63	53	1	1
<i>Berberis anhweiensis</i>	65	55	0	0
<i>Cercis chiensis</i>	72	62	0	0
<i>Indigofera tinctoria</i>	73	63	0	0
<i>Acer dalmatum</i>	56	46	0	0
<i>Phoebe zhennan</i>	62	52	1	1
<i>Kalopanax septemlobus</i>	52	42	0	0
<i>Cinnamomum japonicum</i>	55	45	0	1
<i>Koelreuteria paniculata</i>	59	49	0	0
<i>Ilex macrocarpa</i>	50	40	1	1
<i>Pittosporum tobira</i>	63	53	1	1
<i>Chimonanthus praecox</i>	52	42	0	1
<i>Cinnamomum camphora</i>	65	55	0	1
<i>Viburnum awabuki</i>	60	50	0	0
<i>Osmanthus fragrans</i>	56	46	2	2
<i>Cedrus deodara</i>	77	67	0	0
<i>Ginkgo biloba</i>	62	52	0	0
<i>Lagerstroemia indica</i>	61	51	0	0
<i>Nerium oleander</i>	66	56	0	1
<i>Podocarpus macrophyllus</i>	60	50	0	1
<i>Prunus yedoensis</i>	55	45	0	0
<i>Ligustrum lucidum</i>	55	45	0	1
<i>Tonna sinensis</i>	65	55	0	2
<i>Prunus pérsica</i>	54	44	2	2
<i>Manglietia fordiana</i>	52	42	1	1
<i>Acer buergerianum</i>	53	43	0	0
<i>Mahonia bealei</i>	55	45	0	0
<i>Magnolia grandiflora</i>	57	47	0	1
<i>Populus × canadensis</i>	64	54	0	0
<i>Liriodendron chinense</i>	53	43	0	1
<i>Citrus reticulata</i>	56	46	1	2
Total	1907	1587	10	22

A Figura 6.11 mostra o tempo de processamento do sistema híbrido em um IC de 95% que produziu os resultados da Tabela 6.10 e da Figura 6.10. A Tabela 6.12 mostra o valor médio das amostras.

O tempo de processamento do sistema híbrido é composto pelos tempos de processamento do AG e da RNA. O tempo de processamento do AG nesse sistema é o mesmo

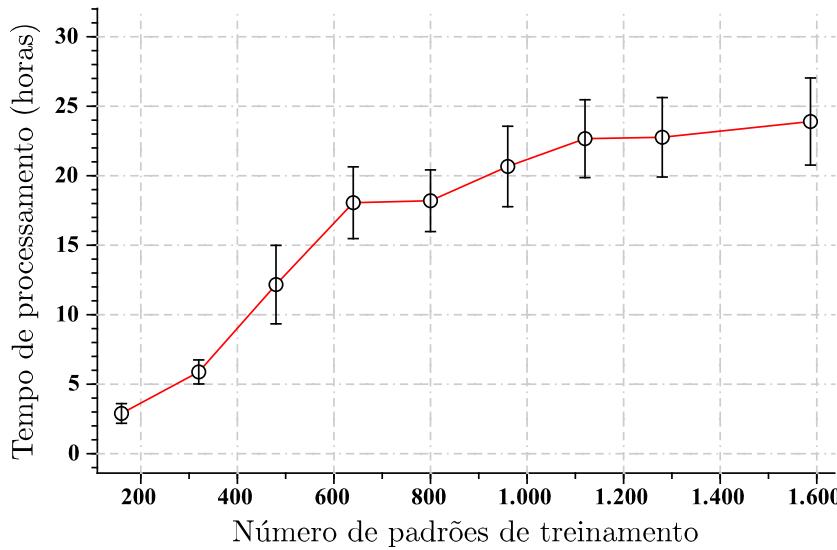


Figura 6.11: *Gráfico do tempo de processamento obtido utilizando como classificador o sistema híbrido em um IC de 95%.*

Tabela 6.12: *Tempo médio de processamento obtido utilizando o sistema híbrido como classificador.*

Padrões de treinamento	Padrões de reconhecimento	Tempo de processamento (horas)
160	1747	3
320	1587	6
480	1427	12
640	1267	18
800	1107	18
960	947	20
1120	787	22
1280	627	23
1587	320	24

tempo de processamento de quando o AG foi utilizado como pré-seletor de plantas (resultados apresentados na Tabela 6.9 e na Figura 6.9). A diferença observada é o tempo de treinamento da RNA que é muito menor se comparado com o tempo de treinamento de uma RNA sem que as entradas sejam pré-selecionadas pelo AG (resultados mostrados na Tabela 6.7 e na Figura 6.7). Dessa forma, esse sistema necessitou em média de 3,75 segundos para pré-selecionar cada um dos 320 padrões desconhecidos do nono ensaio e aproximadamente 4,5 minutos para classificar cada folha.

Assim, para reconhecer 10 padrões por classe utilizando o sistema híbrido, ocorreu em média um erro de 4% quando o algoritmo genético foi utilizado como pré-seletor. A

consequência desse erro é que durante o processamento do AG a verdadeira classe da folha a ser reconhecida não está presente na pré-seleção realizada. Desse modo, conhecendo o erro causado pelo AG é possível obter o erro produzido pela RNA por meio dos gráficos apresentados nesta subseção. Em média esse erro é de 5%. Assim, o erro médio total do sistema híbrido é de 9%.

A taxa de reconhecimento considerando o limite superior do IC foi de 93,2%. Com esse resultado a arquitetura proposta neste trabalho está entre as arquiteturas que apresentam os melhores resultados.

6.5 Comparação dos Resultados

A Tabela 6.13 compara o número de espécies de plantas, o algoritmo de classificação usado e a taxa de acerto obtida por cada um dos trabalhos referenciados da literatura e os proposto nesta dissertação.

Tabela 6.13: *Comparação entre os algoritmos de classificação referenciados e proposto neste trabalho.*

Trabalho	Número de espécies	Algoritmo de reconhecimento	Taxa de acerto
Wu <i>et al.</i> [56]	32	PNN	90,3%
Wang <i>et al.</i> [53]	20	Classificador hiperesférico	92%
Uluturk <i>et al.</i> [51]	32	PNN	92%
Gu <i>et al.</i> [20]	20	1-NN	93,1%
Gu <i>et al.</i> [20]	20	k-NN	85,4%
Gu <i>et al.</i> [20]	20	PNN	91,1%
Kadir <i>et al.</i> [28]	32	PNN	93,7%
Valliammal <i>et al.</i> [52]	-	AG-SMV	94%
AG	32	AG	73%
RNA	32	RNA	75%
Sistema híbrido	32	AG e RNA <i>backpropagation</i>	93,2%

Pode-se observar na Tabela 6.13 que a taxa de acerto obtida no reconhecimento de 10 padrões nos trabalhos do Wu *et al.* [56], do Kadir *et al.* [28] e do sistema híbrido proposto neste trabalho foram de 90,3%, 93,7% e 93,2%, respectivamente. Os demais trabalhos utilizaram métricas diferentes para apresentar seus resultados, como por exemplo, o trabalho de Wang *et al.* [53] utilizou 25 padrões de cada classe para treinar o sistema de reconhecimento e 15 padrões de cada classe para serem reconhecidos, Uluturk *et al.* [51]

apesar de utilizarem o mesmo banco de dados adotado neste trabalho, treinaram a rede PNN com 35 padrões por classe e reconheceram 5 folhas desconhecidas de cada planta, a pesquisa de Gu *et al.* [20] não especificou o número de padrões treinados e reconhecidos e o trabalho de Valliammal *et al.* não informou a quantidade de classes de plantas que eles trabalharam. Valliammal *et al.* apenas disseram que utilizaram 70% dos padrões disponíveis para treinar o classificador SMV e reconheceram os demais 30%.

6.6 Conclusões

Neste capítulo foi apresentado o banco de dados de folhas utilizado neste trabalho, a interface do *software* desenvolvido para realizar o reconhecimento das plantas por meio de suas folhas, os tempos de processamento e os resultados obtidos nas simulações utilizando como classificador o algoritmo genético, a rede neural artificial e o sistema híbrido proposto. Em seguida, as taxas de acerto desses três métodos foram comparadas com as taxas de acerto dos métodos existentes na literatura.

Conclui-se desta comparação que o sistema híbrido proposto por esta dissertação apresenta desempenho satisfatório, pois a taxa de reconhecimento obtida está entre os melhores resultados obtidos por outros pesquisadores.

No próximo capítulo são realizadas as conclusões e as contribuições desta dissertação e as propostas de trabalhos futuros.

Capítulo 7

Conclusões, Contribuições Deste Trabalho e Trabalhos Futuros

7.1 Introdução

Este capítulo apresenta as conclusões, as contribuições e os trabalhos futuros que poderão ser realizados a partir desta dissertação.

7.2 Conclusões

Este trabalho permitiu à implementação de um algoritmo que pode ser utilizado para reconhecer plantas a partir de suas folhas. O desempenho desse algoritmo pôde ser avaliado por meio de testes, cujos resultados foram apresentados no Capítulo 6.

O sistema híbrido de reconhecimento proposto apresentou um desempenho considerado satisfatório, pois a taxa de reconhecimento obtida por ele está entre as melhores taxas obtidas na literatura, e com a proposta de pré-selecionar as classes de maior aptidão com a folha a ser reconhecida o universo de busca pela resposta diminuiu e o tempo de treinamento da rede neural artificial treinada pelo algoritmo *backpropagation* também diminuiu substancialmente quando comparado com o tempo de treinamento da RNA considerando as 32 classes do banco de dados de folhas.

Desde os primeiros ensaios realizados neste trabalho utilizando poucos padrões de treinamentos e muitos padrões a serem reconhecidos o sistema híbrido apresentou resultados melhores do que as arquiteturas tradicionais (AG e RNA), mostrando que mesmo sem uma grande quantidade de padrões de treinamento a pré-seleção proposta possibilita que o reconhecimento usando a RNA seja mais eficiente do que os demais métodos, mas, evidentemente, quanto maior é o número de padrões de treinamento disponíveis melhor é a capacidade do sistema de pré-selecionar as classes, e, consequentemente, de reconhe-

cer corretamente o padrão a ser reconhecido. No último ensaio realizado, em que foram reconhecidos 10 padrões por classe a taxa de acerto obtida com o sistema híbrido chegou a 93,2% com um tempo médio de processamento de 4,5 minutos para reconhecer cada padrão.

7.3 Contribuições Deste Trabalho

Para diversas áreas do conhecimento a busca por técnicas de uso geral de reconhecimento de padrões é de fundamental importância. Biólogos precisam de métodos de classificação de espécimes; programadores buscam formas eficazes de aprendizado para realizar tomadas de decisões baseadas em dados, e profissionais de diversas outras áreas, em que a proximidade com o domínio do reconhecimento de padrões (RP) não é tão grande, já passaram por situações cuja solução por meio da aplicação de técnicas de RP é adequada ou necessária.

Portanto, a contribuição deste trabalho é a apresentação de uma nova arquitetura de reconhecimento de padrões que combina as vantagem do algoritmo genético (baixo tempo de processamento e sua alta capacidade de seleção de classes), com a rede neural artificial treinada pelo algoritmo *backpropagation*.

O sistema de reconhecimento proposto nesta dissertação é aplicado na identificação de plantas por meio de suas folhas. As plantas desenvolvem uma importante função no meio ambiente e sem elas não seria possível existir vida na Terra. Como muitas espécies estão sob risco iminente de extinção o objetivo do *software* desenvolvido é auxiliar os biólogos ou outras pessoas interessadas nesta área de estudos a identificar as espécies promovendo ações para proteger a flora de sua região. O *software* implementado é fácil de ser utilizado e necessita apenas de um computador e uma câmera digital ou um *scanner* para funcionar.

7.4 Trabalhos Futuros

O desenvolvimento de um sistema de reconhecimento utilizando um banco de dados de plantas nacionais para avaliar a capacidade de generalização da arquitetura híbrida e para ser um *software* de apoio de baixo custo para os pesquisadores seria a continuação

mais imediata deste trabalho.

A rede neural utilizada no sistema híbrido para classificar a folha após a pré-seleção realizada pelo AG possui um treinamento considerado muito lento. Portanto, pode-se substituir a RNA *backpropagation* por outros classificadores, tais como, a rede *Learning Vector Quantization* (LVQ) [16] ou a Rede Neural Probabilística (PNN) [15]. A escolha de uma linguagem de programação mais rápida também poderia reduzir o tempo de processamento do sistema.

Com o objetivo de melhorar a capacidade de pré-seleção das classes executada pelo AG é necessário realizar testes utilizando outras características que também podem ser extraídas das folhas, tais como, a análise da forma da folha utilizando o *Active Shape Models* (ASM) [8] ou o *Active Appearance Models* (AAM) [49] que funciona de maneira semelhante ao ASM, porém considera não somente a forma do objeto, mas também a sua aparência, analisando características, tais como, a textura.

Para reduzir as redundâncias das características extraídas, e, consequentemente, diminuir o volume das informações processadas pelo sistema híbrido, podem ser implementadas técnicas como a análise da componente principal (*Principal Component Analysis* - PCA) [19].

O desenvolvimento de um aplicativo de reconhecimento de plantas que utiliza o sistema híbrido para *smartphones* e *tablets* também é proposto com a finalidade de utilizar os recursos computacionais de *hardware* disponíveis nestes aparelhos. A portabilidade destes aparelhos permitem o seu deslocamento no campo possibilitando aos pesquisadores mais precisão no reconhecimento e catalogação de novos tipos de plantas.

7.5 Considerações Finais deste Capítulo

Este capítulo apresentou as conclusões, as contribuições e os trabalhos futuros que poderão ser realizados a partir desta dissertação.

Referências Bibliográficas

- [1] Computação evolucionária. <http://computacaoevolucionaria.blogspot.com.br>, Abr. 2012.
- [2] Estudo das folhas. <http://equipe4.8m.com/2bim/folha.htm>, Fev. 2012.
- [3] Flavia, a leaf recognition algorithm for plant classification using pnn. <http://flavia.sourceforge.net/>, Jan. 2012.
- [4] BARCELOS, E. Z. Recuperação de imagens por conteúdo: Uma abordagem multidimensional de modelagem de similaridade e realimentação de relevância. Dissertação de Mestrado, Universidade Federal de Uberlândia, 2009.
- [5] BEZERRA, M. E. R. Métodos baseados na regra do vizinho mais próximo para reconhecimento de imagens, Janeiro 2006.
- [6] BLICKLE, T. & THIELE, L. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, vol. 4, no., 1997.
- [7] CARNEIRO, M. B. P. Reconhecimento de íris utilizando algoritmos genéticos e amostragem não uniforme. Dissertação de Mestrado, Universidade Federal de Uberlândia, 2010.
- [8] COOTES, T. F.; TAYLOR, C. J.; COOPER, D. H. & GRAHAM, J. Active shape models - their training and application. *Comput. Vis. Image Underst.*, vol. 61, no. 1, Janeiro 1995.
- [9] CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, 1989.

- [10] DARWIN, C. *On the Origin of the Species by Means of Natural Selection: Or, The Preservation of Favoured Races in the Struggle for Life.* John Murray, 1859.
- [11] DE CASTRO, A. A. M. & DO PRADO, P. P. L. Algoritmos para reconhecimento de padrões. *Revista Ciências Exatas*, vol. 8, no., 2002.
- [12] DE CASTRO, N. M. Morfologia vegetal - folha. <http://www.anatomiavegetal.ib.ufu.br/recursosditadicos.htm>, Mai 2012.
- [13] DE CASTRO SILVA, L. N. Análise e síntese de estratégias de aprendizado para redes neurais. Dissertação de Mestrado, UNICAMP, 1998.
- [14] DE PÁDUA BRAGA, A.; DE LEON F. DE CARVALLHO, A. P. & LUDEMRIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações.* LTC, 2000.
- [15] DUDA, R. O.; HART, P. E. & STORK, D. G. *Pattern Classi.* Wiley-Interscience, 2000.
- [16] FAUSETT, L. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications.* Prentice-Hall international editions. Prentice-Hall, 1994.
- [17] FENGHE, J. & YAPING, F. Hybrid genetic algorithm for vehicle routing problem with time windows. In: *Management and Service Science (MASS), 2010 International Conference on*, aug. 2010, p. 1 –4.
- [18] GONZALEZ, RAFAEL C.; WOODS, R. C. *Processamento Digital de Imagens.* Pearson Prentice Hall, 2010.
- [19] GORBAN, A. N.; KGL, B.; WUNSCH, D. C. & ZINOVYEV, A. *Principal Manifolds for Data Visualization and Dimension Reduction.* Springer Publishing Company, Incorporated, 1st ed., 2007.
- [20] GU, X.; DU, J.-X. & WANG, X.-F. Leaf recognition based on the combination of wavelet transform and gaussian interpolation. In: *Proceedings of the 2005 international conference on Advances in Intelligent Computing - Volume Part I*, Berlin, Heidelberg, 2005, ICIC'05, Springer-Verlag, p. 253–262.

- [21] HAUPT, R. & HAUPT, S. *Practical genetic algorithms*. Wiley-Interscience, 2004.
- [22] HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall International Editions Series. Prentice Hall, 1999.
- [23] HAYKIN, S. *Redes Neurais - 2ed*. BOOKMAN COMPANHIA ED, 2001.
- [24] HECHT-NIELSEN. Neurocomputing. *Addison-Wesley*, 1990.
- [25] HOLAND, J. H. Adaptation in natural and artificial systems. *University of Michigan Press*, 1975.
- [26] JACKEL, L.; STENARD, C.; BAIRD, H.; BOSER, B.; BROMLEY, J.; BURGES, C.; DENKER, J.; GRAF, H.; HENDERSON, D.; HOWARD, R.; HUBBARD, W.; LECUN, Y.; MATAN, O.; PEDNAULT, E.; SATTERFIELD, W.; SACKINGER, E. & THOMPSON, T. A neural network approach to handprint character recognition. In: *Compcon Spring '91. Digest of Papers*, feb-1 mar 1991, p. 472 –475.
- [27] JAIN, A.; DUIN, R. & MAO, J. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, jan 2000.
- [28] KADIR, A.; NUGROHO, L. E.; SUSANTO, A. & SANTOSA, P. I. Leaf classification using shape, color, and texture features. *International Journal of Computer Trends and Technology*, 2011.
- [29] LEAFSNAP. Leafsnap: An Electronic Field Guide. <http://leafsnap.com>, Out. 2012.
- [30] LEE, Y.; OH, S.-H. & KIM, M. The effect of initial weights on premature saturation in back-propagation learning. In: *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, jul 1991, vol. i, p. 765 –770 vol.1.
- [31] LINDEN, R. *Algoritmos Genéticos*. BRASPORT, 2008.
- [32] MATHWORKS. Documentation center. <http://www.mathworks.com/help/images/ref/strel.html>, Jun. 2012.

- [33] McCULLOCH, W. S. & PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, vol. 5, no., 1943.
- [34] MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Artificial intelligence. Springer, 1998.
- [35] MINSKY, M. & PAPERT, S. Perceptrons: an introduction to computational geometry. *Massachusetts MIT press*, 1943.
- [36] MITCHELL, M. *An introduction to genetic algorithms*. 1996.
- [37] MORAES, E. C. *Reconhecimento de Padrões e Redes Neurais Artificiais em Predição de Estruturas Secundárias de Proteínas*. Tese de Doutorado, Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro., 2010.
- [38] NG, G. W. *Application of neural networks to adaptive control of nonlinear systems*. Research Studies Press (Taunton, Somerset, England and New York), 1997.
- [39] NGUYEN, D. & WIDROW, B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: *Proceedings of the international joint conference on neural networks*, 1990, vol. 3, Washington, p. 21–26.
- [40] ONISHI, S.; TANAKA, A.; HASEGAWA, H.; KINOSHITA, K. & KISHIDA, S. Construction of individual identification system using voice in three-layered neural networks. In: *Intelligent Signal Processing and Communication Systems, 2009. ISPACS 2009. International Symposium on*, jan. 2009, p. 635 –637.
- [41] QIU, H.; ZHOU, W. & WANG, H. A genetic algorithm-based approach to flexible job-shop scheduling problem. In: *Natural Computation, 2009. ICNC '09. Fifth International Conference on*, aug. 2009, vol. 4, p. 81 –85.
- [42] RAITTZ, R. T. Fan 2002: Um modelo neuro-fuzzy para reconhecimento de padrões. Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2002.
- [43] RIEDMILLER, M. Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms, 1994.

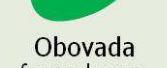
- [44] ROSENBLATT, F. The perceptrons - a perceiving and recognizing automaton. *Cornell Aeronautical Laboratory*, 1957.
- [45] ROSS, T. *Fuzzy Logic with Engineering Applications*. Wiley, 2009.
- [46] SCHALKOFF, R. *Pattern recognition: statistical, structural, and neural approaches*. J. Wiley, 1992.
- [47] SCHWARTZ, H. P. W. R. *Análise de imagens digitais*. Thomson Learning, 2008.
- [48] SONKA, M.; HLAVAC, V. & BOYLE, R. *Image Processing, Analysis, and Machine Vision*, vol. 2. 1998.
- [49] STEGMANN, M. B. Active appearance models: Theory, extensions and cases. Dissertação de Mestrado, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, aug 2000.
- [50] THEODORIDIS, S. & KOUTROUMBAS, K. Pattern recognition. *Academic, San Diego*, 2003.
- [51] ULUTURK, C. & UGUR, A. Recognition of leaves based on morphological features derived from two half-regions. In: *Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on*, july 2012, p. 1 –4.
- [52] VALLIAMMAL, N. & GEETHALAKSHMI, S. N. Efficient feature fusion, selection and classification technique for plant leaf image retrieval system. In: *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, New York, NY, USA, 2012, CCSEIT '12, ACM, p. 132–137.
- [53] WANG, X.-F.; DU, J.-X. & ZHANG, G.-J. Recognition of leaf images based on shape features using a hypersphere classifier. In: *Proceedings of the 2005 international conference on Advances in Intelligent Computing - Volume Part I*, Berlin, Heidelberg, 2005, ICIC'05, Springer-Verlag, p. 87–96.

- [54] WASSERMAN, P. *Neural computing: theory and practice*. Van Nostrand Reinhold, 1989.
- [55] WIDROW, B. & HOFF, M. E. Adaptive switching circuits. *Institute of Radio Engineers. Western Electronic Show and Converstion*, 1960.
- [56] WU, S.; BAO, F.; XU, E.; WANG, Y.-X.; CHANG, Y.-F. & XIANG, Q.-L. A leaf recognition algorithm for plant classification using probabilistic neural network. In: *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, dec. 2007, p. 11 –16.
- [57] YOUYI, J. & XIAO, L. A method for face recognition based on wavelet neural network. In: *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*, dec. 2010, vol. 3, p. 133 –136.

Apêndice A

Forma das folhas

FORMA

			
Aciular forma de agulha	Falciforme forma de foice	Orbicular circular	Rombóide forma de losango
			
Acuminada afilando em ponta longa	Flabelada forma de leque	Ovada forma de ovo, larga na base	Roseta folículos em anéis circulares apertados
			
Alternada folículos dispostos alternadamente	Hastada ou alabardina triangular com lobos basais	Espalmada como uma mão aberta	Espiratulada forma de colher
			
Aristada ponta em forma de espinho	Lanceolada pontas nos dois extremos	Pediforme espalmada, lobos laterais divididos	Sagitada pontuda, base com farpas
			
Bipinulada folículos pinulados	Linear margens paralelas, alongada	Peltada haste inserida ao centro	Subulada ponta afilada, forma de furador
			
Cordiforme forma de coração, haste na fenda	Lobada margens profundamente recortadas	Perfoliada haste parece perfurar a folha	Ternifoliada folículos em três
			
Cuneiforme forma de cunha, base aguda	Obcordata forma de coração, haste na ponta	Pinulada ímpar folículos em fiadas, um na ponta	Tripinulada folículos também bipinulados
			
Deltóide triangular	Obovada forma de ovo, estreita a base	Pinulada par folículos em fiadas, dois na ponta	Truncada ápice em esquadria
			
Digitiforme com lobos em forma de dedos	Obtusa pontas rombas	Pinatilobada lobos profundos e opostos	Unifoliar uma única folha
			
Elíptica forma oval, ponta pequena ou inexistente	Oposta folículos em pares	Reniforme forma de rim	Espiralada anéis de três ou mais folículos

Apêndice B

Formulação das equações do *backpropagation*

Neste apêndice são apresentadas as demonstrações das fórmulas utilizadas pelo algoritmo *backpropagation* para atualizar os pesos da rede MLP.

Embora o erro total E é definido pela soma dos erros dos nodos de saída para todos os padrões, é considerado, sem perda de generalidade, que a minimização do erro para cada padrão individualmente leva à minimização do erro total. Assim, o erro é definido pela Equação (B.1)

$$E = \frac{1}{2} \sum_{j=1}^k (d_j - y_j)^2 \quad (\text{B.1})$$

onde:

- d_j - saída calculada pela rede; e
- y_j - resposta previamente conhecida.

A regra delta sugere que a variação dos pesos (Δw_{ji}) seja definida de acordo com o gradiente descendente do erro com relação ao peso, ou seja, que a variação do peso para um determinado padrão é definida pela Equação (B.2).

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}} \quad (\text{B.2})$$

Agora é necessário definir como cada um dos pesos de cada nodo da rede deve ser ajustado de forma a diminuir o erro total gerado pela rede. Utilizando a regra da cadeia, tem-se que:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} \quad (B.3)$$

onde:

- $net_j = \sum_{i=1}^n x_i w_{ji}$.

A segunda derivada, $\frac{\partial net_j}{\partial w_{ji}}$ é encontrada utilizando a Equação (B.4).

$$\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial \sum_{l=1}^n x_l w_{jl}}{\partial w_{ji}} = x_i \quad (B.4)$$

A primeira derivada, do lado direito da igualdade da Equação (B.3) mede o erro do nodo j . Geralmente, ela é abreviada para δ_j .

$$\delta_j = \frac{\partial E}{\partial net_j} \quad (B.5)$$

O cálculo desta derivada também pode ser definido pela regra da cadeia, como mostrado na Equação (B.6).

$$\delta_j = \frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial net_j} \quad (B.6)$$

A segunda derivada da Equação (B.6), $(\frac{\partial y_j}{\partial net_j})$, é facilmente definida por meio da Equação (B.7).

$$\frac{\partial y_j}{\partial net_j} = \frac{\partial f(net_j)}{\partial net_j} = f'(net_j) \quad (B.7)$$

O cálculo da primeira derivada, $(\frac{\partial E}{\partial y_j})$, que utiliza o erro, depende da camada onde o nodo j se encontra. Se o nodo está na última camada, o seu erro pode ser facilmente definido utilizando a Equação (B.2). Assim, a equação

$$\frac{\partial E}{\partial y_j} = \frac{\partial (\frac{1}{2} \sum_{i=1}^k (d_i - y_i)^2)}{\partial y_j} = (d_j - y_j) \quad (B.8)$$

assume a mesma fórmula da regra delta original. Substituindo os dois termos do lado direito da Equação (B.6), obtém-se a Equação (B.9).

$$\delta_j = (d_j - y_j) f'(net_j) \quad (B.9)$$

Se o nodo j não é um nodo de saída, a regra da cadeia é utilizada para escrever a Equação (B.10).

$$\frac{\partial E}{\partial y_j} = \sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} \frac{\partial \text{net}_l}{\partial y_j} = \sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} \frac{\partial \sum_{i=1}^n y_i w_{il}}{\partial y_j} = \sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} w_{jl} \quad (\text{B.10})$$

onde:

$$\sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} w_{jl} = \sum_{l=1}^M \delta_l w_{jl} \quad (\text{B.11})$$

Substituindo mais uma vez os dois termos do lado direito da Equação (B.6), obtém-se que, para os nodos das camadas intermediárias, o erro é definido pela Equação (B.12).

$$\delta_j = f'(\text{net}_j) \sum_l \delta_l w_{lj} \quad (\text{B.12})$$

Assim, pode-se generalizar a fórmula de ajuste dos pesos mostrada na Equação (B.2), pela forma mostrada na Equação (B.13).

$$\Delta w_{ji} = \alpha \delta_j x_i \quad (\text{B.13})$$

A Equação (B.14) apresenta a formula completa de ajuste dos pesos.

$$w_{ji} (\text{novo}) = w_{ji} (\text{antigo}) + \Delta w_{ji} \quad (\text{B.14})$$

Se o nodo é de saída, o erro δ_j é definido pela Equação (B.9), caso contrário, δ_j é obtido pela Equação (B.12) [14].