UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Luciene de Oliveira Marin

Investigações sobre Redes Neurais Artificiais para o Reconhecimento de Faces Humanas na Forma 3D

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de mestre em Ciência da Computação.

Prof. Jorge Muniz Barreto, D. Sc. A.

Antonio Carlos Zimmermann, Dr. Eng.

Florianópolis, Agosto de 2003

Investigações sobre Redes Neurais Artificiais para o Reconhecimento de Faces Humanas na Forma 3D

Luciene de Oliveira Marin

Esta Dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação, área de concentração Inteligência Artificial e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Fernando Alvaro Ostuni Gauthier, Dr.
Coordenador do Curso

Banca Examinadora

Prof. Jorge Muniz Barreto, D. Sc. A.

Antonio Carlos Zimmermann, Dr. Eng.

Prof. Jovelino Falqueto, Dr.

Prof^{a}. Maria Augusta Soares Machado, Dr. Eng.

Prof. Mauro Roisenberg, Dr. Eng.

Queremos saber, o que vão fazer
Com as novas invenções
Queremos notícia mais séria
Sobre a descoberta da antimatéria
E suas implicações
Na emancipação do homem
Das grandes populações
Homens pobres das cidades
Das estepes, dos sertões

Queremos saber, quando vamos ter
Raio laser mais barato
Queremos, de fato, um relato
Retrato mais sério do mistério da luz
Luz do disco-voador
Pra iluminação do homem
Tão carente, sofredor
Tão perdido na distância
Da morada do Senhor

Queremos saber, queremos viver
Confiantes no futuro
Por isso se faz necessário prever
Qual o itinerário da ilusão
A ilusão do poder
Pois se foi permitido ao homem
Tantas coisas conhecer
É melhor que todos saibam
O que pode acontecer

Queremos saber
Queremos saber
Queremos saber
Todos queremos saber

Gilberto Gil

A todos aqueles que tiveram a boa vontade de ler este trabalho, dando suas opiniões e contribuições e também aqueles que terão a curiosidade de saber no que deu tudo isto...

Agradecimentos

Agradeço a Deus a oportunidade de vir pra Floripa fazer o mestrado e por todas as coisas boas e ruins que me aconteceram no decurso do mesmo.

Aos meus pais Antônio Mauro & Dilma pelo amor, carinho e apoio que sempre me dedicaram e também a toda minha família em especial aos meus manos Dulcilene, Estela e Wagner, meu lindo sobrinho José Luiz e meu cunhado Gilberto.

Aos meus amigos de vivência e convivência, que sempre fizeram me sentir em família: Alessandra, Alessandro, Valquíria, Gisele, Letícia, Milena.

Ao meu amigo do peito, irmão, "xodó" :), gauchesco, gaudério Glaucio pela amizade, amor, companheirismo, pelas caminhadas na Beira Mar, muitas gargalhadas, músicas...

Aos meus pais adotivos de Floripa: Roberto & Denise e ao maninho André, pela grande acolhida e afeto que sempre me destinaram.

Ao Prof. Mauro Roisenberg, pela solicitude, pelo prazer e entusiasmo em ensinar as disciplinas de IA, pelas críticas sempre construtivas com relação a este trabalho...

Ao Prof. Jovelino Falqueto, pelo apoio, pelos "ouvidos" e conselhos preciosos, pelas várias observações apontadas...

Ao Prof. Paulo Sérgio da Silva Borges pela maneira apaixonada e filosófica com que discursa sobre a Inteligência Computacional, pelos "banhos" de cultura...

Ao A. C. Zimmermann pelas infinitas conversações e indagações a respeito dos mistérios das redes neurais artificias e seu exemplo de pesquisador... à sua linda família: esposa Rosana, mãe Íris, e filhos Michael e Richard.

Ao Prof. Jorge Muniz Barreto por ter me escolhido para participar da equipe do Projeto Sorface, pela sua orientação com relação a este trabalho, seu apoio e dedicação e a grande carga de conhecimento com que esbanja. À sua esposa Solange, por ser seus outros braços e pernas, não medindo esforços para ajudar nas funções administrativas e outras..., e também à Cláudia.

Aos amigos Cíntia & Rafael, Kathya, Reinaldo, Fátima, Cláudia, Renato, Leonardo, Procópio, Rafael, Maria, Andréa, Flavinho e outros que por falta de papel não citei...

Ao CNPq pelo apoio financeiro.

Sumário

Lista de Figuras **ix**

Lista de Tabelas **xi**

Lista de Siglas **xii**

Publicações **xiii**

Resumo **xiv**

Abstract **xv**

1 Introdução **1**

1.1 Motivação 1

1.2 Objetivos 3

1.2.1 Objetivo geral 3

1.2.2 Objetivos específicos 3

1.3 Reconhecimento de faces: homem $\times$ máquina 4

1.4 História e o estado-da-arte 6

1.5 Métodos de aquisição 8

1.5.1 Sistemas 2D 9

1.5.2 Sistemas 3D 10

1.6 O sistema de aquisição SORFACE 10

1.7 A base de dados SORFACE 11

1.8 Organização deste documento 12

2 Fundamentação Teórica **16**

2.1 Reconhecimento de padrão 16

2.1.1 Introdução 16

2.1.2 O que é reconhecimento de padrões? 16

2.1.3 Técnicas para reconhecimento de padrões 18
2.2 Reconhecimento de faces 56
2.2.1 Introdução 56
2.2.2 Medida de desempenho de um sistema de verificação 57
2.2.3 Detecção de faces 60
2.2.4 Reconhecimento 61
2.2.5 RNAs e o reconhecimento de faces 65

3 Justificativas para a metodologia empregada 67
3.1 Introdução 67
3.2 Conexionismo versus reconhecimento de faces 3D 67
3.2.1 Complexidade de RNAs $\times$ reconhecimento de faces 68
3.2.2 O Perceptron 69
3.2.3 O Adaline 74
3.3 Computação evolucionária (IA evolucionária) 76
3.3.1 Estratégias evolucionárias 77
3.3.2 Programação Evolucionária 78
3.3.3 Algoritmos genéticos 80
3.4 Computação evolucionária & RNAs 81
3.4.1 Evolução simultânea da arquitetura e conexão de pesos 83

4 Implementações Prévias 86
4.1 Introdução 86
4.2 Ensaios preliminares 86
4.3 A quantidade de exemplos por pessoa 90
4.4 Comparação das velocidades de convergência 92

5 Resultados 99
5.1 Introdução 99
5.2 PE para evolução de arquitetura 99
5.2.1 A base de dados para o experimento 100
5.2.2 O experimento 101
5.2.3 O cálculo da aptidão 102
5.2.4 “Herança de conhecimento” na evolução de RNA com PE 103
5.3 Problema linearmente separável 104
5.3.1 Análise de desempenho 105

6 Conclusão **107**

6.1 Considerações finais 107

6.2 Trabalhos futuros 109

Referências Bibliográficas **110**

Lista de Figuras

1.1 A imagem à esquerda é a visão de uma pessoa normal e a imagem à direita é a visão de uma pessoa com *prosopagnosia*. 5
1.2 Esquema do método de aquisição de faces utilizado neste trabalho. 11
1.3 Exemplo prático de uma aquisição de face. 12
1.4 Exemplos das expressões faciais adquiridas. 13
1.5 Representação no espaço 3D das alturas de uma face. 14
1.6 Um exemplo de representação no espaço 3D das alturas da face com níveis de ruído. 14

2.1 Esquema para se obter a correlação de $f(x, y)$ e $w(x, y)$ no ponto (s, t). Adaptada de: [32]. 20
2.2 Blocos funcionais para o reconhecimento de padrão na abordagem estatística. Adaptada de: [40]. 22
2.3 As várias abordagens estatísticas para o reconhecimento de padrão. Adaptada de: [40]. 24
2.4 Redes auto-associativas para encontrar um subespaço tri-dimensional. (a) linear, (b) não-linear (nem todas as conexões são mostradas). Adaptada de: [40]. 30
2.5 (a) Segmentos de linha de tamanho fixo. (b) Exemplos de padrões de uma classe. Adaptada de: [15]. 34
2.6 Árvore de decisão nebulosa. Adaptada de: [66]. 41
2.7 Diagrama de transição de estados de um modelo de reconhecimento nebuloso generalizado. Adaptada de: [66]. 42
2.8 Diferentes tipos de aglomerados. Adaptada de: [66]. 44
2.9 Nível semântico de uma rede neuro-nebulosa. Adaptada de [13]. 47
2.10 Uma rede função base radial. Adaptada de [45]. 48
2.11 Raciocínio nebuloso. Adaptada de [45]. 50
2.12 Representação da rede adaptativa. Adaptada de [45]. 51

2.13 Uma típica distribuição das populações de ovelhas e lobos. 58
2.14 FAR e FRR versus Limiar. 59
2.15 FAR versus FRR. 60
2.16 Imagens de faces para o uso da técnica NLC. Adaptada de [79]. 63
2.17 Técnica de imagem modelo para reconhecimento HMM. Adaptada de [64]. 65

3.1 O descriminador linear separa o espaço em duas regiões A e B. Adaptada de [53]. 71
3.2 Redes Perceptron (esquerda) e Adaline (direita). Adaptada de [24]. 75
3.3 Rede Madaline. Adaptada de [24]. 75
3.4 A estrutura principal da EPNet. Adaptada de [93]. 84

4.1 Faces com projeções de franjas e uma forma de face 3D. 87
4.2 Seleção da arquitetura de rede ótima. 88
4.3 Treinamento da rede neural MLP ótima 89
4.4 Um exemplo de ocorrência de sobre-treinamento. 90
4.5 Tentativa de simular o sobre-treinamento. 91
4.6 Distribuição normal das duas classes de exemplo. 92
4.7 Distribuições FAR e FRR versus limiar. 93
4.8 Gráficos de alguns treinamentos mediante a quantidade de faces *ovelha*. . . 94
4.9 Erro no conjunto de teste $\times$ faces *ovelha* no conjunto de treinamento. . . . 95
4.10 Pessoas utilizadas para o reconhecimento. 96
4.11 Desempenhos dos algoritmos de retropropagação e suas variantes. 98

5.1 Exemplo de classes linearmente e não linearmente separáveis no espaço 2D. 105

Lista de Tabelas

1.1 Exemplos de expressões faciais. 15

2.1 Exemplos de aplicações para o RP. Adaptada de: [40]. 17
2.2 Ligações entre métodos estatísticos e neurais para o RP. Adaptada de: [78]. 37
2.3 Comparando abordagens estatística, sintática e neural de RP. Adaptada de: [78]. . 38
2.4 Exemplos de biometrias 57

3.1 Notação utilizada na descrição do algoritmo evolutivo genérico. Adaptada de [27]. 77

4.1 Ensaio de expressões faciais para o conjunto de treinamento 97
4.2 Ensaio de expressões faciais para o conjunto de validação 98

5.1 Taxa de acertos, para o conjunto de teste, da rede Adaline onde η é a taxa de aprendizado e ϵ o erro mínimo para o conjunto de treinamento. 106
5.2 Taxa de acertos, para o conjunto de teste, da rede Perceptron onde η é a taxa de aprendizado e ϵ o erro mínimo para o conjunto de treinamento. 106

Lista de Siglas

3D	Tridimensional
ADALINE	*Adaptive linear element*
AG	Algoritmo genético
ATM	*Asynchronous transfer mode*
DARPA	*Defense advanced research projects agency*
EPNet	Programação evolucionária para evolução de redes neurais artificiais
FAR	*False acceptance rate*
FERET	*Face recognition technology*
FRR	*False rejection rate*
HMM	*Hidden Markov model*
IAC	Inteligência artificial conexionista
IAE	Inteligência artificial evolucionária
IEEE	*Institute of electrical and electronics engineers*
LMS	Erro Médio Quadrático Mínimo
LQV	*Learning vector quantization*
MBP	*Modified backpropagation*
MIT	*Institute of Technology Media Lab*
MLP	*Multi-layer perceptron*
NN	*Nearest neighbor*
PCA	*Principal components analysis*
PE	Programação evolucionária
RBF	*Radial basis function*
RBFN	*Radial basis function network*
RNA	Rede neural artificial
RP	Reconhecimento de padrões.
SOM	*Self-Organizing Map*
SORFACE	Sistema óptico de reconhecimento de faces humanas
TRF	Tecnologia de reconhecimento de face
UMD	*University of Maryland*
USC	*University of Southern California*

Publicações

1. MARIN, Luciene de Oliveira; ENCINAS, Leonardo Soliz; ZIMMERMANN, Antônio Carlos; BARRETO, Jorge Muniz. TRABALHO ACEITO: 3D Human Face Recognition as a Linearly Separated Problem. In: IASTED INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND APPLICATIONS (AIA 2004), 2004, Innsbruck, Austria. 2004.

2. MARIN, Luciene de Oliveira; ENCINAS, Leonardo Soliz; STEIN, Procópio Silveira; ZIMMERMANN, Antônio Carlos; BARRETO, Jorge Muniz. Separação linear de faces humanas representadas na forma 3D. In: I2TS'2003 - 2ND INTERNATIONAL INFORMATION AND TELECOMMUNICATION TECHNOLOGIES SYMPOSIUM, 2003, Florianópolis. I2TS'2003 - 2nd International Information and Telecommunication Technologies Symposium. Florianópolis-SC: Azzedine Boukerche, Mirela Sechi Moretti Annoni Notare, 2003. p. 23-23.

3. MARIN, Luciene de Oliveira; ZIMMERMANN, Antônio Carlos; BARRETO, Jorge Muniz. Plausibilidade biológica e aceleração do aprendizado em um sistema de reconhecimento de faces. In: SCPDI 2002 - II SIMPÓSIO CATARINENSE DE PROCESSAMENTO DIGITAL DE IMAGENS, 2002, Florianópolis. 2002.

4. MARIN, Luciene de Oliveira; ZIMMERMANN, Antônio Carlos; BARRETO, Jorge Muniz. Inteligência artificial para o reconhecimento da forma 3D de faces humanas. In: 3A SEPEX - SEMANA DE ENSINO, PESQUISA E EXTENSÃO DA UNIVERSIDADE FEDERAL DE SANTA CATARINA, 2003, Florianópolis, SC. 2003.

Resumo

Reconhecimento de faces humanas é uma área de grande interesse no mundo científico. A maioria das tecnologias desenvolvidas utiliza imagens com informação 2D. Este trabalho contou com um método inédito de processamento para a obtenção da forma 3D de uma face. Por meio dele se produziu várias bases de dados, com diferentes resoluções e níveis de ruído aceitáveis. Elas foram utilizadas na construção de um sistema de reconhecimento de face baseado em redes neurais artificiais. A vantagem de se utilizar a forma 3D da face está na exclusão de problemas ocasionados pela iluminação e desalinhamento.

Um sistema de reconhecimento de faces é um problema de verificação e/ou classificação de padrões, por isso, a abordagem de redes neurais artificiais foi escolhida devido ao seu sucesso de atuação nestes tipos de problema. As justificativas desta escolha foram apresentadas.

Várias técnicas de reconhecimento de padrões foram estudadas, e para um melhor entendimento do processo de reconhecimento de faces foram vistos seu estado-de-arte e como se desenvolveram seus primeiros sistemas.

Neste trabalho empregou-se uma metodologia de programação evolucionária, outro paradigma da inteligência artificial, para a definição da arquitetura de rede neural ótima para o problema de reconhecimento de face. Baseado nestes estudos se chegou a arquiteturas de redes simples porque os padrões puderam ser classificados com grande facilidade. Isto não era esperado no início deste trabalho, por isso esta pesquisa começou testando redes com arquiteturas e algoritmos de aprendizado mais complexos.

Com base nos resultados dos experimentos realizados, nos quais as faces foram facilmente classificadas por redes Adaline e Perceptron conclui-se que os padrões utilizados neste trabalho são linearmente separáveis.

Abstract

Human face recognition is an interesting area in the scientific world. The majority of the developed technologies use 2D images information. In this work is used an inedited processing method to obtain the 3D form of the face. Through it was produced some databases, with different resolutions and acceptable levels of noise. They had been used in the construction of a face recognition system based on artificial neural networks. The advantage of the 3D forms face used here is the exclusion of the problems caused for different illumination conditions and misalignment.

A face recognition system is a pattern verification and/or classification problem, therefore, the artificial neural network approach was chosen due its success to solve these problems. The justifications for this choice had been presented.

Several pattern recognition techniques had been studied and for better agreement of recognition faces process had been seen its state-of-art and as they had developed its first systems.

In this work was used the methodology of evolutionary programming, another artificial intelligence paradigm, for the definition of the optimum neural network architecture to the face recognition problem. Based in it were reached simple networks architectures because the patterns could be classified with great easiness. This was not waited in the beginning of this work therefore this research begin tested networks with more complex architectures and algorithms of learning.

Because the results of the carried through experiments had been classified by Adaline and Perceptron networks, was possible to see that face's patterns treated in this work are linearly separable.

Capítulo 1

Introdução

1.1 Motivação

A face humana é uma imagem fascinante, serve de infinita inspiração a artistas há milhares de anos. Uma das primeiras e mais importantes habilidades humanas é a capacidade de reconhecer faces, pois bebês demonstram identificar a face de suas mães com apenas meia hora de nascimento [35].

A face de uma pessoa é o principal elemento que faz com que os outros indivíduos a reconheça, ou seja, ao visualizar a face de alguém é desencadeado um processo de identificação natural.

Com as mais diversas tecnologias da vida moderna, tais como câmeras de vigilância, terminais de auto-atendimento bancário, comércio eletrônico entre outras, torna-se cada vez mais necessária a construção de métodos seguros para se comprovar a identidade de alguém. Meios de identificação, tais como, carteira de identidade (uma das formas mais tradicionais de identificação), cartões magnéticos e senhas têm demonstrado o quanto são vulneráveis a roubos e clonagens, por exemplo.

A Biometria trabalha para que máquinas sejam capazes de capturar características individuais de uma pessoa, de forma a identificá-la, sem confundi-la com seus semelhantes, uma vez que não existem pessoas idênticas, ainda que univitelinas. Um dos objetivos da Biometria é a construção de sistemas que reconheçam padrões biológicos e estabeleçam a autenticidade a partir de uma característica fisiológica ou comportamental possuída de um indivíduo [69].

Dentre as técnicas de reconhecimento biométrico utilizadas atualmente, as mais precisas são as baseadas em imagem da íris [69]. A desvantagem está em seu caráter invasivo, pois é necessário que o usuário permaneça parado em uma posição definida e com os olhos abertos,

enquanto o *scanner* de íris ou uma câmera captura a imagem. Por isso o interesse pelo reconhecimento de face permaneceu, devido à sua natureza não invasiva e por ser um método básico de identificação de pessoas [71].

Para complementar à biometria a área de reconhecimento de padrões (RP) estuda como máquinas podem observar o ambiente, aprender e distinguir padrões de interesse do que está sendo visto, para a partir disto verificar e justificar decisões baseadas nas categorias dos padrões observados [40]. Segundo Pentland [71], muito do sucesso de sistemas de reconhecimento de face vem da combinação bem estabelecida de técnicas de RP com um sofisticado entendimento do processo de geração de imagem, que produz métodos capazes de capturar regularidades que são peculiares às pessoas, tais como cor de pele, geometria facial, entre outras.

O reconhecimento automático de faces a partir de imagens estáticas e imagens de vídeo vem emergindo como uma área ativa de pesquisa no domínio de RP. A Tecnologia de Reconhecimento de Face - TRF tem numerosas aplicações comerciais e de imposição de lei. Estas aplicações vão desde a comparação estática de imagens, de formato compatível com passaportes, cartões de crédito, foto da carteira de identidade, carteira de habilitação e foto de arquivo policial, à comparação em tempo-real de imagens de vigilância de vídeo. Tais situações apresentam diferentes dificuldades em termos de requerimentos de processamento.

A área de segurança é a que mais tem se favorecido com o desenvolvimento de sistemas de controle de acesso a locais restritos, estações de trabalho, monitoramento, sistemas de pagamento e autenticação biométrica baseadas em reconhecimento de faces. Como exemplos temos [8], [55], [80].

O interesse pela área de segurança já havia sido despertado na autora desta dissertação desde o seu trabalho de conclusão de curso na graduação. Com a existência do projeto de pesquisa Sorface (Reconhecimento de Faces Humanas Através de Técnicas de Inteligência Artificial Aplicadas à Formas 3D), financiado pelo CNPq e então coordenado por seu orientador de mestrado, surgiu a oportunidade de trabalho neste assunto. A equipe do projeto produziu até o momento os seguintes artigos científicos: [101] e [100], demonstrando o desenvolvimento de um método inovador de aquisição da geometria de face. Este método gerou as bases de dados que foram utilizadas neste trabalho.

Com esta pesquisa acredita-se que será dado um passo na direção de evitar tragédias como a do atentado terrorista de 11 de setembro de 2001, nos EUA, bem como facilitar o acesso seguro a lugares públicos como bancos e aeroportos, auxiliar a buscas de terroristas e traficantes, dentre outros.

Numa primeira etapa, o projeto Sorface propõe a construção de um sistema de reconhecimento baseado em verificação que procederá da seguinte maneira: o indivíduo declarará

sua identidade e a mesma será certificada por uma rede neural artificial (RNA) que receberá como entrada sua imagem facial, adquirida em tempo real, através de uma câmera. O principal objetivo deste trabalho de dissertação é encontrar a RNA mais adequada a este problema. Dada esta proposta, um banco demonstrou muito interesse no produto final que o projeto irá produzir. O projeto SORFACE também propõe sistemas que atuarão em outros cenários, tais como, bancos, aeroportos, trânsito e hospitais.

Todos os experimentos realizados se utilizaram da base de dados de faces construída através dos trabalhos de Zimmermann [102], cujo método de extração e pré-processamento consistem de técnicas inovadoras para a obtenção de informações da forma da face. Com o emprego de faces 3D espera-se que o sistema desenvolvido apresente vantagens com relação a técnicas baseadas somente em informações bidimensionais. Tendo em vista que a forma geométrica da face contém muito mais informação e com isto proporciona uma maior precisão ao sistema de reconhecimento, as possibilidades de fraude, com relação aos sistemas que se utilizam de informação 2D, diminuem.

1.2 Objetivos

1.2.1 Objetivo geral

Estudar maneiras de unir soluções baseadas nas abordagens conexionista e evolucionária da Inteligência Artificial para serem aplicadas ao problema de reconhecimento da forma 3D de faces humanas. Desta maneira, pretende-se chegar a soluções ótimas, ou seja, soluções mais econômicas (com relação à complexidade do classificador) e de elevado desempenho (com relação ao reconhecimento).

1.2.2 Objetivos específicos

Para se chegar ao objetivo geral foi necessária a execução dos seguintes objetivos específicos:

- Estudar as técnicas mais usuais para RP;
- Entender como se dá o processo dos sistemas de reconhecimento automático de faces já existentes;
- Identificar as principais dificuldades encontradas por técnicas tradicionais;
- Examinar diferentes arquiteturas de redes neurais e diferentes algoritmos de aprendizado;

- Estudar e desenvolver uma aplicação que utilize técnicas de Programação Evolucionária, para determinação da arquitetura ótima de uma rede neural artificial para o problema de reconhecimento de faces na forma 3D.

1.3 Reconhecimento de faces: homem $\times$ máquina

Segundo Chellappa [19], embora humanos reconheçam faces com relativa facilidade, mesmo em cenas confusas, o reconhecimento através de máquina é uma tarefa difícil de se desempenhar. Há mais de 20 anos, pesquisadores em psicofísica, ciências neurais, engenharia, processamento de imagem, análise e visão computacional têm investigado vários assuntos relacionados ao reconhecimento de face por humanos e máquinas.

Pouco sinergismo existe entre os pesquisadores. A pesquisa do reconhecimento automático de faces trabalha independente dos estudos de psicofísicos e neurofisiologistas, porém muitas de suas descobertas causam importantes impactos na pesquisa de engenheiros.

Os psicofísicos e neurocientistas entram em acordo no estudo dos seguintes assuntos [19]:

- A exclusividade de cada face;
- Se o reconhecimento de face é feito de forma holística e/ou por análise de características locais;
- A análise e o uso de expressões faciais para o reconhecimento;
- Como bebês percebem faces;
- A organização de memória para faces;
- A não habilidade de reconhecimento preciso a faces invertidas;
- A existência de um neurônio “mãe” para o reconhecimento de face;
- A função do hemisfério direito do cérebro na percepção da face;
- E a incapacidade de reconhecer faces devido a condições neurológicas tais como prosopagnosia.

A prosopagnosia é uma doença conhecida como cegueira de face. Um indivíduo com esta deficiência é incapaz de reconhecer pessoas através de suas faces, devido a danificações no centro específico do cérebro responsável pelo reconhecimento de faces. Isto faz com que

o indivíduo não veja com precisão as regiões que mais caracterizam uma face, que são as regiões dos olhos, nariz e boca, ex. Fig.1.1. Porém, ele pode ser capaz de identificar pessoas usando outros centros do cérebro, como o centro geral de reconhecimento de padrões, que a maiorias das pessoas usam para reconhecer objetos [16].

Figura 1.1: A imagem à esquerda é a visão de uma pessoa normal e a imagem à direita é a visão de uma pessoa com *prosopagnosia*.

Relativo a todos estes assuntos de estudo, algumas das hipóteses e teorias propostas para explicar resultados experimentais observados são contraditórias. Muitas têm sido baseadas em pequenos conjuntos de imagens. O problema é devido à não existência de um sistema de avaliação ou *benchmarking* com grandes bases de dados de imagens, de qualidade compatível com aplicações comerciais e de imposição de lei.

Para os engenheiros interessados no projeto de algoritmos e sistemas para reconhecimento de face, os numerosos estudos encontrados na literatura de psicofísica e neurofisiologia são guias úteis. Por exemplo, os projetistas deveriam incluir ambas características globais e locais para representação e reconhecimento de faces. Dentre as características, algumas (cabelos, olhos, boca) são mais significantes ou úteis do que outras (nariz, testa). Estas são informações importantes para imagens frontais de faces, enquanto para imagens laterais e de perfil, o nariz é uma característica importante [19].

Estudos sobre a unicidade da face e caricaturas podem ajudar na descoberta de informações sobre características especiais da face, para serem utilizadas na percepção e reconhecimento.

Investigações a respeito de como humanos reconhecem melhor faces de pessoas de sua própria raça do que de outra, e como bebês reconhecem faces, são muito importantes no projeto de sistemas especialistas para identificação e reconstrução de faces testemunhadas.

Outro assunto, tal como organização de memória, é muito pertinente para o projeto de grandes bases de dados, por exemplo, para construção de álbuns de fotografias policiais. A

utilidade das expressões faciais no reconhecimento de face também é um tópico importante a ser investigado no reconhecimento automático.

Com o passar do tempo, tem havido grande interesse entre os desenvolvedores de algoritmos de visão computacional e projetistas de sistemas em aprender como o processo de reconhecimento visual humano trabalha, para então implantar estes mecanismos em sistemas reais. O paradigma de Marr[1] para visão computacional [58] é um exemplo pioneiro de tais esforços. Hoje eles são mais conscientes da relevância do estudos de psicofísicos e neurofisiologistas. Contudo é necessário prudência em distinguir e aplicar somente as descobertas relevantes do ponto de vista prático e de implementação.

Dentro do projeto SORFACE, o assunto que chama maior atenção está relacionado ao problema da prosopagnosia, por isso uma sugestão seria aumentar a quantidade de informações nas regiões dos olhos, nariz e boca ao serem enviadas à rede neural para o reconhecimento, ou seja, a rede apresentaria um maior número de neurônios nestas regiões. Neste trabalho de dissertação isto não foi implementado, porém experimentos futuros serão feitos com a continuidade das pesquisa realizadas no projeto.

1.4 História e o estado-da-arte

Segundo Chellapa [19], durante o início e meados dos anos 70, foram usadas técnicas típicas de reconhecimento de padrão, onde os atributos eram medidas entre pontos da face ou de um perfil de face. Durante os anos 80, a pesquisa em reconhecimento de face ficou praticamente adormecida. A partir dos anos 90, o interesse em TRF cresceu muito significativamente. Isto se deve a muitas razões: interesse comercial; emergência de classificadores baseados em redes neurais com ênfase em computação de tempo-real e adaptação; viabilidade de hardware; e aumento da necessidade de aplicações relacionadas à vigilância devido ao tráfico de drogas, atividades terroristas, etc.

Chellappa fez uma análise de trinta anos de pesquisa em reconhecimento de faces onde são citados 221 trabalhos. Outra evidência do crescimento desta área de pesquisa é a existência de conferências específicas de reconhecimento de face e gestos [9], [25], [88], [23], bem como a existência de revistas e seções técnicas [47].

[1] Marr definiu visão como *um processo que produz uma descrição, a partir de imagens do mundo externo, que é útil ao visualizador e não repleta de informações irrelevantes*. A palavra "processo" refere-se ao mapeamento das diferentes representações de uma cena, presente no mundo externo, obtidas a partir das matrizes dos valores de intensidade de brilho aos diferentes padrões que descrevem esta cena. As matrizes são obtidas nos primeiros estágios da visão computacional, enquanto que os diferentes padrões são obtidos nos últimos estágios da mesma [62].

O primeiro exemplo mais conhecido de sistema de reconhecimento de face foi desenvolvido por Kohonen [51]. Ele demonstrou que uma simples rede neural poderia realizar reconhecimento de face através de alinhamento e normalização das imagens. O tipo de rede usada computava uma descrição da face por aproximação de auto-vetores da matriz de auto-correlação da imagem facial; estes auto-vetores são os agora conhecidos como "auto-faces" (*eigen-face*). O sistema de Kohonen não foi um sucesso prático, por causa da necessidade de alinhamento preciso e normalização. Nos anos seguintes, muitos pesquisadores tentaram esquemas de reconhecimento de face baseados em bordas, distâncias entre pontos da face, e outras abordagens de redes neurais [71]. Enquanto muitos tiveram sucesso com pequenas bases de dados de imagens alinhadas, ninguém teve sucesso ao tratar com grandes bases de dados e com problemas mais realísticos, onde a localização e escala da face eram desconhecidas.

Em 1990 foi introduzido um método de manipulação algébrica, tornando fácil o cálculo direto das auto-faces. Mostrou-se que menos de 100 instruções eram suficientes para codificar o alinhamento e normalização das imagens. Este método foi o de reconhecimento de faces utilizando a transformada de Karhunen-Loève, proposto em [49]. Em Turk [84] demonstrou-se que o erro residual na codificação das auto-faces poderia ser usado para determinar precisamente a localização, a escala e a orientação das faces na imagem. Isto, acrescentado ao método de reconhecimento auto-face, produziria então um sistema mais confiável de reconhecimento de imagens com poucas restrições.

Segundo Pentland [71], a partir de 1993 surgiram muitos algoritmos para imagens com poucas restrições e seus desempenhos declarados, muitos deles testados em conjuntos de dados relativamente pequenos, tipicamente menores do que 100 imagens. Então, para melhor entender o potencial de cada um deles, o DARPA[2] e o *Army Research Laboratory* elaboraram o programa FERET[3] [73] com o objetivo de avaliar desempenho e encorajar avanços nesta tecnologia. Ele é o conjunto de testes mais abrangente proposto até o momento, com bases de imagens estáticas, ou seja, elas contém apenas informação bidimensional. Ele possui faces com variações de translação, escala e iluminação compatíveis com as fotografias 3×4 ou as de carteira de habilitação para motoristas americanos. Há fotos de pessoas tiradas em datas diferentes (a diferença chega a um ano). O maior teste do FERET possui imagens de 1196 pessoas [71].

Em 2000, três algoritmos demonstraram os mais altos níveis de precisão em reconhecimento com grandes bases de dados (1.196 pessoas ou mais). Eles foram os algoritmos da USC [87], UMD [26] e MIT [61]. Todos eles basearam-se no programa FERET. Somente

[2] *Defense advanced research projects agency*

[3] *Face recognition technology*

dois deles, da USC e MIT, foram capazes de detecção e reconhecimento em imagens com pouca restrição; o sistema da UMD necessitava de aproximação das regiões dos olhos para operar. Houve um quarto algoritmo, desenvolvido pela *Rockefeller University* [70], cujos desenvolvedores desistiram do teste para formarem uma empresa comercial. Os algoritmos MIT e USC também tornaram-se bases para sistemas comerciais.

Para base de dados abaixo de 200 pessoas e imagens adquiridas sob condições similares, todos os quatro algoritmos produziram desempenhos próximos. Entretanto, usando-se o método de casamento por correlação para esta mesma base, pode-se alcançar, às vezes, precisão similar, dado que este é um método lento. Por isso Pentland [71], sugere que, para um novo algoritmo ser considerado potencialmente competitivo, o mesmo deve ser testado com uma base de dados de no mínimo 200 pessoas e alcançando o desempenho acima de 95% de acerto.

Nos últimos 13 anos, muitas pesquisas têm se concentrado nos problemas de segmentação e localização de uma face em uma imagem, e também extração de características tais como forma e distâncias de olhos, boca, etc. Muitos avanços foram alcançados no projeto de classificadores estatísticos e de redes neurais.

A pesquisa atual está dividida dentro das que usam informação bidimensional e as que usam informação tridimensional. Uma excelente análise de ambas as áreas foi publicada nos *Proceedings* da IEEE [12] e análises sobre tipos mais específicos de processamento de face também são encontrados, tais como a análise de modelos conexionistas de processamento de face por Valentin, et al [45].

1.5 Métodos de aquisição

Há diversas formas de aquisição e pré-processamento de faces. Elas podem ser classificadas em dois grupos, as que produzem informação bidimensional e as que produzem informação tridimensional. O método de aquisição utilizado nesta dissertação pertence ao segundo grupo pois produz informação a respeito da forma 3D de faces humanas. Ele é um método de aquisição inovador e foi empregado por Zimmermann [101]. As subseções seguintes darão uma breve revisão de como os sistemas atuais estão classificados mediante os processos de aquisição, bem como um melhor detalhamento a respeito do método de aquisição utilizado neste trabalho.

1.5.1 Sistemas 2D

Segundo Tibbalds [81], a maioria da pesquisa atual em reconhecimento de face está baseada em informação bidimensional. Isto se deve à facilidade do processo de aquisição. A maioria das fontes são câmeras de vídeo, e as informações podem ser reunidas em fitas de vídeo ou fotografias.

Para o tratamento e reconhecimento da informação 2D há uma variedade de diferentes abordagens, elas estão amplamente divididas em duas estratégias, casamento de modelos e casamento de características. Os artigos [14] e [75] fazem comparações entre elas.

No casamento por características extraídas das imagens de faces, informações sobre o tamanho, a forma e a localização das mesmas formam o conjunto de informação para o sistema de reconhecimento que sempre fará uma busca à biblioteca de base de dados para cada indivíduo. Tais características podem ser a forma do olho ou cor, a localização dos olhos relativo aos cantos da boca, etc. Estes sistemas tendem a ter pequenas quantias de informação armazenada para cada indivíduo, que fazem deles mais eficientes em termos de minimização do tamanho do banco de dados e o tempo de busca. Muitos exemplos destes tipos de sistemas tem sido descritos na literatura tais como os trabalhos sobre autofaces de Turk e Pentland [84], perfis de face [63] e distâncias entre características [46]. O reconhecimento é feito através do uso de um sistema de reconhecimento de padrões, seja estatístico, seja por redes neurais, usando como espaço de características essas informações.

Em casamento de modelos, a imagem como um todo é comparada com uma série de modelos armazenados em uma biblioteca. Há muitas variações a partir desta idéia básica, tais como usar somente pequenos modelos de olhos e boca, ou usar amplos modelos de grupo de faces dentro de tipos similares (tais como grupos racial e de sexo).

A maioria dos problemas que afetam sistemas de reconhecimento/identificação em informações 2D são principalmente a orientação e a iluminação. Como a maioria das pesquisas está voltada a imagens em tons de cinza (embora recentemente alguns trabalhos foram baseados em imagens coloridas) mudanças na iluminação da face pode resultar em grande brilho, contraste e mudanças de sombreado em imagens escaneadas. Estas mudanças causam significantes problemas no desenvolvimento destes sistemas. A mudança de orientação da cabeça também pode causar problemas pela mudança da forma e dimensões da face como vista na imagem escaneada. Por exemplo se, em um sistema de casamento por características, os olhos estão separados a “2cm”, logo esta medida somente será precisa se o indivíduo estiver olhando retamente para a câmera. Se a cabeça estiver levemente inclinada para um lado, então o valor medido será menor que o verdadeiro, desta forma fornecendo informação incorreta para o processo de identificação/reconhecimento. Isto tem inspirado alguns

pesquisadores no estudo de sistemas de características geometricamente invariantes [46], sem utilizar-se de sistemas tridimensionais.

1.5.2 Sistemas 3D

A principal vantagem da utilização de informação tridimensional da face é que ela não está sujeita a problemas de iluminação e e os de orientação podem ser facilmente compensados. Da mesma maneira que nos sistema 2D, a forma da face pode ser processada para identificação/reconhecimento através de técnicas de casamento de modelos ou características. Em [81] tem-se uma descrição detalhada de algumas técnicas de aquisição de dados 3D de faces.

Uma primeira categoria de técnicas de extração de formas 3D são os sistemas de escaneamento. Este processo é demorado pois ou a face ou a câmera tem que girar, ou seja, um ou outro terá que ser móvel, o que é lento e complicado mecanicamente. Portanto sua desvantagem é que os equipamentos necessários para adquirir a informação 3D são complexos e caros. Serviços de escaneamento comerciais são fornecidos através de muitas companhias tais como *Cyberware Inc* [38] e a *3D Scanners Ltd* [57] que descreveram um de seus sistemas em *IEE Synposium on '3D Imaging and analysis of depth/range images'* [18].

Outra técnica é a utilizada nos sistemas de luz estruturada. É um método alternativo que utiliza princípios de projeções de franjas de luz para a partir da análise da curvatura destas obter a informação 3D do objeto. As vantagens deste método são: necessita somente uma câmeras e um sistema de projeção de luz estruturada. A princípio este método foi empregado para a determinação da forma de objetos geométricos simples. Neste trabalho aplica-se este método para a determinação de uma geometria mais complexa que é a face humana.

Existe também a técnica utilizada por sistemas de visão estéreo. Visão estéreo é o ramo da visão computacional que analisa o problema da reconstrução da informação tridimensional de objetos a partir de um par de imagens capturadas simultaneamente, mas com um pequeno deslocamento lateral. No caso da tentativa de estabelecer a forma de uma face humana, estes sistemas não são capazes de gerar uma forma de superfície precisa. Alguns exemplos de sistemas que utilizaram esta técnica foram os de Ayache [4], Pollard [74] e Urquhart [85].

1.6 O sistema de aquisição SORFACE

Neste sistema de aquisição são utilizadas técnicas de iluminação estruturada para permitir a extração da geometria da face. O indivíduo é posicionado frente a uma câmera que captura a imagem da face com as franjas de luz obliquamente projetadas sobre ela. A partir desta

imagem é aplicado o método de análise de fase das franjas projetadas, produzindo assim a informação da forma 3D da face. Esta técnica é chamada de *Perfilometria de Fourier*. Na Fig. 1.2 pode-se ver o diagrama esquemático deste sistema. Para maiores detalhes desta implementação tem-se Zimmermann [102]. A Fig. 1.3 mostra um exemplo real da entrada e saída deste sistema de aquisição.

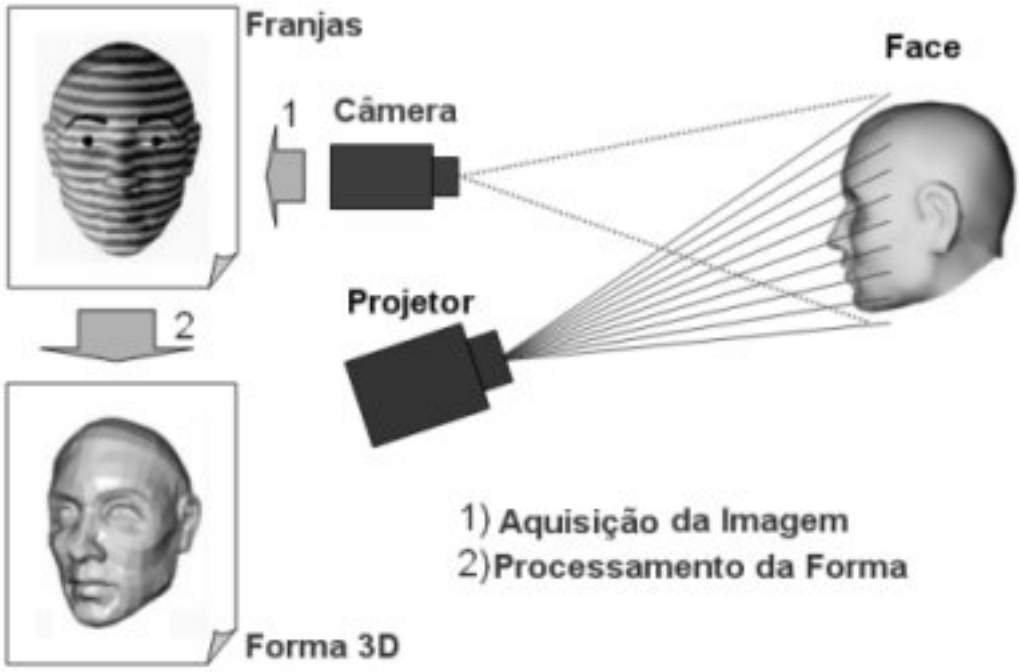

Figura 1.2: Esquema do método de aquisição de faces utilizado neste trabalho.

1.7 A base de dados SORFACE

Na construção da base de dados, concebida pelo método de aquisição descrito acima, o projeto SORFACE contou com a colaboração de 51 pessoas de diferentes raças e de ambos os sexos. Cada uma delas foi submetida à extração de sua imagem de face mediante as expressões faciais descritas na Tab. 1.1. No total de 22 expressões foram obtidas com a interação do indivíduo (como exemplo tem-se a Fig. 1.4), enquanto que as 10 restantes foram obtidas a partir de algoritmos que realizaram rotação da forma da face nos eixos X, Y e Z e escalonamento no eixo Z. Isto levou a 32 expressões por pessoa.

Portanto a partir do sistema de aquisição acima foram geradas matrizes com as medidas das alturas das regiões da face. Estas medidas foram normalizadas no intervalo de $[0\,;1]$, e representam a distância de cada ponto da superfície da face a um plano de referência. Assim, a ponta do nariz corresponde sempre ao local de valor máximo, 1, veja a Fig. 1.5. Todas

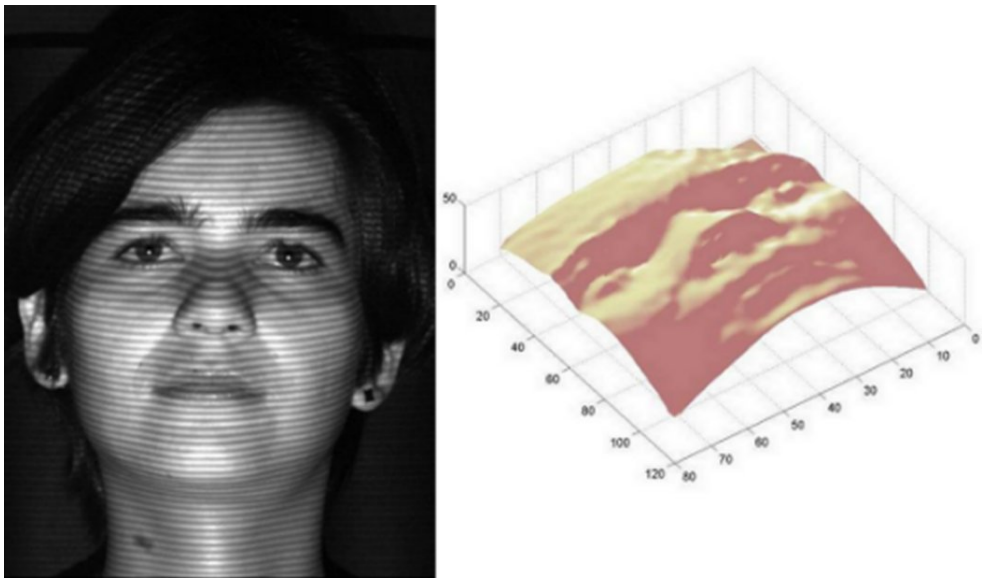

Figura 1.3: Exemplo prático de uma aquisição de face.

estas matrizes de alturas foram submetidas a 5 níveis de ruídos (perturbações nas medidas das alturas), Fig. 1.6, com isso temos o total de 32 expressões $\times$ 5 níveis de ruído $=$ 160 padrões de forma de face por pessoa.

Também disponibilizou-se 9 resoluções $(linha \times coluna)$ destas matrizes, sendo elas (21×15), (31×22), (41×29), (51×36), (61×43), (71×50), (81×57), (91×64) e (101×71). Com isso a base de dados SORFACE contempla o total de 73.440 amostras de faces humanas (51 indivíduos $\times$ 32 expressões e variações nas faces $\times$ 5 níveis de ruído $\times$ 9 resoluções).

1.8 Organização deste documento

No Capítulo 2 é dada uma visão geral a respeito das várias técnicas de reconhecimento de padrões e alguns exemplos de aplicações. Tópicos sobre reconhecimento de faces também foram vistos a fim de serem investigados os aspectos de como começou e como se desenvolveram os sistemas de reconhecimento atuais.

No Capítulo 3 foram apresentadas justificativas a respeito da escolha da abordagem de RNAs, mais propriamente redes diretas e redes de única camada (Perceptron e Adaline), para o desenvolvimento deste trabalho. Mostra também que foi usada a técnica de programação evolucionária para a busca da estrutura da rede neural ótima, ou seja, aquela com o melhor desempenho e a menor complexidade.

Experimentos prévios foram realizados e são apresentados no Capítulo 4. Eles são frutos

Figura 1.4: Exemplos das expressões faciais adquiridas.

de algumas dúvidas que surgiram durante o desenvolvimento deste trabalho. Estas dúvidas foram sanadas, dando assim continuidade à linha de raciocínio que conduziram também os experimentos que chegaram aos resultados finais e conclusão deste trabalho.

O Capítulo 5 mostra o andamento dado aos experimentos até se chegar à conjectura de que o problema de reconhecimento de faces abordado é linearmente separável. Nele é mostrado como a evolução de arquitetura de redes diretas levou a redes com apenas um neurônio na camada intermediária desempenharem bem a tarefa de classificação das faces. E para constatar realmente o fato, os padrões de formas 3D foram apresentados a redes de única camada como o Perceptron e a Adaline, que desempenharam muito bem a tarefa de classificação.

E finalmente o Capítulo 6 consiste da conclusão obtida através do trabalho realizado e da sugestão de tópicos que podem ser explorados em trabalhos futuros.

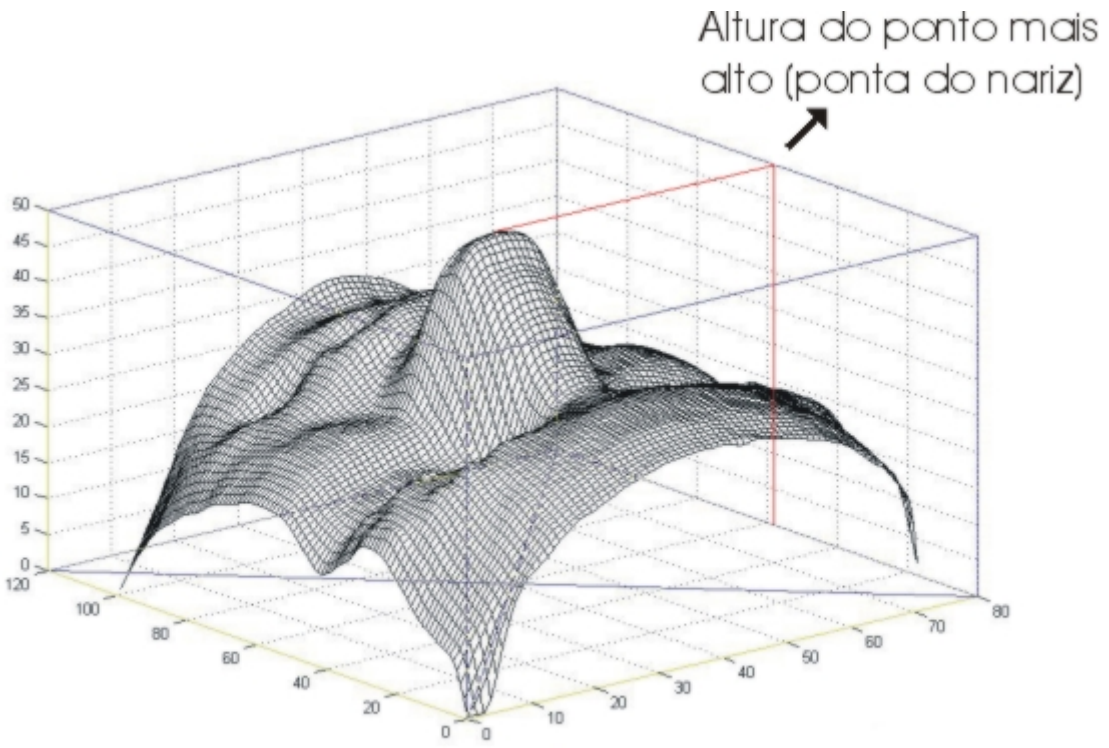

Figura 1.5: Representação no espaço 3D das alturas de uma face.

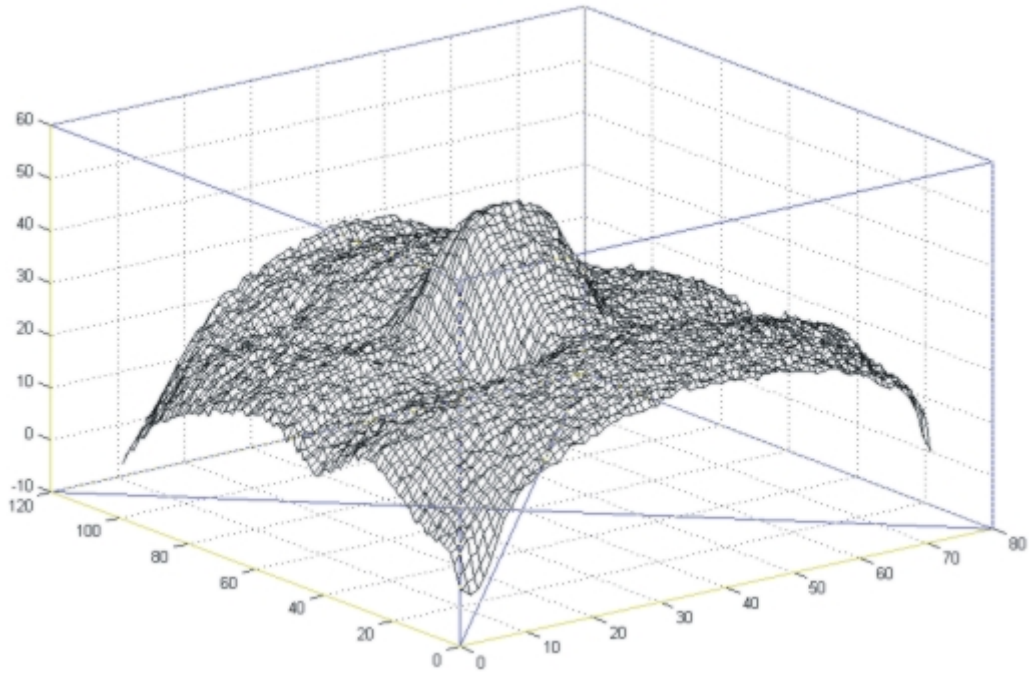

Figura 1.6: Um exemplo de representação no espaço 3D das alturas da face com níveis de ruído.

Expressão facial	Sigla	Indice
normal olhos abertos	$NOAB$	1
normal olhos semi-abertos	$NOSA$	2
normal olhos fechados	$NOFE$	3
normal sobrancelhas altas	$NSAL$	4
normal sobrancelhas baixas	$NSBA$	5
normal sobrancelhas relaxadas	$NSRE$	6
normal boca fechada	$NBFE$	7
normal boca semi-aberta	$NBSU$	8
normal boca aberta	$NBAB$	9
normal bochecha inflada	$NBIN$	10
normal bochecha sugada	$NBIN$	11
normal com óculos	$NCOC$	12
normal olhos abertos - rotação eixo X	$RX-2$	13
normal olhos abertos - rotação eixo X	$RX-1$	14
normal olhos abertos - rotação eixo X	$RX+0$	15
normal olhos abertos - rotação eixo X	$RX+1$	16
normal olhos abertos - rotação eixo X	$RX+2$	17
normal olhos abertos - rotação eixo Y	$RY-2$	18
normal olhos abertos - rotação eixo Y	$RY-1$	19
normal olhos abertos - rotação eixo Y	$RY+0$	20
normal olhos abertos - rotação eixo Y	$RY+1$	21
normal olhos abertos - rotação eixo Y	$RY+2$	22
normal olhos abertos - rotação eixo Z	$RZ-2$	23
normal olhos abertos - rotação eixo Z	$RZ-1$	24
normal olhos abertos - rotação eixo Z	$RZ+0$	25
normal olhos abertos - rotação eixo Z	$RZ+1$	26
normal olhos abertos - rotação eixo Z	$RZ+2$	27
normal olhos abertos - escalonamento eixo Z	$EZ-2$	28
normal olhos abertos - escalonamento eixo Z	$EZ-1$	29
normal olhos abertos - escalonamento eixo Z	$EZ-0$	30
normal olhos abertos - escalonamento eixo Z	$EZ+1$	31
normal olhos abertos - escalonamento eixo Z	$EZ+2$	32

Tabela 1.1: Exemplos de expressões faciais.

Capítulo 2

Fundamentação Teórica

2.1 Reconhecimento de padrão

2.1.1 Introdução

A tarefa de reconhecer padrões está presente na vida humana, desde os primeiros dias de sua existência, e é desempenhada com excelente qualidade. Porém, devido à parcial compreensão deste mecanismo biológico, se torna difícil a construção de uma máquina, baseada em instruções, que "aprenda" a fazer o mesmo [40]. Por este motivo, segundo Barreto [6], técnicas conexionistas têm funcionado muito bem para solução de problemas mais complexos como o reconhecimento de padrões envolvendo voz, caligrafia, diagnóstico médico, faces, dentre outros.

Diversas áreas de engenharia e ciências tais como biologia, psicologia, medicina, marketing, visão computacional e inteligência artificial, apresentam problemas importantes relacionados ao reconhecimento de padrões (RP). Ele é o fator crítico na maioria das tarefas automáticas de tomada de decisão e quanto mais relevante o padrão à disposição, ou seja, quão melhor ele for representado, melhor será realizada a decisão. Por meio dele se alcança uma melhor utilização de tecnologias disponíveis tais como sensores, câmeras, processadores, entre outros. As técnicas de RP têm, assim, um vasto leque de aplicações em um grande número de áreas científicas e tecnológicas, principalmente no projeto e desenvolvimento de sistemas inteligentes, que constituem o cerne do investimento tecnológico atual [40].

2.1.2 O que é reconhecimento de padrões?

Há duas maneira de se reconhecer e/ou classificar um padrão [40]: (i) classificação supervisionada: o padrão de entrada é identificado como um membro de uma classe pré-definida,

ou seja, a classe é definida pelo projetista do sistema, e (ii) classificação não supervisionada: onde o padrão é determinado por uma "fronteira" de classe desconhecida até o momento.

Então observa-se que um problema de RP consiste de uma tarefa de classificação ou categorização, onde as classes ou são definidas pelo projetista do sistema (classificação supervisionada) ou são "aprendidas" de acordo com a similaridade dos padrões (classificação não supervisionada).

O interesse na área de RP tem crescido muito devido a aplicações que são não somente desafiantes mas também computacionalmente mais exigentes. A Tab. 2.1 mostra exemplos de domínios de problema com suas respectivas classes de padrões.

Domínio do Problema	Aplicação	Padrão de Entrada	Classes de Padrão
Bioinformática	Análise de Seqüência	DNA/Seqüência de Proteína	Tipos conhecidos de genes/padrões
Mineração de dados	Busca por padrões significantes	Pontos em um espaço multi-dimensional	Compactar e bem separar grupos
Classificação de documento	Busca na Internet	Documento texto	Categorias semânticas(p.e. negócios, esportes, etc.)
Análise de documento de imagem	Máquina de leitura para cego	Documento imagem	Caracteres alfanuméricos, palavra
Automação industrial	Inspeção de circuito impresso de placas	Intensidade ou alcance de imagem	Natureza do produto defeituosa ou não
Recuperação de base de dados multimídia	Busca Internet	Video clip	Gêneros de vídeo (p.e. ação, diálogo, etc.)
Reconhecimento biométrico	Identificação pessoal	Face, íris, impressão digital	Usuários autorizados para controle de acesso
Sensoriamento remoto	Prognóstico da produção de colheita	Imagem multi-espectral	Categorias de aproveitamento de terra, desenvolvimento de padrões de colheita
Reconhecimento de voz	Inquérito por telefone sem assistência de operador	voz em forma de onda	Palavras faladas

Tabela 2.1: Exemplos de aplicações para o RP. Adaptada de: [40].

Com o avanço e disponibilidade de vários recursos computacionais tornou-se fácil o projeto e utilização de elaborados métodos de análise e classificação de padrões. Em muitas aplicações, não há somente uma única abordagem para classificação que seja "ótima" e por isso a combinação de várias abordagens de classificadores é uma prática bastante usada.

O projeto de um sistema de RP essencialmente envolve as três etapas seguintes:

(i) aquisição de dados (extração de características) e pré-processamento (seleção das

características mais discriminativas),

(ii) representação de dados, e

(iii) tomada de decisão (construção de um classificador ou descritor).

A escolha de sensores, técnicas de pré-processamento, esquema de representação e método para a tomada de decisão depende do domínio do problema. Um problema bem definido e suficientemente detalhado, onde se tem pequenas variações intra-classes e grandes variações inter-classes, produzirá representações compactas de padrões e conseqüentemente a estratégia de tomada de decisão será simplificada. Aprender, a partir de um conjunto de exemplos (conjunto de treinamento), é um atributo importante e desejado na maioria dos sistemas.

A escolha de uma abordagem para o RP não é uma tarefa simples e muitas vezes ela conta com a experiência do projetista. A seguir várias abordagens para o RP são apresentadas. Vale observar que elas não são necessariamente independentes, pois desde os primórdios da pesquisa em RP várias são as tentativas para o projeto de sistemas híbridos [29]. E na literatura de RP, às vezes a mesma abordagem possui diferentes interpretações.

2.1.3 Técnicas para reconhecimento de padrões

2.1.3.1 "Casamento" de modelos (*Template Matching*)

Uma das primeiras e mais simples abordagens para reconhecer padrões é a técnica de casamento de modelos. O "casamento" é uma operação genérica usada para determinar a similaridade entre duas entidades do mesmo tipo. O modelo é tipicamente uma forma 2D ou um protótipo.

O padrão a ser reconhecido é comparado com os modelos armazenados, observando todas as variações possíveis em termos de: translação, rotação e mudanças de escalas. A medida de similaridade é freqüentemente uma correlação ou uma função de distância. Muitas vezes, o modelo por si mesmo é aprendido a partir do conjunto de treinamento. Este método é computacionalmente exigente, mas a disponibilidade de recursos computacionais de hoje permite com que estas abordagens se viabilizem mais facilmente. [40].

O casamento de modelos faz parte das abordagens de decisão teórica que se baseiam na utilização de *funções de decisão* (ou discriminantes). Seja $x = (x_1, x_2, ..., x_n)^T$ um vetor de padrão n-dimensional. Para M classes de padrões $w_1, w_2, ..., w_M$, o problema básico é encontrar M funções de decisão $d_1(x), d_2(x), ..., d_M(x)$, com a propriedade de que, se o padrão x pertencer à classe w_i, então

$$d_i(x) > d_j(x) \quad j = 1, 2, ..., M; \;\; j \neq i. \tag{2.1}$$

Ou seja, um padrão desconhecido x pertencerá à *i*-ésima classe de padrões se a substituição de x em todas as funções de decisão fizer com que $d_i(x)$ tenha o maior valor numérico. Empates são resolvidos arbitrariamente.

A fronteira de decisão que separa as classes w_i e w_j é dada pelos valores de x para os quais $d_i(x) = d_j(x)$ ou, equivalentemente, pelos valores de x para os quais

$$d_i(x) - d_j(x) = 0. \tag{2.2}$$

É comum identificar a fronteira de decisão entre duas classes pela função $d_{ij}(x) = d_i(x) - d_j(x) = 0$. Portanto, $d_{ij}(x) > 0$ para padrões de classe w_i e $d_{ij}(x) < 0$ para padrões de classe w_j [32].

2.1.3.1.1 Classificador de distância mínima

Suponha que cada classe de padrões seja representada por um vetor *protótipo* (ou *médio*):

$$m_j = \frac{1}{N_j} \sum_{x \in w_j} x \quad j = 1, 2, ..., M \tag{2.3}$$

em que N_j é o número de vetores de padrões da classe w_i, e a soma é realizada sobre esses vetores. Uma maneira de definir a pertinência de um vetor padrão x desconhecido é atribuí-lo à classe de seu protótipo mais próximo. A distância euclidiana, ou a de Hamming, pode ser usada para determinar a proximidade, reduzindo o problema à computação das distâncias:

$$D_j(x) = \|x - m_j\| \quad j = 1, 2, ..., M \tag{2.4}$$

em que $\|a\| = (a^T a)^{1/2}$ é a norma euclidiana. Atribuímos, então, x à classe w_i se $D_i(x)$ for a menor distância. Ou seja, a menor distância implica no melhor casamento nessa formulação. Não é difícil mostrar que isso é equivalente a avaliar as funções

$$d_j(x) = x^T m_j - \frac{1}{2} m_j^T m_j \quad j = 1, 2, ..., M \tag{2.5}$$

e a atribuir x à classe w_i se $d_i(x)$ levar ao maior valor numérico. Essa formulação está de acordo com o conceito de função de decisão, como definido na Equação (2.1).

A partir das Equações (2.2) e (2.5), pode-se ver que a fronteira de decisão entre as classes w_i e w_j para o classificador de distância mínima é

$$\begin{aligned} d_{ij}(x) &= d_i(x) - d_j(x) \\ &= x^T(m_i - m_j) - \frac{1}{2}(m_i - m_j)^T(m_i - m_j) = 0. \end{aligned} \tag{2.6}$$

A superfície dada pela Equação (2.6) é a bisseção perpendicular do segmento de linha entre m_i e m_j. Para $n = 2$ a bisseção perpendicular é uma linha, para $n = 3$ é um plano, e para $n > 3$ é chamado de *hiperplano* [32].

2.1.3.1.2 Casamento de modelos por correlação

Segundo [32], o conceito básico de correlação de imagem é considerado como a base para encontrar casamentos de uma sub-imagem $w(x, y)$ de tamanho $J \times K$ dentro de uma imagem $f(x, y)$ de tamanho $M \times N$, supondo-se que $J \leq M$ e $K \leq N$. Embora a abordagem por correlação possa ser formulada na forma vetorial, o tratamento direto com uma imagem ou sub-imagem é mais intuitivo.

Em sua forma mais simples, a correlação entre $f(x, y)$ e $w(x, y)$ é

$$c(s,t) = \sum_x \sum_y f(x,y)w(x-s, y-t) \tag{2.7}$$

em que $s = 0, 1, 2, \ldots, M-1$ e $t = 0, 1, 2, \ldots, N-1$, e a soma é realizada sobre a região da imagem em que f e w se sobreponham. A Fig. 2.1 ilustra este procedimento, sendo considerada a origem de $f(x, y)$ o topo à esquerda e a de $w(x, y)$ a região de seu centro. Para qualquer valor de (s, t) dentro de $f(x, y)$, a aplicação da Equação (2.7) leva a um valor c. Na medida que s e t são varridos, $w(x, y)$ é movido na área da imagem, fornecendo uma função $c(s, t)$. O valor máximo de $c(s, t)$ indica a posição em que $w(x, y)$ melhor se casa com $f(x, y)$. Note que se perde precisão para valores de s e t perto das bordas de $f(x, y)$, com a amplitude de erro sendo proporcional ao tamanho de $w(x, y)$.

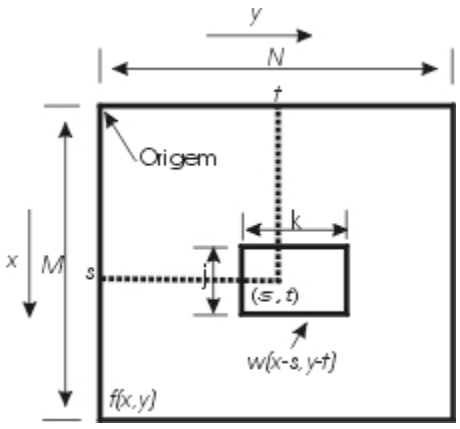

Figura 2.1: Esquema para se obter a correlação de $f(x, y)$ e $w(x, y)$ no ponto (s, t). Adaptada de: [32].

A função de correlação dada na equação (2.7) possui a desvantagem de ser sensível a mudanças na amplitude de $f(x, y)$ e de $w(x, y)$. Por exemplo, dobrando-se todos os valores de $f(x, y)$, se dobrarão todos os valores de $c(s, t)$. Uma abordagem freqüentemente usada para evitar essa dificuldade é realizar o casamento através do *coeficiente de correlação*, que

é definido como

$$\gamma(s,t) = \frac{\sum_x \sum_y [f(x,y) - \overline{f}(x,y)][w(x-s,y-t) - \overline{w}]}{\left\{ \sum_x \sum_y [f(x,y) - \overline{f}(x,y)]^2 \sum_x \sum_y [w(x-s,y-t) - \overline{w}]^2 \right\}^{1/2}} \quad (2.8)$$

em que $s = 0, 1, 2, ..., M-1$ e $t = 0, 1, 2, ..., N-1$, $\overline{w}$ é o valor médio dos pixels em $w(x,y)$ (computado apenas 1 vez), $\overline{f}(x,y)$ é o valor médio de $f(x,y)$ na região coincidente com a posição corrente de w, e as somas são realizadas sobre as coordenadas comuns, tanto a f como a w. O coeficiente de correlação $\gamma(s,t)$ tem sua escala no intervalo -1 a 1, independentemente de mudanças na amplitude de $f(x,y)$ e $w(x,y)$.

Embora a função de correlação possa ser normalizada para mudanças de amplitude através do coeficiente de correlação, a obtenção da normalização para mudanças de tamanho e rotação pode ser difícil. A normalização em relação ao tamanho envolve mudança de escala espacial, um processo que acrescenta um custo computacional considerável. Se uma pista em relação à rotação puder ser extraída de $f(x,y)$, então bastará rotacionar $w(x,y)$ de maneira que ela mesma se alinhe com o grau de rotação de $f(x,y)$. Entretanto, se a natureza da rotação for desconhecida, a busca pelo melhor casamento requererá rotações exaustivas de $w(x,y)$. Esse procedimento é impraticável e, por conseguinte, a correlação é raramente usada em casos em que rotação arbitrária ou sem restrições esteja presente [32].

Muitos pesquisadores atualmente se utilizam da abordagem de casamento de modelos em diversas áreas de aplicações: i) para determinar a presença de uma imagem ou objeto dentro de uma cena [21] e ii) para reconhecimento de caracteres [22]. O aspecto da segurança em sistemas que utilizam técnicas de casamento de modelos, em aplicações de reconhecimento de pessoas, é investigado em [12], pois eles são mais vulneráveis a ataques de força bruta[1]. Isto resulta em invasões de privacidade que acarretam grandes problemas pois o usuário tem registrado uma imagem de parte de seu corpo no banco de dados do sistema.

2.1.3.2 Técnicas estatísticas

Em RP com abordagem estatística, um padrão é representado por um conjunto de características chamado de vetor de característica *d*-dimensional. Os conceitos da teoria de decisão estatística são utilizados para estabelecer fronteiras de decisão entre classes de padrões. O

[1]O ataque de força bruta é uma das técnicas mais antigas de tentativas de invasão em sistemas de segurança. Todo sistema de acesso restrito é acessível através do conjunto de nome de usuário e senha, e um ataque de força bruta significa tentar adivinhar este conjunto por meio de tentativa e erro. É inviável conceber um ataque de força bruta manualmente. Muitos são os programas usados para automatizar este processo de tentativa de acesso.

sistema de reconhecimento é operado em dois modos: treinamento (aprendizagem) e classificação (teste) (veja a Fig. 2.2) [40].

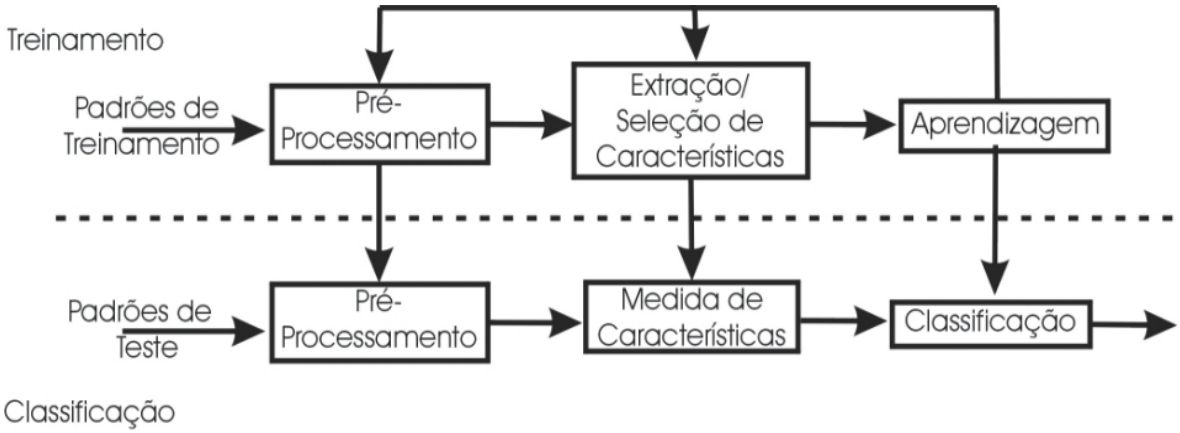

Figura 2.2: Blocos funcionais para o reconhecimento de padrão na abordagem estatística. Adaptada de: [40].

A função do módulo de pré-processamento é capturar o padrão de interesse, remover ruído, normalizar, e qualquer outra operação que contribua para a definição de uma representação compacta do padrão.

Um dos problemas óbvios encontrados, principalmente quando o padrão se trata de uma imagem, é a alta dimensionalidade dos dados de entrada. Técnicas, que combinam as variáveis (características) de entrada mais próximas para produzir um menor número das mesmas, ajudam a aliviar tais problemas. Estas técnicas podem ser construídas "manualmente", baseadas em algum entendimento do problema particular, ou podem ser derivadas dos dados, a partir de procedimentos automáticos [10]. Estes métodos são chamados de extração e seleção de características e serão vistos com mais detalhes nas seções seguintes. Eles estão presentes no módulo de treinamento, parte superior da Fig. 2.2, para encontrar características apropriadas às representações de padrões de entrada e o classificador é treinado para particionar o espaço de características. Otimizações do pré-processamento e das estratégias de extração e seleção de características são realizadas no caminho recorrente da Fig. 2.2. No modo classificação, o classificador treinado mapeia o padrão de entrada a uma das classes de padrões sob considerações, baseado nas características medidas.

O processo de tomada de decisão estatística em RP pode ser sintetizado como segue: Seja um padrão representado por um vetor $x = (x_1, x_2, ..., x_d)$ com d características, ele será determinado a uma das c classes $w_1, w_2, ..., w_c$. Supõe-se que cada característica apresente uma densidade probabilidade ou função massa (dependendo das características serem contínuas ou discretas) condicionada à cada classe. Assim, um padrão x pertencente a uma classe w_i é

visto como uma observação extraída aleatoriamente a partir de uma função de probabilidade classe-condicional $p(x|w_i)$. As regras de decisão, incluindo a regra de decisão Bayes, a regra da probabilidade máxima (que pode ser vista como um caso particular da regra Bayes) e a regra Neyman-Peason são eficazes para definir a fronteira de decisão. A regra de decisão "ótima" de Bayes para a minimização do risco condicional pode ser declarada como segue:

$$R(w_i|x) = \sum_{j=1}^{c} L(w_i, w_j).P(w_j|x) \tag{2.9}$$

Ela determina a classe w_i para o padrão de entrada x onde o risco condicional é mínimo, $L(w_i, w_j)$ é a função de perda causada na decisão de w_i quando a classe verdadeira é w_j e $P(w_j|x)$ é a probabilidade posterior [44] apud [40]. No caso da função perda ser $0/1$, como definido na Eq. 2.10, o risco condicional torna-se a probabilidade condicional de falsa classificação.

$$L(w_i, wj) = \begin{cases} 0, & \text{i=j} \\ 1, & \text{i} \neq \text{j} \end{cases} \tag{2.10}$$

Para a escolha da função de perda, a regra de decisão Bayes pode ser simplificada como segue: ela determina o padrão de entrada x para a classe w_i se

$$P(w_i|x) > P(w_j|x),\ para\, todo\, j \neq i. \tag{2.11}$$

Várias estratégias são utilizadas para projetar um classificador para o RP com abordagem estatística, dependendo da espécie de informação disponível a respeito de densidades de classe-condicional. Se todas elas são especificadas, então a regra de decisão ótima de Bayes pode ser usada para a classificação. Entretanto, densidades de classe-condicional são freqüentemente desconhecidas na prática e devem ser aprendidas a partir dos padrões de treinamento disponíveis. Se a forma da densidade classe-condicional é conhecida, por exemplo, uma Gaussiana multivariada, mas alguns dos parâmetros de densidades, por exemplo, vetores médio e matrizes de covariância, são desconhecidos, então tem-se um problema de decisão paramétrico. Uma estratégia comum para estes tipos de problemas é substituir os parâmetros desconhecidos na função densidade por seus valores estimados. Se a forma da densidade classe-condicional não é conhecida, então opera-se em um modo não paramétrico. Neste caso, ou estima-se a função densidade (ex.: abordagem janela Parzen) ou constrói-se diretamente a fronteira de decisão baseada nos dados de treinamento (ex.: regra do k-ésimo vizinho mais próximo). O perceptron multicamada pode ser visto como um método supervisionado não paramétrico que constrói uma fronteira de decisão.

Outra dicotomia na abordagem estatística para o RP é a do aprendizado supervisionado versus o aprendizado não supervisionado. Em um problema de aprendizado não supervisionado, algumas vezes o número e as estruturas de classes devem ser aprendidas mediante

o conjunto de exemplos de treinamento. As várias dicotomias são mostradas na árvore de estruturas da Fig. 2.3.

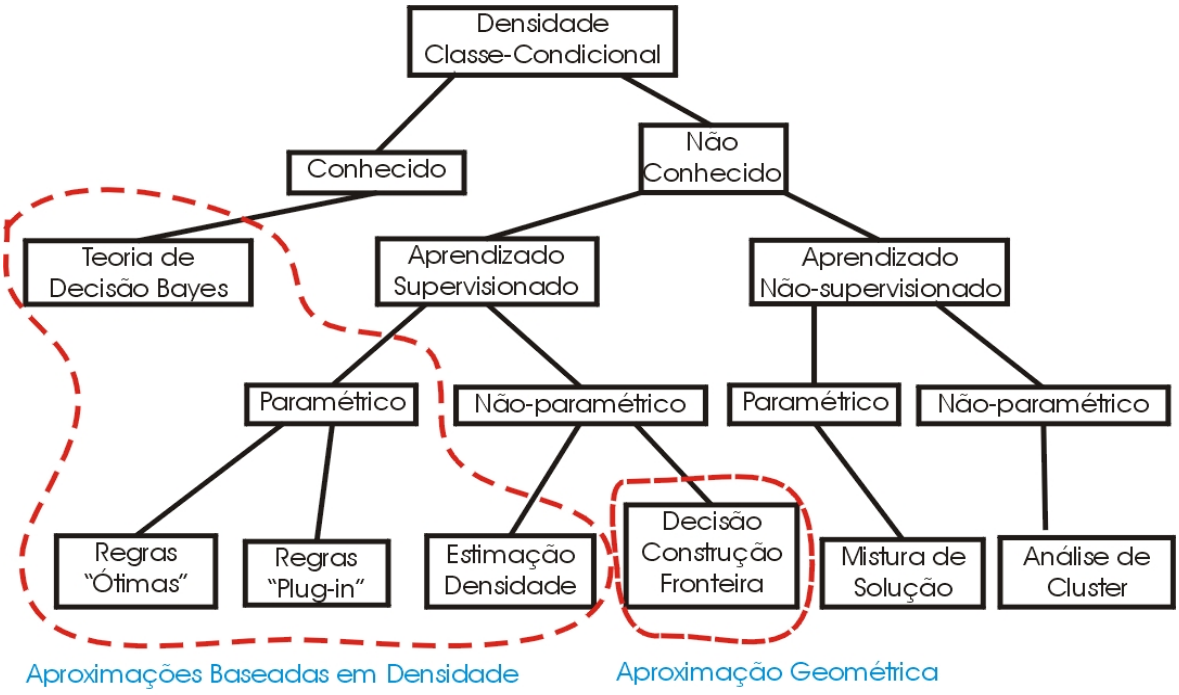

Figura 2.3: As várias abordagens estatísticas para o reconhecimento de padrão. Adaptada de: [40].

A medida que se percorre a árvore de cima para baixo e da esquerda para a direita, menos informações a respeito das características e classes de padrões são disponíveis e como resultado a dificuldade de classificação aumenta. Em alguns casos, a maioria dos métodos (nas folhas da árvore da Fig. 2.3) são tentativas de implementar a regra de decisão Bayes. A análise de aglomerado (cluster) trata com problemas de tomada de decisão no modo não paramétrico e aprendizado não supervisionado [39], onde o número de categorias ou clusters não é especificado; a tarefa é descobrir uma categorização razoável dos dados (se existir alguma). Algoritmos de análise de aglomerado junto com várias técnicas para visualização e projeção de dados multi-dimensionais são também referidas como métodos de *análise exploratória de dados*.

Ainda outra dicotomia se baseia na maneira como as fronteiras de decisão são obtidas, direta (abordagem geométrica) ou indireta (abordagem baseada em densidade probabilística), Fig. 2.3. A abordagem probabilística requer primeiro que a função de densidade seja estimada, para então construir as funções discriminantes que especificam as fronteiras de decisão. Por outro lado, a abordagem geométrica freqüentemente constrói fronteiras de decisão diretamente, através de funções de custo fixo.

Não importa qual seja a regra de classificação ou decisão usada, ela deve ser treinada usando os exemplos de treinamento disponíveis e o desempenho de um classificador dependerá disto e da quantidade destes exemplos. Ao mesmo tempo, o objetivo principal de um sistema de reconhecimento é classificar exemplos de testes futuros, os quais são provavelmente diferentes dos exemplos vistos durante o treinamento.

2.1.3.2.1 Sobre-treinamento e sobre-adaptação

Otimizar um classificador para maximizar seu desempenho sobre o conjunto de treinamento pode nem sempre resultar no desempenho desejado para um conjunto de teste. A habilidade de generalização de um classificador refere-se ao seu desempenho em classificar padrões testes que não foram usados durante o estágio de treinamento. Uma habilidade pobre de generalização pode ser atribuída a qualquer um dos seguintes fatores:

(i) número de características muito grande relativo ao número de exemplos de treinamento,

(ii) grande número de parâmetros desconhecidos associados ao classificador (ex.: classificadores polinomiais ou uma rede neural larga (rede direta com número excessivo de neurônios na camada intermediária)), e

(iii) um classificador ser intensivamente otimizado no conjunto de treinamento (sobre-treinamento[2]).

O sobre-treinamento também é análogo ao fenômeno de sobre-adaptação[3] em regressão, quando existem muitos parâmetros livres. Estes fenômenos são teoricamente investigados através de classificadores que minimizam a taxa de erro aparente (o erro no conjunto de treinamento). Há várias fases no fenômeno de sobre-treinamento, por exemplo, dependendo da relação entre o número de t exemplos e o número m de parâmetros modificáveis. Quando t é menor ou quase igual a m, os exemplos podem, em princípio, ser memorizados e a sobre-adaptação é elevada nesta fase, principalmente quando $t \approx m$.

O sobre-treinamento pode ser dividido em duas categorias:

(i) **Absoluto**, quando o desempenho de classificação degrada para todas as categorias de padrões e

(ii) **Relativo**, quando o desempenho de classificação degrada para algumas categorias, enquanto para outras permanece inalterado ou até mesmo melhora.

[2] *overtraining*
[3] *overfitting*

Às vezes, há dominância de padrões de algumas categorias no conjunto de treinamento, ocasionando um sobre-treinamento do classificador que se adaptará as mesmas. Isto é considerado um sobre-treinamento relativo. O sobre-treinamento absoluto ocorre principalmente devido ao conjunto de treinamento ser um limiar representativo para o conjunto de teste. Por outro lado, o sobre-treinamento relativo ocorre usualmente devido ao conjunto de treinamento apresentar padrões "confusos" nas regiões do envoltório da fronteira de decisão [20].

Os estudos clássicos de Cover [33] e Vapnik [162] apud [40], sobre capacidade e complexidade de classificadores, provêem um bom entendimento dos mecanismos que levam ao sobre-treinamento. Classificadores complexos, por exemplo, aqueles tendo muitos parâmetros independentes, podem ter uma grande capacidade, isto é, eles são hábeis para representar muitas dicotomias para um dado conjunto de dados.

As armadilhas da sobre-adaptação em estimadores, para um dado conjunto de treinamento, são observadas em muitos estágios de um sistema de RP, tais como na redução de dimensionalidade, estimativa de densidade, e construção do classificador. O conceito de sobre-adaptação refere-se à demasiada adaptação e ajuste do classificador a exemplos específicos, perdendo assim sua capacidade de generalização. Em alguns casos consiste de uma distorção local da fronteira de decisão, ou seja, não cabe supor que sua ocorrência é simultânea em todo o espaço de características, a distorção pode ocorrer em diferentes locais e em diferentes momentos. Isto implica que em alguns locais a fronteira de decisão é contínua, enquanto em outras áreas a sobre-adaptação já está presente [76]. Uma solução certa é sempre usar um conjunto teste independente do conjunto de treinamento para avaliação. Para evitar a necessidade de muitos conjuntos de testes independentes, estimadores são freqüentemente baseados em subconjuntos dos dados rotacionados, preservando diferentes partes dos dados para otimização e avaliação [166] apud [40].

2.1.3.2.2 O problema da dimensionalidade e o fenômeno de máximo

O desempenho de um classificador depende do inter-relacionamento entre o tamanho do conjunto de exemplos, o número de características dos padrões e a sua complexidade. Seja o exemplo de uma simples técnica de tabela de consulta, onde se particiona o espaço de características em células e se associa um nome de classe a cada célula. Isto requer que o número de exemplos de treinamento seja uma função exponencial da dimensão de características [18] apud [40]. Este fenômeno é chamado de "maldição da dimensionalidade"[4], que conduz ao "fenômeno de máximo"[5] em um projeto de classificador [40].

[4] Course of dimensionality

[5] *Peaking phenomenon*

A probabilidade de classificação falsa de uma regra de decisão não aumenta na mesma proporção que aumenta o número de características, dado que as densidades classe-condicional sejam completamente conhecidas. Entretanto, tem-se freqüentemente observado que, na prática, o aumento de características pode degradar o desempenho de um classificador, se o número de exemplos de treinamento, que foi usado para projetar o classificador, é relativamente pequeno com relação ao número de características. Este é um comportamento paradoxal, referido como o fenômeno de máximo [80, 131,132] apud [40]. Uma simples explanação sobre este fenômento é dada a seguir. A maioria dos classificadores paramétricos geralmente usados estima parâmetros não conhecidos e liga-os a parâmetros verdadeiros nas densidades de classe condicional. Em uma amostra de tamanho fixo, quando o número de características cresce (à medida que aumenta o número de parâmetros desconhecidos), a confiança dos parâmetros estimados decresce. Conseqüentemente, o desempenho dos classificadores, para uma amostra de tamanho fixo, pode degradar com um aumento no número de características.

Todos os classificadores geralmente usados, incluindo redes neurais diretas, podem sofrer o problema da dimensionalidade, pois é muito difícil estabelecer um exato relacionamento entre a probabilidade de falsa classificação, o número de exemplos de treinamento, o número de características e os parâmetros verdadeiros das densidades de classe-condicional. Algumas linhas de direção são sugeridas relativas ao tamanho do conjunto de exemplos para dimensionalidade. É geralmente aceitável que o número de exemplos de treinamento por classe seja pelo menos dez vezes o número de características $(n/d > 10)$. Isto seria uma boa prática a se seguir no projeto de um classificador [80] apud [40]. Quanto mais complexo o classificador, maior deveria ser a proporção do tamanho de exemplos para ser evitado o problema da dimensionalidade.

2.1.3.2.3 Redução da dimensionalidade

As vantagens em reduzir a dimensionalidade da representação do padrão refletem-se na medida de custo e precisão do classificador. Além disto, como visto anteriormente, uma pequena quantidade de características pode aliviar o problema da dimensionalidade, quando o número de exemplos de treinamento é pequeno. Porém, um reduzido número de características pode levar a uma fraca discriminação e conseqüentemente a uma precisão inferior no sistema de reconhecimento resultante. Mas a redução de dimensionalidade é necessária quando, por exemplo, é possível construir dois padrões arbitrários *similares* codificando-os a partir de um grande número de características redundantes [86].

Existem diferenças entre seleção e extração de características, embora na literatura elas sejam usadas indistintamente. O termo seleção refere-se a algoritmos que procuram sele-

cionar o melhor subconjunto de um conjunto de caraterísticas de entrada. Já algoritmos de extração são métodos que criam novas características a partir de transformações ou combinações do conjunto de características original. Freqüentemente, a extração precede a seleção, pois primeiro características são extraídas a partir do sentido dos dados (p.e. usando componente principal ou análise discriminante) e então algumas características extraídas, com baixa habilidade de discriminação, são descartadas.

A escolha entre seleção e extração depende do domínio de aplicação e dos dados específicos de treinamento disponíveis. A seleção conduz à economia na medida de custo quando algumas características são descartadas e as que foram selecionadas retêm suas interpretações físicas originais. Além do mais, as mesmas podem ser importantes para o entendimento do processo físico que gera os padrões. Por outro lado, transformações geradas por extração podem prover uma melhor habilidade discriminativa do que o melhor subconjunto de características originais, mas estas novas características podem não ter um claro sentido físico.

O ponto principal da redução de dimensionalidade é a escolha de uma função de critério. Um critério geralmente usado é o erro de classificação segundo um subconjunto de características. Porém, o erro de classificação, por si só, não é confiável quando a quantidade de exemplos de padrões é pequena em relação ao número de características. E ainda mais, para a escolha de uma função critério, é necessário determinar a dimensionalidade apropriada do espaço de características reduzido. E em resposta a isto surge a noção de dimensionalidade intrínseca dos dados, que consiste de determinar se os padrões d-dimensionais originais podem ser descritos adequadamente em um subespaço de dimensionalidade menor do que d. Por exemplo, padrões d-dimensionais ao longo de uma curva aplainada tem uma dimensionalidade intrínseca de um, independentemente do valor de d. Deve-se perceber que dimensionalidade intrínseca não é o mesmo que dimensionalidade linear, que consiste de uma propriedade global dos dados envolvendo o número de autovalores significativos da matriz de covariância dos dados. Apesar de haver muitos algoritmos disponíveis para estimar a dimensionalidade intrínseca [81] apud [40], eles não indicam quão facilmente um subespaço de dimensionalidade pode ser identificado. A seguir veremos alguns dos métodos mais usados para extração e seleção de características.

2.1.3.2.4 Extração de características

Segundo [40], um método de extração de características determina um subespaço apropriado de dimensionalidade m (de uma maneira linear ou não-linear) no espaço de características original de dimensionalidade d ($m \leq d$). A transformada linear, assim como a análise de componente principal (PCA) ou expansão Karhunen-Loève computam os m maiores autove-

tores da matriz de covariância $d \times d$ de n padrões d-dimensionais. A transformação linear é definida como

$$Y = XH, \tag{2.12}$$

onde X é a matriz de padrão $n \times d$, Y é a matriz derivada $n \times m$, e H é a matriz de transformação linear $d \times m$ cujas colunas são autovetores. Visto que PCA usa as características mais expressivas (autovetores com os maiores autovalores), ele efetivamente aproxima os dados para um subespaço linear usando o critério do erro quadrado médio. Existem outros métodos que são mais apropriados para distribuições não-Gaussianas.

Enquanto que PCA é um método de extração de características linear e não supervisionado, análise discriminante usa a informação de categoria associada com cada padrão para extração (linear) da maioria das características discriminatórias, nela a separação interclasses é feita por uma medida de separabilidade que resulta no encontro de autovetores de $S_w^{-1}S_b$ (o produto do inverso da matriz de espalhamento do interior da classe, S_w, e a matriz de espalhamento entre classes, S_b) [58] apud [40].

Existem muitas maneiras de definir técnicas de extração de características não lineares. Um método semelhante e diretamente relacionado ao PCA é chamado de Kernel PCA [73], [145]. A idéia básica do kernel PCA é primeiro mapear os dados de entrada dentro de algum novo espaço de característica F, via uma função não linear Φ (por exemplo, polinomial de grau p) e então executar um PCA linear no espaço mapeado.

Escalonamento multidimensional (MDS) é outra técnica de extração de características não linear. Seu objetivo é representar um conjunto de dados multidimensional em 2 ou 3 dimensões semelhantes onde a matriz distância, no espaço de característica d-dimensional original, é preservada tão fielmente quanto possível no espaço projetado. Um problema com MDS é que ele não dá uma função de mapeamento explícita. Assim, não é possível estabelecer um novo padrão em um mapa já computado por um dado conjunto de treinamento, sem ter que repetir o mapeamento. Muitas técnicas têm sido investigadas para tratar esta deficiência que abrange desde interpolação linear até o treinamento de uma rede neural [38] apud [40].

Uma rede neural direta oferece um procedimento integrado para extração de características e classificação. A saída de cada camada intermediária pode ser interpretada como um conjunto de novas características, freqüentemente não lineares, apresentadas à camada de saída para classificação. Neste sentido, redes multi-camadas servem como extratores de características [100] apud [40]. Por exemplo, as redes que apresentam as então chamadas "camadas de pesos compartilhados", que são de fato filtros para extração de características em imagens bi-dimensionais. Durante o treinamento, os filtros são direcionados para os

dados, de maneira a maximizar o desempenho da classificação.

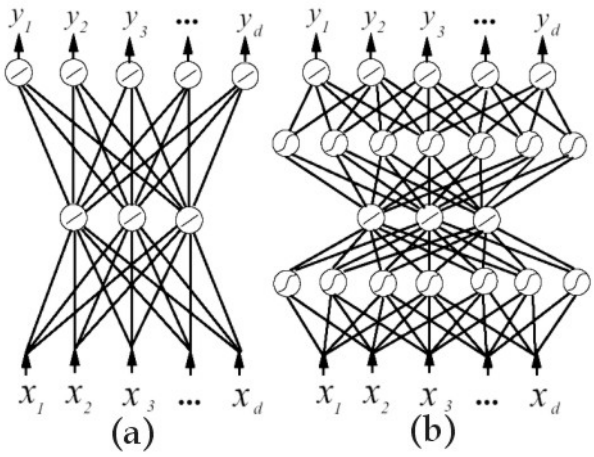

Figura 2.4: Redes auto-associativas para encontrar um subespaço tri-dimensional. (a) linear, (b) não-linear (nem todas as conexões são mostradas). Adaptada de: [40].

Redes neurais também podem ser usadas diretamente para extração de características em um modo não supervisionado. A Fig. 2.4 mostra a arquitetura de uma rede que é hábil para encontrar o subespaço PCA [117] apud [40]. Ao invés de sigmóides, os neurônios têm funções de transferência lineares. Esta rede tem d entradas e d saídas, onde d é o número de características dado. As entradas são também usadas como saídas desejadas, forçando a camada de saída a reconstruir o espaço de entrada usando somente uma camada intermediária. Os três nós na camada intermediária capturam os primeiros três componentes principais [18] apud [40]. Se duas camadas não lineares, com unidades intermediárias contendo funções de transferência sigmoidais, são incluídas também (veja Fig. 2.4(b)), então um subespaço não linear é encontrado na camada intermediária (também chamada de camada gargalo). A não linearidade é limitada pelo tamanho destas camadas adicionais. Estas então chamadas redes auto-associativas, ou redes PCA não lineares, oferecem uma poderosa ferramenta para treinar e descrever subespaços não lineares [98] apud [40].

Mapa auto-organizál, ou mapa de Kohonen [92] apud [40], pode também ser usado para extração de características não lineares. Nesta rede, chamada SOM[6] na literatura em inglês, os neurônios são dispostos em um espaço m-dimensional, onde m é geralmente 1, 2 ou 3. Cada neurônio é conectado a todas as d unidades de entrada. Os pesos das conexões de cada neurônio formam um vetor de pesos d-dimensional. Durante o treinamento, padrões são apresentados à rede de forma aleatória. A cada apresentação o vencedor, que é o vetor peso mais próximo do vetor de entrada, é identificado primeiro. Então, todos os neurônios

[6]Self-Organizing Map

na vizinhança do vencedor são atualizados de modo que seus vetores de pesos movam-se em direção ao vetor de entrada. Depois que o treinamento é feito, os vetores de pesos dos neurônios da vizinhança tornam-se bem parecidos com os padrões de entrada que estão próximos no espaço de características original. Assim, um mapa de "preservação de topologia" é formado, ou seja, a rede SOM oferece um mapa m-dimensional com uma conectividade espacial, que pode ser interpretada como extração de características.

2.1.3.2.5 Seleção de características

O problema da seleção é definido como segue: para um dado conjunto de d características, selecionar um subconjunto de tamanho m que conduza ao menor erro de classificação. O interesse da aplicação de métodos de seleção se deve ao grande número de características encontradas nas seguintes situações: (i) união de multi-sensores e (ii) integração de múltiplos modelos de dados [40].

Seja Y o conjunto de características dado, com cardinalidade d e seja m o número de características desejado no subconjunto selecionado $X, X \subseteq Y$. Seja $J(X)$ a função de critério de seleção para o conjunto X. Supõe-se que o maior valor de J indique um melhor subconjunto de características; a escolha natural da função critério é $J = (1 - P_e)$, onde P_e denota o erro de classificação. O uso de P_e na função critério faz o procedimento de seleção depender do classificador usado e dos tamanhos dos conjuntos de treinamento e teste. A maioria das abordagens diretas para o problema de seleção irá requerer (i) exame de todos os possíveis $\begin{pmatrix} d \\ m \end{pmatrix}$ subconjuntos de tamanho m e (ii) seleção do subconjunto com o maior valor de $J(\cdot)$. Entretanto o número de subconjuntos possíveis cresce combinatorialmente, fazendo desta uma busca exaustiva impraticável mesmo para valores pequenos de m e d. O único método de seleção ótimo que evita a busca exaustiva, pelo uso de resultados intermediários para o valor final de critério, está baseado no algoritmo de ramificação e fronteira [40].

Dado que os procedimentos de extração e seleção de características tenham encontrado uma representação apropriada para os padrões, chega a hora de escolher a abordagem na qual o classificador estatístico será projetado, que na prática é um problema difícil e na maioria das vezes esta escolha é freqüentemente baseada na experiência do projetista e nos acontecimentos ocorridos entre classificador e usuário [40].

2.1.3.3 Análise estrutural-sintática

Em muitos problemas de reconhecimento envolvendo padrões complexos é mais apropriado adotar uma perspectiva hierárquica, onde um padrão é visto como uma composição de simples sub-padrões [28].

Os métodos estruturais-sintáticos usam a estrutura interna do padrão como um elemento de análise. Eles são baseados no fato de que um padrão, por exemplo, um objeto, pode ser descrito recursivamente a partir de formas simples (primitivas), através de sua estrutura. A análise é realizada pela comparação de cada uma das estruturas próprias do objeto com classes referência, ou determinando se o exemplo pertence ou não a famílias de modelos referência, gerados a partir de cada classe.

Nestas técnicas, a representação do padrão não somente faz possível sua discriminação, como também sua reconstrução.

A tarefa de pré-processamento e extração de primitivas é transformar os dados "crus" do sensor para uma forma mais apropriada à inferência de uma descrição estrutural.

Contornos e segmentos de contornos são as primitivas de componentes de padrão mais comuns. Além de contornos, regiões são amplamente usadas. Elas podem ser quantitativamente caracterizadas por vários parâmetros, referindo-se ao tamanho e forma. Também histograma de níveis de cinza baseado em parâmetros de regiões são bem conhecidos.

A representação do padrão na abordagem estrutural está baseada em elementos primitivos e seus relacionamentos. Esta informação é simbólica na sua natureza. Então, ao invés de um vetor de características, como na abordagem estatística, outras estruturas de dados, freqüentemente de grande complexidade, como gráficos etc, são usados.

Ao se projetar um sistema de RP sintático, é muito importante a questão da seleção de padrões primitivos. Na maioria das vezes o processo de seleção de primitivas é guiado por intuição e heurísticas [15].

2.1.3.3.1 Gramáticas formais e linguagens

Segundo [15], a representação de classe na abordagem sintática é freqüentemente baseada em gramáticas formais. Dependendo da estrutura na qual a gramática opera, pode-se distinguir entre diferentes tipos de gramática. Primeiro, será discutido gramática sobre cadeias de caracteres (*strings*). Uma gramática formal de cadeia é uma quádrupla

$$G = (N, T, P, S) \tag{2.13}$$

onde:

N é um conjunto finito de símbolos não terminais

T é um conjunto finito de símbolos terminais com $N \cap T = 0$

P é um conjunto finito de produções ou regras re-escritas, e

$S \in N$ é o símbolo inicial

A informação mais importante está contida no conjunto de produções. Cada produção é da forma $X \rightarrow Y$ onde ambos X e Y são cadeias de símbolos sobre o alfabeto $V = N \cup T$.

O significado das produções $X \rightarrow Y$ é que uma cadeia X que ocorre dentro de qualquer outra cadeia como uma subcadeia, pode ser substituída pela cadeia Y. X é chamado de lado esquerdo e Y é chamado de lado direito da regra. A linguagem $L(G)$ de uma gramática G é definida como o conjunto de cadeias terminais que podem ser derivadas a partir de um símbolo inicial por repetidas aplicações de produções. Como um exemplo, considere a gramática:

$$\begin{aligned} G &= (N, T, P, S), \\ N &= \{S, A, B\}, \\ T &= \{a, b, c\}, \\ P &= \{p_1 : S \rightarrow cAb,\ p_2 : A \rightarrow aBa,\ p_3 : B \rightarrow aBa,\ p_4 : B \rightarrow cb\}. \end{aligned} \tag{2.14}$$

Esta gramática gera a linguagem:

$$L(G) = \{ca^n cba^n b \mid n \geq 1\}. \tag{2.15}$$

Para a geração do elemento $caacbaab \in L(G)$ (i.e. $n = 2$), a seguinte sequência de substituições de subcadeias são aplicadas:

$$S \rightarrow cAb \rightarrow caBab \rightarrow caaBaab \rightarrow caacbaab.$$

A gramática formal G é uma ferramenta adequada para a descrição de um conjunto infinito de cadeias eventuais, i.e. a linguagem $L(G)$. O uso de uma gramática para representação de classe de padrões é governada pela idéia de que os terminais da gramática correspondem a padrões primitivos que podem ser diretamente extraídos de um padrão de entrada por meio de pré-processamento adequado e métodos de segmentação. O conjunto de gramáticas não terminais correspondem a subpadrões de maior complexidade, que são construídos a partir de elementos primitivos. O procedimento de construção de sub-padrões complexos a partir de constituintes mais simples é modelado por produções da gramática. Finalmente, o símbolo inicial representa a classe dos padrões sob estudo como um todo.

Para ilustrar estas idéias, considere de novo a gramática em (2.14). Suponha que os símbolos terminais representam segmentos de linha de tamanho fixo como indicado na Fig. 2.5(a). Observe a classe de figuras semelhantes a flechas mostrada na Fig. 2.5(b). Usando as primitivas da Fig. 2.5(a) é possível construí-las, o ponto de início é a fenda da cauda da flecha; o contorno segue no sentido horário. Portanto esta classe de padrões pode ser exatamente representada pela linguagem na equação (2.15), que novamente é descrita pela gramática da equação (2.14).

As regras de uma gramática formal são uma ferramenta adequada para modelagem de propriedades estruturais de padrões, particularmente pela descrição de como um padrão

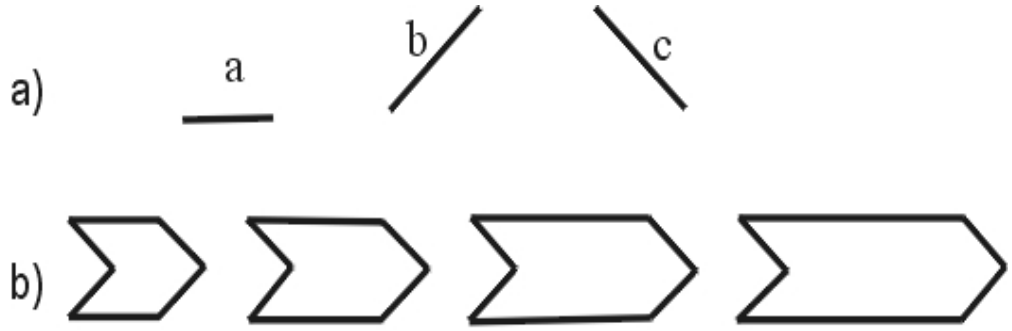

Figura 2.5: (a) Segmentos de linha de tamanho fixo. (b) Exemplos de padrões de uma classe. Adaptada de: [15].

complexo é hierarquicamente composto de elementos mais simples, incluindo relações entre eles. Porém, existem deficiências na adequação de representações de aspecto numérico, tal como tamanho e orientação de linhas, parâmetros de textura de regiões, ou orientação de superfície 3-D. Gramáticas atribuídas determinam uma solução para este problema. A idéia é acrescentar a cada símbolo de gramática $\in V$ um vetor de atributos $\alpha(A) = (\alpha_1(A),\ \alpha_2(A), ...,\ \alpha_n(A))$.

Os componentes $\alpha_i(A)$ podem tomar valores numéricos a partir de um certo domínio. Dado um vetor atributo $\alpha(A)$, ele pode ser interpretado também como um vetor de características (da abordagem estatística), dando uma caracterização numérica e quantitativa aos (sub)padrões representados pelo símbolo A. As propriedades estruturais de uma classe de padrão, que são dadas pelos símbolos da gramática e produções, são usualmente chamadas de componentes sintáticos, enquanto atributos são referidos como informação semântica.

Considerando uma produção $A_1...A_n \rightarrow B_1...B_m,\quad A_i, B_j \in V$, há usualmente um relacionamento entre os atributos de símbolos do lado esquerdo e os atributos de símbolos do lado direito. Dois casos devem ser distinguidos. Primeiro, os atributos do lado esquerdo podem ser dependentes daqueles do lado direito, i.e.

$$\alpha(A_i) = f_i(\alpha(B_1), ...,\ \alpha(B_m)),\ \ i = 1, ..., n, \tag{2.16}$$

ou os atributos do lado direito podem ser dependentes daqueles do lado esquerdo, i.e.

$$\alpha(B_i) = g_i(\alpha(A_1), ...,\ \alpha(A_m)),\ \ i = 1, ..., m. \tag{2.17}$$

Gramáticas atribuídas foram originalmente propostas para modelagem de semânticas de linguagem de programação, e de acordo com [50], os atributos da equação (2.16) são chamados de sintetizados, enquanto os atributos da equação (2.17) são chamados de herdados. Teoricamente, uma gramática pode conter símbolos com atributos de ambos os tipos, mas

por razão de simplicidade, é aconselhável usar somente um tipo de atributo dentro de uma gramática, se possível. O reconhecimento "de baixo para cima[7]" inicia-se com um padrão segmento com todas as primitivas extraídas da mesma maneira que os atributos, para os símbolos terminais que são conhecidos. Então os atributos de símbolos não terminais podem ser sucessivamente computados durante a análise de acordo com a equação (2.16). De modo oposto, no reconhecimento "de cima para baixo[8]" os valores dos atributos para o símbolo inicial da gramática devem ser conhecidos e estes são sucessivamente substituídos por outros símbolos de acordo com a equação (2.17). Observe que f_i e g_i na equação (2.16) e (2.17), respectivamente, podem ser qualquer função matemática de forma fechada ou qualquer outro algoritmo que toma alguns valores de atributos como entrada e produz outros valores de atributos como saída. Em Fu [28] foi introduzida a noção de atributos de gramáticas que unificam as abordagens estatística e sintática para o RP.

2.1.3.3.2 Métodos sintáticos

Classificadores estatísticos, como discutido em seção anterior, são vantajosos em muitos casos. Eles podem ser eficientemente implementados e são poderosos para tratar padrões distorcidos e com ruído. Porém, abordagens estatísticas são limitadas se rica informação estrutural é importante. Outro fator de limitação é que usualmente um grande número de exemplos de padrões são necessários para o projeto de um classificador. Estas limitações podem freqüentemente ser superadas por métodos sintáticos. Existem duas abordagens diferentes para análise sintática, chamadas autômato abstrato [37] e analisador gramatical (*parsers*). O problema da análise tem sido estudado por muitos anos, não somente por pesquisadores em RP mas também no contexto de projeto de compiladores. Um bom achado teórico sobre este assunto encontra-se em Aho [1]. Dependendo do problema de reconhecimento, a abordagem sintática talvez renda uma explosão combinatorial de possibilidades a serem investigadas, exigindo grande conjunto de treinamento e muitos esforços computacionais [72].

2.1.3.4 Redes neurais artificiais

A conectividade de uma rede neural determina sua estrutura. Grupos de neurônios podem ser localmente interconectados para formarem "aglomerados" que são conectados a outros aglomerados de forma solta, imprecisa, ou indireta. Alternativamente, neurônios podem ser organizados em grupos ou camadas que são (direcionalmente) conectadas a outras camadas. Assim sendo, as implementações de abordagem neural para RP requerem uma avaliação inicial de arquiteturas de redes neurais. As possibilidades são [78]: (i) Projetar uma aplicação

[7] *bottom-up*
[8] *top-down*

dependente da estrutura da rede que execute algumas computações desejadas [17]. (ii) Selecionar uma estrutura pré-existente "comumente usada" onde algoritmos de treinamento são disponíveis. Por exemplo, redes diretas *(feedforward)* e Hopfield. (iii) Adaptar uma estrutura do item (ii) para satisfazer uma aplicação específica [44].

Muitas estruturas diferentes de redes neurais "genéricas", indicadas no item (ii), são úteis para uma classe de problemas de RP. Exemplos são:

Associador de Padrão (AP): esta implementação neural é exemplificada pelas redes diretas com seu mecanismo de aprendizado (retropropagação e regra delta generalizada).

Conteúdo Endereçável ou Modelo de Memória Associativa: esta estrutura de rede neural, melhor exemplificada por um modelo de Hopfield, é outra tentativa de construir um sistema de RP com propriedades úteis de associação de padrão.

Redes Auto-organizáveis: estas redes exemplificam implementações neurais de aprendizado não supervisionado, de forma a aglomerar, ou auto-organizar padrões de entrada dentro de classes ou aglomerados, baseadas em alguma forma de similaridade.

Segundo Jain [40], modelos de redes neurais usam alguns princípios organizacionais, tais como aprendizado, generalização, adaptabilidade, tolerância à falha e computação distribuída. A principal diferença entre redes neurais e outras abordagens de RP é sua habilidade de aprender relacionamentos complexos não lineares entre dados de entrada e saída, com o uso de procedimento seqüencial de treinamento. Além disto, elas possuem características gerais de adaptação de si mesmas aos dados.

A família de redes neurais mais usadas para as tarefas de reconhecimento de padrão [41] são as redes diretas, que incluem perceptron multicamada e redes função base radial (RBF[9]). Estas redes são organizadas em camadas, com conexões unidirecionais entre as mesmas. Outra rede popular é o Mapa Auto-Organizável (SOM), ou rede de Kohonen [52], que é usada principalmente para aglomeração de dados e mapeamento de características. O processo de aprendizado envolve atualizações na arquitetura da rede e nos pesos das conexões, de forma que ela possa eficientemente executar uma tarefa específica de classificação/aglomeração.

A popularidade do uso de modelos de redes neurais para problemas de RP é devido sua baixa dependência de conhecimento de domínio específico (relativo às abordagens baseadas em modelos e regras) e devido à disponibilidade de algoritmos eficientes de aprendizado.

Redes neurais provêem vários algoritmos não lineares para extração de características (usando camadas intermediárias) e classificação. Em adição, existem algoritmos de extração

[9] *Radial basis function*

de características e classificação que podem também ser mapeados em arquitetura de redes neurais para implementação eficiente (hardware). Contrariando a aparente diferença de princípios de base , a maioria dos modelos de redes neurais bem conhecidos são implicitamente equivalentes ou similares a métodos estatísticos clássicos de RP (veja Tabela 2.2). Não obstante estas similaridades, redes neurais podem oferecer muitas vantagens tais como,

Abordagem Estatística	**Abordagem Neural**
Função Discriminante Linear	Perceptron
Análise de Componente Principal	Rede Auto-Associativa, e várias redes PCA
Uma Estimativa Probabilística Posteriori	Multicamada Perceptron
Análise de Discriminante Não-linear	Multicamada Perceptron
Classificador baseado em Densidade de Janela Parzen	Redes Função Base Radial
Regras K-NN editadas	Kohonen's LVQ

Tabela 2.2: Ligações entre métodos estatísticos e neurais para o RP. Adaptada de: [78].

unificação de abordagens para extração de características e classificação e procedimentos flexíveis para encontrar soluções não lineares.

2.1.3.4.1 Comparando e relacionando as abordagens estatística, sintática e neural

As fronteiras entre as abordagens estatística, sintática e neural para RP são imprecisas, nebulosas, e fracas, pois elas compartilham características e objetivos comuns. Para um problema de RP específico, escolhe-se uma abordagem em detrimento de outra com base na análise dos fundamentos de componentes estatísticos, dos fundamentos de estrutura gramatical, como também a compatibilidade de uma solução de rede neural, e na ausência de modelo estatístico ou estrutural adequado escolhe-se abordagens do tipo "caixa preta" [78].

Na abordagem estatística ou abordagem de decisão teórica, a estrutura do padrão é freqüentemente desconsiderada. Porém, esta estrutura *pode* ser extraída por uma adequada escolha de características, por exemplo, um vetor de característica binário poderia indicar a presença ou ausência de relações observadas. Similarmente, a abordagem neural para RP é, em alguns casos, uma implementação derivada das abordagens estatística e sintática. Quando se tem disponível informação estrutural explícita sobre o padrão, faz-se a escolha da abordagem sintática. Já quando esta informação não é disponível ou irrelevante, então a abordagem estatística pode ser usada. Muitas aplicações práticas de RP caem entre estes dois extremos. Por exemplo, uma gramática atribuída [83], provê um meio para combinar as abordagens sintática e estatística.

A abordagem de redes neurais para RP é uma área relativamente nova. Porém ela não deve ser considerada como um novo conceito ou meramente um conjunto de técnicas alternativas para a implementação de abordagens estatística ou estrutural. A tabela 2.3 resume estas diferentes abordagens para o RP.

		A. Estatística	A. Sintática	A. Neural
1.	Base de geração (armazenamento) de padrão	Modelos probabilísticos	Gramáticas formais	Estados estáveis ou vetor de pesos
2.	Base de classificação de padrão (Reconhecimento/Descrição)	Teoria de Estimação/Decisão	Parsing	Baseados em propriedades (previsíveis) de NN
3.	Organização de características	Vetor de características	Primitivas e relações observadas	Entradas neurais ou estados armazenados
4.	Treinamento típico			
	Supervisionado:	Estimativa da densidade/distribuição (usualmente paramétrica)	Formação de gramáticas (heurísticas ou inferência gramatical)	Determinação de parâmetros de sistemas de NN (p.e. pesos)
	Não-supervisionado:	Aglomeração	Aglomeração	Aglomeração
5.	Limitações	Dificuldade em expressar informação estrutural	Dificuldade em aprender regras estruturais	Pouca informação semântica da rede

Tabela 2.3: Comparando abordagens estatística, sintática e neural de RP. Adaptada de: [78].

2.1.3.5 Lógica nebulosa

A maioria dos problemas de classificação de padrões são intrinsicamente não adequados a abordagens com formulação matemática precisa. Por causa desta estrutura conceitual, a teoria dos conjuntos nebulosos provê um cenário mais natural para a formulação e solução destes problemas do que as abordagens mais tradicionais baseadas em teoria dos conjuntos clássicos, teoria da probabilidade, e lógica bi-valorada.

A teoria dos conjuntos nebulosos proposta por Zadeh [96], [97], [98] e trabalhos subseqüentes forneceram ferramentas matemáticas adequadas e técnicas para sistemas complexos de análise e processos de decisão, onde a indeterminação do padrão é mais inerente de variabilidade e/ou incerteza do que aleatoriedade. A teoria dos conjuntos nebulosos fornece significante importância em problemas de RP envolvendo outras abordagens tais como teoria de decisão estatística, abordagem sintática e redes neurais. Primeiramente foram desenvolvidos algoritmos nebulosos para diferentes sistemas de RP e problemas de processamento de imagem, seguidos de linguagens nebulosas e teoria de autômatos nebulosos [66].

A relação íntima entre a teoria dos conjuntos nebulosos e a teoria de RP fornece grandes recursos em problemas gerais de tomada de decisão, ambos ambientes randômicos e não-

randômicos, porque a maioria dos problemas de classificação no mundo real são nebulosos por natureza. Uma grande quantidade de literatura relacionada a técnicas nebulosas no reconhecimento de voz, reconhecimento de padrão, processamento de imagem, análise de agrupamento e tópicos relacionados tem sido publicada, mas uma abordagem unificada ainda não está disponível.

2.1.3.5.1 Classificação supervisionada

Um algoritmo de classificação nebuloso quando aplicado a um padrão p, produz $\mu_C(p)$, um grau de pertinência de p com relação à classe C. Isto pode ser observado como o grau de similaridade entre p e um padrão típico ou ideal representando a classe C, conhecido como protótipo (ou modelo). O padrão protótipo pode ser o vetor média no espaço de padrão, fazendo deste um procedimento de similaridade. Quando a descrição explícita do algoritmo de classificação é conhecida, ele é dito ser um algoritmo transparente R_{tr}. Caso contrário, eles podem ser chamados de opacos R_{op}. Dentro do *framework* da teoria dos conjuntos nebulosos, o problema de reconhecimento de padrão pode ser visto como a conversão de um algoritmo de reconhecimento opaco em um algoritmo de reconhecimento transparente.

O RP supervisionado em todas as suas formas tem sido tratado de uma maneira unificada em [7] usando o framework da teoria dos conjuntos nebulosos.

Abstração e generalização

O RP nebuloso supervisionado é realizado em duas operações básicas: "abstração" e "generalização". Dado padrões completamente descritos em um conjunto de medidas como um vetor $X \in \Omega_X \cdot A$, então, um conjunto de treinamento de uma classe nebulosa A de padrões é dada por

$$\{(X_1, \mu_1), (X_2, \mu_2), ..., (X_t, \mu_t)\}$$

onde $\mu_i \in [0, 1]$ é o grau de pertinência do *i*-ésimo padrão X_i, $i = 1, 2, ..., t$.

A abstração, em termos informais, é a identificação das propriedades comuns dos exemplos e a agregação das mesmas para definirem o conjunto A. Mais formalmente, a abstração neste conjunto de treinamento é a função de pertinência μ de A dos exemplos. A generalização é executada quando a estimativa $\langle\ \mu\ \rangle$ de μ é obtida e usada para computar os valores de μ em pontos diferentes de $X_1, X_2, ..., X_t$.

Assim, dados dois conjuntos nebulosos A e B em Ω_X com duas funções de pertinência desconhecidas μ_A, μ_B e também dado um conjunto de treinamento

$$\{(X_1, \mu_1^A, \mu_1^B), (X_2, \mu_2^A, \mu_2^B), ..., (X_i, \mu_i^A, \mu_i^B), ..., (X_t, \mu_t^A, \mu_t^B)\}$$

a abstração envolve a estimativa de μ^A, μ^B, e a generalização envolve o uso de $\langle\ \mu\ \rangle^A$, $\langle\ \mu\ \rangle^B$ para um padrão desconhecido X, não presente no conjunto de treinamento. Isto pode ser facilmente estendido para o caso de m classes ($m > 2$).

É evidente que o problema de classes não nebulosas está incluído como um caso especial onde a função de pertinência é um mapeamento $f : \Omega \longrightarrow \{0, 1\}$ e se distinguir entre duas classes de padrões através uma separação de hiperplano. Para um único conjunto de padrões, o problema em questão é essencialmente encontrar, se existe, um hiperplano L passando pela origem de R^l, supondo que $\Omega_X = R^l$, tal que os pontos dados $X_1, X_2, ..., X_t$ pertencentes a um conjunto A localizam-se todos no mesmo lado do hiperplano.

Zadeh [99] sugere que as características sejam valoradas lingüisticamente, p.e., o tamanho de uma característica pode ter valores 'pequeno', 'muito pequeno', etc. Se existem N características semelhantes, há uma relação nebulosa no universo $Z_1 \times Z_2 \times ... \times Z_n \times ...Z_N$, onde Z_n é o universo da n-ésima característica ϕ_n (que tem valores lingüísticos). Uma classe nebulosa de padrão é uma relação nebulosa sobre $Z_1 \times Z_2 \times ... \times Z_N \times [0, 1]$, isto é, uma classe nebulosa tipo 2. Assim, pode-se visualizar uma tabela da forma:

ϕ_1	$\cdots$	ϕ_n	$\cdots$	ϕ_N	
p_1^1	$\cdots$	p_1^n	$\cdots$	p_1^N	p_1^{N+1}
.					
.					
.					
p_i^1	$\cdots$	p_i^n	$\cdots$	p_i^N	p_i^{N+1}
.					
.					
.					
p_t^1	$\cdots$	p_t^n	$\cdots$	p_t^N	p_t^{N+1}

onde $i = 1, \ldots, t; n = 1, \ldots, N$: p_i^n representa o valor lingüístico da n-ésima característica correspondente ao i-ésimo padrão e p_i^{N+1} representa um valor verdade lingüístico: um conjunto nebuloso sobre [0, 1].

Esta tabela pode ser interpretada como respostas para N questões. A i-ésima linha é uma regra nebulosa que diz: se para um padrão p_i, a primeira característica tem valores lingüísticos p_i^1, a segunda tem $p_i^2, \ldots,$ a n-ésima tem p_i^n, etc., então o grau de pertinência do padrão é p_i^{N+1}.

Logo a classe nebulosa de padrão é dada pela relação

$$\begin{aligned} R &= \bigcup_i \bigcap_n p_i^n \\ i &= 1, 2, \ldots, t \\ n &= 1, 2, \ldots, N+1 \end{aligned} \tag{2.18}$$

Dado um padrão desconhecido X, cujos valores de características são $\phi_1(X), \phi_2(X), \ldots, \phi_N(X)$, seu grau de pertinência à classe é obtido por composições max-min de R.

Quando existirem m classes de padrão, existirão m tabelas relacionais.

2.1.3.5.2 Árvore de decisão nebulosa

Nesta abordagem cada coluna da tabela relacional é interpretada como um conjunto de possíveis respostas para uma questão concernente a n características e assim chega-se a uma árvore de decisão nebulosa para um padrão dado. Uma árvore de decisão nebulosa é uma árvore tal que cada nó não folha i tem uma função k-tupla de decisão

$$f_i : \Omega \rightarrow [0,1]^k$$

e k filhos ordenados. Cada filho não-folha corresponde a possíveis respostas para uma questão prévia e o filho é também associado com uma questão determinada por uma resposta prévia. Para cada ramo na árvore está associado um valor no intervalo de $[0, 1]$. Cada folha corresponde a uma classe de padrão. Cada caminho, partir da raíz até uma folha representa uma tarefa de decisão dos exemplos para a classe correspondente à folha. Cada decisão é valorada pelo mínimo (ou produto) dos valores correspondentes à composição de ramos do caminho. O padrão é determinado à classe cujo caminho da raiz à folha final é o de menor valor, conforme pode ser visto na Fig. 2.6 a seguir.

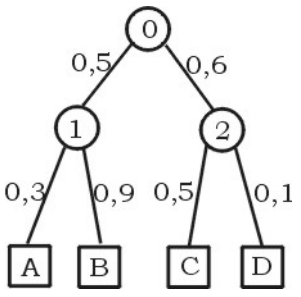

Figura 2.6: Árvore de decisão nebulosa. Adaptada de: [66].

2.1.3.5.3 Distância ponderada e vetores de similaridade

Duas abordagens foram desenvolvidas para classificação de padrões definidos imprecisamente, em problemas com um pequeno número de exemplos, onde a independência estatística não pode ser suposta (aprendizado não-paramétrico). A primeira abordagem requer a computação dos pesos das funções de distância usadas na estimativa do valor de pertinência para cada classe de padrão. A segunda abordagem está baseada na avaliação das propriedades dos conjuntos e encontro dos vetores de similaridade correspondentes a diferentes classes para identificação dos padrões. A Fig. 2.7 mostra um diagrama de transição de estados de um modelo de reconhecimento nebuloso onde $B = (b_1, b_2, \ldots, b_n)$ são os possíveis símbolos de saída para cada entrada. $\mu_1, \mu_2, \ldots, \mu_n$ são as funções de pertinência correspondentes às saídas associadas a cada transição de saída.

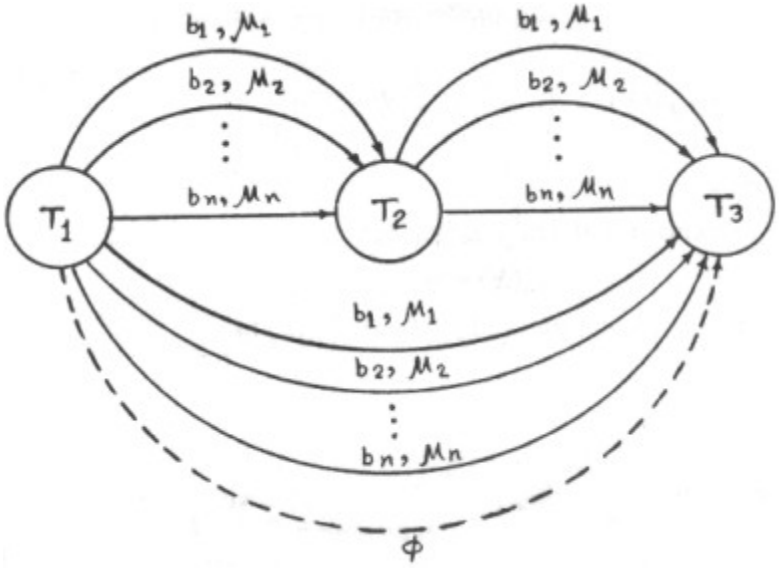

Figura 2.7: Diagrama de transição de estados de um modelo de reconhecimento nebuloso generalizado. Adaptada de: [66].

Transições nulas, chamadas deleções, não têm nenhuma saída e são representadas pelas linhas pontilhadas entre o estado inicial T_1 e o estado final T_3. Outras transições descritas como $T_1 \rightarrow T_3$ representam substituições e as transições descritas como $T_1 \rightarrow T_2 \rightarrow T_3$, produzem duas saídas devido a segmentações incorretas do símbolo de entrada, são denotadas como inserções. Se a segmentação é perfeita (supervisionada), os erros de deleção e inserção não estarão presentes mas o erro de substituição devido à falsa classificação pode ocorrer.

2.1.3.5.4 Abordagem sintática nebulosa

Na seção 2.1.3.3 foram explicados os princípios do RP sintático. Esta abordagem utiliza-se

da representação de um padrão através de uma cadeia de sub-padrões concatenados, chamados primitivas. Estas primitivas são consideradas símbolos terminais do alfabeto de uma gramática formal, cuja linguagem consiste de um conjunto de padrões pertencentes à mesma classe, desta forma o reconhecimento sintático envolve uma análise desta cadeia.

Esta proposta faz que a abordagem sintática incorpore os conceitos dos conjuntos nebulosos em dois níveis. Primeiro, considerando as primitivas de padrões como entidades nebulosas, i.e., tais sub-padrões seriam considerados, p.e., como “arcos quase circulares”, curvas “branda”, “moderada” e “aguda”, etc. Em segundo, com as relações estruturais nebulosas entre os sub-padrões, dado que a gramática formal é nebulisada por regras de produção ponderadas e o grau da função de pertinência de uma cadeia é obtido por composições min-max dos graus das produções usadas nas derivações. A inferência de uma gramática nebulosa que a partir de uma linguagem nebulosa específica infere tanto as produções assim como os pesos das regras, é outro problema interessante. Mais detalhes sobre gramáticas nebulosas, autômato nebuloso, inferência de gramáticas nebulosas e métodos de reconhecimento sintático podem ser vistos em Pal [66].

2.1.3.5.5 Classificação não supervisionada

A classificação não supervisionada pode ser resolvida por um método de análise de dados chamado agrupamento. O objetivo da análise de cluster é particionar o conjunto de dados dado em um certo número de conjuntos naturais e homogêneos, onde os elementos de cada conjunto são tão similares quanto possível e dissimilares entre conjuntos diferentes. O número de tais conjuntos tanto pode ser fixado previamente como pode resultar de uma conseqüência de restrições impostas. Vários algoritmos foram desenvolvidos com o intuito de se obter clusters a partir de um conjunto de dados. Uma análise destes algoritmos convencionais pode ser encontrada em Anderberg [2] e Tou [82].

Em todos os algoritmos clássicos, é implícita a suposição da existência de clusters disjuntos em um conjunto de dados, enquanto, em muitos casos na prática, os clusters não estão completamente disjuntos e a separação dos mesmos é uma noção nebulosa. O conceito de subconjuntos nebulosos oferece especial vantagem sobre técnicas convencionais de agrupamento e permite uma representação adequada a clusters intratáveis tais como os mostrados na Fig. 2.8, cujas dificuldades encontradas são: (a) aglomerados alongados; (b) aglomerados ligados; (c) aglomerados não linearmente separáveis; (d) aglomerados esféricos; (e) aglomerados não compactos e (f) populações de aglomerados desiguais. A técnica a seguir foi desenvolvida com a introdução da teoria dos conjunto nebulosos.

Figura 2.8: Diferentes tipos de aglomerados. Adaptada de: [66].

Agrupamento por partição nebulosa O problema do agrupamento nebuloso foi primeiramente estudado por Ruspini [77]. Ele introduziu a noção de partição nebulosa para representar aglomerados em um conjunto de dados. Ele apresenta o agrupamento nebuloso como a quebra da função densidade probabilidade do conjunto de dados em uma soma ponderada de densidades probabilidade dos aglomerados componentes. Estas densidades são interpretadas como a representação do grau de pertinência de cada ponto a cada aglomerado. Uma partição nebulosa é uma família de conjuntos nebulosos $F_1 \ldots F_m$ sobre $X(=\{x\})$ tal que

$$\begin{aligned} \sum_j \mu_{F_j}(x) &= 1 \quad para\ \ todo\ \ x \in X \\ j &= 1, 2, \ldots, m \end{aligned} \tag{2.19}$$

De acordo com Ruspini as vantagens de uma representação de conjunto nebuloso em análise de aglomerado são que pontos perdidos ou pontos isolados entre aglomerados, bem como outros tipos de incertezas, podem ser classificados. O problema do agrupamento nebuloso é enunciado como segue.

Seja X um conjunto de dados finito e d uma função de valor positivo real (função de distância ou de dissimilaridade) cujo domínio é X^2 tal que

$$d(x,x) = 0 \quad para\ \ todo\ \ x \in X \quad e \tag{2.20}$$

$$d(x,y) = d(y,x) \quad para\ \ todo\ \ x, y \in X \tag{2.21}$$

Uma partição nebulosa $F_1 \ldots F_m$ onde m é conhecido, a priori, é tal que elementos "próximos" em X (no sentido de d) terão classificações similares e elementos dissimilares terão diferentes classificações. A classificação de um elemento x é o vetor $C(x) = [\mu_{F_1}(x), \ldots \mu_{F_m}(x)]$.

Uma das possíveis maneiras de satifazer o requerimento acima é selecionar a função $C(x)$ de forma que ela minimize adequadamente algum requerimento funcional.

Seja υ uma função de $[0,1]^m \times [0,1]^m$ em $\Re^+$ (conjunto dos números positivos reais) tal que $\upsilon(a,a) = 0$ e $\upsilon(a,b) = \upsilon(b,a)$ e seja f uma função real de uma variável real positiva não decrescente e não identicamente zero satisfazendo $f(0) = 0$. Então a função C deveria ser selecionada tal que

$$x, y \in X, \upsilon(C(x), C(y)) = f(d(x,y)) \tag{2.22}$$

Geralmente, esta equação não tem solução e isto torna-se um problema de minimização. Encontrando a minimização de C

$$\sum_{x,y \in X} w(x)w(y)[\upsilon(C(x), C(y)) - f(d(x,y))]^2 \tag{2.23}$$

onde w é uma função de peso apropriada. Usualmente υ é dada como a distância Euclidiana. Outras formas de f são propostas. Uma abordagem levemente diferente usa a medida de associação entre um ponto x e um conjunto nebuloso F sobre X como o inverso de uma distância média ponderada entre x e F, esta distância média entre x e F é definida como

$$d(x,F) = \frac{1}{|F|} \sum_{i=1}^{|X|} \mu_F(x_i) d(x, x_i) \tag{2.24}$$

onde $|F|$ e $|X|$ denotam a cardinalidade, i.e., o número de suportes/elementos em F e X, respectivamente. A idéia básica é que o valor de pertinência de x a um aglomerado nebuloso $F_j (j = 1, 2, \ldots, m)$ varia inversamente proporcional à distância média entre x e F_j.

Ao invés de definir uma partição nebulosa $(F_1 \ldots F_m)$ pela condição de ortogonalidade dada na Eq. 2.19, Zadeh [99] propôs a propriedade de proximidade nebulosa para caracterizar aglomerados nebulosos induzidos por uma relação nebulosa R. Assim as m partições nebulosas $F_1 \ldots F_m$ satisfazem então a propriedade de proximidade nebulosa se as seguintes condições são satisfeitas:

1. Ambos elementos x e y de X tem alto grau de pertinência em algumas F_i, se e somente se, (x, y) tem um alto grau de pertinência em R.

2. Se $x \in X$ tem um alto grau de pertinência em alguma F_i e $y \in X$ tem um alto grau de pertinência em alguma $F_j,\ j \neq i$, então (x, y) não têm um alto grau de pertinência em R.

A partir desta nova visão de partição nebulosa como um conjunto nebuloso de aglomerados nebulosos, foi desenvolvida uma nova abordagem para representação de aglomerado.

Dada uma relação de equivalência R sobre $X \times X$ e denotada por $R(x)$ do conjunto $\{y \in X, \mu_R(x,y) = 1\}$, um subconjunto não nebuloso C de X é dito ser uma representação $R - aglomerado$ de X se e somente se

$$\bigcup_{x \in C} R(x) = X \tag{2.25}$$

Se C não contém nenhum subconjunto próprio, C é dito ser uma representação mínima de X. Quando R é nebulosa e X é finito, um conjunto nebuloso C é uma R-representação de X se e somente se

$$\sum_{x \in X} \mu_R(x,y)\mu_C(x) \geq 1 \ \ para \ \ todo \ \ y \in X \tag{2.26}$$

Dentro da análise classificatória nebulosa ainda existem outros aspectos a serem abordados tais como o da intersecção de partições nebulosas, agrupamento por decomposição de conjuntos nebulosos induzidos e validade de aglomerado, maiores detalhes estão presentes em Pal [66].

2.1.3.6 Abordagem neuro-nebulosa

Redes neurais são projetadas na tentativa de imitar e emular o cérebro humano a fim de se obter um desempenho inteligente. Já a importância dos conjuntos nebulosos situa-se na habilidade de se modelar dados incertos e ambíguos, tão freqüentemente encontrados no mundo real. Portanto, para capacitar um sistema a cuidar de situações da vida real de maneira mais parecida com a humana, evoca-se a necessidade da incorporação do conceito de conjuntos nebulosos dentro de redes neurais. Embora a lógica nebulosa seja um mecanismo natural para propagação de incerteza, ela pode envolver, em alguns casos, um aumento na quantia da computação requerida (comparado com um sistema usando lógica binária clássica). Porém, isto pode ser adequadamente compensado usando-se modelos de redes neurais nebulosas, obtendo assim o potencial da computação paralela com a alta flexibilidade. Conceitos nebulosos já tem sido incorporado dentro de redes neurais em problemas de controle, na modelagem das saídas de distribuições de possibilidades, na aprendizado e extrapolação de relacionamentos complexos entre antecedentes e conseqüentes de regras e em raciocínio nebuloso [9], [10], [11] e [12] apud [67].

Segundo L.M. Brasil [13], as RNAs e a lógica nebulosa podem ser combinadas para formarem sistemas híbridos, provendo dois modos complementares de modelagem da organização complexa e dicotômica do cérebro humano. Onde as RNAs modelam os processos de ordem celular, isto é, a fisiologia do cérebro, enquanto que a lógica nebulosa supre propriedades de uma modelagem psicológica da mente [104] apud [13].

A nível sintático, uma RNA pode ser descrita de forma a mostrar um mapeamento nebuloso. E a nível semântico mostra-se que há uma correspondência entre RNA e a lógica nebulosa, seja na forma de função de pertinência ou operações com os conectivos nebulosos.

Assim, o paradigma simbólico nebuloso pode ser representado nas conexões de uma RNA por uma função de pertinência. Ou pode ser representado ainda em termos das unidades de processamento (neurônios), na função de ativação, na operação de confluência, onde, dependendo do tipo de neurônio, pode ser substituído por um dos operadores de agregação da lógica nebulosa.

Assim, pode-se observar que há uma equivalência semântica entre estes dois paradigmas, conforme é mostrado na Fig.2.9.

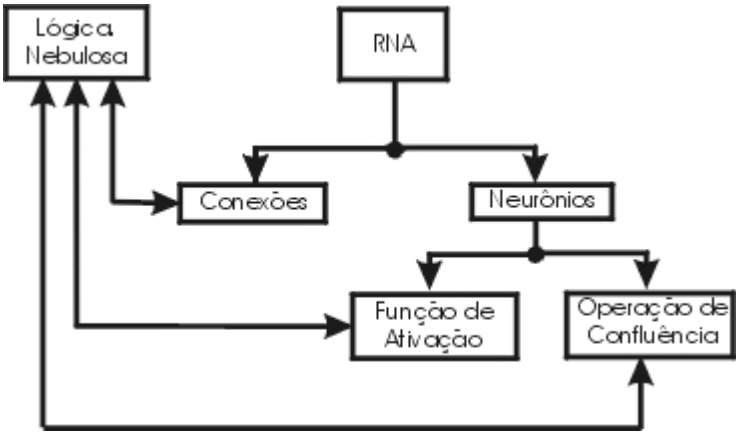

Figura 2.9: Nível semântico de uma rede neuro-nebulosa. Adaptada de [13].

2.1.3.6.1 Redes RBF's e sistemas de inferência nebulosa

Jang e Sun [45] demonstram a equivalência funcional entre uma rede neural de função base radial (RBF) e uma classe de sistemas de inferência nebulosa. Apesar das motivações destes dois modelos terem origens diferentes (RBF's a partir da psicologia e sistema de inferência nebulosa a partir da ciência cognitiva), eles compartilham características comuns não somente em suas operações sobre os dados, mas também em seu processo de aprendizado. O trabalho mostrou que, sob poucas restrições, eles são funcionalmente equivalentes; os algoritmos de aprendizado e o poder representacional de um modelo pode ser aplicado ao outro, e vice-versa.

Redes função base radial

Campo receptivo localmente modularizado e sobreposto é uma estrutura muito estudada em regiões do córtex cerebral, p.e. o córtex visual, etc. Baseado em campos receptivos

biológicos, Moody e Darken [6], [7] apud [45] propuseram uma estrutura de rede, RBF, que emprega campos receptivos locais para executar mapeamentos de funções. A Fig. 2.10 mostra o diagrama esquemático de uma RBF com cinco unidades de campo receptivo; a saída do $i$ésimo campo receptivo (ou unidade intermediária) é

$$w_i = R_i(\vec{x}) = R_i(\|\vec{x} - \vec{c_i}\|/\sigma_i) \;\; i = 1, 2, ..., H \tag{2.27}$$

onde $\vec{x}$ é um vetor de entrada N dimensional, $\vec{c_i}$ é um vetor com a mesma dimensão de $\vec{x}$, H é o número de unidades de campo receptivo, e $R_i(\cdot)$ é a resposta do $i$ésimo campo receptivo com um único máximo na origem. Tipicamente, $R_i(\cdot)$ é uma função do tipo Gaussiana

$$R_i(\vec{x}) = \exp\left[-\frac{\|\vec{x} - \vec{c_i}\|^2}{\sigma_i^2}\right] \tag{2.28}$$

Assim a função base radial w_i computada pela $i$ésima unidade intermediária é máxima quando o vetor de entrada $\vec{x}$ está próximo do centro $\vec{c}$ daquela unidade.

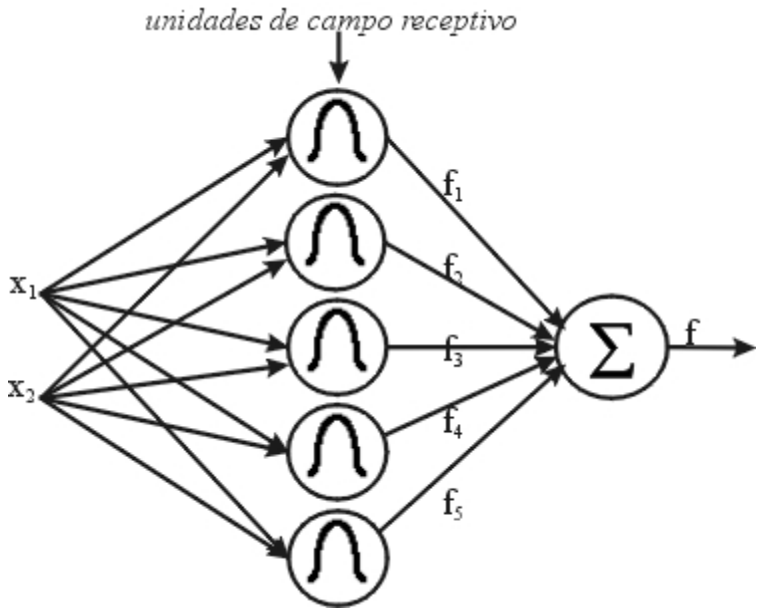

Figura 2.10: Uma rede função base radial. Adaptada de [45].

A saída de uma RBF pode ser computada de duas maneiras. Na forma mais simples, como mostrada na Fig. 2.10, a saída é a soma ponderada do valor da função associada com cada campo receptivo:

$$f(\vec{x}) = \sum_{i-1}^{H} f_i w_i = \sum_{i-1}^{H} f_i R_i(\vec{x}) \tag{2.29}$$

onde f_i é o valor da função, ou força, do $i$ésimo campo receptivo. Com a adição de conexões laterais (não mostradas na Fig. 2.10) entre as unidades de campo receptivo, a rede pode

produzir a resposta normalizada como a média ponderada das forças [6] apud [45]:

$$f(\vec{x}) - \frac{\sum_{i=1}^{H} f_i w_i}{\sum_{i=1}^{H} w_i} = \frac{\sum_{i=1}^{H} f_i R_i(\vec{x})}{\sum_{i=1}^{H} R_i(\vec{x})} \tag{2.30}$$

Para minimizar o erro quadrado entre a saída desejada e a saída do modelo, muitos algoritmos de aprendizagem tem sido propostos para identificar os parâmetros ($\vec{c_i}$, $\vec{\sigma_i}$, e f_i) de uma RBF. Moody *et al.* [6] apud [45] usa técnicas auto-organizáveis para encontrar os centros ($\vec{c_i}$) e as larguras (σ_i) dos campos receptivos, e então emprega o algoritmo supervisionado Adaline ou a regra de aprendizado LMS (algoritmo *mínimo quadrado médio*) para identificar f_i. Por outro lado, Chen *et al.* [1] apud [45] aplica o algoritmo de aprendizado de mínimo quadrado ortogonal para determinar aqueles parâmetros.

Regras "se-então" nebulosas e sistemas de inferência nebulosa

Um exemplo de *regras "se-então" nebulosas* (ou *expressão condicional nebulosa*) é

Se pressão é alta, então volume é pequeno.

onde *pressão* e *volume* são *variáveis lingüísticas*, *alta* e *pequeno* são *valores* ou *rótulos lingüísticos* caracterizados pela função de pertinência apropriada. Outro tipo de regra "se-então" nebulosa, proposta por Takagi e Sugeno[10] apud [45], possui conjuntos nebulosos envolvidos somente na parte premissa. Por exemplo, a dependência da resistência do ar (força) sobre a velocidade de um objeto em movimento pode ser descrita como

Se velocidade é alta, então força $= k * (velocidade)^2$.

onde *alta* é somente o rótulo lingüístico aqui, e a parte conseqüente é descrita por uma equação não nebulosa da variável de entrada, velocidade.

Sistemas de inferência nebulosa são também conhecidos como *sistemas baseado em regras nebulosas, modelos nebulosos, memórias associativas nebulosas*, ou *controladores nebulosos* quando usados como controladores. Um sistema de inferência nebulosa é composto de um conjunto de regras "se-então" nebulosas, um banco de dados contendo funções de pertinência de rótulos lingüísticos, e um mecanismo de inferência chamado *raciocínio nebuloso*. Suponha que tem-se uma regra consistindo de duas regras nebulosas "se-então" do tipo Takagi e Sugeno:

Regra 1 : Se x_1 é A_1 e x_2 é B_1, então $f_1 = a_1x_1 + b_1x_2 + c_1$.

Regra 2 : Se x_1 é A_2 e x_2 é B_2, então $f_2 = a_2x_1 + b_2x_2 + c_2$.

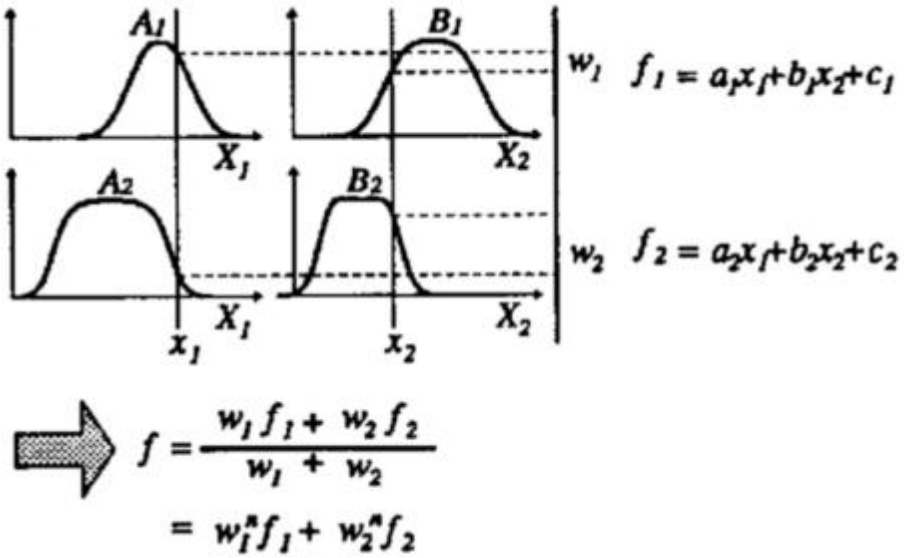

Figura 2.11: Raciocínio nebuloso. Adaptada de [45].

então o mecanismo de raciocínio nebuloso pode ser ilustrado na Fig. 2.11

T-norma (usualmente operador de mínimo ou multiplicação) dos valores de pertinência da parte premissa

$$\begin{aligned} w_i &= \mu_{A_i}(x_1)\mu_{B_i}(x_2), \quad ou \\ &= min\{\mu_{A_i}(x_1), \mu_{B_1}(x_2)\}. \end{aligned} \tag{2.31}$$

Note que a saída completa pode ser escolhida como a soma ponderada de cada saída de regra [9], [2] apud [45]

$$f(\vec{x}) = \sum_{i=1}^{R} w_i f_i \tag{2.32}$$

ou mais convencionalmente, como a média ponderada [10] apud [45] (como mostrado na Fig. 2.11):

$$f(\vec{x}) = \frac{\sum_{i=0}^{R} w_i f_i}{\sum_{i=0}^{R} w_i} \tag{2.33}$$

onde R é o número de regras "se-então" nebulosas.

Modelagem nebulosa preocupa-se com a identificação da estrutura (número de regras, partição de padrões, etc.) e parâmetros de sistemas de inferência nebulosa. Jang [4], [5] e [3] apud [45] propõe um método mais direto que transforma o sistema de inferência nebulosa da Fig. 2.11 em uma rede adaptativa equivalente, Fig. 2.12. A partir daí emprega-se o gradiente descendente (retropropagação) para atualizar os parâmetros da premissa (que determinam as formas e posições das funções de pertinência) e o método do mínimo quadrado para identificar parâmetros conseqüentes (que especificam a saída de cada regra). Na rede adaptativa

proposta (Fig. 2.12), não há peso associado a cada ligação e os nós em diferentes camadas podem ter diferentes funções, correspondentes a cada passo de um mecanismo de raciocínio nebuloso. Mais especificamente, a camada 1 calcula valores de pertinência, a camada 2 executa o operador T-norma, a camada 3 computa pesos normalizados, a camada 4 deriva o produto das saídas de cada regra e pesos normalizados correspondentes, e a camada 5 soma estas entradas produzindo a saída total.

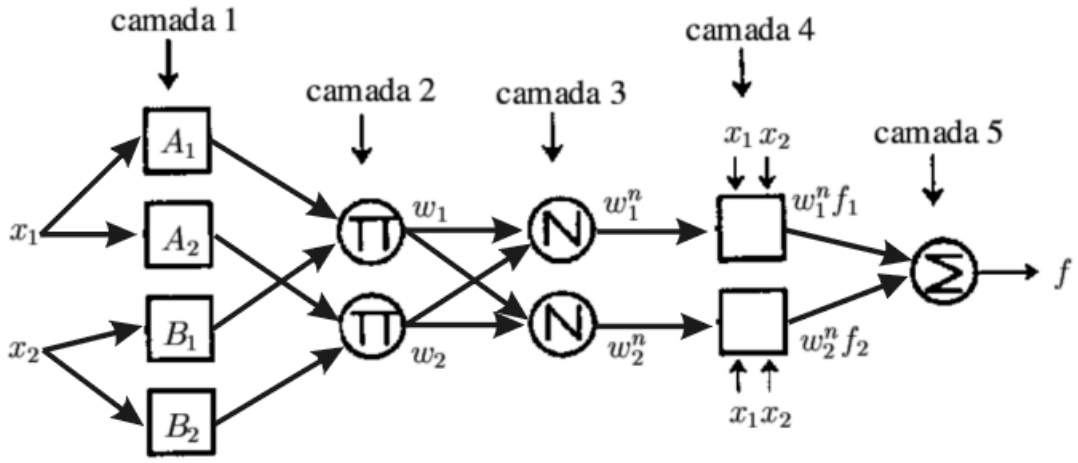

Figura 2.12: Representação da rede adaptativa. Adaptada de [45].

A partir das equações (2.29), (2.30), (2.32) e (2.33), torna-se óbvio que a equivalência funcional entre uma RBF e um sistema de inferência nebulosa pode ser estabelecida se as seguintes condições são satisfeitas:

1. O número de unidades de campo receptivo é igual ao número de regras “se-então” nebulosas.

2. A saída de cada regra “se-então” nebulosa é composta de uma constante (ou seja, a_1, b_1, a_2 e b_2 são zeros na Fig.2.11).

3. As funções de pertinência dentro de cada regra são escolhidas como funções Gaussianas com a mesma variância.

4. O operador T-norma usado para computar cada força de disparo de regra é a multiplicação.

5. Ambos RBF e sistemas de inferência nebulosa usam o mesmo método (i. e., média ou soma ponderada) para derivar suas saídas completas.

Sob estas condições, as funções de pertinência das variáveis lingüísticas $A1$ e $B1$ na Fig. 2.11 podem ser expressas como

$$\begin{aligned} \mu_{A_1}(x_1) &= \exp\left[-\frac{(x_1 - c_{A_1})^2}{\sigma_1^2}\right]. \\ \mu_{B_1}(x_2) &= \exp\left[-\frac{(x_2 - c_{B_1})^2}{\sigma_1^2}\right]. \end{aligned} \tag{2.34}$$

Portanto o peso da regra 1 (a saída do primeiro nó na camada 2) é:

$$w_1(x_1, x_2) = \mu_{A_1}(x_1)\mu_{B_1}(x_2) = \exp\left[-\frac{\|\vec{x} - \vec{c_1}\|^2}{\sigma_1^2}\right] = R_1(\vec{x}). \tag{2.35}$$

onde $\vec{c_1} = (c_{A_1}, c_{B_1})$ é o centro do campo receptivo correspondente. O mesmo argumento aplica-se a w_2. Então sob as condições acima, as saídas das Fig. 2.11 e Fig. 2.12 são exatamente as mesmas de uma RBFN[10] (com dois campos receptivos), onde as unidades de campo receptivo e as unidades saídas são funcionalmente equivalentes para as camadas em cascatas 1, 2 e camadas 3, 4 e 5, respectivamente. Sem as condições acima, as RBF's são somente um caso especial de sistemas de inferência nebulosa.

Por causa da equivalência funcional mostrada acima, aplica-se o que é conhecido sobre um modelo no outro e vice-versa. Em outras palavras, aplica-se o aprendizado de regras das RBF's vistas na Seção 2.1.3.6.1 em um sistema de inferência nebulosa, e o aprendizado de regras do sistema de inferência nebulosa, exposto na Seção 2.1.3.6.1, pode ser utilizado para encontrar a estrutura (i.e., número de unidades de campo receptivo) e parâmetros de RBF's. Além disso, recentemente Wang[12] apud [45] provou que um sistema de inferência nebulosa com função de pertinência Gaussiana escalada, conforme apresentada na Equação 2.36.

$$\mu_A(x) = k * \exp\left[-\frac{(x - c)^2}{\sigma^2}\right] \tag{2.36}$$

é um aproximador universal que pode aproximar arbitrariamente bem quaisquer dados de entrada e saída não lineares sobre um conjunto compacto. Este argumento pode ser aplicado a RBF's se a resposta do campo receptivo em (2.7) é também escalada por uma constante.

Pesquisadores têm proposto redes neurais diretas com neurônios nebulosos [54], que apresentam ótimo desempenho quando usadas para reconhecer padrões de treinamentos alterados e distorcidos. Este método foi utilizado para reconhecimento de letras e números em matrizes de 16×16 pixeis.

Em [33] os autores Grohman e Dhawan questionam a eficiência do algoritmo padrão de retropropagação em sistemas de diagnóstico médico, pois não é inerente ao mesmo a

[10] *Radial basis function network*

análise do espaço de características do problema, durante o treinamento, e isto pode, às vezes, resultar em superfícies de decisão inadequadas. Eles propuseram um novo algoritmo neuro-nebuloso para a classificação de diagnósticos difíceis de casos de câncer em seio. Este método de treinamento, chamado classificador de padrões neuro-nebulosos, primeiro identifica aglomerados dentro do conjunto de treinamento, e então constrói a arquitetura atual da rede. Esta característica aumenta significativamente a robustez minimizando o risco de se cair em mínimos locais, e a classificação do padrão está baseada em dados aglomerados. Isto permite uma imunidade superior a ruído nos padrões de treinamento e a melhora do desempenho do classificador. E o benefício adicional deste método é que ele auto determina a estrutura da rede, algo que elimina a necessidade de heurísticas para determinação de neurônios e camadas, problema comum na utilização de redes diretas.

A definição do conceito de "robustez e invariância" em sistemas de RP conforme [48] diz que primeiro, o modelo deve reconhecer objetos que são transladados, escalonados e rotacionados. Segundo, o sistema deve ter forte resistência a ruído e finalmente, depois de completamente treinado, deve ser capaz de reconhecer novos objetos em outras categorias, sem alterar qualquer parâmetro do modelo. Para isto uma rede neural-nebulosa, *ART Nebulosa*, com aprendizado não supervisionado foi proposta para solução do dilema da estabilidade-plasticidade, produzindo assim robustez ao modelo.

Experimentos comparando redes neurais treinadas com saídas desejadas abruptas e nebulosas foram descritos em Gader [30]. Um algoritmo para reconhecimento de palavra escrita à mão usando rede neural foi testado sobre imagens de palavras extraídas do Serviço Postal dos Estados Unidos. As saídas nebulosas foram definidas usando o algoritmo do k-vizinho mais próximo nebuloso. As redes abruptas foram levemente melhores do que as redes nebulosas em nível de caracter, mas as redes nebulosas tiveram melhor desempenho em nível de palavras.

Também foi proposto em Mitra [60] um modelo de sistema especialista conexionista, baseado em uma versão nebulosa de redes diretas, capaz de interrogar ao usuário informações com relação às características dos padrões, no caso de entradas parciais. As justificativas sobre a decisão inferida podem ser produzidas em forma de regras, quando desejadas pelo usuário. Os valores dos pesos das conexões da rede neural treinada são utilizados em todos os estágios do procedimento de inferência. Partes antecedentes e conseqüentes das justificativas de regras são fornecidas em forma *natural*. A efetividade do algoritmo foi testada com relação ao problema de reconhecimento de voz, alguns dados médicos e classes de padrões intratáveis gerados artificialmente (não linearmente separáveis).

2.1.3.7 Algoritmos genéticos para classificação de padrões

Algoritmo genético (AG) é um processo estocástico de busca em um espaço complexo e multimodal. É um método que se utiliza do domínio de conhecimento específico, em forma de função objetiva, para executar uma busca aleatória direta. AGs estão sendo gradualmente aplicados nas mais variadas áreas. Nesta seção é apresentada a forma como AGs são aplicados à classificação de padrões.

Aplica-se AG em RP para a classificação de um padrão em um espaço N dimensional. O AG localiza H hiperplanos (fronteiras de decisão) em um determinado espaço de características, de modo a obter o mínimo de falsas classificações para os exemplos.

Segundo Pal [68], em AGs, uma solução individual do problema é a codificação em forma de cadeias ou cromossomos. Uma função objetiva é associada com cada cadeia fornecendo um mapeamento do espaço cromossômico ao espaço do problema. Cria-se uma população inicial por seleção aleatória de um número fixo de cadeias. Operadores biologicamente inspirados como *cruzamento* e *mutação* são então aplicados sobre elas para produzirem um nova coleção de cadeias. Baseado no princípio da *sobrevivência do mais adequado*, poucas delas são selecionadas (algumas mais de uma vez) para formar uma nova população. Este ciclo de cruzamento, mutação, e seleção é repetido um número de vezes até que uma condição de término é encontrada. A melhor cadeia verificada na última geração geralmente provê a solução do problema sob consideração.

No domínio de classificação de padrões usando um número fixo de (H) hiperplanos, os cromossomos representam um conjunto de tais hiperplanos. A função objetiva associada é o número de exemplos classificados erroneamente ($erros$) pelo conjunto de hiperplanos codificados na cadeia. A função de adaptação que o AG tenta maximizar é o número de pontos corretamente classificados por uma cadeia, i.e., $adapt = n - erros$, onde n é o número total de exemplos de treinamento.

2.1.3.7.1 Representação da cadeia

Na geometria elementar, a equação de um hiperplano no espaço N dimensional ($X_1 - X_2 - \ldots - X_N$) é dado por

$$x_N \cos \alpha_{N-1} + \beta_{N-1} \sin \alpha_{N-1} = d$$

onde

$$\begin{aligned} \beta_{N-1} &= x_{N-1} \cos \alpha_{N-2} + \beta_{N-2} \sin \alpha_{N-2} \\ \beta_{N-2} &= x_{N-2} \cos \alpha_{N-3} + \beta_{N-3} \sin \alpha_{N-3} \\ &\vdots \end{aligned}$$

$$\beta_1 \quad = \quad x_1 \cos\alpha_0 + \beta_0 \sin\alpha_0$$

Estes parâmetros são descritos a seguir:

$(x_1, x_2, \ldots, x_N)$:	um ponto no hiperplano
α_{N-1}:	o ângulo que a unidade normal do hiperplano faz com o eixo X_N
α_{N-2}:	o ângulo que a projeção da normal no espaço $(X_1 - X_2 - \cdots - X_{N-1})$ faz com o eixo X_{N-1}
$\vdots$	
α_1 :	o ângulo que a projeção da normal no plano $(X1 - X2)$ faz com o eixo X_2
α_0 :	o ângulo que a projeção da normal no plano $(X1)$ faz com o eixo $X_1 = 0$
d :	a distância perpendicular do hiperplano a partir da origem

Assim, a N tupla $< \alpha_1, \alpha_2, \ldots, \alpha_{N-1}, d >$ especifica um hiperplano no espaço N dimensional.

Cada ângulo $\alpha_j, j = 1, 2, \ldots, N-1$ pode variar no intervalo de 0 a 2π, com a precisão definida pelo número de bits usados para representar um ângulo. Para especificar d, o hiper-retângulo contendo os pontos de exemplos é considerado. Para $N-1$ ângulos, $\alpha_1, \ldots, \alpha_{N-1}$ (i. e. para uma dada orientação), o hiperplano passando por um dos vértices do hiper retângulo e tendo uma distância mínima, d_{min}, a partir da origem, é especificado como a base do hiperplano. Seja $diag$ o tamanho da diagonal do hiper retângulo e sejam os bits usados para especificarem d capazes de gerar valores, ditos $offset$, no intervalo de $[0, diag]$. (O número de bits de novo controla a precisão de d). Então $d = d_{min} + offset$ [68].

2.1.3.7.2 Operadores genéticos

Um conjunto de H hiperplanos, cada um compreendendo $N-1$ ângulos variáveis e d, é codificado em um simples cromossomo. Um cromossomo é então representado por uma tupla $H < \mathcal{H}_1, \mathcal{H}_2 \ldots \mathcal{H}_H >$ onde cada $\mathcal{H}_i$, $i = 1, 2, \ldots, H$ é representado por uma N tupla $< \alpha_1, \alpha_2, \ldots, \alpha_{N-1}, d >$.

Para inicializar a população, Pop cadeias binárias são geradas de modo aleatório. A adaptação de uma cadeia é definida como o número de pontos corretamente classificados pelos H hiperplanos codificados nela. Para a computação da adaptação de uma cadeia, os parâmetros $\alpha_1, \alpha_2, \ldots, \alpha_{N-1}$ e d, correspondentes a cada hiperplano são extraídos. Estes são usados para determinar a região onde cada ponto do padrão de treinamento se posiciona.

Uma região demarca a classe i se o número máximo de pontos que posicionam-se nesta região são também da classe i. Qualquer empate é resolvido arbitrariamente. Todos os outros pontos nesta região são considerados mal classificados. As más classificações para todas as regiões são somadas para se obter o número de más classificações, *miss*, para uma cadeia inteira. Sua adaptação é então definida em *n-miss* [68].

Seleção por *Roleta* é usada para implementar a *estratégia de seleção proporcional*. A eleição é incorporada pela substituição da pior cadeia da geração atual pela melhor cadeia vista na última geração. Cruzamento de *ponto-simples* é aplicado com um valor fixo de 0.8 de probabilidade de cruzamento. A operação de mutação é realizada sobre uma base bit-a-bit para um valor de probabilidade de mutação que é inicialmente alto, então decresce gradualmente para um valor mínimo pré-estabelecido e então aumenta novamente mais tarde em outros estágios do algoritmo. Isto assegura que em um estágio inicial, quando o algoritmo tem pouco conhecimento sobre o domínio de busca, ele execute uma busca aleatória através do espaço de características. Esta aleatoriedade é gradualmente decrementada com a morte de gerações, de forma que o algoritmo execute uma busca detalhada na vizinhança de soluções promissoras até o momento. Apesar disto, o algoritmo pode ainda ficar preso em um mínimo local. Este problema pode ser superado através do aumento do valor de probabilidade de mutação, fazendo a busca mais aleatória de novo. O algoritmo termina seu processamento se a população contém ao menos uma cadeia com nenhuma falsa classificação de exemplos e não existe melhora significante na adaptação média da população atual sobre as gerações subseqüentes [68].

2.2 Reconhecimento de faces

2.2.1 Introdução

Um sistema de reconhecimento de face é um sistema biométrico que identifica ou verifica seres humanos através de uma característica exclusiva, por exemplo, a face. Sistemas biométricos capturam atributos inerentes a cada indivíduo em particular e que podem ser medidos. Alguns exemplos de características humanas usadas para biometria são mostradas na Tab. 2.4 [42]:

O reconhecimento de face a partir de imagens estáticas e imagens de vídeo há décadas vem emergindo como uma atividade na área de pesquisa com numerosas aplicações comerciais, industriais e de imposição de lei. Estas aplicações, cada vez mais requerem algoritmos robustos que atuem sobre diferentes condições de iluminação, expressões faciais e orientações.

Grandeza biométrica	Características observadas
Assinatura	o padrão, velocidade, aceleração e a pressão da caneta ao escrever uma assinatura
Impressão digital	padrão dos sulcos cutâneos da superfície da ponta do dedo
Voz	a maneira como humanos geram sons a partir das regiões vocais, boca, cavidades nasais e lábios
Íris	a região circular do olho limitada pela pupila e o glóbulo branco
Retina	o padrão formado pelas veias abaixo da superfície da retina em um olho
Mão	medições da geometria da mão humana
Orelha	medições da geometria da orelha humana
Termografia facial	o calor emitido através do rosto
Face	medições de perfil, frontal e forma

Tabela 2.4: Exemplos de biometrias

O problema do reconhecimento de face é usualmente abordado de duas diferentes maneiras:

Identificação: Dada uma pessoa com sua face a ser investigada e uma galeria de faces de outros indivíduos, a tarefa de identificação consiste em encontrar a classe correta para a face investigada. Em outras palavras: “Quem eu sou?”.

Verificação (autenticação) de face: Dado um conjunto de faces e uma face a ser investigada tendo sua classe declarada, a tarefa de verificação/autenticação é certificar ou não a informação declarada. Em outras palavras: “Confirme que eu sou a pessoa X.”

O trabalho desenvolvido nesta dissertação focaliza-se na abordagem de reconhecimento por verificação (autenticação). Trabalhos futuros poderão ser ampliados para a abordagem de identificação, onde os cenários de aplicação incluirão imagens a partir de cenas em aeroportos, trânsito e etc. Todas estas aplicações utilizando-se de informação da forma 3D da face.

2.2.2 Medida de desempenho de um sistema de verificação

Durante o processo de autenticação, a tarefa de um classificador é essencialmente identificar duas classes de padrões, i.e., se os mesmos pertencem a uma pessoa ou não. Os padrões que pertencem a uma pessoa genuína são chamados de “população ovelha” ou classe positiva. Os

vetores que não pertencem à pessoa, i.e. os impostores, são chamados de “população lobo” ou classe negativa. Quando a saída do classificador for próxima a zero o padrão de entrada é atribuído à classe negativa e quando for próxima de um à classe positiva.

Se cada um dos dois conjuntos de padrões possuem probabilidades de serem classificados por um classificador através de uma distribuição, por exemplo normal, sendo o limite da saída entre zero e um, um classificador bem treinado daria o conjunto de resultados mostrados na Fig. 2.13.

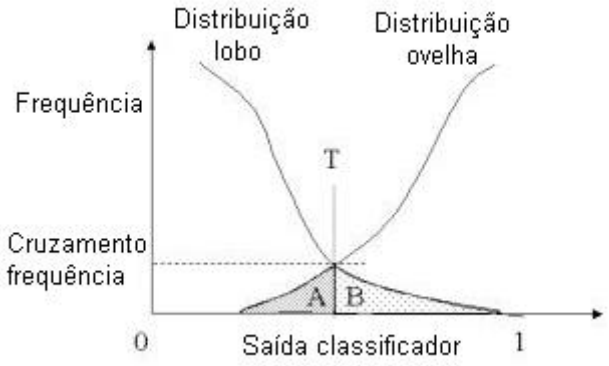

Figura 2.13: Uma típica distribuição das populações de ovelhas e lobos.

Seja T o limiar do classificador, isto implica que, se o classificador responder (por exemplo, a partir de uma saída de neurônio) um valor maior do que T, a pessoa é aceita com a sua identidade declarada, caso contrário ela é rejeitada. A área A na Fig. 2.13 (área limitada pela distribuição ovelha, o limiar T e o eixo da saída do classificador) mostra os casos de falsa rejeição e B (área limitada pela distribuição lobo, limiar T e o eixo da saída do classificador) mostra os casos de falsa aceitação. Esta análise é chamada de Análise de Distribuição Ovelha-Lobo e é útil para checar o quanto um classificador tem aprendido ou não.

A área B na Fig. 2.13 é proporcional à Taxa de Falsa Aceitação ou Taxa de Alarme Falso (FAR - *False Acceptance Rate*). FAR é também chamado de um Erro Tipo I e é definido pela Equação.

$$FAR = \frac{Total\ \ de\ \ Falsa\ \ Aceitacão}{Total\ \ de\ \ Tentativas\ \ Falsas} \tag{2.37}$$

Da mesma maneira, a área A na Fig. 2.13 é proporcional à Taxa de Falsa Rejeição (FRR - *False Rejection Rate*). FRR é também chamada de erro Tipo II e é definida pela Eq. 2.38

$$FFR = \frac{Total\ \ de\ \ Falsa\ \ Rejeicão}{Total\ \ de\ \ Tentativas\ \ Verdadeiras} \tag{2.38}$$

O resultado do deslocamento de T ao longo do eixo de resultados resulta em diferentes FAR e FRR que, quando plotados, geram um gráfico parecido com o da Fig. 2.14. Esta análise é chamada de Análise de Limiar e é útil para checar a Taxa de Erro Idêntico (ERR), dada por $FAR = FRR$.

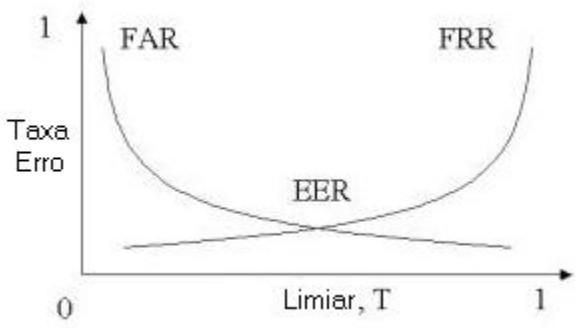

Figura 2.14: FAR e FRR versus Limiar.

Quando T aumenta de zero a um, o FAR decrementa de um a zero e o FRR aumenta de zero a um. Um alto FAR significa que um impostor teria grandes chances de ser aceito como um usuário verdadeiro, enquanto um alto FRR significa que um usuário genuíno teria grandes chances de ser rejeitado, quando sua identidade declarada é autêntica. Um alto FRR causará baixa segurança aos usuários em um sistema de controle de acesso utilizando biometria. Por outro lado, um alto FAR seria catastrófico pois um impostor poderia facilmente obter um acesso ilegal.

A *Freqüência de Cruzamento* é usualmente expressa como $(1 : X)$, onde X é arredondado para inteiro. A Freqüência de Cruzamento e o ERR são freqüentemente usados para comparar a qualidade de diferentes classificadores e/ou dados biométricos. EER pode ser usado para comparar os resultados de dois classificadores ou duas características biométricas, dependendo do contexto da comparação. O classificador, ou característica biométrica, com o mais baixo ERR é o melhor pois pode discriminar melhor as duas classes.

Plotando FAR versus FRR tem-se o gráfico de Características de Operação de Receptores (ROC - Receiver's Operating Characteristics), que é mostrado na Fig. 2.15

O gráfico ROC deve este nome ao seu uso original em gerenciamento. Ele também é chamado de *detecção do erro da curva de concessão* pelo *National Institute of Standards and Tecnology* (NIST) ou em termos mais gerais como *curva de desempenho*. É desejável que aplicações judiciais tenham um alto FAR para então tentar maximizar a chance de obter um suspeito. Por exemplo, uma aplicação para capturar suspeitos, onde um banco de dados criminal pode ser consultado e retornar uma lista de suspeitos que mais se parecem com o criminoso, ordenados em ordem crescente de similitude. Por outro lado, para aplicações

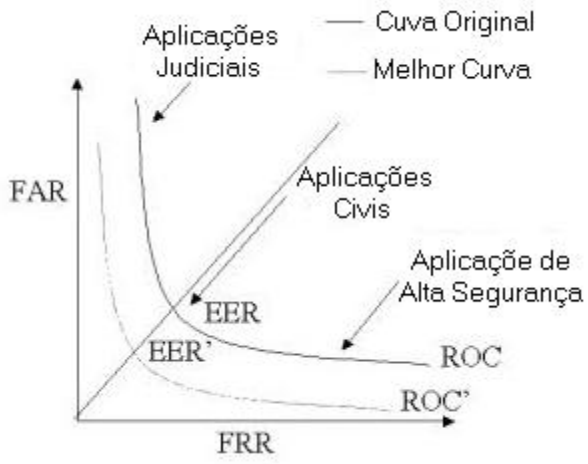

Figura 2.15: FAR versus FRR.

de alta segurança tal como uma aplicação para ATM[11], é desejável um FAR extremamente baixo pois tais aplicações não podem tolerar a aceitação de um impostor.

A Fig. 2.15 mostra que não é possível minimizar FAR e FRR ao mesmo tempo. Entretanto, um acordo pode ser alcançado, quando $FAR = FRR$, que é o chamado ponto de EER. Em aplicações civis este acerto é alcançável. Um exemplo é o acesso a informação geral mas em terminal de computador de identidade específica. A linha pontilhada da curva da Fig. 2.15 mostra um sistema de melhor qualidade do que o com linha sólida. A curva mais próxima à origem é a de melhor qualidade para o sistema.

2.2.3 Detecção de faces

O trabalho Yang [89] mostra que a interação entre o ser humano e o computador é uma área de intenso nível de pesquisa onde são desenvolvidas interfaces cada vez mais amigáveis. A face humana é um dos meios mais efetivos para se atingir esta meta pois ela carrega grande número de informações, de modo que computadores poderiam reagir de acordo. Por exemplo, computadores podem ajustar seu comportamento pelo conhecimento das emoções do usuário, através de suas expressões faciais. Atenção visual é outro exemplo onde computadores podem reagir baseados nos interesses de seus usuários. Rumo a este objetivo, o reconhecimento de faces e expressões faciais têm atraído muita atenção recentemente, embora tendo sido estudado há mais de vinte anos por psicólogos, neurocientistas e engenheiros. Muitas aplicações interessantes e úteis foram desenvolvidas com estes esforços. A maioria dos métodos existentes consideram que faces humanas devem ser extraídas de imagem

[11] ATM (*Automatic Teller Machines*) equivale a máquina de atendimento automático ou caixa automático.

estática ou de uma seqüência de imagens e focalizadas através de algoritmos de reconhecimento. Porém, esta é uma tarefa muito desafiante e não mais fácil do que o reconhecimento de faces. Detecção de face é consideravelmente difícil porque envolve a localização da face sem nenhum conhecimento prévio de suas escalas, localizações, orientações (ereta, rotacionada) com ou sem oclusões, com diferentes posições (frontal, perfil). Expressões faciais e condições de iluminação também alteram por completo aparências de faces, tornando difícil detectá-las. Além disto, a aparência de faces humanas em uma imagem depende da posição das pessoas e do campo de visão dos dispositivos de aquisição de imagem.

2.2.4 Reconhecimento

2.2.4.1 As primeiras abordagens

Os primeiros trabalhos em reconhecimento de faces por computador necessitavam de operadores humanos para a localização de pontos da face cujas posições eram fornecidas como entrada. Dado um conjunto de distâncias de pontos, por exemplo, uma delas seria a distância da ponta do nariz ao queixo, de uma pessoa desconhecida, a técnica do vizinho mais próximo ou outras regras de classificação eram usadas para identificá-la. Como a extração de características era feita manualmente, o sistema era indiferente a grandes variações na rotação da cabeça, inclinações, qualidade de imagem, e contraste. Em seguida vieram trabalhos similares, mas sem intervenção humana para a aquisição dos dados de entrada. [19]

2.2.4.2 Abordagem estatística

Segundo Nefian [64], as técnicas estatísticas em reconhecimento de faces são utilizadas tanto na extração de características como na tarefa de classificação. Dentre as abordagens estatísticas para reconhecimento de faces, as mais usadas são: i) Métodos de correlação; ii) Métodos de decomposição de valor singular; iii) Métodos baseados em expansão Karhune-Loeve; iv) Métodos baseados em discriminante linear Fisher; v) Métodos baseados em modelo de Markov escondido.

2.2.4.2.1 Métodos de correlação

A maioria direta dos procedimentos usados para reconhecimento de faces é o casamento entre as imagens teste e um conjunto de treinamento de imagens baseado numa medida de correlação. O método de correlação é computacionalmente muito caro e a dependência do reconhecimento sobre a resolução da imagem tem sido investigada. Estudos mostram que o reconhecimento baseado na correlação tem um bom nível de desempenho usando vetores de

características pequenos.

2.2.4.2.2 Métodos baseados em expansão Karhunen-Loeve

O método de reconhecimento utilizando auto-faces tem sido implementado com o propósito de comparações, pois ele foi um dos melhores e mais sucedidos algoritmos. Este método foi desenvolvido no M.I.T. por [Turk and Pentland, 1991] apud [79]. É chamado de Análise de Componente Principal onde poucos parâmetros extraídos da face são usados para representação. Estes parâmetros são obtidos pela projeção da face em um sistema de coordenadas dadas por autovetores da matriz de covariância do conjunto de treinamento. Estes autovetores, imagens deles mesmos, são chamados auto-faces e transpõe um vetor de espaço chamado espaço face. Cada face é então codificada por meios de suas coordenadas no espaço face. A comparação de duas faces corresponde a um cálculo da distância Euclidiana entre suas representações do espaço de face [79].

O trabalho de Stan [56] apresenta uma nova abordagem para classificação chamada combinação linear mais próxima (NLC) para reconhecimento de faces baseado em auto-face. Ele considera múltiplos vetores de padrão disponíveis em classes, cada vetor começando de um ponto em um espaço auto-face. Uma combinação linear de vetores pertencentes a uma classe de face é usada para definir uma medida de distância entre um vetor consulta e a classe, a medida é definida como sendo a distância Euclidiana do vetor consulta para a combinação linear mais próxima (portanto NLC). Isto contrasta à classificação por vizinho mais próximo (NN) onde um vetor consulta é comparado com cada vetor exemplo individualmente. Usando uma combinação linear de vetores exemplos, ao invés de cada um deles individualmente, estende-se a capacidade de representação dos exemplos pela generalização, através de interpolação e extrapolação. Experimentos mostraram que isto conduziu a melhores resultados do que os métodos de classificação existentes. A Fig. 2.16 ilustra o uso da técnica NLC para deduzir a posição de y (ângulo de visão, iluminação, ou expressão) relativa a dois exemplos x_k $(k = 1, 2)$. Nesta Figura tem-se na primeira linha: faces sob mudanças no ângulo de visão. A face consulta y (à esquerda) está a um ângulo relativamente central das duas faces exemplos x_1 e x_2, vistas a um ângulo direito e esquerdo respectivamente. Na segunda linha: faces sob mudanças de iluminação. A face consulta y (à esquerda) é iluminada por uma luz à direita e é comparada a duas faces exemplos x_1 e x_2, uma é iluminada pela esquerda e outra pelo centro, respectivamente. E na última linha: faces sob mudanças de expressões. Com isto ele mostra que a abordagem NLC reduz significativamente a taxa de erro com relação à abordagem de classificação NN em auto-face para reconhecimento de faces. Este aprimoramento se deve à capacidade de representação da técnica NLC com relação ao padrões exemplos na base de dados: variações na iluminação, ângulo de visão

e na expressão entre as imagens de faces exemplos são consideradas por variações em seus pesos que determinam a combinação linear.

Figura 2.16: Imagens de faces para o uso da técnica NLC. Adaptada de [79].

2.2.4.2.3 Reconhecimento sob condições gerais de visualização

Esta é uma abordagem paramétrica e estende a capacidade do método auto-face para reconhecimento de objetos em imagem 3D sob diferentes condições de iluminação e visualização. Dadas N imagens de objetos tidas sob P condições de visão e L condições de iluminação, um conjunto de imagem universal é construído de forma a conter todos os dados disponíveis. Desta maneira um simples espaço paramétrico descreve a identificação do objeto, bem como as condições de visualização ou iluminação. A decomposição auto-face deste espaço é usada para extração e classificação de características. Entretanto, para garantir a discriminação entre diferentes classes de objeto o número de autovetores usados neste método é maior, comparado ao método auto-face clássico [64].

2.2.4.2.4 Reconhecimento usando auto-características

Consiste do uso de características faciais para reconhecimento de faces. Isto pode ser visto como uma representação modular ou por camadas da face onde uma descrição grosseira (baixa resolução) de toda a face é definida por detalhes adicionais (alta resolução) salientando as regiões das características. A técnica auto-face foi estendida para detectar características faciais. Para cada característica da face, um espaço de características é construído pela seleção da maioria de auto-características mais significativas (autovetores

correspondentes para grandes autovalores da matriz de correlação de características). Na representação auto-característica a equivalente distância do espaço de característica (DFFS) pode ser efetivamente usada para detecção de características faciais. [64].

2.2.4.2.5 Método discriminante linear - Fisherfaces

Neste método há a redução da dimensionalidade do espaço de características pelo uso do Discriminante Linear Fisher (FLD) [21] apud [64]. O FLD usa a informação de um grupo de classes e desenvolve um conjunto de vetores de características nos quais variações de diferentes faces são enfatizadas, enquanto diferentes exemplos de faces, devido a condições de iluminação, expressões faciais e orientação, são “não-enfatizadas”.

2.2.4.2.6 Método baseado no modelo Markov escondido

O modelo Markov escondido (HMM) é um conjunto de modelos estatísticos usados para caracterizar propriedades estatísticas de um sinal. HMM é feito a partir de dois processos inter-relacionados: (1)uma cadeia de Markov secreta e não observável com finitos números de estados, uma matriz de probabilidade de transição de estado e uma distribuição de probabilidade de estado inicial. (2)um conjunto de funções de densidade probabilidade associado a cada estado.

O modelo HMM tem sido usado extensivamente para reconhecimento de voz, neste caso os dados têm naturalmente uma dimensão (1D) ao longo do eixo do tempo. Entretanto, uma cadeia HMM equivalente e completamente conectada em duas dimensões dominaria uma grande quantidade de problemas computacionais. Tentativas foram realizadas para usar representações multi-modelos que conduzem a um pseudo HMM 2D. Estes modelos são atualmente usados no reconhecimento de caracteres. Foi proposto em [Samaria *et al*] apud [64] o uso de 1D HMM contínuo para reconhecimento de faces. Supondo-se que cada face está em uma posição ereta e frontal, características ocorrerão em uma ordem previsível, isto é, testa, olhos, nariz etc. Esta ordenação sugere o uso de um modelo “top-bottom”, onde somente transições entre estados adjacentes no modo de cima para baixo são permitidos. Os estados do modelo correspondem a características faciais como testa, olhos, nariz, boca e queixo. A sequência de observação abaixo é gerada a partir de uma imagem X x Y usando uma janela de amostra X x L com X x M pixels sobrepostos (Fig. 2.17) [64].

Cada vetor de observação é um bloco de L linhas. Há uma linha M sobreposta entre observações sucessivas. A sobreposição permite que as características sejam capturadas de maneira que a posição vertical seja independente, enquanto um particionamento disjunto da imagem poderia resultar em características nas fronteiras dos blocos.

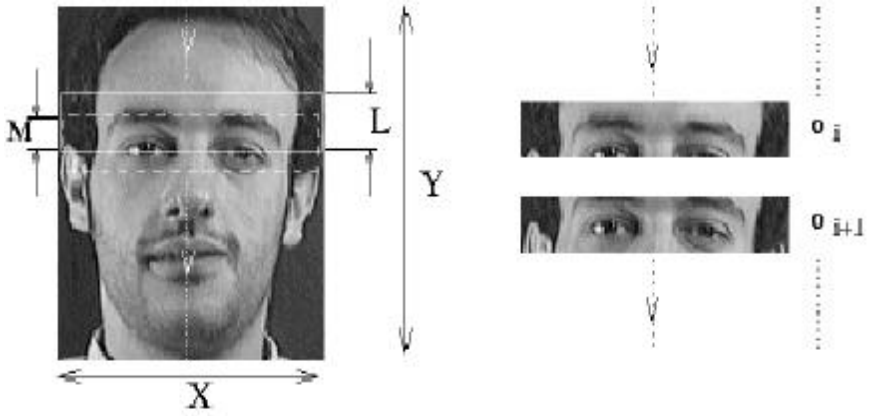

Figura 2.17: Técnica de imagem modelo para reconhecimento HMM. Adaptada de [64].

2.2.5 RNAs e o reconhecimento de faces

A maioria das aplicações em RNAs divide-se em três categorias: *classificação* (*reconhecimento de padrões*), onde a rede classifica o padrão de entrada em categorias pré-definidas ou não; *predição*, onde a rede tenta extrapolar uma série de entrada, e *controle*, onde a rede é usada para guiar interativamente alguns processos externos ou dispositivos. Os dois primeiros são basicamente casos de aproximação, onde se deseja aproximar alguma função tipicamente numérica[34].

O uso de RNAs em reconhecimento de faces vem sendo aplicado a muitos problemas: classificação por sexo, reconhecimento de faces, e classificação de expressões faciais. Uma das primeiras demonstrações foi realiza por Kohonen [52] através dos chamados mapas associativos. Com um pequeno conjunto de imagens eram realizados experimentos onde a rede dava respostas precisas mesmo quando as imagens de entrada apresentavam muito ruído ou quando partes da imagem estavam faltando [19].

Como já dito, atualmente existem numerosos estudos explorando vários conceitos e problemas no processo de reconhecimento de faces e muitos esforços são empregados na construção de sistemas eficientes com o uso de redes neurais artificiais e outras abordagens. Entretanto, o desempenho da maioria dos esquemas propostos geralmente é sensível a transformações em coordenadas 2D na imagem (por exemplo, escalonamento, translação) [43]. Como neste trabalho apresentado exploramos a natureza tridimensional da face através de sua forma, o sistema produzido possui uma maior habilidade ao tratar os fatores acima.

2.2.5.1 Tópicos para o projeto experimental

O sucesso de uma aplicação em reconhecimento de faces depende de como a informação é apresentada à rede neural. Dependendo da aplicação, um pré-processamento deverá ser feito

para produzir uma representação que acentue a informação mais relevante. No contexto de redes neurais como classificadores, duas restrições devem ser consideradas: i) a dimensão do classificador (número de graus de liberdade) que está associada ao número de conexões em uma rede neural e ii) o número de exemplos no conjunto de treinamento (tamanho do conjunto de dados) [43].

Em geral, é aconselhável que o tamanho do conjunto de dados exceda ao número de conexões, em ordem de magnitude. Aumentar o número de neurônios na camada de entrada e o número de exemplos de treinamento ajuda a capacidade de generalização da rede neural. Porém, quanto a isso se deve tomar cuidados pois o aumento do tamanho da rede neural (número de neurônios, camadas e pesos), afeta o comportamento de convergência da rede [43].

Em aplicações envolvendo processamento de imagem, algumas decisões devem ser tomadas para reduzir o tamanho do vetor do padrão de entrada a ser apresentado à rede neural. Entretanto, uma rede neural apresentará um melhor desempenho como classificador quando for treinada com dados brutos (sem nenhum processo de redução de dimensionalidade). Este é um exemplo clássico dos ajustes necessários entre tamanhos de redes neurais e o tamanho do vetor de padrão [43].

Geralmente, redes neurais provêem um mecanismo adaptativo para combinação de um conjunto de não-linearidades a fim de aproximar uma determinada transformação não-linear. A maioria das aplicações em reconhecimento de faces são implementadas usando redes neurais de primeira ordem, o tradicional perceptron multi-camada. Enquanto em nível de treinamento, os algoritmos de otimização destas categorias de redes neurais tentam reduzir o tempo de treinamento e eliminar mínimos locais [43].

Os classificadores não lineares, que são versões interpolativas dos classificadores vizinho mais próximo, constituem a segunda categoria mais popular de redes neurais em reconhecimento de faces. Elas requerem pouco treinamento, mas a maioria impõe restrições com relação a funções básicas como memória e velocidade de classificação [43].

Capítulo 3

Justificativas para a metodologia empregada

3.1 Introdução

Neste capítulo são apresentadas algumas justificativas para a utilização de RNAs, especialmente as redes diretas, aplicadas ao problema de reconhecimento de faces humanas em formas 3D. Além do paradigma conexionista buscou-se também o emprego de técnicas de inteligência artificial evolucionária, para a definição da arquitetura da rede ótima a ser utilizada no problema.

3.2 Conexionismo versus reconhecimento de faces 3D

Segundo Barreto [6], o reconhecimento de padrões é uma das primeiras aplicações de redes neurais dado que o Perceptron de Rosenblatt foi o primeiro instrumento capaz de reconhecer letras.

O sucesso de RNAs tem-se verificado por sua capacidade de aprender, seu comportamento emergente, capacidade de adaptação, evolução de excitação em redes com ciclos e sua importância para a psicologia e etc. O reconhecimento de padrões é uma tarefa geralmente desempenhada muito melhor usando as capacidades cognitivas do homem do que executando um algoritmo. Sendo assim, como redes neurais são modelos matemáticos inspirados no sistema neural biológico, esta é uma aplicação em potencial para RNAs.

O paradigma de aprendizado supervisionado é o mais utilizado em reconhecimento de padrões juntamente com as redes neurais diretas multi-camadas, o maior exemplo é o algoritmo de retropropagação e suas variantes. Entretanto bons resultados são obtidos também

com o aprendizado competitivo do tipo redes de Kohonen onde o paradigma de aprendizado é não supervisionado, e onde as classes de padrões não são previamente identificadas.

Considerando que o objetivo principal deste trabalho de dissertação é produzir um sistema de reconhecimento de face com baixo custo computacional, as RNAs se tornam muito atrativas. Outro objetivo também é verificar se eventualmente estas técnicas são superiores a outras técnicas empregadas tais como estatísticas e *casamento* de modelo.

3.2.1 Complexidade de RNAs $\times$ reconhecimento de faces

Barreto em [6] afirma que, com relação à complexidade dos problemas tratados por RNAs existem poucos estudos. Ela diz respeito a dois pontos: i) definição da topologia de rede necessária e ii) tamanho mínimo da rede a ser usada para o problema. A classificação dos problemas tratados por RNAs pode ser dada da seguinte forma:

1. **Problemas estáticos linearmente separáveis.**
   São problemas envolvendo a implementação de uma função (por ser um problema estático) e que podem ser resolvidos por um perceptron de uma camada.

2. **Problemas estáticos linearmente não separáveis.**
   São problemas envolvendo a implementação de uma função (por ser um problema estático) e que podem ser resolvidos por uma rede direta, com neurônios estáticos[1], exigindo ao menos uma camada de neurônios internos.

3. **Problemas dinâmicos com dinâmica finita.**
   Os problemas com dinâmica finita são aqueles onde a duração da resposta do sistema após uma entrada dura um tempo finito. Um exemplo são os filtros FIR ("Finite Impulse Response"). Estes problemas podem ser resolvidos por rede direta com neurônios dinâmicos[2].

4. **Problemas dinâmicos com dinâmica infinita.**
   Os problemas com dinâmica infinita são aqueles onde a duração da resposta do sistema após uma entrada pode durar um tempo infinito. Um exemplo são os filtros IIR ("Infinite Impulse Response"). Estes problemas podem ser abordados por rede com

[1] Um neurônio é estático quando o valor de sua entrada e saída referem-se ao mesmo instante das excitações, ou seja, o retardo é nulo [6].

[2] Um neurônio é dinâmico quando para o cálculo de sua saída em um dado instante é necessário o conhecimento de sua saída no instante anterior [6].

retroação ou rede estática e conjunto de retardos. Neste caso o problema da estabilidade da rede, ou seja, se a rede encontrará ou não solução e quanto tempo será necessário é um problema em aberto.

De início supunha-se que o problema de reconhecimento da forma 3D de faces humanas tratado neste trabalho era um problema de auto-associação de padrões de entrada a padrões de saída, logo um problema estático linearmente não separável. Porém, assim que foram aplicadas técnicas de programação evolucionária na busca da definição da rede ótima (número de neurônios ideal na camada intermediária) constatou-se que o problema de reconhecimento de faces se classifica no primeiro caso, ou seja, um problema estático linearmente separável, dado que foi resolvido primeiramente por uma rede direta com apenas um neurônio e em seguida por uma rede perceptron, de apenas uma única camada. Como isto foi realizado será mostrado com mais detalhes no Capítulo 5.

Portanto, segundo Grönroos [34], embora muitos métodos eficientes são desenvolvidos para o treinamento de RNAs, nenhum método definitivo existe para se determinar a arquitetura de RNA mais portável a problemas particulares. Logo, o que existe para esta definição são somente heurísticas, como por exemplo citada em Kovács [53]. Muitas delas são sugestões dadas a partir da experiência adquirida por meio de experimentações.

Com isto optou-se pela técnica de programação evolucionária, um ramo da IA evolutiva[3] que tem apresentado eficientes soluções no que diz respeito à "evolução de RNAs", por exemplo [90], [92] e [95]. Portanto supôs-se que, para este trabalho, a utilização desta técnica para a evolução da arquitetura de RNAs seria muito apropriada. A seguir, na Seção 3.3, tem-se uma breve revisão do que é a computação evolutiva. E como esta abordagem levou a redes de única camada a resolverem o problema de reconhecimento da forma 3D de faces humanas, na Seção seguinte é apresentada um resumo da teoria das redes Perceptron e Adaline.

3.2.2 O Perceptron

Segundo Kovács [53], no final da década de 1950, Rosenblatt na Universidade de Cornell, deu prosseguimento às idéias de McCulloch ao criar uma rede de múltiplos neurônios do tipo *discriminadores lineares* que chamou de rede *perceptron.*

[3]É válido lembrar que relativo a este assunto na literatura, as palavras "evolutiva" e "evolucionária" são sinônimos. Com relação à "Computação Evolucionária" a mesma é considerada, por alguns pesquisadores, como métodos de otimização apenas, enquanto nesta dissertação ela é considerada um paradigma de IA e é equivalente a dizer "IA evolucionária".

Dentre as muitas variações de percetrons criadas por ele, a mais simples foi uma rede de única camada cujos pesos e bias poderiam ser treinados. A técnica de treinamento usada foi chamada de “regra de aprendizado perceptron”.

O Perceptron gerou grande interesse devido a sua habilidade de generalização e capacidade aprender a partir dos pesos de conexões inicializados de forma aleatória. Perceptrons são especialmente portáveis para problemas simples em classificação de padrão. Eles são redes rápidas e confiáveis. Em adição, um entendimento das operações do Perceptron provê uma boa base para o entendimento de redes mais complexas.

O problema que Rosenblatt propôs resolver foi o seguinte: para casos simples como a implementação das funções booleanas **E** e **OU** de duas variáveis é relativamente trivial escolher os pesos sinápticos e o valor do limiar. Porém, para a implementação de uma função discriminatória arbitrária esta escolha não é trivial, e se o número de variáveis envolvidas for grande, a existência de um *método* é imprescindível.

Por simplicidade, o autor considerou um Perceptron de n entradas formado por um único descriminador linear cuja a função era classificar vetores de *padrões* em duas categorias: $y = 1$ correspondendo à categoria A e $y = 0$ à categoria B. Rosenblatt efetivamente construiu um Perceptron deste tipo, batizado de Mark 1, de 400 fotocélulas arranjadas em uma matriz de 20×20 pixels, que constituíam as componentes do vetor de entrada e que servia para o reconhecimento de caracteres grafados nesta matriz. Assim por exemplo, se a função do Perceptron fosse reconhecer o símbolo A, a sua resposta deveria ser $y = 1$ sempre que alguma variante deste fosse apresentado na sua entrada e $y = 0$ em caso contrário. Supondo que efetivamente estas duas classes de padrões A e $não\ A$ fossem linearmente separáveis, resta a questão de como escolher os pesos das conexões de entrada w_i e o limiar Θ, parâmetros que definem unicamente um descriminador linear, cujo um dos primeiros exemplos foi estabelecido pelo neurônio de McCulloch. Genericamente, um descriminador linear de n entradas $x_1, x_2, \ldots x_n$ e uma saída y é definido pela expressão:

$$\begin{aligned} y &= H\left(\sum_{i=1}^{n} w_i x_i - \Theta\right) = H(w^t x - \Theta) \rightarrow y \in [0;1] \\ y &= sgn\left(\sum_{i=1}^{n} w_i x_i - \Theta\right) = sgn(w^t x - \Theta) \rightarrow y \in [-1;1] \end{aligned} \tag{3.1}$$

onde os componentes do vetor w, $w1, w_2, \ldots w_n$ são os pesos associados às entradas x_i, Θ é o valor de limiar, $H(\nu)$ é a função degrau unitário e $sgn(\nu)$ o operador sinal.

A expressão 3.1 representa um hiperplano que divide o espaço euclidiano $\Re^n$, de dimensão n, em duas regiões A e B. Assim, um vetor **x** de componentes $x_1, x_2, \ldots x_n$ estará em

uma destas regiões na medida em que se verificar:

$$\begin{cases} w^t x - \Theta > 0 & \Rightarrow \quad x \in A \\ w^t x - \Theta < 0 & \Rightarrow \quad x \in B \end{cases} \tag{3.2}$$

Enquanto o valor da saída y será:

$$y = 1 \; se \; \mathbf{x} \in A \;\; e \;\; y = 0 \; ou \; -1 \;\; se \;\; \mathbf{x} \in B \tag{3.3}$$

Esta situação está representada no diagrama da Fig. 3.1 para o caso em que a dimensão do espaço é $n = 2$, ou seja no plano euclidiano.

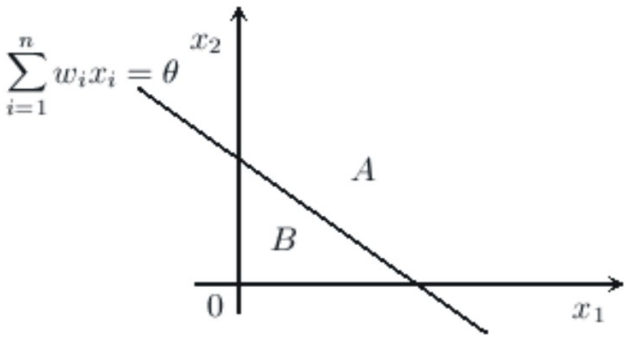

Figura 3.1: O descriminador linear separa o espaço em duas regiões A e B. Adaptada de [53].

Em vista deste comportamento o descriminador linear pode ser utilizado, em certos casos, como um *classificador de padrões* ou *separador de aglomerados* de pontos. Para ilustrar como isto poderia ser feito: sejam duas coleções $\Phi 1 = u_1, u_2 \ldots u_k$ de k vetores $n-$dimensionais e $\Phi 2 = z_1, z_2, \ldots z_m$ de m vetores $n-$dimensionais. Suponha que se pretende implementar um descriminador linear que separe estas duas coleções de vetores conforme a expressão 3.3, isto é: $y = 1 \; se \; \mathbf{x} \in \Phi 1 \;\; e \;\; y = 0 \; ou \; -1 \;\; se \;\; \mathbf{x} \in \Phi 2$.

Naturalmente isto só será possível se as coleções $\Phi 1$ e $\Phi 2$ formarem aglomerados no espaço $\Re^n$, tal que seja possível passar um hiperplano:

$$\sum_{i=1}^{n} w_i x_i = w^r x = \Theta \tag{3.4}$$

que separe os dois aglomerados. Se as coleções $\Phi 1$ e $\Phi 2$ forem tais que isto é possível então são ditas *coleções linearmente separáveis*. Caso contrário, não serão separáveis linearmente e não existirá nenhum descritor linear capaz de executar esta função. Na Fig. 5.1 é ilustrada esta situação.

3.2.2.1 Método de treinamento

Para formalizar melhor o método de treinamento através de exemplos, definindo inicialmente um conjunto Ψ com L *exemplos de treinamento*. Cada exemplo é um par do tipo $(\mathbf{x}_l^d, y_l^d)$, onde os $\mathbf{x}_l^d$ são entradas que devem gerar as saídas y_l^d. Estas saídas y_l^d serão $+1$ ou -1 conforme o vetor $\mathbf{x}_l^d$ pertença à classe $\Phi 1$ ou $\Phi 2$ que se pretende separar. Portanto o conjunto de treinamento é:

$$\Psi = \left\{ \left(x_l^d, y_l^d \right) \right\}_{l=1}^{L} \tag{3.5}$$

O descritor linear é inicializado com parâmetros arbitrários $\{\mathbf{w}_0, \Theta_0\}$ e os vetores $\mathbf{x}_l^d$ dos exemplos de treinamento são aplicados seqüencialmente à sua entrada. Utilizando-se de *algum algoritmo* para ajuste dos parâmetros $\{\mathbf{w}, \Theta\}$ procura-se convergir para valores $\{\mathbf{w}^*, \Theta^*\}$ tais que as saídas especificadas no conjunto de treinamento para todos os exemplos em Ψ seja:

$$y = sgn\left(w^{*^t} x_{i,l}^d - \Theta^* \right) = y_l^d \;\; para \;\; l = 1, 2, \ldots L \tag{3.6}$$

Quando o descriminador linear exibe o comportamento expresso em (3.6), diz-se que está treinado. Resta determinar algum algoritmo que ajuste estes parâmetros e que assegure esta convergência, se possível da maneira mais rápida possível.

3.2.2.2 Princípio de aprendizado de Hebb

Em 1949, Hebb, um biólogo que estudava o comportamento de animais, propôs um princípio pelo qual o aprendizado em sistemas nervosos complexos poderia ser reduzido a um processo puramente *local*. Com as devidas adaptações ao descriminador linear, este princípio pode ser traduzido no seguinte algoritmo: ao se apresentar o $l-$ésimo exemplo, os parâmetros w_i devem ser atualizados segundo a regra:

$$\begin{aligned} & w_i^{novo} = w_i^{velho} + \Delta w_i \\ & com: \\ & \Delta w_i = \eta.(y_l^d - y_l) x_{i,l}^d \end{aligned} \tag{3.7}$$

Por esta regra, a alteração do $i-$ésimo parâmetro depende unicamente do produto da $i-$ésima entrada pelo erro de saída $e_l = (y_l^d - y)$, sendo y a saída com os parâmetros "velhos" do descriminador, ou seja:

$$y_l = sgn\left(\Sigma_{i=1}^{n} w_i^{velho} x_{l,i}^d - \Theta^{velho} \right) \tag{3.8}$$

O parâmetro η em (3.7) é referido como *taxa de aprendizado*, na medida em que se reflete a taxa com que os pesos são alterados em conseqüência dos erros. Esta é uma regra *local* à medida em que não depende dos valores das demais variáveis espalhadas pelo sistema. Para

aplicar esta regra de atualização ao parâmetro Θ, note que este pode ser incorporado à soma ponderada como um peso w_{n+1}, associado a uma entrada x_{n+1}, que está constantemente no valor $x_{n+1} = -1$, resultando a regra de atualização:

$$\Theta^{novo} = \Theta^{velho} - \eta.(y_l^d - y_l) \tag{3.9}$$

Com a incorporação do limiar na soma ponderada, $w_{n+1} = \Theta$, pode-se daqui por diante simplificar a notação do descriminador linear para:

$$y = sgn\left(\Sigma_{i=1}^{n+1} w_i x_i\right) = sgn(w^t x) \tag{3.10}$$

3.2.2.3 Lei de aprendizado do Perceptron

O princípio hebbiano de treinamento expresso por (3.7) e (3.8) foi empregado por Rosenblatt para alterar os parâmetros $\mathbf{w}$ de um descriminador linear. A expressão (3.8) revela que se a saída y_l for igual à saída desejada y_l^d, os parâmetros não sofrem alteração alguma uma vez que os Δw_i serão todos nulos. Por outro lado se a saída desejada e a saída do descriminador forem diferentes, tem-se que:

$$\begin{aligned} y_l^d &= 1,\ y_l = -1 \rightarrow \Delta w_i = 2\eta.x_{i,l}^d \\ y_l^d &= -1,\ y_l = 1 \rightarrow \Delta w_i = -2\eta.x_{i,l}^d \Rightarrow \\ &\begin{cases} \Delta w_i = 0\ \ para\ \ y_l^d = y_l \\ \Delta w_i = 2\eta.y_l^d x_{i,l}^d\ \ para\ \ y_l^d \neq y_l \end{cases} \end{aligned} \tag{3.11}$$

o que pode ser resumido na expressão:

$$\Delta w_i = \eta.(1 - y_l^d y_l) y_l^d x_{i,l}^d \tag{3.12}$$

Ao se implementar um descriminador linear cujo hiperplano além de separar as duas coleções de pontos mantém também uma distância de pelo menos τ de qualquer dos pontos x_l^d do conjunto de treinamento. Isto significa impor uma *zona de exclusão* de largura τ em torno do hiperplano. Evidentemente, o parâmetro τ deverá ser escolhido suficientemente pequeno para que isto seja possível. Lembrando que em um espaço euclidiano $\Re^n$ a distância $d(\mathbf{w}, \mathbf{x})$ de um ponto x a um hiperplano $\mathbf{w}^t\mathbf{x} = 0$ é dado por $d(\mathbf{w}, \mathbf{x}) = |\mathbf{w}^t\mathbf{x}|/|\mathbf{w}|$, este requisito impõe que:

$$\frac{|w^t x_l^d|}{|w|} \geq \tau \quad para\ \ todo\ \ l = 1, 2, \ldots L \tag{3.13}$$

Considerando, além disto, que os pontos das duas categorias $\Phi 1$ e $\Phi 2$ a serem separadas devem estar dos *respectivos lados do hiperplano*, aplicando a relação (3.2) vem que:

$$\begin{cases} w^t x_l^d > 0\ \ para\ \ y_l^d = 1 \rightarrow x \in \Phi 1 \\ w^t x_l^d < 0\ \ para\ \ y_l^d = -1 \rightarrow x \in \Phi 2 \end{cases} \tag{3.14}$$

o que pode ser reunido na condição:

$$y_l^d w^t x_l^d \geq |w|\tau = \delta \tag{3.15}$$

Com este requisito adicional, o algoritmo da expressão (3.12) passa a ser:

$$\Delta w_i = \eta H \left(\delta - y_l^d w^t x_l^d\right) y_l^d x_{i,l}^d \tag{3.16}$$

onde $H(u)$ é a função degrau unitário. A expressão (3.16) é a *lei de aprendizado do Perceptron* proposto por Rosenblatt. Em [53] foi mostrado que se as duas categorias $\Phi 1$ e $\Phi 2$ representadas no conjunto de treinamento forem *linearmente separáveis* então o algoritmo (3.16) *sempre converge em um número finito de iterações*.

3.2.3 O Adaline

Devido à separabilidade linear do problema de reconhecimento da forma 3D de faces abordado neste trabalho, utilizou-se também a rede Adaline, que obteve grande desempenho no reconhecimento, por isso ela será brevemente exposta nesta seção.

Segundo [53], na mesma época que Rosenblatt trabalhava no Perceptron, Widrow na Universidade de Stanford desenvolveu um modelo neural linear, muito simples conceitualmente, que ele batizou de Adaline (acrônimo do inglês: *ADAptive LINear Element*) e que mais tarde sua generalização multidimensional, o Madaline (Múltiplos Adalines). A contribuição realmente importante do trabalho de Widrow foi a invenção de um princípio de treinamento extremamente poderoso para rede Adaline conhecido como a *Regra Delta*, que foi mais tarde generalizado para redes com modelos neurais mais elaborados.

Adalines são similares ao Perceptron, mas sua função de transferência é linear e não degrau. Isto permite que suas saídas tenham qualquer valor, enquanto que a saída do Perceptron é limitada em ser somente 0 e/ou 1. Ambos Adaline e Perceptron podem somente resolver problemas linearmente separáveis. Entretanto, a regra de aprendizado por Erro Médio Quadrático Mínimo (LMS), é mais poderosa do que a regra de aprendizado Perceptron. O LMS ou regra de aprendizado Widrow-Hoff minimiza o erro quadrado médio e, assim, move a fronteira de decisão tanto quanto possível dos padrões de treinamento.

A Fig. 3.2 mostra uma rede Perceptron e Adaline respectivamente. Como se vê a rede Adaline tem a mesma estrutura básica do Perceptron. A única diferença é que o neurônio linear usa uma função de transferência linear e a mesma calcula a saída do neurônio, simplesmente retornando o valor passado a ela.

Este neurônio pode ser treinado para aprender uma função que atribui valores de entrada a tamanhos, ou encontrar uma aproximação linear para uma função não linear. A rede linear não se aplica a uma computação não linear [24].

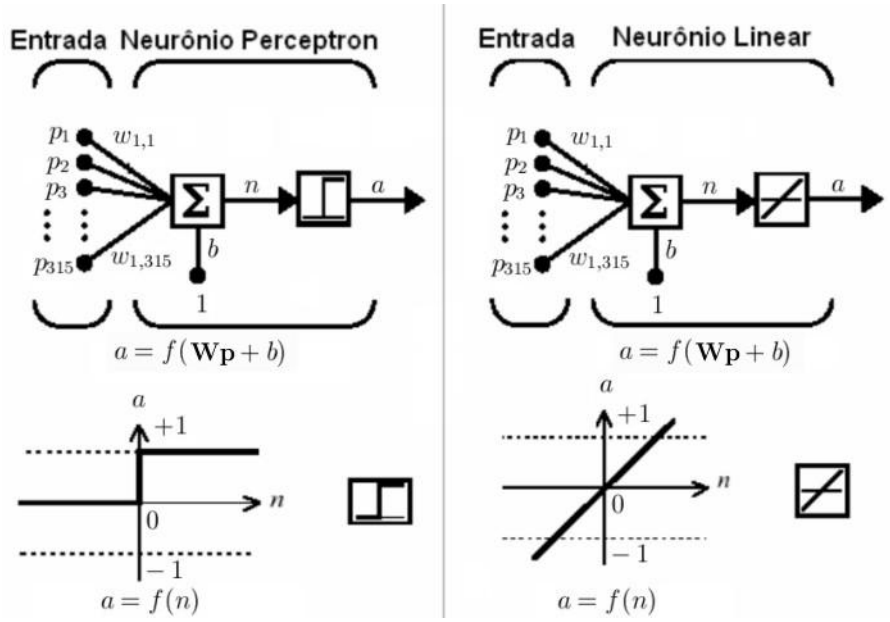

Figura 3.2: Redes Perceptron (esquerda) e Adaline (direita). Adaptada de [24].

Quando se tem mais de um neurônio Adaline na camada da rede ela é dita ser uma rede Madaline (Fig. 3.3). Note o vetor de saída a de tamanho S. A regra Widrow-Hoff somente pode treinar redes de única camada linear. Isto não é lá uma desvantagem, dado que redes lineares de única camada são tão capazes quanto redes lineares multi-camadas. E para toda rede linear multi-camada há sempre uma rede linear de única camada equivalente [24].

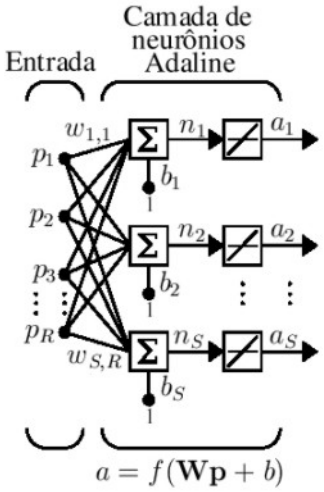

Figura 3.3: Rede Madaline. Adaptada de [24].

3.3 Computação evolucionária (IA evolucionária)

Nesta seção uma breve apresentação de outro paradigma da IA, a Inteligência Artificial Evolucionária, é exposta. Ela consiste de algoritmos de otimização probabilística baseados no modelo natural de evolução, por isso são chamados de algoritmos evolutivos ou evolucionários. As três principais correntes são os algoritmos de *estratégia evolucionária* (EEs), *programação evolucionária* (PE), e *algoritmos genéticos* (AGs) [5].

Os algoritmos evolucionários baseiam-se no processo de aprendizado coletivo de uma população de indivíduos, cada um representando um ponto de busca no espaço de soluções de um determinado problema. A população é arbitrariamente inicializada, e ela evolui em direção à melhor ou melhores regiões de busca no espaço de soluções por meios de processos aleatórios de *seleção*, *mutação* e *reprodução*. A análise do ambiente provê informação sobre valores de adaptação dos indivíduos nos pontos de busca, e o processo de seleção favorece aqueles indivíduos de alta adaptação para reproduzir mais freqüentemente do que aqueles de pouca adaptação. O mecanismo de recombinação permite misturar a informação dos pais para ser transmitida aos seus descendentes, e mutação introduz inovação dentro da população.

Segundo Falqueto [27] um algoritmo evolutivo genérico pode ser formalizado da seguinte maneira:

ALGORITMO EVOLUTIVO GENERICO

$t := 0$;
inicializar: $P(0) := \{\overrightarrow{a_1}(0), \ldots, \overrightarrow{a_\mu}(0)\} \in I^\mu$;
calcular: $P(0) := \{f(\overrightarrow{a_1}(0)), \ldots, f(\overrightarrow{a_\mu}(0))\}$;
ENQUANTO (criterio de parada) FACA
    recombinar: $P'(t) := r(P(t))$;
    mutar: $P''(t) := m(P'(t))$;
    calcular: $P''(t) := \{f(\overrightarrow{a''_1}(0)), \ldots, f(\overrightarrow{a''_\mu}(0))\}$;
    selecionar: $P(t+1) := s(P''(t))$;
    $t := t + 1$;
FIM ENQUANTO

A notação utilizada no algoritmo acima é descrita na Tab. 3.1.

As três principais correntes de exemplos deste algoritmo geral, foram desenvolvidos independentemente um do outro; eles podem hoje em dia ser identificados por: *programação evolucionária* (PE), originalmente desenvolvido por L.J. Fogel, A. J. Owens, e M. J. Walsh nos Estados Unidos (1966) e recentemente redefinido por D. B. Fogel (1991); *estratégias*

Notação	Descrição
I	Espaço de indivíduos
$\mathbf{R}$	Conjunto dos reais
$\vec{a} \in I$:	Um indivíduo qualquer de I (geralmente $\vec{a}$ é definido por um vetor $\vec{x} \in \mathbf{R}$
$f : I \rightarrow R$	Função de aptidão. Mapeia cada $\vec{a}$, através do seu genótipo, no seu fenótipo
$P(t) = (\vec{a_1}(t), \vec{a_\mu}(t))$	Subconjunto de I, chamado população em um instante t
$P'(t), P''(t), P'''(t)$	Populações intermediárias, auxiliares no algoritmo
μ	Número de indivíduos da população genitora
λ	Número de indivíduos da população gerada
$r : P^\mu \rightarrow P^\lambda$	Operador de recombinação (cruzamento), aplicado sobre indivíduos da população genitora (P^μ)
$m : P^\lambda \rightarrow P^\lambda$	Operador de mutação, aplicado sobre indivíduos da população gerada (P^λ)
$s : (P^\mu \cup P^{\mu+\lambda})$	Operador de seleção, aplicado sobre a população gerada, acrescida ou não a geradora

Tabela 3.1: Notação utilizada na descrição do algoritmo evolutivo genérico. Adaptada de [27].

evolucionárias (EEs), desenvolvidas na Alemanha por I. Rechenberg (1965) e H. P. Schwefel (1977); e *algoritmos genéticos* (AGs) por J. Holland (1975) nos Estados Unidos, bem como, com refinamentos por K.De Jong (1975), J. Grefenstette(1986) e D. Goldberg (1989). Cada uma destas correntes tem claramente demonstrado sua capacidade de produzir boas soluções aproximadas para problemas de otimização, mesmo nos casos de superfícies de resposta multimodais, descontínuas, não diferenciáveis, com ruído ou movimento [5].

3.3.1 Estratégias evolucionárias

Segundo Falqueto [27], em 1964 e 1965, Ingo Rechenber e seu colega Has-Paul Schwefe, da Technicshe Universität Berlin, idealizaram a metodologia e batizaram-na de Estratégias Evolucionárias, para resolver problemas técnicos de otimização de perfis aerodinâmicos. Hoje, após diversas modificações, a sistemática é empregada em muitos problemas de otimização com variáveis reais, pois necessita de pouca informação sobre o problema, não incorrendo em derivadas da função a otimizar, e sendo aplicável a modelos tanto lineares como não lin-

eares. Os primeiros algoritmos construídos utilizavam uma política conhecida como $(1+1)$, em que apenas os operadores de seleção e mutação era utilizados e *um pai gerava um filho* a cada geração. A mutação sofrida pelos indivíduos obedecia à distribuição normal. Posteriormente este processo foi ampliado por Schwefel, para estratégias "mais globais" denominadas $(\mu + \lambda)$. Uma nova proposta, a (μ, λ), dita μ vírgula $lambda$, com μ pais e λ filhos, e também incorporando o operador de recombinação: nesta proposta um novo indivíduo poderia ser formado potencialmente herdando características de todos os progenitores existentes na população.

Na forma $(\mu + \lambda)$ eram selecionados apenas os melhores indivíduos encontrados no conjunto união dos pais e dos filhos, indicando um caráter altamente eletivo e, segundo Schwefel, inapta para operar em ambientes mutantes, o que o leva a recomendar a estratégia (μ, λ), em que os pais são eliminados antes da seleção, isto é, não convivem e não concorrem como os filhos. Esta característica não é a que ocorre com mais freqüência nos fenômenos biológicos.

Uma característica importante da metodologia é que a seleção é feita sempre de forma determinística, ficando vivo o subconjunto dos melhores indivíduos, sem qualquer possibilidade de um elemento "menos agraciado" sobreviver. Schwefel recomenda a estratégia (μ, λ), com a sincronização entre o nascimento dos filhos e a morte dos pais. Em [BÄC 93] apud [27] cita-se que as mutações não podem ser completamente aleatórias, o que implicaria serem os filhos totalmente independentes dos pais e na Estratégia Evolucionária este é o operador de maior importância. A população deve ter tamanho adequado para proporcionar suficiente riqueza genética, o que previne o empobrecimento a cada geração; também deve haver cooperação entre os indivíduos, pois não se pode esperar que apenas o melhor concentre todo o conhecimento. Ainda mais, o processo deve permitir que haja deterioração em algum ponto, pois isto significa a possibilidade de escapar de um ótimo local e prosseguir no encalço do ótimo global da superfície de busca.

Os embasamentos teóricos da metodologia foram propostos por Rechenberg, Schwefel e Rudolph, tendo este demonstrado analiticamente que a metodologia $(1 + 1)$ converge. Este processo dedutivo pode ser adaptado para mostrar que a sucedânea $(\mu+\lambda)$ também converge, mas não se tem uma demonstração formalizada para a política (μ, λ).

3.3.2 Programação Evolucionária

Segundo Falqueto [27], após um ano da publicação de Rechenberg, divulgando a metodologia de Estratégias Evolucionárias, os pesquisadores Lawrence J. Fogel, A. J. Owens e M. J. Walsh publicaram um livro, descrevendo o paradigma da Programação Evolucionária. Com

outro ferramental de desenvolvimento, mas ainda baseado na inspiração evolucionista darwiniana, este grupo pesquisava nos EUA o que o outro, independentemente, também procurava fazer em Berlin. O problema inicialmente tratado nesta metodologia era o de evoluir máquinas de estado finito para predição de símbolos. A aptidão de cada máquina era medida pelo número de símbolos corretamente previsto pela mesma, sendo que cada genitor originava, por mutação, um descendente. Dentre os ascendentes e descendentes, os melhores 50% eram escolhidos para continuar o processo, na próxima iteração. Esta, aliás, é a política $(\mu + \lambda)$ de Estratégias Evolucionárias.

Mais tarde, Daniel B. Fogel, filho de Lawrence, generalizou a metodologia, tornando-a aplicável a funções de variáveis reais, além de ter publicado o que consiste na atual fundamentação teórica do processo.

Deve-se notar que a Programação Evolucionária não usa recombinação de diferentes indivíduos progenitores para formação de prole, como em Estratégia Evolucionária e AG. Os criadores da metodologia argumentam que a recombinação não é usada porque a Programação Evolucionária se inspira na evolução inter-espécies, onde não existe cruzamento, e conseqüentemente, recombinação.

Outra singularidade: em AG o problema é, em grande parte das aplicações, codificado em cadeia de caracteres que mimetizam as variáveis sob pesquisa, enquanto em Programação Evolucionária a representação dos indivíduos é função do caso sob análise. Assim, o único operador evolucionário, a mutação, na Programação Evolucionária, possibilita que pequenas variações no comportamento dos filhos ocorram de forma muito mais freqüente que grandes variações e é o único operador genético considerado. Uma população de 2μ indivíduos é formada a partir de μ pais e μ filhos, cada um deles advindo da mutação de seu pai. É feito um torneio estocástico, em que cada pai e cada filho, concorre com um subgrupo aleatório da população de pais e filhos. Os melhores μ indivíduos ordenados no torneio sobrevivem, para formar a nova população.

Como fundamentação teórica, além da citada na obra de D. B. Fogel, pode-se dizer que Bäck & Schwefel afirmam ser a convergência da Programação Evolucionária demonstrável, aplicando a mesma sistemática adotada para o caso $(1 + 1)$ de Estratégias Evolucionárias.

O campo de aplicação desta metodologia é especialmente aquele em que a superfície de solução da função de aptidão é muito acidentada, com muitos pontos de ótimo locais. Se assume que esta superfície de solução pode ser descrita em função de variáveis reais e tenha soluções ótimas, que podem ser atingidas com os passos gerais da metodologia:

- Escolher de maneira aleatória uma população inicial de soluções experimentais;

- De cada indivíduo da população inicial gerar nova população, sendo seus componentes

mutações do indivíduo original, e variando de um para outro continuamente, em um intervalo definido;

- Calcular a adaptação de cada indivíduo, e, via torneio estocástico (podendo também ser determinístico, segundo o caso), classificar os indivíduos que comporão a próxima geração. O número de indivíduos pode variar de geração a geração.

Note-se que a Programação Evolucionária e as Estratégias Evolucionárias, apesar de terem nascido de forma independente, se assemelham por três aspectos principais:

1. Trabalham com os próprios valores das variáveis - em lugar de suas codificações, como no AG;

2. Usam mutação com o mesmo método Gaussiano (multivariado com média zero);

3. Adotam seleção entre pais e filhos para formar nova população no caso da política $(\mu + \lambda)$. Por outro lado as diferenças mais marcantes são:

   - A Programação Evolucionária seleciona de forma aleatória em um torneio, promovido entre os indivíduos da próxima população;
   - Cada nova solução compete com certo número de oponentes e permanece na população em função de seu desempenho;
   - Em contrapartida, em Estratégias Evolucionárias, usa-se seleção determinística, simplesmente descartando os piores indivíduos.

Estas duas metodologias, por trabalharem com os valores reais das funções a otimizar e se apoiarem sobretudo no operador de mutação, formam um todo à parte bastante diferenciado, com relação à sistemática dos Algoritmos Genéticos.

3.3.3 Algoritmos genéticos

Segundo Bittencourt [11], os algoritmos genéticos (AGs) são o ramo mais conhecido da Computação Evolucionária, e tiveram origem no trabalho de Holland, também nos anos sessenta. Ao contrário dos dois esquemas vistos acima - EE e PE, os AG's conceitualmente apresentam um escopo mais amplo do que a simples otimização. Eles são apresentados como um modelo para a aprendizagem de máquina. Há uma explicação para este fato: originalmente, os AG's estavam muito fortemente ligados a modelos de aprendizado automático, como o demonstra a ênfase dada por Holland aos chamados sistemas classificadores, que são um modelo de aprendizado de máquina usando AGs. Só mais tarde, a partir da publicação do livro *Genetic algorithms in search, optimization, and machine learning*, que a idéia de

otimização passou a ocupar o lugar central na teoria dos AGs. No livro *Adaptation in Natural and Artificial Systems*, Holland introduz o assunto no âmbito da genética, economia, teoria de jogos, pesquisa, reconhecimento de padrões e inferência estatística, controle e otimização de funções e sistema nervoso central.

Segundo Falqueto [27] dos três paradigmas básicos da Computação Evolucionária, os AGs formam a estratégia mais biologicamente inspirada, por simular mais fielmente o processo evolucionário, muito embora tenha inúmeros pontos em que se distancia muito da Natureza.

3.4 Computação evolucionária & RNAs

O uso da computação evolucionária já tem sido feito com a finalidade de enfrentar a complexidade de problemas a serem tratados por métodos conexionistas e com algum sucesso. A maioria dos trabalhos em que algoritmos evolutivos são usados simultaneamente com redes neurais tratam do problema de treinamento, onde os mesmos são usados como um paradigma de treinamento de RNAs. Exemplos são a utilização de algoritmos genéticos para treinamento de RNAs, a combinação do algoritmo de retropropagação e AGs para evitar mínimos locais e AGs para treinamento de RNAs com neurônios nebulosos [6].

Algoritmos evolutivos podem ser aplicados às RNAs de duas formas: i) para o ajuste dos pesos das conexões e ii) e para otimização de topologia de rede.

Hoje em dia muitos trabalhos em que algoritmos evolutivos são usados para a escolha da arquitetura de RNA que melhor se adapte à solução de um problema estão sendo utilizados. Pode-se citar por exemplo [3], [65], [31] e principalmente os trabalhos de Xin Yao [94], [90], [92], [95] que "evoluiu" tanto a arquitetura quanto os pesos das então chamadas redes neurais evolucionárias, através de algoritmos de programação evolucionária (PE).

O algoritmo evolucionário descrito abaixo realiza o treinamento de uma população de redes até chegar a redes bem treinadas onde os pesos ideais são alcançados [95].

Para evoluir os pesos das conexões de uma rede neural através de PE tem-se os passos a seguir:

1. Gerar uma população inicial de μ indivíduos aleatórios, e atribuir $k = 1$. Cada indivíduo é um par de vetores com valores reais, (w_i, η_i), $\forall i \in \{1, \ldots, \mu\}$, onde w_i's são vetores de pesos das conexões e η_i's são vetores das variâncias para as mutações Gaussianas (também conhecidos como parâmetros de estratégia em algoritmos evolucionários (AEs) auto-adaptativos). Cada indivíduo corresponde a uma rede neural artificial.

2. Cada indivíduo (w_i, η_i), $1, \ldots, \mu$, cria um único descendente $(w_i^{'}, \eta_i^{'})$ por meio das equações: para $j = 1, \ldots, n$,

$$\eta_i^{'}(j) = \eta_i(j) \exp(\tau^{'} N(0,1) + \tau N_j(0,1)) \quad (3.17)$$

$$w_i^{'}(j) = w_i(j) + \eta_i^{'}(j) N_j(0,1), \quad (3.18)$$

onde $w_i(j)$, $w_i^{'}(j)$, $\eta_i(j)$, e $\eta_i^{'}(j)$ denotam a $j-$ésima componente dos vetores w_i, $w_i^{'}$, η_i e $\eta_i^{'}$, respectivamente. $N(0,1)$ denota um número aleatório de uma dimensão gerado a partir da uma distribuição normal com média 0 e variância 1. $N_j(0,1)$ indica que o número aleatório é gerado novamente para cada valor de j. Para os parâmetros τ e $\tau^{'}$ são geralmente atribuídos os valores $(\sqrt{2\sqrt{n}})^{-1}$ e $(\sqrt{2n})^{-1}$ respectivamente. $N_j(0,1)$ na Eq. 3.18 pode ser substituído por mutação Cauchy [91] para uma evolução mais rápida.

3. Determinar a aptidão para todos os indivíduos, incluindo todas as redes mães e filhas, baseada no erro de treinamento. Diferentes funções de erro podem ser usadas aqui.

4. Fazer a comparação unindo mães (w_i, η_i) e filhas $(w_i^{'}, \eta_i^{'})$, $\forall i \in \{1, \ldots, \mu\}$. Para cada indivíduo, q oponentes são escolhidos aleatoreamente através de uma distribuição uniforme para todas as mães e filhas. Em cada comparação, se a aptidão do indivíduo não é menor do que de todos os seus oponentes, ele recebe um "win". Selecionar μ indivíduos de (w_i, η_i) e $(w_i^{'}, \eta_i^{'})$, $\forall i \in \{1, \ldots, \mu\}$, que tem os maiores números de "wins" para formar a próxima geração. (Este esquema de torneio de seleção pode ser substituído por outros.)

5. Parar se o critério de alcance foi satisfeito; senão, $k = k + 1$ e vá para o Passo 2.

Similar à evolução dos pesos de conexões, as duas maiores fases envolvidas na evolução de arquiteturas são o esquema de representação do genótipo das arquiteturas (que pode ser direto ou indireto) e o algoritmo evolucionário usado para evoluir as arquiteturas de RNAs. Um dos assuntos chaves na codificação de arquiteturas de RNAs é decidir quantas informações sobre uma arquitetura deverá ser codificada. Em um extremo, todos os detalhes, i.e., todas as conexões e neurônios de uma arquitetura podem ser especificados. Este tipo de esquema de representação é chamado de *codificação direta*. Por outro lado, somente a maioria dos parâmetros importantes de uma arquitetura, tal como o número de camadas intermediárias e neurônios intermediários em cada camada são codificados. Outros detalhes sobre arquitetura são deixados para o processo de treinamento decidir. Este tipo de esquema de representação é chamado *codificação indireta*. Depois do esquema de representação ter

sido escolhido, a evolução das arquiteturas pode progredir de acordo com o ciclo mostrado abaixo. O ciclo pára quando uma RNA satisfatória é encontrada.

1. Decodificar cada indivíduo na geração atual em uma arquitetura. Se o esquema de codificação indireto é usado, os detalhes da arquitetura serão especificados por regras ou pelo processo de treinamento.

2. Treinar cada RNA, com a arquitetura de-codificada por uma regra de aprendizado pré-definida (alguns parâmetros da regra de aprendizagem podem ser evoluídos durante o treinamento), iniciando a partir de diferentes conjuntos de pesos de conexões aleatórios e, se existe, parâmetros de regra de aprendizagem.

3. Computar a aptidão de cada indivíduo de acordo com o resultado do treinamento acima e outros critérios de desempenho tal como a complexidade das arquiteturas.

4. Selecionar os pais da população baseados em seus valores de aptidão.

5. Aplicar operadores de busca nos pais e gerar descendentes que formarão a próxima geração.

A maioria das pesquisas têm se concentrado na evolução da estrutura topológica de RNAs. Relativamente poucas pesquisas são realizadas a respeito da evolução das funções de transferências dos neurônios [95].

3.4.1 Evolução simultânea da arquitetura e conexão de pesos

A maioria das abordagens evolucionárias evolui a arquitetura somente, sem qualquer conexão de pesos. O treinamento é realizado somente depois que a rede ótima é encontrada. Principalmente se o esquema de codificação utilizado é o indireto. O maior problema com a evolução das arquiteturas sem as conexões de pesos é o *ruído na avaliação da aptidão* [194] apud [95].

Yao e Liu [93] desenvolveram um sistema automático, EPNet, baseados em PE tanto para evolução da arquitetura de RNAs como conexões de pesos. A EPNet não usa quaisquer operadores de cruzamento e sim operadores de mutação para modificar arquiteturas e pesos. A evolução comportamental (i.e. funcional) foi usada e enfatiza na EPNet ao invés da evolução genética. Técnicas foram desenvolvidas para manter a ligação comportamental entre um indivíduo e seus descendentes. A Fig. 3.4 mostra a estrutura principal da EPNet.

A EPNet usa seleção baseada em ordenação e cinco mutações: treinamento híbrido, deleção de nós, deleção de conexão, adição de conexão e adição de nós. O treinamento

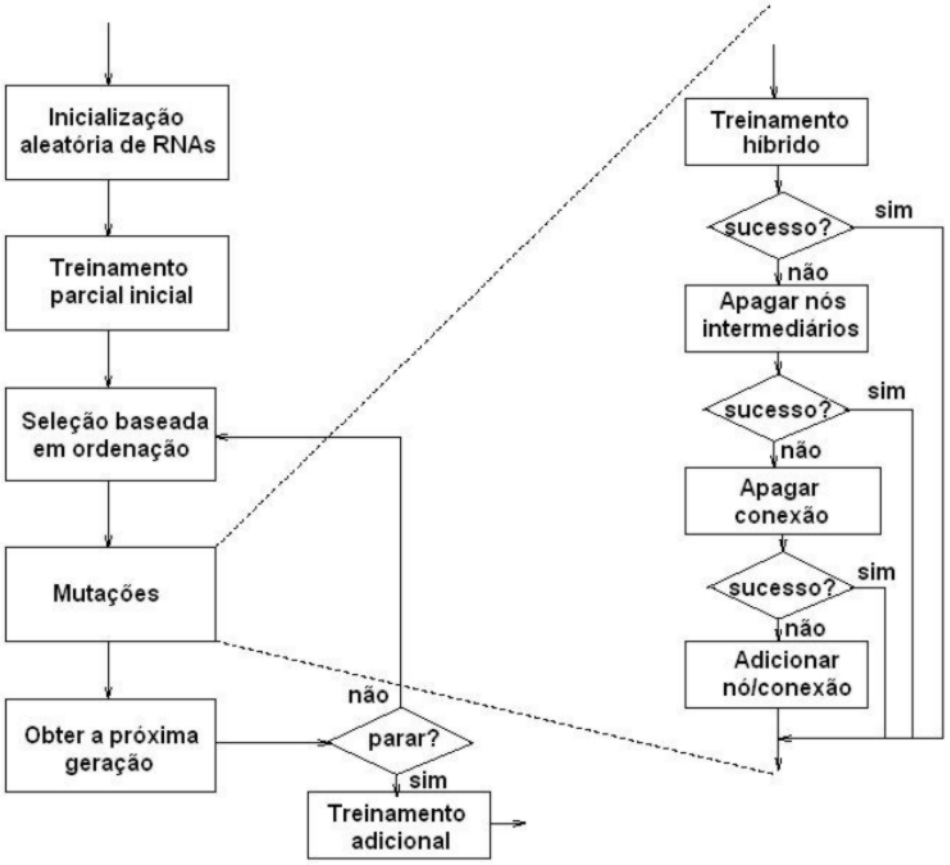

Figura 3.4: A estrutura principal da EPNet. Adaptada de [93].

híbrido é a única mutação na EPNet que modifica os pesos das RNAs, os autores denominaram como MBP (*modified backpropagation*). Esta metodologia está baseada numa modificação do algoritmo de retropropagação com uma taxa de aprendizado adaptativa e têmpera simulada[4]. As outras quatro mutações são usadas para acrescentar ou cortar nós intermediários e conexões.

O número de épocas usadas pelo MBP para treinar cada RNA em uma população é definida por dois parâmetros especificados pelo usuário. Não há garantia de que uma RNA irá convergir a um ótimo local depois destas épocas. Então este treinamento é chamado de treinamento parcial. Ele é usado para construir pontes de ligação entre um indivíduo e seus descendentes.

As cinco mutações são realizadas seqüencialmente. Se uma mutação leva ao melhor

[4]Têmpera simulada é uma metodologia de solução de problemas de busca e otimização não pertencente à "área evolucionária", mas que também é inspirada pela Natureza. Este método seleciona um ponto aleatório e a partir daí faz um movimento de valor também aleatório na superfície de busca. O novo ponto é aceito caso leve a uma solução melhor. Inicialmente, mesmo movimentos que leve à piora são aceitos. À medida que o processo segue estes movimentos são paulatinamente proibidos [27].

descendente, então ela é tida por bem sucedida. Nenhuma das mutações adiante serão aplicadas. De outra forma a próxima mutação é realizada. A motivação na questão da ordem das mutações é para encorajar a evolução de RNAs compactas sem sacrificar a generalização. O conjunto de validação é usado na EPNet para medir a aptidão de um indivíduo.

A EPNet foi testada extensivamente com vários problemas padrões de comparação (*benchmarks*) e obteve excelentes resultados. Os problemas foram o da paridade de tamanho de quatro a oito, o problema dos dois espirais, o problema de câncer do seio, o problema diabetes, o problema de doença do coração, o problema tiróide, o problema do cartão de crédito australiano, o problema de predição de séries temporais de Mackey-Glass, etc. Por meio deste sistema muitas RNAs compactas e com boa capacidade de generalização foram encontradas.

Com isto, este sistema serviu de inspiração para o trabalho realizado nesta dissertação. Dado que reconhecimento de faces é uma aplicação potencial de RNAs, e dado que evolução de pesos e arquiteturas de RNAs são aplicações pontenciais para algoritmos de PE, foi então implementado um sistema semelhante e mais simplificado, que evolui arquitetura de redes com ou sem herança de conhecimento. Isto será mostrado com mais detalhes no Capítulo 5.

Capítulo 4

Implementações Prévias

4.1 Introdução

Este capítulo visa a exposição dos experimentos preliminares que foram realizados a fim de dar início ao desenvolvimento de soluções baseadas em RNAs para o problema de reconhecimento de faces na forma 3D.

4.2 Ensaios preliminares

Inicialmente, os experimentos se utilizaram da primeira base de dados de faces construída por Zimmermann [102]. Ela contém informações de quatro pessoas, as primeiras a submeterem suas faces a técnicas de iluminação estruturada. Foram produzidas então as primeiras matrizes de alturas com a informação da superfície 3D de cada face. Estas matrizes por sua vez serviram de entradas para RNAs do tipo multi-camadas de perceptrons (MLP) ou redes diretas.

O objetivo deste experimento foi analisar o comportamento de uma RNA mediante vários aspectos: diferentes tamanhos das matrizes de altura, tempo de treinamento, quantidade de neurônios na camada intermediária e análise dos diferentes desempenhos de reconhecimento.

Esta primeira base de dados foi produzida a partir de um sistema óptico de aquisição desenvolvido junto ao pacote de ferramentas MATLAB[1]. Ao total são 240 exemplos de 4

[1]MATLAB (Language of Technical Computing) é um pacote de ferramentas que integram computação matemática, visualização, e uma poderosa linguagem para prover um ambiente flexível para computação técnica. Sua arquitetura aberta proporciona facilidade de uso tanto dele quanto dos produtos que o acompanham para explorar dados, criar algoritmos, e criar ferramentas voltadas a aplicações específicas. A versão utilizada para realizar todos os experimentos deste trabalho de dissertação foi a 6.1, esta versão foi comprada pelo Projeto SORFACE. Mais informação sobre o MATLAB podem ser encontradas no site

indivíduos mostrados na Fig. 4.1 (à esquerda). Para cada indivíduo foram produzidas 12 imagens que processadas geraram as matrizes de alturas. Estas matrizes apresentam 5 categorias de resoluções (quantidade de $linhas \times colunas$). Exemplo do mapeamento 3D é ilustrado na Fig. 4.1 (à direita). Os exemplos para cada indivíduo diferem quanto a variações na expressão facial (com relação a olhos, sobrancelhas, boca, bochecha) e posicionamento espacial (rotação em torno dos eixos X, Y e Z e escalonamento em Z). As implementações de redes neurais também foram realizadas no programa MATLAB.

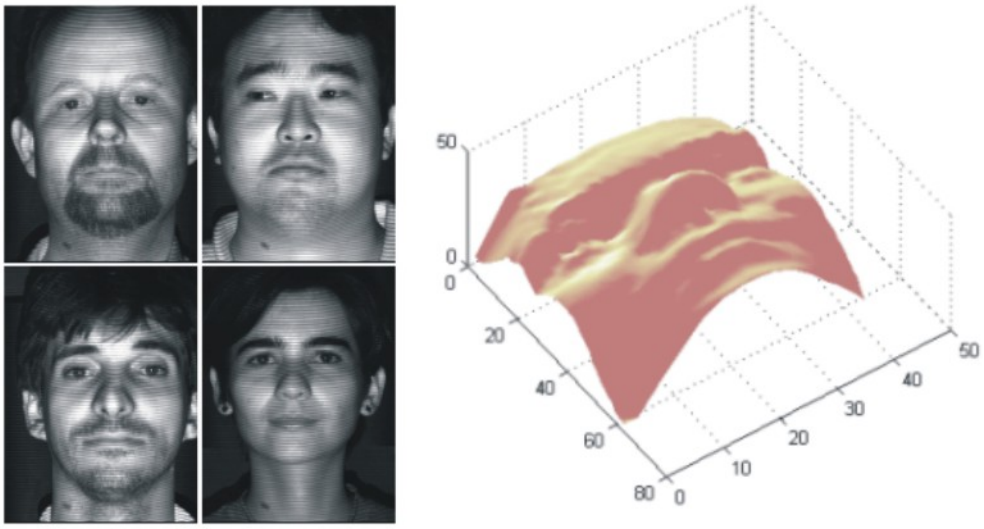

Figura 4.1: Faces com projeções de franjas e uma forma de face 3D.

Como o objetivo do sistema é de verificação, o conjunto de treinamento foi constituído da seguinte maneira: tomou-se um indivíduo como verdadeiro, aquele que interessa ser reconhecido e os outros 3 indivíduos como falsos (os que não podem ser aceitos pelo sistema). Para cada indivíduo no conjunto de treinamento tem-se 8 variações, portanto o conjunto apresenta 32 exemplos. O conjunto de validação possui 4 exemplos para cada indivíduo, dando o total de 16 exemplos.

Através da ferramenta MATLAB foi utilizada uma rede direta de três camadas. O algoritmo de aprendizado utilizado foi uma versão modificada do algoritmo padrão retropropagação. O mesmo apresenta as taxas de momento e aprendizagem adaptáveis durante o treinamento. A rede teve a configuração de $r \times h \times 1$, onde r é o número de neurônios na camada de entrada (conforme as 5 resoluções diferentes das matrizes de altura da face) e h o número de neurônios na camada intermediária. A rede neural foi aleatoriamente inicializada antes de iniciar o treinamento e a quantidade de épocas foi configurada em 200. A resolução das imagens foi modificada (21×15, 31×23, 41×29, 51×37 e 61×43) e o número de nós

http://www.mathworks.com/products/matlab/.

intermediários foi incrementado de 5 a cada experimento. O erro de generalização, isto é, o erro quadrado médio para os padrões não vistos pela rede durante a fase de treinamento (padrões do conjunto de teste) foi anotado. A resolução r e o número de nós intermediários h que provêem a menor complexidade e o menor erro de generalização foram escolhidos para o sistema. A seleção da melhor rede baseou-se no menor erro médio alcançado no conjunto de validação (teste).

Na Fig. 4.2 o erro de generalização oscila quanto ao número de neurônios na camada intermediária e respectivas resoluções, correspondentes ao tamanho da entrada da rede. Através desta figura percebe-se o menor erro de generalização em $r = 51 \times 37$ e $h = 30$. Assim este experimento se baseou nesta rede ótima com a arquitetura de $1887 \times 30 \times 1$, onde a entrada corresponde a um vetor de 1887 posições contendo as alturas da face e uma saída para classificar se o indivíduo é verdadeiro ou não. As funções de transferência utilizadas foram tangente hiperbólica para a camada intermediária e sigmoidal ou logística para a camada de saída.

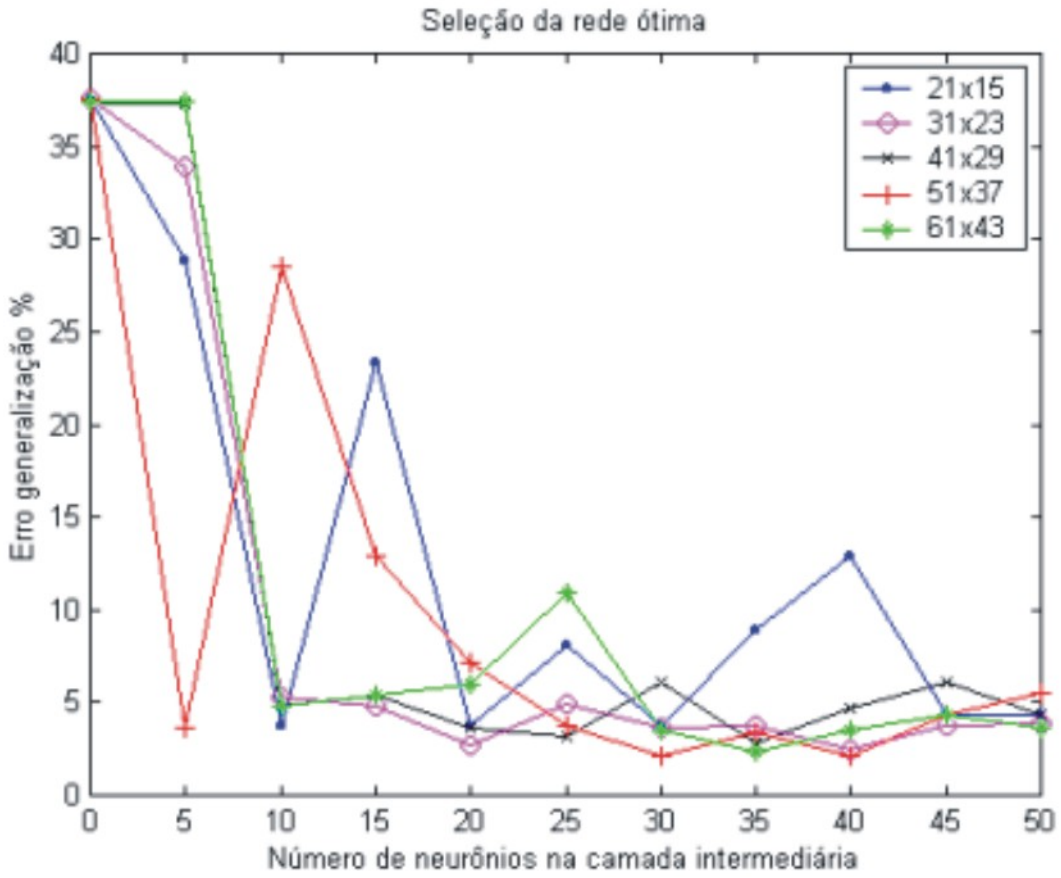

Figura 4.2: Seleção da arquitetura de rede ótima.

Para o treinamento da rede foi utilizada a função “traingdx” do programa MATLAB que

implementa um algoritmo de treinamento retropropagação modificado, onde os valores de pesos e bias são atualizados de acordo com o gradiente descendente do momento e taxa de aprendizagem adaptativa. Com o treinamento desta rede neural ótima, obteve-se o seguinte gráfico de desempenho mostrado na Fig. 4.3. Nota-se sua rápida convergência ao erro pré-estabelecido de $1 * 10^{-3}$ em apenas 123 épocas.

Na tentativa de simular um sobre-treinamento, situação semelhante ao gráfico da Fig. 4.4, configurou-se o número de épocas para 77.000, resultando uma evolução de treinamento mostrada no gráfico da Fig. 4.5. Concluiu-se que o fato da não ocorrência de sobre-treinamento seja devido ao algoritmo de treinamento utilizado através da ferramenta MATLAB ser otimizado. Isto é, ele monitora o processo de treinamento para evitar que um sobre-treinamento venha a ocorrer. E é por isso que se estabiliza em um erro de aproximadamente $1 * 10^{-31}$, sendo sua convergência neste erro muito rápida.

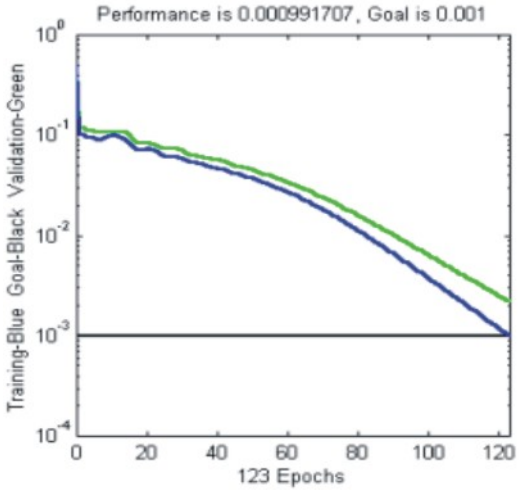

Figura 4.3: Treinamento da rede neural MLP ótima

A avaliação do desempenho do sistema proposto é uma maneira de medir sua qualidade quanto ao aspecto de autenticação, avaliando suas respostas mediante a imagens de faces nunca vistas por ele. A tarefa deste classificador neural é essencialmente identificar duas classes de padrões, isto é, os exemplos de face pertencentes ao usuário que se quer identificar (autêntico) ou não. Os exemplos pertencentes ao usuário verdadeiro são coletivamente chamados de uma *população de ovelhas* ou uma *classe positiva*, enquanto os exemplos não pertencentes ao usuário, isto é, falsos, são coletivamente chamados de uma *população de lobos* ou uma *classe negativa*. Neste experimento convencionou-se atribuir 0 à classe negativa e 1 à classe positiva, conforme visto na Seção 2.2.2.

Cada uma, das duas classes de exemplos, possui sua própria probabilidade de ser identificada pela rede neural devidamente treinada. Esta probabilidade é definida por uma dis-

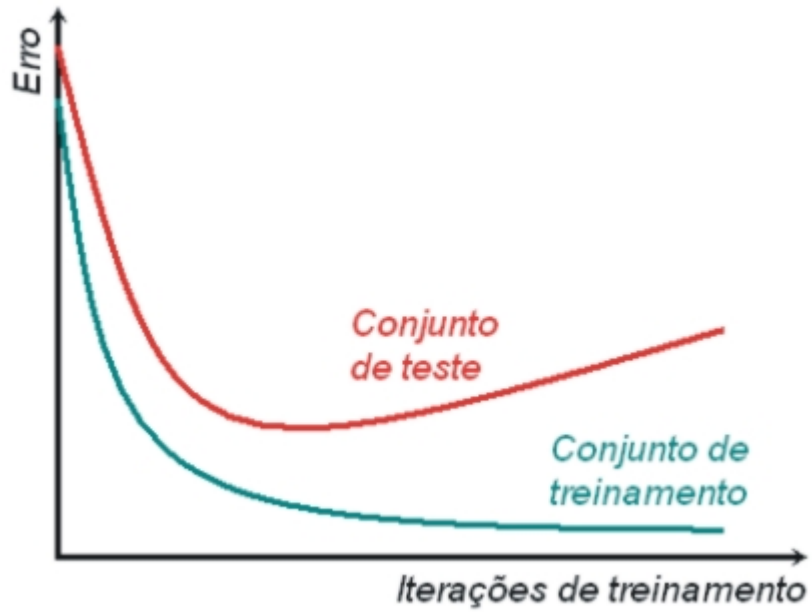

Figura 4.4: Um exemplo de ocorrência de sobre-treinamento.

tribuição normal onde os resultados estão no limite de zero a um, como mostrado na Fig. 4.6. A rede foi treinada até convergir ao erro de $1 * 10^{-2}$ e depois avaliada com os exemplos de teste para as classes positiva e negativa. Aos padrões de teste foram acrescentados vários níveis de ruído, como exemplificado na Fig. 1.6.

Ao se executar a análise da distribuição de lobos e ovelhas para a checagem do desempenho deste classificador neural escolhe-se $T = 0,5$. Isto é, se a resposta do classificador é maior do que $0,5$, a pessoa é aceita mediante sua identificação declarada, senão é rejeitada. A área A na Fig. 4.6, como visto na Seção 2.2.2, mostra os casos de falsa rejeição e B os casos de falsa aceitação.

Aplicando-se a análise de limiar (Seção 2.2.2), que é o resultado do deslocamento de T ao longo do eixo de resultados, produz-se o gráfico da Fig. 4.7. Este gráfico é importante pois leva à checagem da taxa de erro (EER) onde $FAR = FRR$. Assim, conclui-se que o limiar T ideal para este experimento estaria muito próximo a $0,6$, conforme é visivelmente percebido.

4.3 A quantidade de exemplos por pessoa

Houve a princípio a necessidade de determinar a quantidade ideal de exemplos de faces por pessoa. Em uma segunda etapa dispunha-se de uma base de dados maior, com amostras de 50 pessoas. Para cada indivíduo foram produzidos 160 exemplos de faces constituídos de 32

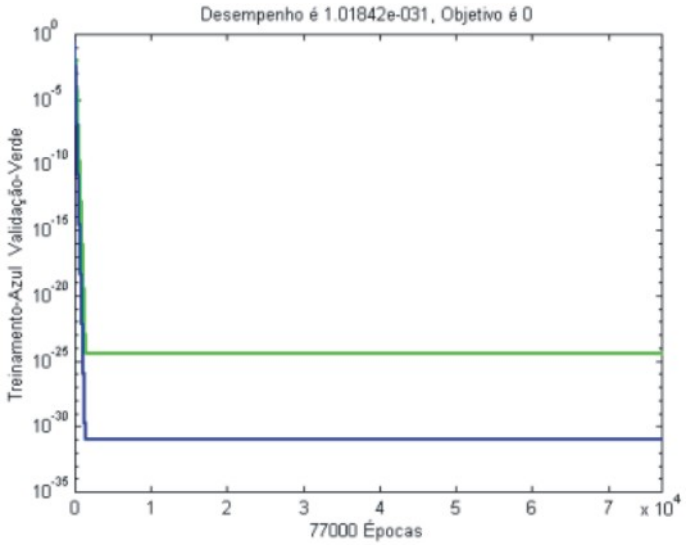

Figura 4.5: Tentativa de simular o sobre-treinamento.

expressões submetidas a 5 níveis de ruído. Com isto, surgiu a dúvida de que esta quantidade fosse demasiada. Então realizou-se o experimento descrito a seguir a fim de determinar a quantidade ideal de exemplos por pessoa para compor o conjunto de treinamento.

O experimento apresentou 33 passos, onde a cada passo foram construídos 33 conjuntos de treinamento diferentes. Com isto analisou-se a capacidade de generalização da rede em relação ao conjunto de teste (que é sempre o mesmo). Ou seja, a princípio se desejava encontrar um número de exemplos da face *ovelha* (< 160) onde, a partir do mesmo, não houvesse decréscimo algum na taxa de erro para o conjunto de teste. Os conjuntos de treinamento se estabeleceram da seguinte maneira:

Passo 1:	1 exemplo da pessoa a reconhecer	$+\ 49$ pessoas diferentes
Passo 2:	2 exemplos da pessoa a reconhecer	$+\ 49$ pessoas diferentes
Passo 3:	5 exemplos da pessoa a reconhecer	$+\ 49$ pessoas diferentes
Passo 4:	10 exemplos da pessoa a reconhecer	$+\ 49$ pessoas diferentes
Passo 5:	15 exemplos da pessoa a reconhecer	$+\ 49$ pessoas diferentes
$\vdots$	$\vdots$	$\vdots$
Passo 33:	160 exemplos da pessoa a reconhecer	$+\ 49$ de pessoas diferentes

Já o conjunto de teste se manteve o mesmo para todos os passos, com o total de 100 exemplos (50 exemplos da pessoa que se deseja reconhecer $+\ 50$ exemplos de diferentes pessoas).

A mesma rede neural direta foi utilizada em todos os passos do algoritmo e apresentou a configuração de $(315 \times 100 \times 2)$. O algoritmo de treinamento utilizado foi o de retropropagação com taxa de momento e aprendizado adaptáveis (*traingdx*).

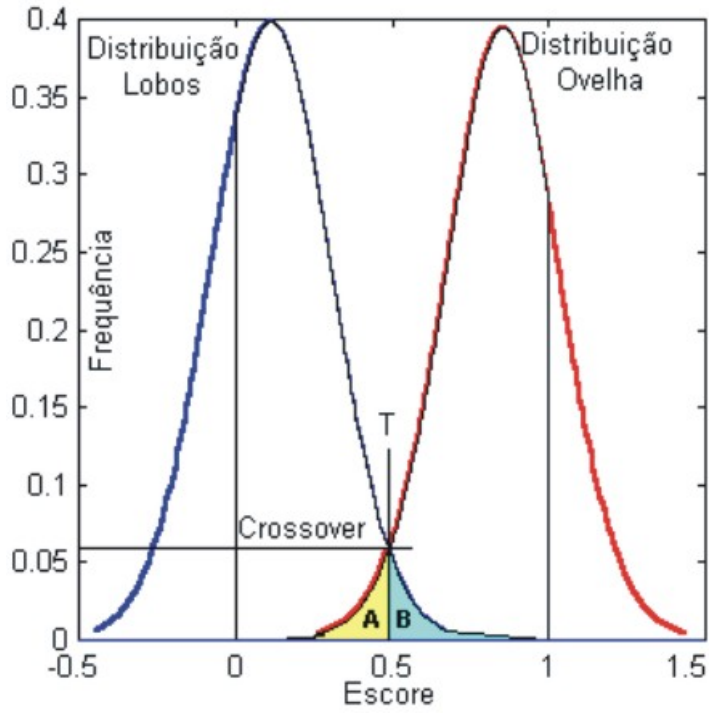

Figura 4.6: Distribuição normal das duas classes de exemplo.

Nos 33 ensaios, iniciou-se a rede com os mesmos pesos aleatórios e o erro médio a alcançar foi configurado em $0,05$. A Fig. 4.8 apresenta alguns gráficos dos treinamentos realizados, onde a curva em azul denota e evolução do erro quadrado médio no conjunto de treinamento e a curva em verde o erro no conjunto de teste. Nota-se que quando a quantidade de exemplos por face para o conjunto de treinamento se aproxima de 160 a convergência do erro no conjunto de teste é mais rápida. Com o gráfico da Fig. 4.9 conclui-se que 160 é uma boa quantidade de exemplos por pessoa, dado que produziu-se o menor erro alcançado pelo conjunto de teste.

4.4 Comparação das velocidades de convergência

Neste outro experimento também foi usada a mesma base de dados anterior. Todo o processo que vai desde a obtenção das formas 3D das faces até o reconhecimento com redes neurais foi implementado através do programa MATLAB. Com exceção da implementação do algoritmo de retropropagação modificado, *quickprop*, feito por Scott Fahlman em linguagem Lisp e traduzido para a linguagem C por Terry Regier da Universidade da California, Berkeley. Este algoritmo quickprop foi adaptado para ser usado neste experimento.

Para efeito de comparação, foram usados 4 tipos do algoritmo de aprendizado com

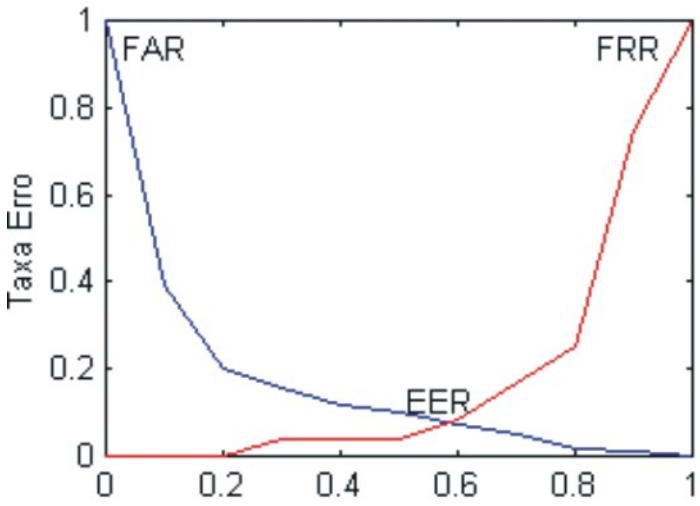

Figura 4.7: Distribuições FAR e FRR versus limiar.

retropropagação: sua implementação padrão (*traingd*) e três implementações modificadas (*traingdx*, *traingdx-fuzzy* e *quickprop*), suas modificações tem o intuito de acelerar a convergência do algoritmo padrão. Os resultados mostram o quanto a velocidade de aprendizado da rede neural artificial pode aumentar.

A base de dados disponível apresentava faces de 50 pessoas, com 5 resoluções diferentes para as matrizes de alturas (21×15, 31×23, 41×29, 51×37 e 61×43). Utilizou-se apenas os 4 indivíduos mostrados na Fig. 4.10 em virtude da necessidade de economia de recurso de máquina e tempo de processamento. Além de não apresentar grande informação adicional a utilização dos exemplos de face das 50 pessoas. Cada um dos 4 indivíduos possui 32 exemplos (variações faciais), resultando assim em 128 exemplos de padrões. Os exemplos para cada um também diferem quanto a variações na expressão facial (com relação a olhos, sobrancelhas, boca, bochecha) e posicionamento espacial (rotação em torno dos eixos X, Y e Z e escalonamento no eixo Z). A resolução da matriz de alturas utilizada foi a menor (21 linhas $\times$ 15 colunas), resultando assim a necessidade de 315 neurônios na camada de entrada da rede.

Como o objetivo é construir um sistema de verificação, espera-se que a saída da rede, que possui 2 neurônios, responda da seguinte forma: $(1\ 0)$ para o indivíduo *ovelha*, ou seja, o primeiro neurônio da camada de saída deve ser ativado, $(0\ 1)$, para os outros 3 indivíduos restantes (*lobos*), sendo ativado o segundo neurônio da camada de saída. Estes são os que não poderiam ser aceitos pelo sistema numa simulação real.

O conjunto de treinamento foi montado tendo em vista as 20 variações por pessoa, como descrito na Tab. 4.1. Tem-se então o total de 80 exemplos que são previamente embaralhados

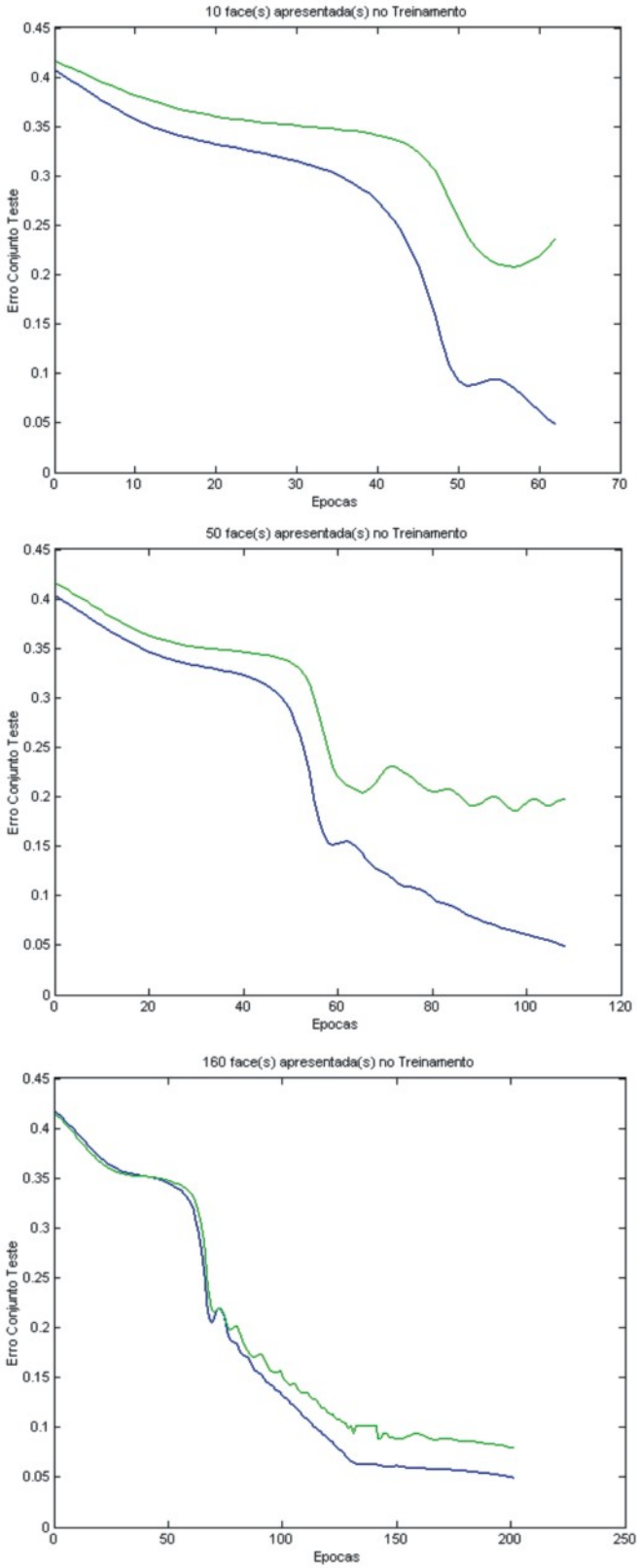

Figura 4.8: Gráficos de alguns treinamentos mediante a quantidade de faces *ovelha*.

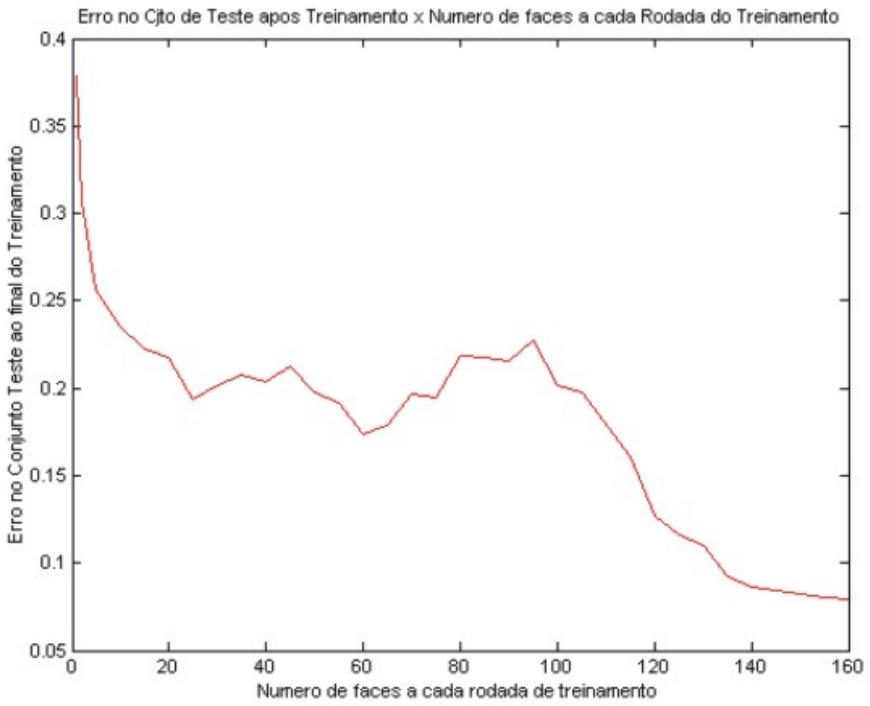

Figura 4.9: Erro no conjunto de teste $\times$ faces *ovelha* no conjunto de treinamento.

antes de serem apresentados à rede na etapa de treinamento. Já o conjunto de validação, que no treinamento é usado para monitorar a capacidade de generalização da rede, apresentou 12 variações por pessoa, como descrito na Tab. 4.2, resultando assim em 48 exemplos.

A rede neural artificial apresenta a topologia de uma rede direta com a seguinte configuração: $315 \times 15 \times 2$. A seguir serão apresentadas os gráficos dos desempenhos das quatro modalidades do algoritmo de aprendizado com retropropagação, o padrão *traingd*, o *traingdx*, com taxa de aprendizado e momento adaptáveis, o *traingdx-fuzzy*, com taxa de aprendizado e momento adaptáveis por controlador de lógica nebulosa, conforme descrito em [36], e finalmente o *quickprop*, desenvolvido por Fahlman. Os algoritmos de aprendizado *traingd* e *traingdx* são implementações pertencentes ao pacote de redes neurais do programa MATLAB, enquanto o *traingdx-fuzzy*, foi uma modificação feita no algoritmo *traingdx* para incorporar as funções do motor de inferência nebulosa para controle das taxas de aprendizado e momento. Já o *quickprop*, desenvolvido em linguagem C, foi obtido através do trabalho de Fahlman e aplicado ao problema de reconhecimento de faces 3D. A cada ensaio a rede é inicializada com valores aleatórios para as matrizes de peso e bias e treinada com os respectivos algoritmos de retropropagação até o erro pré-estabelecido de $1 * 10^{-2}$. O objetivo é fazer um comparativo entre as velocidade de convergência de cada um.

A seguir tem-se os gráficos dos respectivos desempenhos, mostrados na Fig. 4.11. As

Figura 4.10: Pessoas utilizadas para o reconhecimento.

curvas em azul denotam a evolução do erro no conjunto de treinamento e as curvas em verde no conjunto de validação. Nas quatro modalidades houve 100% de acerto no reconhecimento das faces *ovelha* $\times$ *lobo*.

Percebe-se que, para o problema de reconhecimento automático de faces, o algoritmo de treinamento que apresenta um aprendizado mais rápido é o *traingdx-fuzzy*. Ele chega ao erro estabelecido com apenas 132 épocas, e com o erro no conjunto de validação ainda menor, indicando um ótimo aprendizado da rede. Isto se deve à união de duas poderosas abordagens da inteligência artificial que são redes neurais e a teoria dos conjuntos nebulosos.

Os algoritmos *traingdx* e *quickprop* também foram muito rápidos, porém o algoritmo de retropropagação padrão, *traingd*, apresentou o pior desempenho. Foram necessárias 10766 épocas para ele convergir até o erro desejado. Com isso notamos a importância das modificações no algoritmo padrão de retropropagação para um aprendizado mais rápido.

Todos estes experimentos relatados serviram como um impulso inicial para se chegar ao resultado final deste trabalho de dissertação. A partir dos mesmos concluiu-se que: i) não seria necessária uma rede muito complexa (elevado número de neurônios na camada intermediária) para o problema de reconhecimento de faces tratado, apesar do número de neurônios na camada de entrada da rede ser de no mínimo 315, ii) quanto maior a quantidade

Expressão facial	Sigla	Indice
normal olhos abertos	$NOAB$	1
normal olhos fechados	$NOFE$	2
normal sobrancelhas altas	$NSAL$	3
normal sobrancelhas relaxadas	$NSRE$	4
normal boca fechada	$NBFE$	5
normal boca aberta	$NBAB$	6
normal bochecha inflada	$NBIN$	7
normal com óculos	$NCOC$	8
normal olhos abertos - rotação eixo X	$RX-2$	9
normal olhos abertos - rotação eixo X	$RX+0$	10
normal olhos abertos - rotação eixo X	$RX+2$	11
normal olhos abertos - rotação eixo Y	$RY-2$	12
normal olhos abertos - rotação eixo Y	$RY+0$	13
normal olhos abertos - rotação eixo Y	$RY+2$	14
normal olhos abertos - rotação eixo Y	$RZ-2$	15
normal olhos abertos - rotação eixo Z	$RZ+0$	16
normal olhos abertos - rotação eixo Z	$RZ+2$	17
normal olhos abertos - escalonamento eixo Z	$EZ-2$	18
normal olhos abertos - escalonamento eixo Z	$EZ-0$	19
normal olhos abertos - escalonamento eixo Z	$EZ+2$	20

Tabela 4.1: Ensaio de expressões faciais para o conjunto de treinamento

de exemplos de faces por pessoa no conjunto de treinamento, melhor seria sua capacidade de generalização para os exemplos não vistos pela rede e iii) a utilização dos algoritmos de retropropagação modificados coopera para uma fase de treinamento mais rápida. Portanto estas informações foram consideradas durante a realização dos experimentos finais.

Expressão facial	Sigla	$Indice$
normal olhos semi-abertos	$NOSA$	1
normal sobrancelhas baixas	$NSBA$	2
normal boca semi-aberta	$NBSU$	3
normal bochecha sugada	$NBIN$	4
normal olhos abertos - rotação eixo X	$RX-1$	5
normal olhos abertos - rotação eixo X	$RX+1$	6
normal olhos abertos - rotação eixo Y	$RY-1$	7
normal olhos abertos - rotação eixo Y	$RY+1$	8
normal olhos abertos - rotação eixo Z	$RZ-1$	9
normal olhos abertos - rotação eixo Z	$RZ+1$	10
normal olhos abertos - escalonamento eixo Z	$EZ-1$	11
normal olhos abertos - escalonamento eixo Z	$EZ+1$	12

Tabela 4.2: Ensaio de expressões faciais para o conjunto de validação

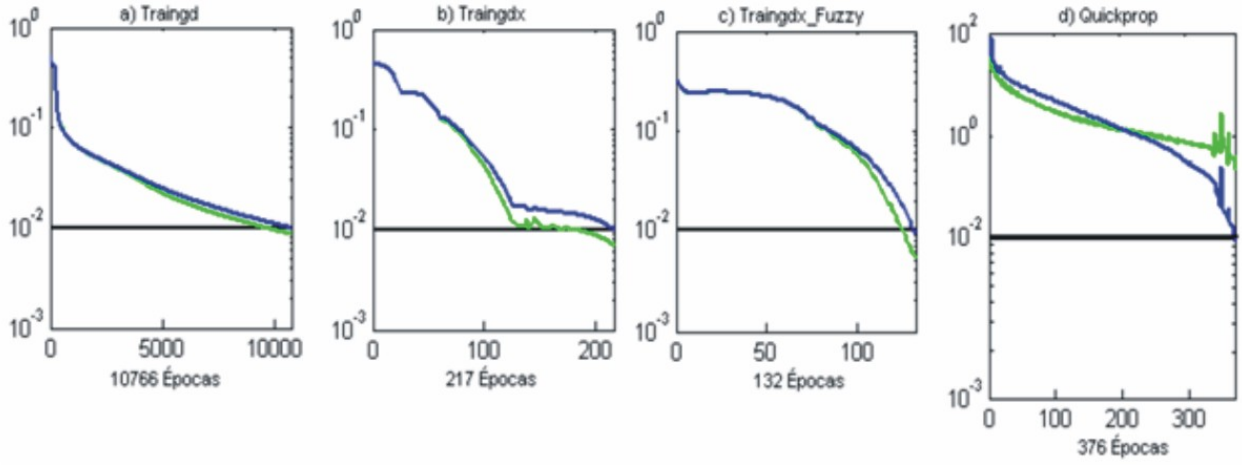

Figura 4.11: Desempenhos dos algoritmos de retropropagação e suas variantes.

Capítulo 5

Resultados

5.1 Introdução

Este capítulo tem por objetivo mostrar os passos que foram dados até se chegar à inusitada conjectura de que o problema de reconhecimento de faces abordado neste trabalho é linearmente separável. Durante a utilização da técnica de programação evolucionária para a evolução da arquitetura de redes diretas de três camadas (entrada, intermediária e saída), foram encontradas redes com apenas um neurônio na camada intermediária que desempenharam perfeitamente a tarefa de classificação das faces em classes *ovelha* e *lobo* (seção 2.2.2). Isto levou à suposição de que o problema abordado seria linearmente separável. Para constatar de fato que os padrões de formas 3D de faces humanas são linearmente separáveis, foram utilizadas redes de uma camada como Perceptron e Adaline, que somente conseguem classificar padrões nesta situação.

5.2 PE para evolução de arquitetura

Como mostrado na seção 3.4, algoritmos de PE são utilizados com bastante eficiência na evolução tanto dos pesos como da arquitetura de RNAs. Eles foram aplicados em diversos problemas tais como XOR, paridade de bits, o problema do cartão de crédito Australiano [94], entre outros. Os experimentos realizados para este trabalho foram inspirados nestes algoritmos, porém de uma forma mais simplificada, haja visto que foram evoluídas somente as arquiteturas das RNAs. Os pesos das conexões foram ajustados por meio de treinamento com o algoritmo de aprendizado de retropropagação, devido à sua maior velocidade de convergência em relação aos algoritmos de PE para esta mesma finalidade. Por outro lado, algoritmos de PE para aprendizado de RNAs demandam uma maior complexidade. Também

existe o fato de que as redes da população sempre teriam a arquitetura de redes diretas completamente interconectadas, tornou-se mais prático o uso do algoritmos de retropropagação. O experimento realizado será exposto com maiores detalhes nas seções seguintes.

5.2.1 A base de dados para o experimento

Como primeiro passo, construiu-se três bases de dados (compreendendo os conjuntos de treinamento e teste) de diferentes tamanhos. Optou-se por primeiramente simular os experimentos na menor base (1^a Base), devido a grande quantidade de dados envolvidos em algoritmos evolucionários. O fato é que os indivíduos da população se tratam de RNAs com a configuração mínima de $(315 \times N \times 2)$, sendo N o número de neurônios intermediários. As 315 entradas se devem à escolha dos dados de face com a menor resolução, maiores informações a respeito disto foram expostas na seção 1.7. A partir do momento que se chegou a resultados satisfatórios para as mesmas, as bases maiores foram utilizadas para a confirmação dos resultados. Assim, fez-se a construção das bases de dados da seguinte maneira:

- 1^a Base:

  Conjunto Treinamento: 120 padrões - (20 ovelhas $\times$ 100 lobos)

  Conjunto Teste: 24 padrões - (12 ovelhas x 12 lobos)

- 2^a Base:

  Conjunto Treinamento: 600 padrões - (100 ovelhas $\times$ 500 lobos)

  Conjunto Teste: 120 padrões - (60 ovelhas x 60 lobos)

- 3^a Base:

  Conjunto Treinamento: 5100 padrões - (100 ovelhas $\times$ 5000 lobos)

  Conjunto Teste: 3060 padrões - (60 ovelhas $\times$ 3000 lobos)

A separação de exemplos por pessoa para os conjuntos de treinamento e teste foi realizada de maneira arbitrária. Expressões faciais e variações rotacionais e de escala na forma 3D da face foram escolhidas da seguinte maneira: a cada grupo de três variações, as duas extremas foram tomadas para o conjunto de treinamento e a intermediária para o conjunto de teste. Por exemplo, levando em consideração as variações de expressões faciais com relação aos olhos (Tab. 1.1), os padrões de face com as expressões NOAB (*normal olhos abertos*) e NOFE (*normal olhos fechados*) foram selecionadas para o conjunto de treinamento, enquanto que o padrão com a expressão NOSA (*normal olhos semi-abertos*) foi selecionado para o conjunto

de teste. Com esta forma de separação deseja-se perceber o quanto uma RNA é capaz de generalizar após uma certa quantidade de treinamento. A separação dos padrões conforme estas observações (de expressão, rotação e escala) para os conjuntos de treinamento e teste é melhor entendida através das Tabelas 4.1 e 4.2 respectivamente.

5.2.2 O experimento

O objetivo inicial do experimento realizado foi a busca da quantidade ideal de neurônios na camada intermediária de uma RNA, pois isto leva a um melhor desempenho no reconhecimento da forma 3D de faces humanas, englobando principalmente uma melhor capacidade de generalização.

Até o momento a quantidade de neurônios ideal para a camada intermediária de uma rede direta é um objeto de investigação. Neste caso somente uma camada intermediária é considerada pois para qualquer número maior do que um de camadas, em uma RNA direta e completamente conectada, existe sempre uma rede equivalente de única camada intermediária.

Sabe-se que ela deve estar relacionada à quantidade de exemplos no conjunto de treinamento e à quantidade de neurônios de entrada e saída da rede. Porém muitas vezes esta descoberta fica a cargo da experiência do pesquisador ou é obtida por métodos de tentativa e erro. Várias heurísticas já foram propostas, tal como apresentado por Kovács [53], porém não existe nada até o momento que seja genérico, ou seja, aplicável a qualquer tipo de problema.

A priori, para o problema de reconhecimento de faces tratado neste trabalho, supôs-se que o número de neurônios na camada intermediária deveria estar no intervalo de $[1 \ldots 120]$, por exemplo, devido à utilização da $1^a Base$. Quanto ao número máximo de neurônios ser 120 (número total de exemplos no conjunto de treinamento) se justifica levando-se em consideração o paradigma neural de aprendizado chamado *Contrapropagação*. Nele diz-se que a camada interna deve ter tantos neurônios quantos forem os exemplos disponíveis para treinamento [6]. É válido observar que para o problema que está sendo tratado, este paradigma não é apropriado pois com este número de neurônios na camada intermediária faria com que a RNA perdesse toda sua capacidade de generalização.

Portanto neste experimento, utilizando a abordagem de programação evolucionária, a população consistiu de 10 indivíduos (RNAs). A princípio a quantidade de neurônios iniciais na camada intermediária foi definida de forma arbitrária. O algoritmo é dado a seguir:

- Passo 1: Determinar a quantidade de neurônios na camada intermediária para cada RNA

- Passo 2: Inicializar as RNAs com pesos aleatórios;
- Passo 3: Treinar as redes até um número determinado de épocas;
- Passo 4: Ordenar as RNAS conforme seus valores de aptidão;
- Passo 5: Descartar a metade das RNAs, as que apresentarem as piores aptidões;
- Passo 6: Mutar a outra metade de RNAs, originando as redes descendentes;
- Passo 7: Tendo chegado à condição de parada terminar, senão voltar ao Passo 3.

O número de neurônios da camada intermediária de cada rede foi, inicialmente, retirado do conjunto $H = [1, 5, 10, 15, \ldots, 120]$, portanto adotados de forma totalmente arbitrária. A partir daí, as redes foram inicializadas com pesos aleatórios no intervalo de $[-1, 1]$. O número de épocas de treinamento determinado para este experimento foi de 100. A ordenação das redes conforme o valor da aptidão foi realizada de forma decrescente. A maneira como foi feito o cálculo da aptidão está descrita na Seção seguinte. A mutação das redes foi realizada mediante interação com o usuário, ela foi realizada de duas maneiras, com e sem herança dos pesos da rede ascendente. Não houve muita diferença com relação ao desempenho devido a estas características. Mas algumas considerações a respeito serão investigadas na Seção 5.2.4.

5.2.3 O cálculo da aptidão

Para cada rede calculou-se o valor de sua aptidão, ou seja, esta medida revela o desempenho do reconhecimento para o conjunto de teste. As melhores redes são selecionadas para gerar descendentes. Este cálculo se baseou nas respostas (saídas) dadas pelas redes mediante à apresentação do conjunto de teste (exemplos de padrões nunca visto por elas). Considerando-se que as redes possuem 2 neurônios na camada de saída: (N_1, N_2). O primeiro neurônio correspondendo a "sim" e o segundo correspondendo a "não". Para as respostas da rede cuja a saída desejada era "ovelha", ou seja $(1, 0)$ a aptidão foi calculada da seguinte maneira:

$$Aptidão\_Ovelha = \frac{N_2}{N_1}$$

E também para as respostas da rede cuja a saída desejada era "lobo", ou seja $(0, 1)$ a aptidão foi calculada da forma:

$$Aptidão\_Lobo = \frac{N_1}{N_2}$$

Sendo a aptidão da rede:

$$Aptidão\_RNA = 1 - \frac{(Aptidão\_Ovelha + Aptidão\_Lobo)}{2}$$

Portanto o cálculo da $Aptidão\_RNA$ reflete o quanto a rede está respondendo corretamente, quanto mais próxima a 1 melhor o seu desempenho.

5.2.4 "Herança de conhecimento" na evolução de RNA com PE

Herança de conhecimento pode ser simulada como a possibilidade de transmissão, através das operações de algoritmos evolucionários tais como mutação e evolução, do conhecimento já adquirido pelo indivíduo gerador. No caso de RNAs seria transmitir aos descendentes os pesos sinápticos, que representam todo o conhecimento da rede adquirido até o momento. No trabalho de Falqueto [27], foi utilizada a técnica de AGs para evolução das redes com e sem herança de conhecimento. Neste trabalho de dissertação implementou-se um algoritmo de PE onde a principal operação envolvida foi a mutação, que foi executada com e sem herança de conhecimento. Isto faz refletir um maior ou menor grau de inter-relacionamento entre os indivíduos que constituem as populações ascendentes e descendentes.

5.2.4.1 Evolução de RNA sem "herança de conhecimento"

Neste caso supõe-se que as redes geram novos indivíduos sem a transmissão de conhecimento que são os pesos sinápticos, já alterados pelo algoritmo de aprendizado. Ou seja, cada nova geração de indivíduos, não usufrui de nenhum conhecimento anterior, pois para cada nova rede, seus pesos não são herdados de seu ascendente, e sim, são aplicados aleatoriamente, portanto, o conhecimento começará do zero, a partir do ponto em que a mesma for treinada e somente a estrutura/topologia da rede, é herdada. Cada população de descendentes parte de um conjunto de pesos sinápticos totalmente aleatórios e o resto do algoritmo de PE atua, a cada geração, na seleção dos indivíduos com números de neurônios na camada intermediária de maior aptidão para evoluí-los, dando-lhes maiores chances de gerarem a próxima população.

Esta opção de implementação apesar de parecer menos eficiente e menos biologicamente plausível, gerou um desempenho pouco inferior à implementação apresentada a seguir que é a melhor opção e foi comprovada no exemplo estudado em Falqueto [27].

5.2.4.2 Simulação de evolução de RNA com "herança de conhecimento"

Nesta implementação a herança de conhecimento (pesos sinápticos) foi transmitida a todos os indivíduos descendentes de forma total ou parcial. Para os indivíduos descendentes cuja

arquitetura era menor do que a de sua rede geradora, sua matriz de pesos foi totalmente preenchida, desconsiderando-se os pesos excedentes da rede ascendente. Para os indivíduos cuja arquitetura era maior, toda a matriz de peso da rede ascendente foi herdada, sendo os pesos restantes preenchidos de maneira aleatória. Tudo isto para que os pesos "evoluídos" tenham influência no processo evolutivo das redes descendentes. O desempenho desta implementação foi sensivelmente melhor ao da implementação anterior, ou seja, a população de redes rapidamente convergiu a aptidões consideráveis.

Segundo [27], supõe-se que esta experimentação é meramente ideal e indica que uma espécie transmitiria aos descendentes todo o conhecimento armazenado pelos antecedentes, que, por sua vez, aumentá-lo-iam com novos conhecimentos durante sua existência.

Portanto, através do algoritmo de PE exposto chegou-se a conclusão de que o problema de reconhecimento de forma 3D abordado é linearmente separável, pois o algoritmo faz chegar rapidamente a uma rede que com apenas um neurônio na camada intermediária é capaz de separar as classes *ovelha* e *lobo*. Com isso partiu-se para utilização de redes mais simples e mais eficientes para resolver este tipo de problema.

5.3 Problema linearmente separável

A busca da melhor solução para um problema através de RNAs consiste na escolha da rede ideal a ser utilizada. O primeiro passo é verificar se o problema é estático ou é dinâmico. O problema será estático se ele for, por exemplo, um problema de associar padrões de entrada a padrões de saída. O problema será dinâmico se para determinar a saída em relação à entrada é preciso ter algo mais, tal como um estado inicial do sistema. Além disto, um problema será dinâmico quando precisar da entrada e também de um estado que evolui com o tempo para se produzir uma saída, ou seja, se o tempo muda, a solução também varia. Todo sistema dinâmico tem que ter o conceito de estado que permite traduzir uma relação em uma função e que evolui com o tempo.

Um problema estático pode ser resolvido por uma rede direta que associa padrões. Uma rede direta é um hetero-associador e também um sistema estático. Ela age como um auto-associador de função que associa entrada com saída. Se a auto-associação é um problema estático ela poderá ser linearmente separável ou não. Se ela for linearmente separável pode-se resolvê-la com um Perceptron de única camada [59]. Caso seja linearmente não separável será preciso no mínimo de uma camada intermediária com no mínimo dois neurônios. Como no início dos experimentos supunha-se que a tarefa de reconhecimento de faces era não linermente separável, partiu-se para utilização de redes diretas com três camadas (de entrada,

intermediária e de saída). A grande questão era determinar a quantidade de neurônios na camada intermediária.

Como dito acima, a busca pela solução baseada em RNAs evolutivas para o problema abordado por este trabalho de dissertação levou a uma descoberta surpreendente, haja visto que as demais pesquisas sobre o problema de reconhecimento de faces nunca chegaram a esta conclusão. A princípio supunha-se que ele era um problema não linearmente separável, porém os experimentos com PE levaram a redes com um único neurônio na camada intermediária a resolver o problema. A partir daí foram utilizadas as redes Perceptron e Adalaine para a constatação de que o problema é de fato linearmente separável. Exemplos de padrões linearmente separáveis e não linearmente separáveis no espaço 2D são mostrados na Fig. 5.1.

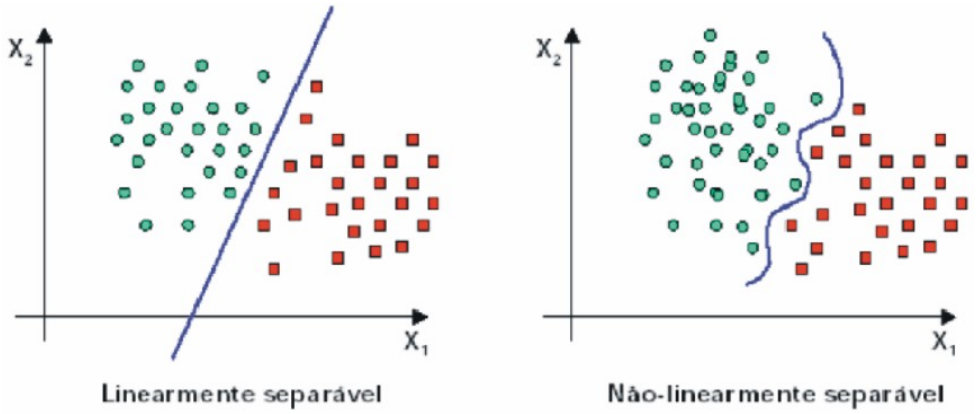

Figura 5.1: Exemplo de classes linearmente e não linearmente separáveis no espaço 2D.

5.3.1 Análise de desempenho

Os desempenhos das redes Adaline e Perceptron para o reconhecimento de faces em forma 3D são mostrados nas Tabelas 5.1 e 5.2. O desempenho equivale à porcentagem de respostas corretas para os padrões apresentados no conjunto de teste. A rede Adaline foi implementada em linguagem C/C++, enquanto que a rede Perceptron consistiu de uma implementação da ferramenta Matlab. Percebe-se que os desempenhos da rede Perceptron para as três bases de dados foram de 100%, isto se deve a algoritmos de otimização implícitos nas implementações Matlab, tal como métodos de inicialização de pesos da rede baseado nos padrões de entrada. Já com a rede Adaline houve um decréscimo no desempenho com relação à maior base, devido à grande quantidade de padrões e à facilidade com que ela caía em mínimos locais, durante a fase de treinamento, porém ainda assim apresentou um ótimo desempenho.

Portanto, constata-se que ambas as redes Perceptron e Adaline executaram, com alta

Rede Adaline	η	ϵ	taxa de acertos
1^a Base	0.001	0.01	100%
2^a Base	0.001	0.4	100%
3^a Base	0.0001	1.4	$98,76\%$

Tabela 5.1: Taxa de acertos, para o conjunto de teste, da rede Adaline onde η é a taxa de aprendizado e ϵ o erro mínimo para o conjunto de treinamento.

Rede Perceptron	η	ϵ	taxa de acertos
1^a Base	0.05	10^{-5}	100%
2^a Base	0.05	10^{-5}	100%
3^a Base	0.05	10^{-5}	100%

Tabela 5.2: Taxa de acertos, para o conjunto de teste, da rede Perceptron onde η é a taxa de aprendizado e ϵ o erro mínimo para o conjunto de treinamento.

precisão, a tarefa de classificação dos padrões apresentados. E para um melhor entendimento destes resultados, nas Seções 3.2.2 e 3.2.3 foram apresentados os aspectos teóricos destas abordagens.

Capítulo 6

Conclusão

6.1 Considerações finais

O reconhecimento de faces humanas é um problema do domínio de reconhecimento de padrões e da biometria. Ele é um objeto de pesquisa muito importante no meio acadêmico nacional e internacional, pois a indústria, o comércio e a "Justiça" têm grande interesse no desenvolvimento desta área. Sistemas de reconhecimento de face vem auxiliar ou até mesmo resolver muitos problemas relacionados à segurança em amplos aspectos.

As tecnologias de reconhecimento de faces já obtiveram algumas contribuições dos estudos de neurocientistas e psicofísicos. É importante que esta sinergia sempre ocorra, pois o reconhecimento automático só teria a ganhar na tentativa de "imitar" o reconhecimento humano de faces, haja visto que uma face é a informação de maior relevância que se pode ter de alguém e é através disto que se realiza a melhor forma de identificação entre as pessoas.

Uma das vantagens do reconhecimento automático de faces, comparado a outros sistemas de reconhecimento, como íris e impressão digital, por exemplo, é a mínima interação que ocorre entre homem e máquina.

A maioria dos sistemas desenvolvidos para o reconhecimento de faces se utiliza de imagens de face em 2D, ou seja, sem a informação da forma facial e sim informações de cor e distâncias entre pontos na face, por exemplo. Com isto, os maiores problemas encontrados estão relacionados à iluminação e à orientação.

O programa de reconhecimento desenvolvido neste trabalho contou com as vantagens do uso dos dados tridimensionais de faces humanas, onde uma face é representada por uma matriz com os valores das suas alturas (distâncias perpendiculares de pontos no relevo da face até um plano imaginário passando relativamente próximo às duas orelhas). Com isso, problemas comuns relacionados à utilização de informação 2D não existem. Estas formas

foram obtidas através do método de processamento de projeções de luz estruturada chamado perfilometria de Fourier.

A busca pela melhor solução para o problema de reconhecimento de faces 3D levou à escolha da abordagem de redes neurais artificiais, as justificativas foram apresentadas. Uma das vantagens de sua utilização é que RNAs são pouco sensíveis a variações espaciais, minimizando assim o problema da má orientação, sempre presente na fase de aquisição. E é melhor que sejam apresentados à rede padrões com certo nível de deslocamento espacial, durante a fase de treinamento, para proporciona-lhe uma boa capacidade de generalização.

Dentro da abordagem de RNAs a questão foi determinar a arquitetura de rede que melhor atendesse aos requisitos de complexidade e taxa de acertos no reconhecimento. Problemas de classificação supervisionada de padrões são aplicações em potencial para RNAs, por isso primeiramente se deu a escolha de uma rede direta com treinamento supervisionado. Definida a arquitetura, o objetivo foi descobrir qual a quantidade de neurônios na camada intermediária ideal para se obter a rede ótima, ou seja, aquela que levaria a uma solução mais econômica computacionalmente e mais eficiente relativo às taxas de classificação correta.

Ainda não existe um método genérico que determine a arquitetura ótima de uma RNA, assim, esta investigação é sempre realizada através de heurísticas ou mesmo por tentativa e erro. Neste trabalho arbitrou-se pela utilização da técnica de programação evolucionária, outro paradigma da inteligência artificial, para realização desta tarefa.

Por meio do algoritmo de programação evolucionária aplicado a uma população de RNAs, chegou-se a redes diretas que, com um único neurônio na camada intermediária, executaram com alta precisão a tarefa de classificação dos padrões apresentados. Foi a partir daí que passou-se a considerar a separabilidade linear dos padrões de faces.

Para a comprovação experimental deste fato, foram utilizadas as redes Perceptron e Adaline. Elas também determinaram as classes de padrões com ótimo desempenho (elevada taxa de classificações corretas). Devido a isto, pode-se então conjecturar que os padrões tratados neste trabalho são de fato linearmente separáveis. Isto não deixa de ser uma surpreendente descoberta visto que esta declaração tão pouco já foi cogitada na literatura de reconhecimento de faces com redes neurais.

É importante considerar que o uso das redes Perceptron e/ou Adaline, para a construção de um sistema de reconhecimento automático de faces implicará em uma maior eficiência pois estas redes são as de menor complexidade possível. E com relação à base de dados utilizada neste trabalho, estas redes superam principalmente as técnicas mais comumente empregadas, tais como, técnicas estatísticas e de casamento de modelos, por exemplo. Haja visto que estas demandam muito mais recursos computacionais.

Portanto, a solução proposta por meio desta dissertação de mestrado é extremamente

eficiente, tanto em nível de complexidade computacional quanto em nível da capacidade de reconhecimento. O único ponto negativo seria com relação ao tempo gasto na etapa de treinamento da rede, porém isto pode ser facilmente superado pela utilização de máquinas com maiores recursos computacionais.

6.2 Trabalhos futuros

Como trabalhos futuros se propõe:

- Realizar um demonstração matemática sobre a separabilidade linear dos padrões de faces tratados.
- Utilizar bases de dados com um maior número de pessoas (e menor miscigenação entre as mesmas) do que a utilizada neste trabalho.
- Implementação do sistema de reconhecimento de faces levando-se em consideração uma maior resolução para as regiões dos olhos nariz e boca. Regiões estas que melhor caracterizam uma face.
- Levar em consideração, além da informação da forma da face também a informação de cor das regiões da face.
- Utilizar para o reconhecimento redes de Kohonen e mapas auto-organizáveis, com treinamento não supervisionado e a fim de fazer uma comparação de dempenho com as redes diretas.
- Implementação de um sistema de reconhecimento baseado na forma 3D para um cenário de identificação de faces (onde o indivíduo não declara sua identidade).

Referências Bibliográficas

[1] AHO, A. V. & ULLMAN, J. D. *The theory of parsing, translation and compiling*, vol. 1: Parsing. Prentice Hall, Englewood Cliffs, N.Y., 1972.

[2] ANDERBERG, M. R. *Cluster Analysis for Applications*. Academic Press, New York, 1973.

[3] ANGELINE, P. J. *Evolutionary algorithms and emergente intelligence*. Tese de Doutorado, Ohio State University, 1993.

[4] AYACHE, N. & LUSTMAN, F. Trinocular stereo vision for robotics. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 1 (1991), 73–85.

[5] BÄCK, T. & SCHWEFEL, H. P. An overview of evolutionary algorithms for parameter optimization. *The Massachusetts Institute of Techonology - Evolutionary Computation 1*, 1 (1993), 1–23.

[6] BARRETO, J. M. *Inteligência Artificial no Limiar do Século XXI*, 3 ed. J. M. Barreto $\rho\rho\rho$ Edições, Florianópolis, 2001.

[7] BELLMAN, R. E.; KALABA, R. & ZADEH, L. *Abstraction and Pattern Classification*. J. Math. Anal. Appl., 1966.

[8] BEYMER, D. J. Face recognition under varying pose. *IEEE Proceedings of Computer Vision and Pattern Recognition* (1994), 556–761.

[9] BICHSEL, M., Ed. *First International Conference on Face and Gesture Recognition*, Zurich, 1995, MultiMedia Laboratory Department of Computer Science University of Zurich.

[10] BISHOP, C. M. *Neural Networks for Pattern Recognition*, 2 ed. Oxford University Press Inc. - Bookcraft Ltd, Walton Street, Oxford OX2 6DP, New York, 1996.

[11] BITTENCOURT, G. *Inteligência Artificial: Ferramentas e Teorias*. Editora da UFSC, 1998.

[12] BOLLE, R. M.; CONNELL, J. H. & RATHA, N. K. Biometric perils and patches. *Elsevier Science Ltd. Pattern Recognition Society.* (October 2001), 1–12.

[13] BRASIL, L. M. Proposta de arquitetura para sistema especialista híbrido e a correspondente metodologia de aquisição do conhecimento. Doutorado, Universidade Federal de Santa Catarina, 1999.

[14] BRUNELLI, R. & POGGIO, T. Face recognition: features versus templates. *IEEE Trans. Pattern Anal. Mach. Intell. 15* (1993), 1042–1052.

[15] BUNKE, H. & SANFELIU, A. *Statistical and Syntatic Models and Pattern Recognition Techniques*, springer verlag ed. in: C. Torras (Ed.), Computer Vision: Theory and Industrial Applications pp 215-266, New York, 1992.

[16] BURMAN, C. *Documento eletrônico HTML em 19/07/2003*. URL=http://www.prosopagnosia.com/.

[17] CARPENTER, G. A. & GROSSBERG, S. Art 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics 26*, 3 (December 1987), 4919–4930.

[18] CHAMP, P. Reverse engineering in industrial applications using laser stripe triangulation. *In Cooloquium: 3D Imaging and Analysis fo Depth/Range Images 4*, IEE Electronics Division (March 1994), 1–4.

[19] CHELLAPPA, R.; WILSON, C. L. & SIROHEY, S. Human and machine recognition of faces: A survey. *Proceedings of the IEEE 83*, 5 (May 1995), 703–740.

[20] CHI, Z. Mlp classifiers: Overtraining and solutions. *In Proceedings of IEEE International Conference on Neural Networks, Perth, Australia 5*, 4 (November-December 1995), 2821.

[21] CHOI, M.-S. & KIM, W.-Y. A novel two stage template matching method for rotation and illumination invariance. *Pattern Recognition Society. Published by Elsevier Science Ltd* (2000), 119–129.

[22] CONNELL, S. D. & JAIN, A. K. Template-based online character recognition. *Pattern Recognition Society. Published by Elsevier Science Ltd* (2001), 1–14.

[23] CROWLEY, J. L., Ed. *Fourth IEEE International Conference on Face and Gesture Recognition*, Grenoble, France, March 2000, IEEE Computer Society Press.

[24] DEMUTH, H. & BEALE, M. *Neural Network Toolbox For Use with MATLAB*. Neural Network Toolbox User's Guide.

[25] ESSA, I., Ed. *Second IEEE International Conference on Face and Gesture Recognition*, Killington, USA, 1995, IEEE Computer Society Press.

[26] ETEMAD, K. & CHELLAPPA, R. Discriminant analysis for recognition of human face images. *J. Optical Soc. Am. A. 14* (1997), 1724–1733.

[27] FALQUETO, J. *Inspiração biológica em IA*. Tese de Doutorado, Universidade Federal de Santa Catarina - Departamento de Computaçao e Estatística, 2002.

[28] FU, K. S. *Syntactic Pattern Recognition and Applications*. Englewood Cliffs - Prentice-Hall, 1982.

[29] FU, K. S. A step towards unification of syntactic and statistical pattern recognition. *IEEE Trans Pattern Analysis and Machine Intelligence 5*, 2 (March 1983), 200–205.

[30] GADER, P.; MOHAMED, M. & CHIANG, J.-H. Comparison of crisp and fuzzy character neural networks in handwritten word recognition. *IEE Transactions on Fuzzy Sistems 3*, 3 (August 1995), 357–363.

[31] GARCIA, R.; DE AZEVEDO, F. M. & BARRETO, J. M. Genetic algorithms in the optimal choice of neural networks for signal processing. In *38 IEEE Midwest Symposium on Circuits and Systems*, Rio de Janeiro, 1995, vol. 8, p. 13–16.

[32] GONZALEZ, R. C. & WOODS, R. E. *Digital Image Processing*. Addison-Wesley Publising Company, Inc., 1992.

[33] GROHMAN, W. M. & DHAWAN, A. P. Fuzzy convex set-based pattern classification for analysis of mammographic microcalcifications. *Elsevier Sciense Ltd on behalf of Pattern Recognition Socity*, 34 (2001), 1469–1482.

[34] GRÖNROOS, M. A. Evolutionary design of neural networks. Dissertação de Mestrado, University of Turku, 1998.

[35] HALLINAN, P. L.; GORDON, G. G.; YUILLE, A. L.; GIBLIN, P. & MUMFORD, D. *Two- and Three- Dimensional Patterns of the Face*. A K Peters, Ltd, 1999.

[36] HAYKIN, S. *Neural Networks, A Comprehensive Foundation*. Upper Saddle River. Prentice Hall, Inc., New Jersey 07458, July 6,1998.

[37] HOPCROFT, J. E. & ULLMAN, J. D. *Introduction to Automata Theory, Languages and Compilation*. Addison-Wesley, Reading, Ma, 1979.

[38] INC., C. URL=http://www.cyberware.com/.

[39] JAIN, A. K. & DUBES, R. C. *Algorithms for Clustering Data*. Englewood Cliffs - Prentice Hall, 1988.

[40] JAIN, A. K.; DUIN, R. P. W. & MAO, J. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 1 (January 2000), 4–37.

[41] JAIN, A. K.; MAO, J. & MOHIUDDIN, K. M. Artificial neural networks: a tutorial. *IEEE Computer* (March 1996), 31–44.

[42] JAIN, A.; HONG, L. & PANKANTI, S. Biometric identification. *Communications of the ACM 43*, 2 (February 2000), 90–98.

[43] JAIN, L. C.; HALICI, U.; HAYASHI, I.; LEE, S. B. & TSUTSUI, S. *Intelligent Biometric Techniques in Fingerprint and Face Recognition*. The CRC Press, 1999.

[44] JAMISON, T. A. & SCHALKOFF, R. J. Image labelling via a neural network approach and a comparison with existing alternatives. *Image and Vision Computing 6*, 4 (November 1988), 3–214.

[45] JANG, J. S. R. & SUN, C. T. Funcional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks 4*, 1 (January 1993), 156–159.

[46] KAMEL, M. S.; SHEN, H. C.; WONG, A. K. C.; HONG, T. M. & CAMPEANU, R. I. Face recognition using perspective invariant features. *Pattern Recognition Letters 15* (1994), 877–883.

[47] KASTURI, R., Ed. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1997, Theme Section of the Journal, IEEE Computer Society Press.

[48] KIM, M.-H.; JANG, D.-S. & YANG, Y.-K. A robust-invariant pattern recognition model using fuzzy art. *Elsevier Sciense Ltd on behalf of Pattern Recognition Socity*, 34 (2001), 1685–1696.

[49] KIRBY, M. & SIROVICH, L. Application of the karhunen-loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 1*, 12 (January 1990), 103–108.

[50] KNUTH, D. E. *Semantics of context-free languages 2(2), 127-145, 1968*, springer verlag ed. in: Mathematical System Theory 5(1) pp. 95-96, 1971.

[51] KOHONEN, T. *Self-Organization an Associative Memory*. Springer-Verlag Berlin, May 1989.

[52] KOHONEN, T. Self-organizing maps. *Springer Series in Information Sciences 30* (1995).

[53] KOVÁCS, Z. L. *Redes Neurais Artificiais: Fundamentos e Aplicações*. Edição acadêmica São Paulo, 1996.

[54] KWAN, H. K. & CAI, Y. A fuzzy neural network and its application to pattern recognition. *IEEE Transactions on Fuzzy Systems 2*, 3 (August 1994), 185–193.

[55] LAWRENCE, S.; GILES, C. L.; TSOI, A. C. & BACK, A. D. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks 8*, 1 (1997), 98–113.

[56] LI, S. Z. Face recognition based on nearest linear combinations. *Scholl of EEE, Nanyang Technological University, Singapore*.

[57] LTD., D. S. URL=http://www.edscanners.com/.

[58] MARR, D. *Vision*. CA: Freeman, San Francisco, 1982.

[59] MINSKHY, M. L. & PAPERT, S. A. *Perceptrons: an introduction to computational geometry*, 3^a ed. The MIT Press, Massachussets, 1988. Impressão modificada do original de 1969.

[60] MITRA, S. & PAL, S. K. Fuzzy multi-layer perceptron, inferencing and rule generation. *IEEE Transactions on Neural Network 6*, 1 (January 1995), 51–63.

[61] MOGHADDAM, B. & PENTLAND, A. Probabilistic visual recognition for object recognition. *IEEE Transactions Pattern Analysis and Machine Intelligence 19*, 7 (July 1997), 696–710.

[62] MOLZ, R. F. *Uma Metodologia para o Desenvolvimento de Aplicações de Visão Computacional utilizando um projeto conjunto de Hardware e Software*. Tese de Doutorado, Universidade Federal do Rio Grande do Sul, Setembro 2001.

[63] NAJMAN, L.; VAILLANT, R. & PERNOT, E. From face sideview to identification. *In Image Processing: Theory and Application. Elsevier Science Publishers* (1993), 299–302.

[64] NEFIAN, A. V. *Statistical Approaches To Face Recognition*. Degree of doctor of philosophy in electrical engineering, Georgia Institute of Technology - Scholl of Electrical Engineering, December 1996.

[65] NOTES IN COMPUTER SCIENCE, L., Ed. *Use of genetic algorithms in neural networks definition*, 1991, no. 540 p. 196-203, Springer Verlag.

[66] PAL, S. K. & MAJUMBER, D. K. D. *Fuzzy Mathematical Approach to Pattern Recognition*. Wiley Eastern Limited, 1987.

[67] PAL, S. K. & MITRA, S. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transaction on Neural Networks 3*, 5 (September 1992), 683–697.

[68] PAL, S. K. & WANG, P. P. *Genetic Algorithms for Pattern Recognition*. The CRC Press, 1996.

[69] PANKANTI, S.; BOLLE, R. M. & JAIN, A. Biometrics: The future of identification. *Computer* (February 2000), 46–49.

[70] PENEV, P. & ATICK, J. Local feature analysis: A general statistical theory for object representation. *Network Computation in Neural Systems, 7:477, 500, 1996. 7* (1996), 477–500.

[71] PENTLAND., A. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 1 (January 2000), 107–119.

[72] PERLOVSKY, L. I. Conundrum fo combinatorial complexity. *IEEE Trans. Pattern Analysis and Machine Intelligence 20*, 6 (1998), 666–670.

[73] PHILLIPS, P.; WECHSLER, H.; HUANG, J. & RAUSS, P. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing 16*, 5 (1998), 295–306.

[74] POLLARD, S. B.; PARRILL, J. & MAYHEW, J. E. W. Recovering partial 3d wire frames descriptions from stereo data. *Image and Vision Computing 1*, 9 (1991), 58–65.

[75] ROBERTSON, G. & CRAW, I. Testing face recognition systems. *In Proceedings of the British Machine Vision Conference* (1993).

[76] ROSIN, P. L. & FIERENS, F. Improving neural network generalisation. *In Proceedings of IGARSS'95, Firenze, Italy* (1995). July.

[77] RUSPINI, E. H. *Numerical Methods for Fuzzy Clustering*, vol. 2. Inf. Sci., 1970.

[78] SCHALKOFF, R. *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc., Clemson University, 1992.

[79] SPIES, H. Face recognition - a novel technique. Mather thesis summary.

[80] SUNG, K. K. & POGGIO, T. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 1 (1998), 39–51.

[81] TIBBALDS, A. D. *Three Dimensional Human Face Acquisition for Recognition*. Doctor of philosophy, Signal Processing and Communications Laboratory. Department of Engineering. University of Cambridge, March 1998.

[82] TOU, J. T. & GONZALEZ, R. C. *Pattern Recognition Principles*. Addison-Wesley Publishing Co., London, 1974.

[83] TSAI, W. H. & FU, K. S. Attributed grammar - a tool for combining syntactic and statistical approaches to pattern recognition. *IEEE Transactions on Systems, Man and Cybernectics SMC-10*, 12 (1980), 873–885.

[84] TURK, M. A. & PENTLAND, A. P. Face recognition using eigenfaces. *In Proc. of the IEEE con Computer Society Conferece. IEEE Computer Society Press* (1991).

[85] URQUHART, C. W. *The active stereo probe: The design and implementation of an active videometrics system*. Phd thesis, The Department of Computing Science. The University of Glasgow, June 1997.

[86] WATANABE, S. *Pattern Recognition: Human and Mechanical*. Wiley, New York, 1985.

[87] WISKOTT, L.; FELLOUS, J.-M.; KRÜGER, N. & VON DER MALSBURG, C. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 7 (July 1997), 775–779.

[88] YACHIDA, M., Ed. *Third IEEE International Conference on Face and Gesture Recognition*, Nara, Japan, April 1998, IEEE Computer Society Press.

[89] YANG, M. H.; AHUJA, N. & DRIEGMAN, D. A survey on face detection methods.

[90] YAO, X. & LIU, Y. Evolving artificial neural networks for medical applications. Relatório técnico., University of New South Wales. Department of computer Science, Australian Defence Force Academy, Canberra, ACT, Australia 2600, 1995.

[91] YAO, X. & LIU, Y. Fast evolution strategies. In *Evolutionary Programming VI*, Berlin, 1997, P. J. Angeline, R. G. Reynolds, J. R. McDonnell, & R. Eberhart, Eds., Springer, p. 151–161.

[92] YAO, X. & LIU, Y. Making use of population information in evolutionary artificial neural networks. Relatório técnico., University of New South Wales. Department of computer Science, Australian Defence Force Academy, Canberra, ACT, Australia 2600, 1998.

[93] YAO, X. & LIU, Y. A new evolutionary system for evolving artificial neural networks. Relatório técnico., School of Computer Science. The University of Birmingham. Edgbaston. United Kingdom, Australian Defence Force Academy, Canberra, ACT, Australia 2600, 1999.

[94] YAO, X. A review of evolutionary artificial neural networks. Relatório técnico, Commonwealth Scientific and Industrial Research Organization, Division of Building, Constrution and Enginnering, PO Box 56, Highett, Victoria 3190, Australia, 1992.

[95] YAO, X. Evolving artificial neural networks. Relatório técnico., School of Computer Science. The University of Birmingham. Edgbaston. United Kingdom, Australian Defence Force Academy, Canberra, ACT, Australia 2600, 1999.

[96] ZADEH, L. A. *Fuzzy sets*, vol. 8. Inform. and Control, 1965.

[97] ZADEH, L. A. *Fuzzy algorithms*, vol. 12. Inform. and Control, 1966.

[98] ZADEH, L. A. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst., Man and Cyberns. SMC-3* (1973), 28–44.

[99] ZADEH, L. A. *Fuzzy Sets and Their Application to Pattern Classification and Cluster Analysis*. Memo UCB/ERL, M-607, Univ. Calf., Berkeley, 1976.

[100] ZIMMERMANN, A. C.; GONÇALVES JR., A. A. & BARRETO, J. M. A 3d object extraction and recognition method. *Sixth International Conference on Control, Automation, Robotics and Vision - ICARV* (December 2000).

[101] ZIMMERMANN, A. C.; GONÇALVES JR., A. A. & BARRETO, J. M. General non-invasive shape reconstruction and recognition method applied to 3d external biologic morphologies. *Proceedings of 6th IEEE International Symposium on Bio-Informatics & Biomedical Engineering - BIBE2000* (November 2000).

[102] ZIMMERMANN, A. C. *Reconhecimento de Faces Humanas Através de Técnicas de Inteligência Artificial Aplicadas a Formas 3D*. Tese de Doutorado, Universidade Federal de Santa Catarina, Março 2003.