Universidade Federal do Ceará (UFC)
Centro de Tecnologia
Departamento de Engenharia de Teleinformática (DETI)
Programa de Pós-Graduação em Engenharia de Teleinformática (PPGETI)

Autor: ISAQUE QUEIROZ MONTEIRO

MÉTODOS DE APRENDIZADO DE MÁQUINA PARA RECONHECIMENTO DE FACES: UMA COMPARAÇÃO DE DESEMPENHO

FORTALEZA
2009

MONTEIRO

DISSERTAÇÃO DE MESTRADO

2009

Autor: **Isaque Queiroz Monteiro**

Orientador: **Prof. Dr. Guilherme de Alencar Barreto**

MÉTODOS DE APRENDIZADO DE MÁQUINA PARA RECONHECIMENTO DE FACES: UMA COMPARAÇÃO DE DESEMPENHO

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como parte dos requisitos exigidos para obtenção do grau de **Mestre em Engenharia de Teleinformática**.

Área de Concentração: **Sinais e Sistemas**

FORTALEZA

2009

Ficha catalográfica elaborada pela bibliotecária Umbelina Caldas Neta - CRB558-CE

M776m Monteiro, Isaque Queiroz

Métodos de aprendizado de máquina para reconhecimento de faces: uma comparação de desempenho / Isaque Queiroz Monteiro, 2009.

152 f.; il.; enc.

Orientador: Prof. Dr. Guilherme de Alencar Barreto

Área de concentração: Sinais e Sistemas

Dissertação (mestrado) Universidade Federal do Ceará, Departamento de Engenharia de Teleinformática, Fortaleza, 2009.

1.Teleinformática 2. Sistemas de reconhecimento de padrões. 3.Biometria 4. Redes Neurais. I. Barreto, Guilherme de Alencar (Orient.) II. Universidade Federal do Ceará - Programa de Pós - Graduação em Engenharia de Teleinformática. III. Título.

C.D.D. 621.38

Esta dissertação sob o título *Métodos de Aprendizado de Máquina para Reconhecimento de Faces: Uma Comparação de Desempenho*, foi submetida como parte dos requisitos necessários à obtenção do Grau de Mestre em Engenharia de Teleinformática, outorgado pela Universidade Federal do Ceará, e encontra-se à disposição dos interessados na biblioteca da referida universidade.

A citação de qualquer trecho desta dissertação é permitida, desde que feita em conformidade com as normas da ética científica.

Isaque Queiroz Monteiro

DISSERTAÇÃO APROVADA EM 05/06/2009.

Prof. Dr. Guilherme de Alencar Barreto
Universidade Federal do Ceará - Orientador

Prof. Dr. Paulo César Cortez
Universidade Federal do Ceará

Prof. Dr. Cláudio M. de Sá Medeiros
Instituto Federal de Educação,
Ciência e Tecnologia do Ceará

Prof. Dr. José Manoel de Seixas
Universidade Federal do Rio de Janeiro

Dedico este trabalho aos meus pais
Morais (in memorium) e Antonia,
e à minha esposa, Manuelina,
pelo amor e apoio de sempre.

Agradecimentos

Ao meu pai Morais (in memorium) pelos ensinamentos e exemplo de vida.

À minha Mãe pelo zelo e carinho, sempre que os dias eram curtos demais.

À minha Esposa pelo amor, apoio e compreensão durante esta tarefa.

Ao meu orientador e amigo, Guilherme de Alencar Barreto, pela paciência e solicitude dedicadas neste trabalho.

À minha família.

Aos professores e funcionários do Departamento de Engenharia de Teleinformática que de forma direta ou indireta contribuíram para o desenvolvimento deste trabalho.

Aos demais colegas de graduação e pós-graduação, pelas críticas e sugestões, e, em especial, aos amigos de laboratório pela ajuda nas dúvidas, companheirismo nas dificuldades.

À FUNCAP (Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico) por ter custeado meus estudos de mestrado.

"Quando alguém tem força para vencer a si mesmo,
nasceu para grandes empreendimentos."

Jean Baptiste Henri Lacordaire

Resumo

O objetivo principal desta dissertação é realizar uma ampla análise comparativa de desempenho entre dezoito métodos de aprendizagem de máquina em tarefas de reconhecimento de faces sob diferentes poses, condições de iluminação, configurações e expressões faciais. Os desempenhos destes algoritmos foram avaliados para cinco bancos de faces usados para fins de *benchmarking*: YALE-1, CMU, ORL, STIRLING e BARTLETT. Análise de Componentes Principais (PCA) é usada para reduzir a dimensionalidade dos padrões de entrada gerados a partir da vetorização das imagens digitais das faces.

Após uma primeira avaliação do desempenho global de cada um dos classificadores, os de melhor desempenho são adicionalmente avaliados por meio de testes de hipóteses e matrizes de confusão com o intuito de determinar o grau de similaridade estatística entre suas taxas de acerto global e por classes, respectivamente. Todos os classificadores foram ainda avaliados quanto ao seus desempenhos para imagens degradadas por ruído sal-e-pimenta. O tempo de execução de cada arquitetura durante as etapas de treinamento e teste foi também determinado para fins de comparação entre as diferentes arquiteturas.

Os resultados experimentais indicam, considerando apenas as taxas de acerto, que os discriminates não-lineares baseados na rede MLP e em máquinas de vetor-suporte (SVM) apresentam desempenhos melhores que os dos demais classificadores, para todos os bancos de faces utilizados. Entretanto, quando o tempo de execução também é usado como figura de mérito, os desempenhos de discriminantes lineares, tais como as redes Perceptron Logístico e Madaline, passam a rivalizar com os obtidos por discriminantes não-lineares.

Palavras–Chave: Redes Neurais Artificiais, Perceptron Simples, Perceptron Multicamada, Rede Auto-Organizável de Kohonen, Máquinas de Vetores Suporte, Reconhecimento de Faces.

Abstract

The main goal of this dissertation is to perform a broad performance comparison analysis among eighteen different machine learning methods for human face recognition under different poses, illumination conditions, configurations and facial expressions. All the algorithms were evaluated on three benchmarking face databases (BARTLETT, CMU, ORL, STIRLING and YALE-1). Principal Component Analysis (PCA) is used to reduce the data dimensionality generated from the vectorization of the face images.

After preliminary evaluations of the global performance of each classifier, the best ones are then evaluated by hypotheses testing and confusion matrix in order to highlight statistical similarities among their classification performances. All classifiers were further evaluated by thpresenting to them images corrupted by salt-and-pepper noise. The execution time of each classifier during training/testing runs were also determined.

When only the recognition rates are analyzed, the simulation results indicate that nonlinear discriminants based on the MLP network and support vector machines (SVM) present better performances than the other classifiers, for all image datasets. However, when the execution time is also used as a figure of merit, the performances of linear discriminants, such as the logistic perceptron and Madaline networks, become competitive with respect to those obtained by nonlinear discriminants.

Keywords: Artificial Neural Networks, Simple Perceptron, Multilayer Perceptron, Self-Organizing Map, Support Vector Machines, Face Recognition.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

GRNN	*Generalized Regression Neural Network*
LMS	*Least Mean Squared*
MADALINE	*Multiple Adaptive Linear Element*
MLG	*Maximum Likelihood Gaussian Classifier*
MLP	*Multi-layer Perceptron*
MLP-1C	*Classificador MLP com uma camada escondida*
MLP-2C	*Classificador MLP com duas camada escondida*
OLAM	*Optimal Linear Associative Memory*
PCA	*Principal Component Analysis*
PL	*Perceptron Logístico*
PL-MEKA	*Perceptron Logístico treinado pelo algoritmo MEKA*
PS	*Perceptron Simples*
RBF	*Radial Basis Function*
SOM	*Self-Organizing Map*
SOM-C1	*Classificador SOM de Rotulação por Voto Majoritário*
SOM-C2	*Classificador SOM de Rotulação por Mapas Individuais*
SOM-C3	*Classificador SOM de Rotulação Auto-Supervisionada*
SOM-C4	*Classificador SOM de Rotulação pelo Centróide mais Próximo*
SVM	*Support Vector Machines*
SVM-LOO	*Classificador com kernel linear e aprendizado um contra um*
SVM-LOA	*Classificador com kernel linear e aprendizado um contra todos*
SVM-GOO	*Classificador com kernel gaussiano e aprendizado um contra um*
SVM-GOA	*Classificador com kernel gaussiano e aprendizado um contra todos*
VC	*Vapnik-Chervonenkis dimension*

Lista de Símbolos

t	Tempo discreto
$\mathcal{C}_b$	Conjunto de Faces b
C	Parâmetro de regularização das máquinas de vetores suporte
ξ	Variável de flexibilização adotada em SVM
$k(\cdot)$	Função kernel
$\mathbf{x}(t)$	Vetor de entrada da rede no instante t
p	Dimensão dos vetores de entrada
$p(\mathbf{x})$	Densidade de probabilidade do vetor de entrada $\mathbf{x}(t)$
$\mathcal{X}$	Espaço contínuo dos dados de entrada
Φ	Transformação não-linear
$\mathcal{A}, \mathcal{Y}$	Topologia do espaço de saída
q	Número de neurônios existentes: ou na camada escondida do classificador MLP-1C ou na rede SOM
$\mathbf{w}(t), \mathbf{m}(t)$	Vetores de pesos
$d(t)$	Saída desejada
$h(i^*, i; t)$	Função de vizinhança da rede SOM
$L(\mathbf{w}, b, \alpha)$	Função Lagrangiano
$\eta(t)$	Taxa de aprendizagem usada pela rede MLP
α, ν	Multiplicadores de Lagrange
ρ	Margem de separação entre hiperplanos e vetores suporte
ρ_r	Probabilidade de um pixel ser corrompido por ruído
$\alpha(t)$	Taxa de aprendizagem usada pela rede SOM
α_0, α_T	Valores inicial e final de $\alpha(t)$, respectivamente
σ	Raio das funções de base das redes RBF
$\sigma(t)$	Abertura da vizinhança topológica da função gaussiana
σ_0, σ_T	Valores inicial e final de $\sigma(t)$, respectivamente
$\mathbf{c}_j$	Centróide da classe j usado pelo Classificador SOM-C1
c	Dimensão de saída
$\mathbf{c}^*$	Centro das funções de base para redes RBF
$\mathbf{d}$	Saída desejada
$\mathbf{y}$	Saída obtida
$\mathbf{e}$	Erro de saída

$\delta(t)$	Gradiente local da rede MLP
$\mathbf{P}(t)$	Estimação da inversa da matriz de covariância de $\mathbf{q}(t)$
$\mathbf{k}(t)$	Ganho de Kalman
λ	Fator de esquecimento

Sumário

1 Introdução

"A vantagem dos míopes é enxergar onde as grandes vistas não pegam."

Joaquim Maria Machado de Assis

Neste Capítulo, a tarefa de reconhecimento de faces humanas é apresentada juntamente com a motivação, os objetivos e as contribuições deste trabalho.

1.1 Introdução

Quando se vê uma face, tipos celulares especializados se reportam a uma certa parte do cérebro: o centro distribuído de processamento - o córtex visual - o qual se encarrega de reconhecê-la. Esta característica humana tem motivado cientistas e engenheiros a criarem dispositivos capazes de mimetizar este comportamento. E esta busca tem se intensificado nas últimas décadas, amparada pelo avanço tecnológico para processamento de sinais e estimulada por setores comerciais que vislumbram inúmeras aplicações na área de biometria (DORIZZI, 2006).

Dentre as aplicações para o reconhecimento de faces humanas destacam-se as seguintes: sistemas de segurança para identificação de indivíduos em multidões e sistemas de controle de acesso, sejam estes no âmbito residencial e empresarial ou até mesmo no acesso a contas bancárias. Em Chellappa et al. (1995) e Zhao et al. (2003) tem-se uma descrição mais detalhada sobre estas aplicações.

Mesmo com o avanço conseguido dos sistemas de processamento de sinais, a complexidade inerente ao problema de reconhecimento impõe desafios ainda em aberto para os métodos de aprendizado de máquina. Vários estudos, aplicando classificadores estatísticos e neurais, têm procurado solucionar tais questões, contudo nota-se uma ausência de estudos comparativos mais abrangentes, tanto em termos de um maior número de arquiteturas avaliadas quanto pelo uso de diferentes conjuntos de faces.

Em geral, os trabalhos tratam do desempenho de um único classificador (ou alguns poucos!) comparado com algum classificador de referência, normalmente um de desempenho inferior. Os artigos seguintes exemplificam esta situação:

- Perceptron Multicamadas: Gaspar (2006) e Xu & Ahmadi (2007).
- Funções de Base Radial: Feitosa et al. (2000), Howell (2001) e Toh & Yau (2005).
- Máquinas de Vetor-Suporte: Guo et al. (2001) e Soto (2005).
- Mapas Auto-Organizáveis de Kohonen: Tan (1995) e Santos (2005).
- Métodos Estatísticos: Krol & Florek (2005).

Tendo em mente a realização de um estudo mais abrangente sobre o desempenho de classificadores de padrões em reconehcimento de faces, são apresentados a seguir os objetivos desta dissertação.

1.2 Objetivo Geral

Este trabalho busca realizar uma comparação entre 18 classificadores quando aplicados à tarefa de reconhecer faces humanas sob diferentes configurações como angulação facial, diferentes expressões, níveis de iluminação, uso de acessórios (e.g. óculos) e até mesmo quando as imagens encontram-se degradadas por ruído. Esta avaliação é conduzida sobre 05 conjuntos diferentes de faces, cada um destes caracterizando-se por algum dos aspectos mencionados anteriormente.

1.2.1 Objetivos Específicos

O escopo desta dissertação envolve um subconjunto de problemas que serão abordados individualmente para construir o conjunto final de resultados. Estes problemas estão listados a seguir:

1. comparar classificadores oriundos de paradigmas de aprendizado distintos (i.e. supervisionados ou não), sejam eles neurais ou estatísticos;
2. comparar classificadores que geram superfícies de decisão diferentes, ou seja, classificadores lineares e não-lineares;
3. avaliar o desempenho do Algoritmo do Filtro de Kalman Estendido para treinar a rede Perceptron Logístico;

4. aplicar teste de hipóteses para estabelecer a significância estatística entre os classificadores sob avaliação;

5. estudar o comportamento dos melhores classificadores através de suas matrizes de confusão e, com isto, estabelecer o grau de similaridade entre estes e as classes que apresentem maior dificuldade de classificação;

6. avaliar o custo computacional destes classificadores.

Na próxima seção são listados os trabalhos desenvolvidos antes e durante a preparação desta dissertação.

1.3 Produção Científica

Durante o período de graduação, enquanto bolsista de iniciação científica e tecnológica, e durante o desenvolvimento desta dissertação, os seguintes artigos foram produzidos e apresentados nos referidos eventos:

- **Isaque Q. Monteiro** & Guilherme A. Barreto, "*A Performance Comparison of Neural Network Based Classifiers for Face Recognition*", II Workshop on Computational Intelligence (WCI'2008), realizado de 26 a 30 de de Outubro de 2008, Salvador−Bahia.

- **Isaque Q. Monteiro**, Samy A. Queiroz, Alex T. S. Carneiro, Luiz G. M. Souza & Guilherme A. Barreto, "*Face Recognition Independent of Facial Expression Through SOM-based Classifiers*", VI International Telecommunications Symposium (ITS'2006), realizado de 03 a 06 de setembro de 2006, Fortaleza−Ceará;

- Samy A. Queiroz, **Isaque Q. Monteiro**, Luiz G. M. Souza & Guilherme A. Barreto, "*Classificação Robusta de Faces Usando a Rede de Kohonen*", XVI Congresso Brasileiro de Automática (CBA'2006), realizado de 03 a 06 de agosto de 2006, Salvador−Bahia.

- **Isaque Q. Monteiro**, Patrícia V. Nascimento & Guilherme A. Barreto, "*Dynamic LVQ Models for Classification of Spatiotemporal Patterns*",VII Congresso Brasileiro de Redes Neurais (CBRN'2005), realizado de 16 a 19 de outubro de 2005, Salvador−Bahia.

1.4 Organização da Dissertação

A seguir, um breve resumo de cada um dos Capítulos e apêndices que compõem o restante deste documento é realizado.

- No Capítulo 2 apresentam-se os conjuntos de faces utilizados neste trabalho juntamente com a metodologia de classificação empregada.
- No Capítulo 3 descrevem-se as principais características do mapa auto-organizável de Kohonen e em seguida os classificadores dele decorrentes. Ao fim deste capítulo, os resultados de classificação destas redes são apresentados e comentados.
- No Capítulo 4 descrevem-se os classificadores baseados na rede Perceptron. Suas variantes lineares e não-lineares. Além disso descreve-se, grosso modo, o Algoritmo de Kalman Estendido Múltiplo o qual pode ser uma alternativa ao tradicional *Backpropagation* no treinamento destas redes e que é usado para o treinar a rede Perceptron Logístico. Do mesmo modo, ao fim desse capítulo apresentam-se os resultados de classificação obtidos juntamente com uma breve análise dos mesmos.
- O Capítulo 5 aborda os classificadores de kernel. Descrevem-se de modo sucinto as características das redes RBF, GRNN e SVM e as condições de implementação adotadas para elas. Encerra-se este Capítulo com os resultados de classificação obtidos e a análise dos mesmos.
- No Capítulo 6 realizam-se as comparações finais. Em princípio, todos os classificadores são avaliados para a tarefa de classificar faces ruidosas. Em seguida, testes de hipóteses são empregados para comparar os melhores classificadores, tomados dois à dois. Nos casos em que estes são considerados iguais, avaliam-se suas matrizes de confusão no intuito de discriminar as dificuldades encontradas para cada classe. Ao término são apresentados dados preliminares do custo computacional destas redes.
- No Capítulo 7 são feitas as conclusões e recomendações finais desta dissertação.
- No Apêndice A apresentam-se de modo sucinto o classificador Mémoria Associativa Linear Ótima (OLAM) e o classificador Gaussiano de Máxima Verossimilhança (MLG) juntamente com o desempenho destes na tarefa sob análise.
- O Apêndice B contém apenas uma tabela reduzida da distribuição t de Student para auxílio nas comparações feitas via teste de hipóteses.

2 *Descrição dos Bancos de Dados e Metodologia Utilizados*

"Quando escrevo uma equação na lousa vejo
os números e as letras de cores diferente.
E eu me pergunto: que diabos meus alunos vêem?"

Richard Feynman

Este Capítulo descreve os bancos de imagens utilizados neste trabalho, assim como a metodologia preliminar à tarefa de classificação em si e à análise posterior de desempenho das arquiteturas que serão descritas nos próximos Capítulos.

2.1 Bancos de Dados de Faces Utilizados

Neste trabalho, cinco bancos de faces humanas foram utilizados. Eles são aqui denominados de BARTLETT, CMU, ORL, STIRLING e YALE e são caracterizados por combinar imagens de indivíduos obtidas em situações não-ideais. Por situações não-ideais entendem-se aquelas em que as imagens digitais são obtidas com variações de expressões, do ângulo do rosto em relação à câmera e/ou de iluminação, dentre outras que dificultam consideravelmente o reconhecimento automático do indivíduo.

Embora muitas combinações de uso destes bancos de dados seja possível, optou-se por utilizá-los em separado e analisar o desempenho das arquiteturas de classificação para um dos conjuntos por vez. Em outras palavras, não consta neste trabalho experimentos onde se tem, na fase de treinamento ou de teste, imagens provenientes de bancos distintos (e.g. CMU + ORL) ou quaisquer combinações com os demais.

Como ponto de partida para as descrições que se seguem, as seguintes definições serão úteis à melhor organização deste trabalho:

- **Conjunto de Faces** - Coleção $\mathcal{C}_b$ contendo todas as imagens de faces de um banco de dados que está sendo utilizado. Deste modo, trabalha-se com cinco conjuntos, a saber, $b = \{BARTLETT, CMU, ORL, STIRLING, YALE\}$.

- **Subconjunto de Faces** - Conjunto de todas as imagens de faces que pertencem a um indivíduo em particular. Assim, subconjunto e classe adquirem sentido equivalente para o objetivo de classificação presente neste trabalho.

2.1.1 Banco de Faces BARTLETT

Este primeiro conjunto de faces humanas foi fornecido pela pesquisadora Marian Stewart Bartlett (BARTLETT, 2001) com permissão de David Beymer (D., 1994) e é composto por 100 imagens formadas por 60×60 pixels, em escala de cinza e no formato JPEG, divididas, de modo igual, para 20 indivíduos. A Figura 2.1 mostra os diferentes indivíduos que compõem este banco de dados.

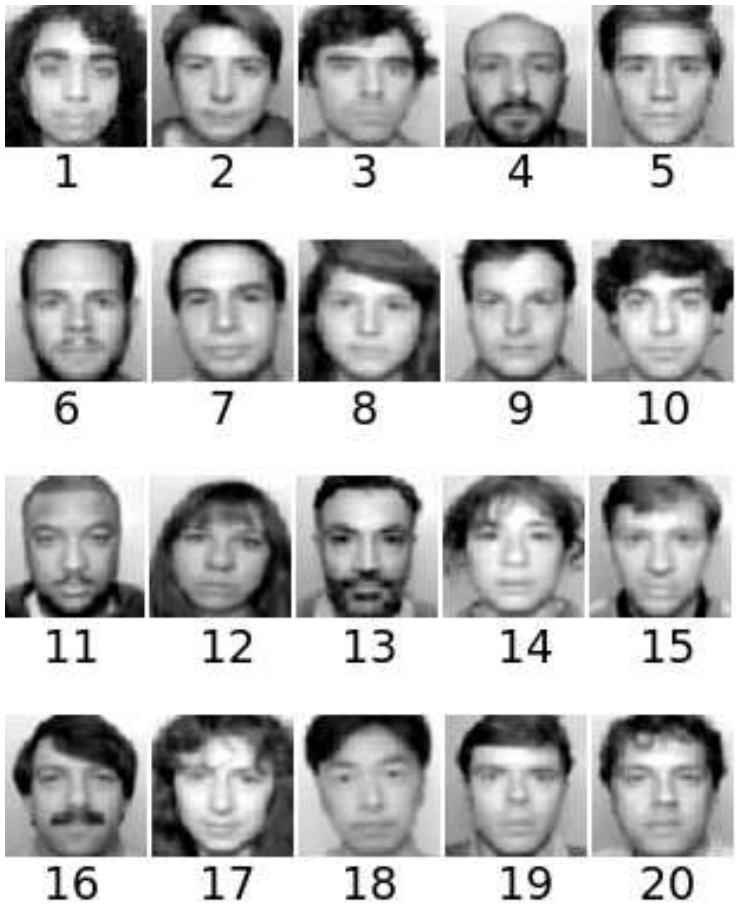


Figura 2.1 : indivíduos que compõem o banco de dados Bartlett.

As cinco imagens de cada subconjunto variam em pose[1] com relação ao eixo vertical como mostra a Figura 2.2. Para cada indivíduo existe uma vista frontal de sua face, além de duas variações simétricas de pose, tanto para a esquerda quanto para a direita da posição frontal. O incremento angular para cada pose é de 15°, em relação à posição frontal.

-30° -15° 0° 15° 30°

Figura 2.2 : amostra das variações das posições faciais presentes no banco de imagens Bartlett.

2.1.2 Banco de Faces CMU

Este conjunto de imagens, disponibilizado por Tom Mitchell da Carnegie Mellon University (MITCHELL, 1997), caracteriza-se por ser equilibrado, ou seja, é composto de 20 subconjuntos contendo o mesmo número de imagens (ver Figura 2.3). Todas as 640 imagens estão em escala de cinza no formato PGM.

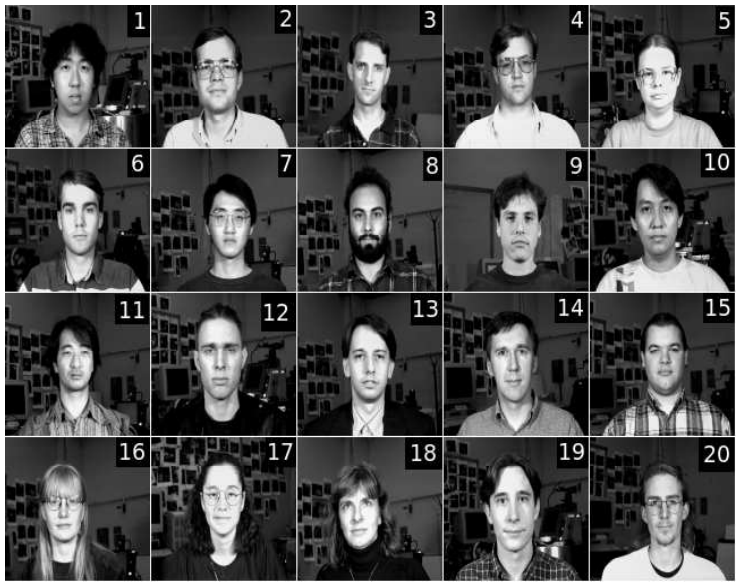


Figura 2.3 : indivíduos que compõem o banco de imagens CMU.

[1]Neste caso, corresponde ao ângulo da face em relação à câmera.

Cada subconjunto é resultado da combinação de poses (rosto em vista frontal, lateral direita, lateral esquerda, e inclinado para cima), expressões (neutralidade, alegria, tristeza e raiva) e acessórios (com e sem óculos de sol). A Figura 2.4 exemplifica a variedade de cada subconjunto. O tamanho das imagens utilizadas é de 128×120 pixels embora seja disponível também outros dois conjuntos com tamanhos inferiores (64×60 e 32×30 pixels, respectivamente).

Figura 2.4 : variações faciais existentes no banco de imagens CMU para um dado indivíduo.

2.1.3 Banco de Faces ORL

O terceiro conjunto de faces, denominado de ORL (SAMARIA; HARTER, 1994), é composto de uma coleção de imagens que foram tiradas entre abril de 1992 e abril de 1994 no laboratório de pesquisas da Olivetti em Cambridge, Reino Unido. Este conjunto divide-se em 40 subconjuntos contendo 10 imagens cada. Cada imagem é formada por 92×112 pixels. A Figura 2.5 mostra os indivíduos que compõem este banco de imagens.

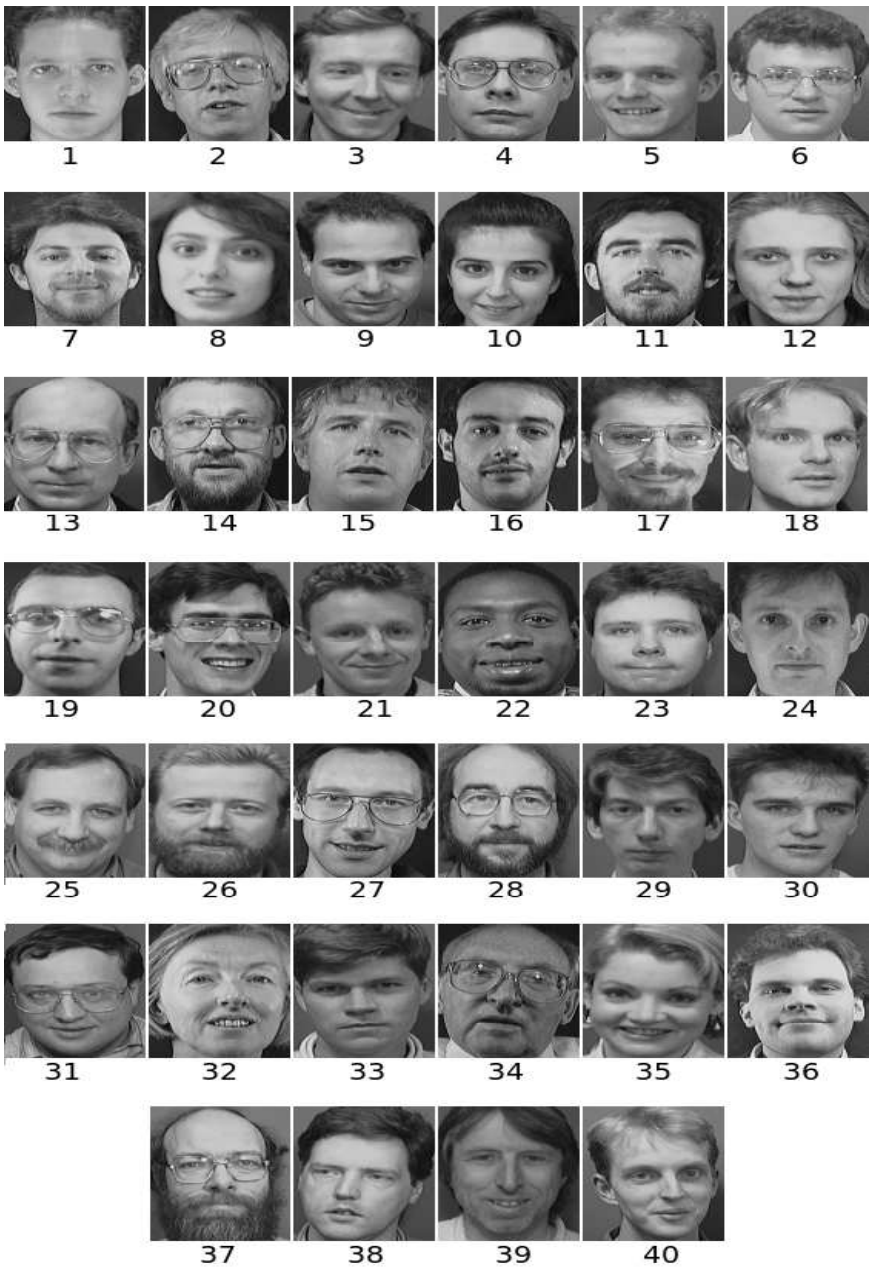


Figura 2.5 : indivíduos que compõem o banco de imagens ORL.

Esses exemplares foram registrados em dias diferentes, com condições distintas de iluminação, de expressões faciais (e.g. de olhos abertos ou fechados, sorrindo, etc.) e uso de acessórios, tais como óculos. Todas as imagens foram registradas contra um fundo escuro e houve uma variação na pose das faces (poses frontais e inclinadas) com tolerância para algum movimento lateral das mesmas. A Figura 2.6 exemplifica essas diferentes condições para um determinado indivíduo.

Figura 2.6 : variações presentes no banco de imagens ORL para um dado indivíduo.

2.1.4 Banco de Faces STIRLING

Este conjunto de faces STIRLING (STIRLING, 2008), originalmente produzido para auxiliar pesquisas relacionadas à psicologia, mais especificamente, percepção visual, memória e processamento, é composto por 36 subconjuntos contendo 09 imagens para cada indivíduo (Figura 2.7). O tamanho de todas as imagens é 270×350 pixels no formato GIF.

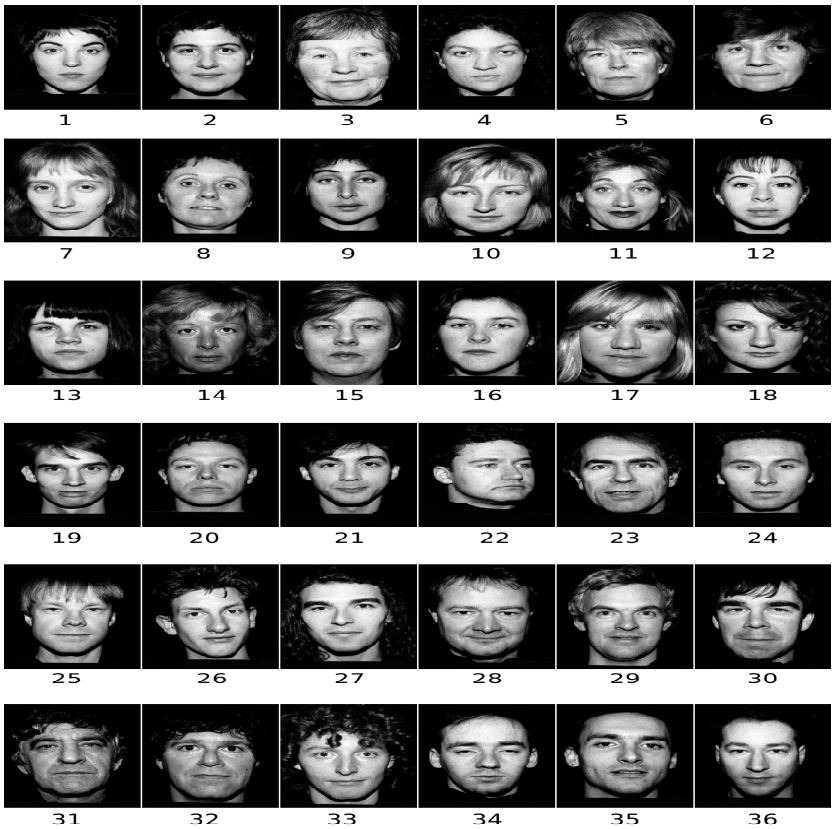


Figura 2.7 : indivíduos que compõem o banco de imagens STIRLING.

Cada subconjunto das imagens do banco STIRLING resulta da combinação de três ângulos de enquadramento da face (vistas frontal, oblíqua e lateral a 90°) e três condições de expressão (neutralidade, sorrindo e falando). A Figura 2.8 exemplifica o conteúdo de um subconjunto (i.e. indivíduo) específico.

Figura 2.8 : variações presentes no banco de imagens STIRLING para um dado indivíduo.

2.1.5 Banco de Faces YALE-1

O último banco de imagens, denominado de YALE-1 (BELHUMEUR et al., 1997), é composto de 165 imagens em escala de cinza, no formato GIF. Cada indivíduo (ver Figura 2.9), de um total de 15, possui 11 imagens em diferentes situações.

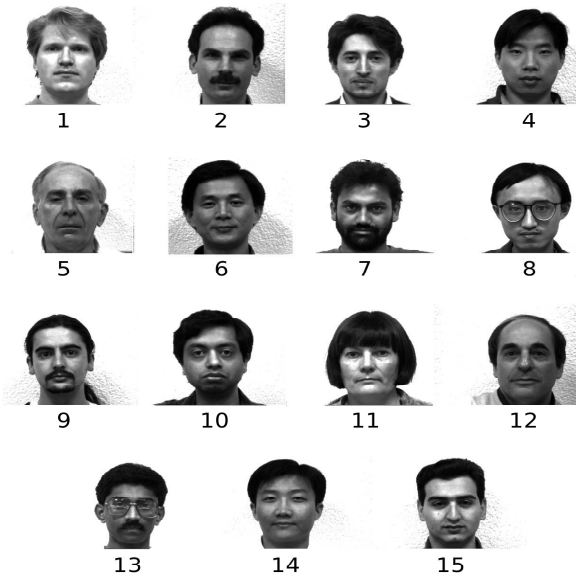


Figura 2.9 : indivíduos que compõem o banco de imagens YALE-1.

As onze situações encontradas para cada indivíduo do conjunto YALE-1 compõem-se de distintas expressões faciais: tristeza, sonolência, surpresa, piscar um olho e alegria; bem como configurações diferentes: iluminação frontal, expressão normal sem óculos, expressão normal com óculos, iluminação lateral à esquerda e iluminação lateral à direita. A Figura 2.10 exemplifica as variações encontradas. O tamanho de todas as imagens é 243×320 pixels.

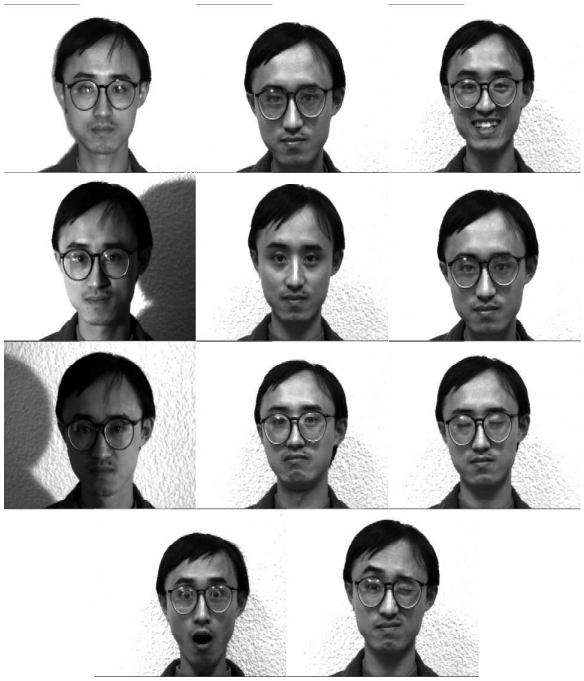


Figura 2.10 : variações faciais existentes no banco de dados YALE-1 para um dado indivíduo.

2.2 Metodologia

Os experimentos realizados nesta dissertação obedeceram à metodologia de projeto e validação ilustrada na Figura 2.11.

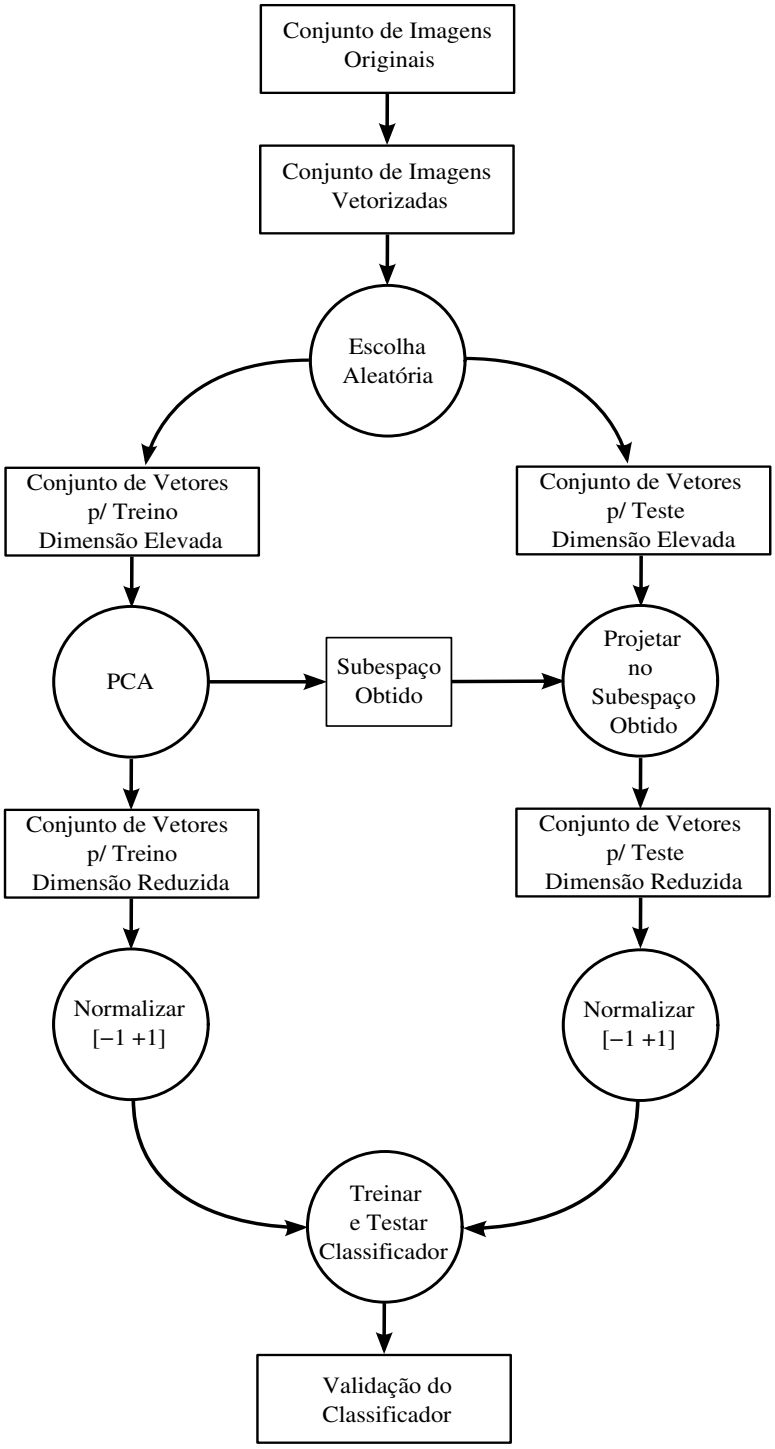


Figura 2.11 : metodologia de projeto e validação de classificadores usada nesta dissertação.

A seqüência de etapas apresentadas neste fluxograma pode ser resumida nas seguintes etapas:

- As imagens que compõem o conjunto utilizado são vetorizadas; ou seja, cada imagem é transformada de uma matriz de pixels para um vetor coluna.
- Obtidos todos os vetores de imagens, são selecionados aleatoriamente c vetores para compor o conjunto de teste de modo que haja sempre um representante de cada classe compondo este conjunto. O restante dos vetores formam o conjunto de treinamento do classificador.
- Nas rotinas de randomização (seleção aleatória de exemplos e inicialização de pesos) foi utilizado o comando `RAND()` do Matlab©, que implementa por default o algoritmo *Mersenne Twister* [2].
- A técnica de Análise por Componentes Principais é aplicada aos vetores de treinamento para reduzir a dimensionalidade destes.
- Os vetores de teste, previamente selecionados, são projetados no subespaço obtido na etapa anterior e reduz-se a dimensionalidade destes.
- Antes das etapas de treinamento e teste a serem aplicados ao classificador, estes vetores de dimensão reduzida são normalizados.
- Ao término destes procedimentos o classificador é treinado, para logo em seguida o mesmo ser testado e ter seu desempenho validado com base em medidas estatísticas que serão descritas mais adiante.

Do fluxograma, quatro das etapas que o compõe (vetorização, seleção de exemplos de treinamento e teste, redução de dimensionalidade e normalização) são detalhadas a seguir:

Vetorização

Usualmente, uma imagem digital é representada como uma matriz de intensidade de pixels de resolução $M \times N$. Antes de ser apresentada ao classificador, cada imagem digital $\mathbf{I}$ precisa ser vetorizada, ou seja, convertida em um vetor $\mathbf{x}$ de dimensão $M \cdot N$. Matematicamente, o processo de vetorização consiste em um mapeamento f do espaço das matrizes de dimensão $M \times N$ para o espaço de vetores de dimensão $N \cdot M$, ou seja

$$f : \mathbf{I} \in \mathbb{R}^M \times \mathbb{R}^N \rightarrow \mathbf{x} \in \mathbb{R}^{M \cdot N} \tag{2.1}$$

[2]
- http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/randstream.list.html
- http://www.math.sci.hiroshima-u.ac.jp/∼m-mat/MT/emt.html

Esse mapeamento se faz necessário porque classificadores de padrões manipulam vetores na entrada, e não matrizes. Nesta dissertação, optou-se pelo procedimento de *vetorizar por colunas*, através do qual as colunas da imagem original são empilhadas uma embaixo da outra: a segunda abaixo da primeira, a terceira abaixo da segunda e assim sucessivamente, de modo a se ter no final apenas um vetor coluna.

É importante notar que, mesmo para imagens de baixa resolução (e.g. 60×60), tais como aquelas presentes no banco de faces BARTLETT, as respectivas imagens vetorizadas têm dimensão 3600, que é um número bastante elevado. Por isso, faz-se também necessário reduzir a dimensão dos vetores a serem apresentados aos classificadores.

Seleção de Exemplos de Treinamento e Teste

O critério adotado para a seleção destes conjuntos de treino e teste é um misto entre *Leave-One-Out* (HAYKIN, 1994) e *Bootstrap* (GOOD, 2006). O procedimento é melhor descrito nos seguintes passos:

Passo 1 - Escolhe-se aleatoriamente uma imagem vetorizada de cada indivíduo para compor o grupo de teste. Ao final desta seleção o grupo de teste contará com c exemplares para testar o classificador. O restante dos vetores compõe o conjunto de treinamento.

Passo 2 - Repete-se o Passo 1 por um número R ($R \gg 1$) de realizações.

Redução de Dimensionalidade

Trabalhar com dados de alta dimensionalidade é um problema notório em Reconhecimento de Padrões (BELLMAN, 1961), problema este que pode ocorrer em tarefas tão diversas quanto predição de séries temporais (VERLEYSEN; FRANÇOIS, 2005) ou classificação de padrões (JIMENEZ; LANDGREBE, 1998; FUKUNAGA; HAYES, 1989). Uma alternativa comumente utilizada para tratar este problema é a técnica de Análise por Componentes Principais (*Principal Component Analysis*, PCA) (JOLLIFFE, 1986).

PCA é uma transformação linear aplicada aos vetores de dados pela qual a primeira dimensão obtida (i.e. primeiro componente principal ou hiperplano) é alinhada com a direção de maior variância dos dados, o segundo componente principal é alinhado com a direção de maior variância dentre todas as direções ortogonais à primeira direção obtida, e assim por diante. Estes componentes podem ser, então, usadas para descrever os dados originais e a quantidade destes a ser utilizada é uma escolha de projeto. Isto resulta da quantidade de variância que se deseja explicar dos dados originais.

Assim, aplica-se PCA[3] às imagens vetorizadas, doravante chamadas de vetores ou pa-

[3]Neste trabalho é utilizada a função *pca.m* implementada por C. Merkwirth (MERKWIRTH et al., 2005). O método das potências é adotado.

drões de entrada, selecionados para treino dos classificadores e são utilizados tantos componentes principais quanto fossem necessários para explicar 90% da variância do conjunto original. A escolha por esta porcentagem deu-se após avaliações preliminares de desempenho dos algoritmos para 70, 75, 80, 85, 90 e 95 porcento de variância explicada dos dados.

Técnicas alternativas de redução da dimensionalidade, tais como Wavelets, Curvas Principais (HASTIE; STUETZLE, 1989) ou PCA não-linear (OJA, 1989), poderiam também ser igualmente aplicadas, porém como o objetivo maior é analisar o desempenho dos classificadores e não de técnicas de pré-processamento de imagens, a abordagem adotada foi de escolher uma técnica de redução de dimensionalidade amplamente aceita e concentrar esforços na avaliação de desempenho dos classificadores de padrões. Esta abordagem mostrou-se adequada para todos os bancos de imagens utilizadas, visto que as taxas de reconhecimento produzidas pelos classificadores são bastante elevadas, mesmo para imagens adquiridas sob condições adversas de iluminação, *background* e expressões faciais.

Antes de aplicar PCA, todas as imagens vetorizadas de um certo conjunto (e.g. BARTLETT) são organizadas como colunas de uma matriz de dados $\mathcal{X} \in \mathbb{R}^{MN} \times \mathbb{R}^{P}$, em que P é o número de exemplos de treinamento. Ao aplicar PCA, faz-se necessário calcular $M \cdot N$ autovalores e os autovetores correspondentes da matriz de autocovariância $\mathbf{C}_{\mathcal{X}}$, dada por

$$\mathbf{C}_{\mathcal{X}} = E[\mathcal{X}\mathcal{X}^{T}] \approx \frac{1}{P}\mathcal{X}\mathcal{X}^{T}, \tag{2.2}$$

em que $E[\cdot]$ é o operador valor esperado e o sobrescrito T denota a matriz transposta. O número de componentes principais L ($L \ll MN$), a serem escolhidos, define a nova dimensão dos vetores de entrada a serem usados para treinar ou testar os classificadores avaliados neste trabalho.

A aplicação de PCA à matriz de dados $\mathcal{X}$ gera uma nova matriz $\mathcal{Y} \in \mathbb{R}^{L} \times \mathbb{R}^{P}$ de dados transformados. No contexto de redução de dimensionalidade, isto equivale a dizer que cada coluna $\mathbf{x}_i \in \mathbb{R}^{MN}$, $i = 1, ..., P$, da matriz de dados $\mathcal{X}$ é linearmente transformada em um vetor $\mathbf{y}_i \in \mathbb{R}^{L}$ que corresponde a i-ésima coluna da matriz $\mathcal{Y}$.

Na Figura 2.12 à 2.16 são mostrados o valores de cada componente principal (autovetor) para uma dada matriz de covariância, ou seja, quanto um dado componente principal explica da variância original para um certo conjunto de imagens.

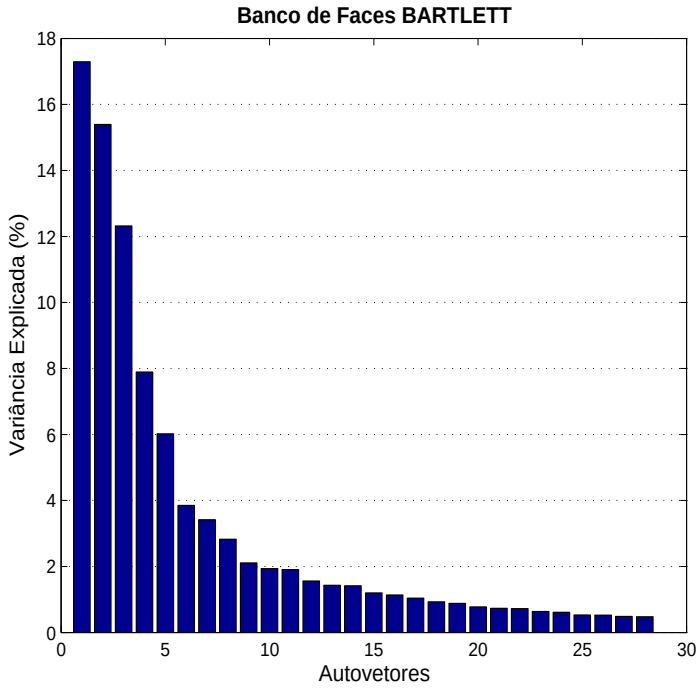


Figura 2.12 : importância relativa dos primeiros 28 componentes principais associados às imagens do conjunto BARTLETT selecionadas para treinamento.

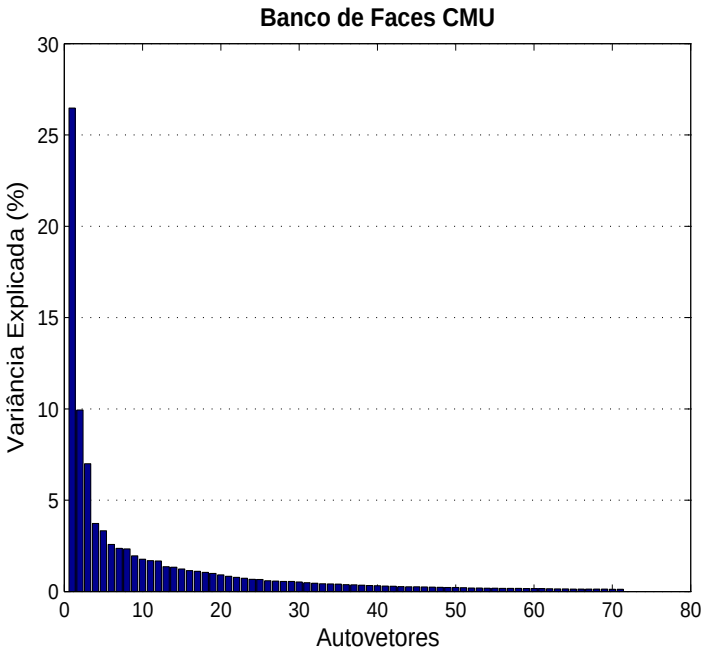


Figura 2.13 : importância relativa dos primeiros 71 componentes principais associados às imagens do conjunto CMU selecionadas para treinamento.

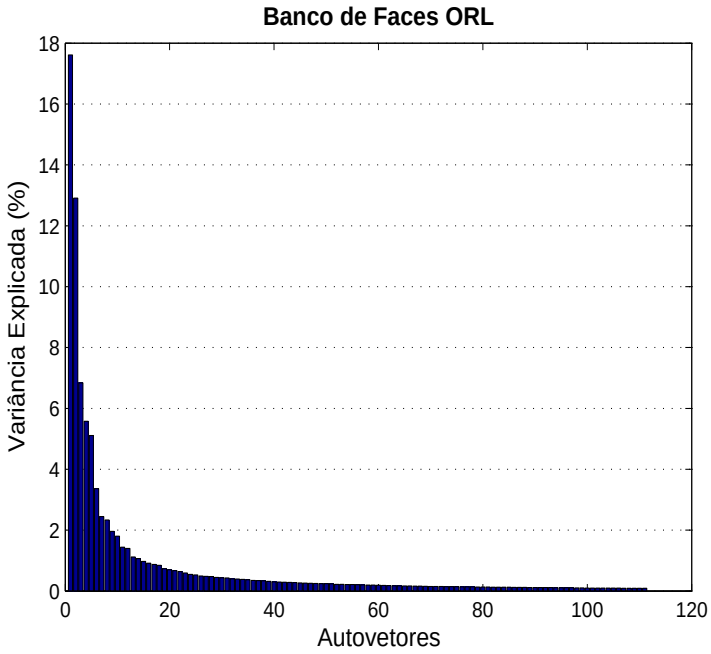


Figura 2.14 : importância relativa dos primeiros 111 componentes principais associados às imagens do conjunto ORL selecionadas para treinamento.

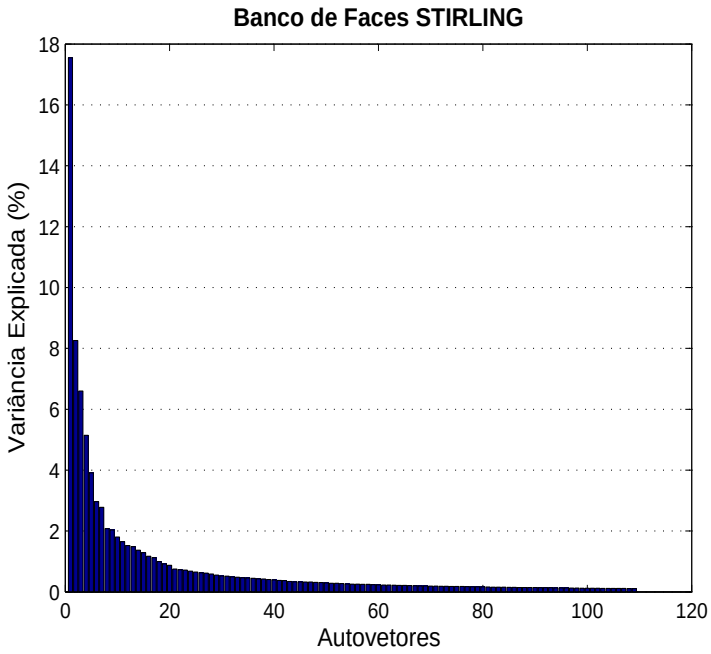


Figura 2.15 : importância relativa dos primeiros 109 componentes principais associados às imagens do conjunto STIRLING selecionadas para treinamento.

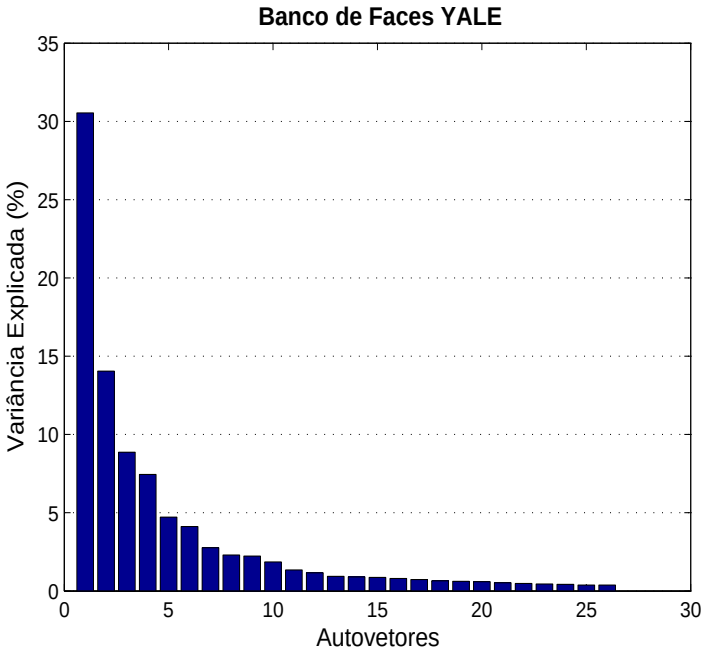


Figura 2.16 : importância relativa dos primeiros 26 componentes principais associados às imagens do conjunto YALE-1 selecionadas para treinamento.

Observa-se pelas Figuras anteriores como a técnica PCA é capaz de representar bem todos os conjuntos de imagens avaliados. Em outras palavras, todos os conjuntos de faces requerem um número particularmente reduzido de componentes principais para explicar 90% da variância dos dados originais. Este resultado é indicativo de um alto grau de redundância nos dados originais, fato este já esperado para o caso de imagens digitais. A Tabela 2.1 apresenta os valores das dimensões dos vetores antes e depois da aplicação de PCA para redução de dimensionalidade das imagens vetorizadas para todos os bancos de faces avaliados.

Tabela 2.1 : dimensões das imagens vetorizadas antes e depois da aplicação de PCA.

Database	#vetores	pré PCA	pós PCA
BARTLETT	100	3.600	28
CMU	640	15.360	71
ORL	400	10.304	111
STIRLING	324	94.500	109
YALE	165	77.760	26

A última coluna da tabela 2.1 representa, na prática, a quantidade de componentes principais (combinação linear de pixels das imagens) que foram obtidos das imagens selecionadas para treino. Assim, após dividir-se as imagens do conjunto, uma parte para

treino e outra para teste (um exemplar de cada cada classe), estas últimas foram projetadas sobre o espaço encontrado com as primeiras. Com esta variância explicada, alguns bancos de imagens tiveram taxas de compressão bem elevadas (e.g. STIRLING).

Normalização

Uma vez obtida a matriz de dados transformados $\mathcal{Y}$, a etapa seguinte consiste em efetuar uma normalização isotrópica[4] dos componentes dos vetores $\mathbf{y}_i$, $i = 1, ..., P$, condicionando-os ao intervalo $[-1 \quad 1]$. Com isto, busca-se basicamente evitar que uma dimensão se sobreponha em relação às outras, distorcendo os valores providos pelas métricas de distância (e.g. distância euclidiana).

O procedimento de normalização pode ser resumido da seguinte forma. Para cada componente y_{il} do vetor $\mathbf{y}_i$, o seu novo valor y_{il}^* é dado por

$$y_{il}^* = 2\left(\frac{y_{il} - y_{il}^{min}}{y_{il}^{max} - y_{il}^{min}}\right) - 1 \tag{2.3}$$

em que

$$y_{il}^{max} = \max_{k=1,...,P}\{y_{il}(k)\} \qquad \text{e} \qquad y_{il}^{min} = \min_{k=1,...,P}\{y_l(k)\}. \tag{2.4}$$

[4]Isotropia no espaço: não existem direções no espaço privilegiadas, ou, equivalentemente, identificáveis. Qualquer direção do espaço é equivalente às outras direções.

2.3 Resumo do Capítulo

Neste Capítulo, foram brevemente descritos os cinco conjuntos de faces utilizados neste trabalho e a metodologia adotada para resolver este problema de classificação de faces. Na metodologia adotada, quatro etapas merecem destaque:

1. vetorizar as imagens;
2. seleção de exemplos de treinamento e teste;
3. reduzir a dimensionalidade dos dados;
4. normalizar as suas componentes por vetor.

A justificativa para este Capítulo deve-se à escolha feita em dividir os resultados de classificação por grupos de classificadores e apresentá-los à medida que cada arquitetura seja descrita. No Capítulo seguinte serão apresentados os classificadores baseados na rede auto-organizável de Kohonen.

3 A Rede Auto-Organizável de Kohonen

"Sempre que ensinares, ensina a duvidar do que ensinas"

Ortega y Gosset

Este Capítulo traz uma introdução à **Rede Auto-Organizável de Kohonen** (*Kohonen's Self-Organizing Map*, SOM), proposta por Kohonen (1982). Esta rede, que pertence à classe das redes neurais não-supervisionadas competitivas, é uma das principais arquiteturas neurais e possui aplicações em um número variado de áreas da Engenharia e da Ciência, conforme pode ser averiguado no levantamento bibliográfico feito em Oja et al. (2003). Com base neste levantamento, percebe-se que a rede SOM tem sido aplicada principalmente em tarefas de análise de agrupamentos (*c*lustering), quantização vetorial e visualização de dados multidimensionais, que são tarefas de natureza inerentemente não-supervisionadas. Contudo, um dos objetivos desta dissertação é avaliar o desempenho de classificadores de padrões baseados na rede SOM, destacando o fato menos conhecido de que esta rede, com poucas modificações, pode também ser aplicada na categorização de dados rotulados. Tais classificadores baseados na rede SOM são também apresentados neste Capítulo.

3.1 Sobre a Rede SOM

Modelos de aprendizagem competitiva constituem uma das principais classes de redes neurais não-supervisionadas (PRINCIPE et al., 2000; HAYKIN, 1994). Este tipo de algoritmo de aprendizagem é baseado no conceito de *neurônio vencedor*, definido como o neurônio cujo vetor de pesos é o mais próximo ao vetor de entrada atual. Usando distância euclidiana como medida de proximidade, o neurônio vencedor $i^*(t)$ é determinado pela seguinte expressão

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|, \tag{3.1}$$

em que $\mathbf{x}(t) \in \mathbb{R}^m$ denota o vetor de entrada atual, $\mathbf{w}_i(t) \in \mathbb{R}^m$ é o vetor de pesos do neurônio i e t simboliza a iteração atual do algoritmo. Durante a fase de aprendizagem, os vetores de pesos dos neurônios vencedores são modificados incrementalmente no tempo a fim de extrair *propriedades estatísticas médias* do conjunto de padrões de entrada.

Como exemplo destes modelos, a rede auto-organizável de Kohonen (SOM) é um conhecido algoritmo de aprendizagem competitiva que foi proposto com o intuito de modelar algumas características essenciais dos mapas computacionais existentes no cérebro humano sem ter, contudo, a ambição de ser um modelo biologicamente plausível. A popularização da rede SOM deve-se mais propriamente às aplicações do que a sua teoria, esta ainda não totalmente desenvolvida (FORT, 2006).

Essa rede tem como objetivo transformar um sinal de entrada de dimensão qualquer em um arranjo discreto de unidades de processamento, ou, como em muitas de suas aplicações, ela aprende a mapear (i.e. projetar) um espaço de entrada $\mathcal{X}$, contínuo e de alta dimensionalidade, em um espaço de saída $\mathcal{A}$, discreto e de baixa dimensionalidade, em que N neurônios estão dispostos em estruturas geométricas fixas, em geral, grades uni-, bi- ou tridimensionais. Vale ressaltar que esta projeção é executada de forma adaptativa, buscando-se conservar, à medida do possível, relações de similaridade topológica entre os dois espaços envolvidos. As Figuras 3.1 e 3.2 ilustram tais projeções.

Para entender melhor como se dá esse processo, supõe-se que $\mathcal{X}$ é um espaço contínuo de dados de entrada tal que sua topologia seja definida por certas relações métricas entre vetores $\mathbf{x} \in \mathcal{X}$. Sem perda de generalidade, supõe-se também que deste espaço $\mathcal{X}$ conheçam-se apenas um conjunto finito de vetores (amostras) $\mathbf{x} \in \mathcal{X}$ organizados segundo uma densidade de probabilidade $p(\mathbf{x})$ desconhecida. Por sua vez, a topologia do espaço de saída $\mathcal{A}$ é definida pelo arranjo geométrico de um conjunto de neurônios $i \in \mathcal{A}$ e define-se, nesta condição, uma transformação não-linear Φ, chamada *mapeamento de características* (*feature map*, em inglês), que leva do espaço de entrada $\mathcal{X}$ ao espaço de saída $\mathcal{A}$ (ver Figura 3.1). Matematicamente, tem-se a seguinte representação

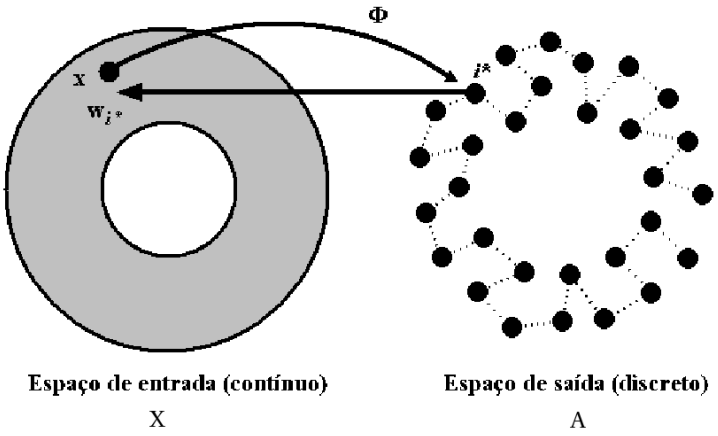


Figura 3.1 : esboço do mapeamento de características Φ implementado por uma rede SOM do tipo unidimensional.

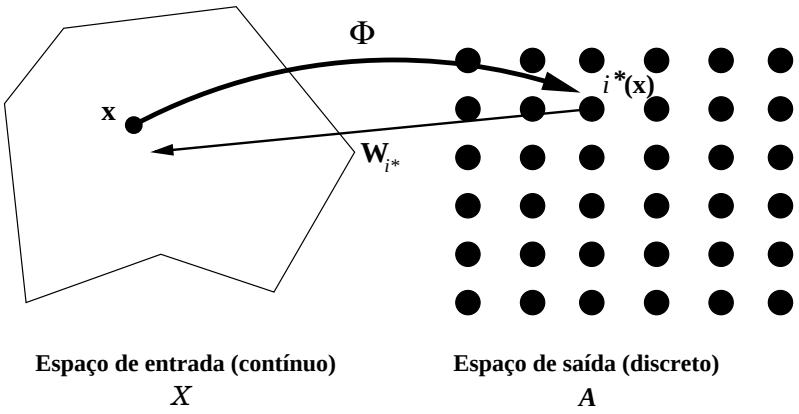


Figura 3.2 : esboço do mapeamento de características Φ implementado por uma rede SOM do tipo bidimensional.

$$\Phi : \mathcal{X} \rightarrow \mathcal{A}. \tag{3.2}$$

Assim, sendo m a dimensão do vetor de espaço de entrada $\mathcal{X}$, um vetor de entrada $\mathbf{x}(t) \in \mathcal{X} \subset \mathbb{R}^m$, selecionado de forma aleatória, é representado por

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_m(t) \end{pmatrix}. \tag{3.3}$$

E, nestas condições, um vetor de pesos associado a cada neurônio da rede tem a mesma dimensão do vetor de entrada. Sendo representado para um dado neurônio i da seguinte

forma

$$\mathbf{w}_i = \begin{pmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{im} \end{pmatrix}, \quad i = 1, 2, \ldots, q, \tag{3.4}$$

na qual q é o número total de neurônios da rede.

> Observação: a atribuição de valores iniciais aos pesos pode ser realizada de diversas formas (KOHONEN, 1998) como, por exemplo, valores aleatórios ou amostrados do espaço de entrada, valores obtidos de um subespaço linear gerado com os dois autovetores mais significativos oriundos da matriz de auto-correlação dos dados de entrada, dentre outros modos possíveis. A escolha do método de iniciação dos pesos é uma decisão de projeto, ou seja, uma decisão do projetista ou usuário da rede.

Na seqüencia, para se obter um mapa auto-organizável, três condicionantes principais podem ser identificadas neste processo.

- *Competição.* Existe uma função que avalia cada neurônio da rede e identifica aquele mais próximo ao vetor de entrada corrente. Este neurônio é dito, então, *vencedor* da competição. Metaforicamente, este processo pode ser entendido em analogia aos diversos estímulos sensoriais captados pelos órgãos sensoriais e que "alimentam" o córtex cerebral continuamente, ativando ou não neurônios, ou conjuntos específicos destes, mais dedicados a tratar a informação que carregam.

- *Cooperação.* Ao se obter o neurônio vencedor, não apenas este, mas também seus vizinhos no arranjo geométrico de saída "interagem" com o estímulo de entrada corrente através de uma *função vizinhança.* Esta interação intensifica a resposta deste grupo de neurônios fisicamente próximos ao padrão de entrada corrente, condicionando-os a melhor representar padrões de entrada similares que porventura venham a ocorrer no futuro.

- *Adaptação.* Os pesos sinápticos da rede são ajustados ao término da competição e cooperação aplicadas ao mapa. Esta força de ajuste, que atua no neurônio vencedor e seus vizinhos, decresce com o tempo e faz com que o mapa por eles formado se distribua pelo espaço de entrada e o melhor represente. Medidas globais de erro de quantização do mapa ajudam a explicar como se dá esta adequação com o tempo.

Nas subseções seguintes apresentam-se, em mais detalhes, estas etapas de competição, cooperação e adaptação, que caracterizam estes mapas auto-organizáveis de Kohonen,

então tratadas. Além disso, questões relativas à convergência da rede SOM e a preservação da topologia conseguida por esta arquitetura são abordadas.

3.1.1 Competição - Uma medida de similaridade

Na busca pelo neurônio vencedor, mensura-se a distância entre todos os pesos sinápticos do mapa e o padrão de entrada corrente. Aquele neurônio que apresentar maior similaridade ou, de outro modo, esteja mais próximo do padrão apresentado é considerado *vencedor* neste processo de escolha. Várias métricas podem ser adotadas neste sentido, de acordo com o problema com o qual se lida. Dentre estas, distância Manhattan, distância Hamming, distância de Mahalanobis, ou Entropia, por exemplo. Tendo em vista o freqüente uso da distância euclidiana, e também por ter sido adotada neste trabalho, este processo de competição pode ser representado da seguinte forma

$$i^*(\mathbf{x}(t)) = \arg\min_{i \in \mathcal{A}} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|, \tag{3.5}$$

em que $i^*(\mathbf{x}(t))$ é o índice que representa o neurônio vencedor para o padrão de entrada $\mathbf{x}(t)$. A norma euclidiana $\|\cdot\|$ é definida como

$$\|\mathbf{x}(t) - \mathbf{w}_i(t)\| = \sqrt{[\mathbf{x}(t) - \mathbf{w}_i(t)]^T[\mathbf{x}(t) - \mathbf{w}_i(t)]} = \sqrt{\sum_{j=1}^{p}[x_j(t) - w_{ij}(t)]^2}, \tag{3.6}$$

na qual $(\cdot)^T$ denota o vetor transposto. A Equação (3.5) permite a seguinte observação:

> *O espaço contínuo, ao qual pertencem os vetores de entrada, é mapeado em um espaço de saída discreto por meio de um processo de competição entre os neurônios.*

3.1.2 Cooperação - O papel da função de vizinhança

O neurônio vencedor $i^*(t)$ determina o centro de um grupo espacialmente localizado de neurônios no mapa chamado de vizinhança de i^*. O neurônio vencedor e sua vizinhança interagem lateralmente de forma cooperativa. Existe evidência neurobiológica para esta *interação lateral* entre os neurônios excitados (KOHONEN, 1997). Em particular, um neurônio que está disparando tende a excitar neurônios em sua vizinhança imediata *mais* do que aqueles mais distantes. A intensidade da interação lateral entre o vencedor i^* e um neurônio i qualquer é, em geral, descrita matematicamente na forma de uma função vizinhança $h(i^*, i; t)$. Esta função define o que se chama de vizinhança *topológica* centrada

no neurônio vencedor $i^*(\mathbf{x}(t))$.

Considerando o estudo do arranjo discreto assumido pelos neurônios na rede SOM, têm-se dois modelos de grades bastante utilizados (KOHONEN, 1997): (i) a grade unidimensional e, (ii) a grade bidimensional. Analisando o caso de que os neurônios estejam dispostos em uma grade unidimensional, tem-se que $\mathbf{r}_i(t) \in \mathbb{R}$, ou seja, a posição de um neurônio i qualquer coincide com seu próprio índice, ou seja, $\mathbf{r}_i(t) = i$. Neste caso, cada neurônio possui apenas vizinhos à direita e à esquerda (ver Figura 3.1). Contudo, se os neurônios da rede SOM estão dispostos em uma grade bidimensional (como é o caso de todas as implementações deste trabalho), tem-se que $\mathbf{r}_i(t) \in \mathbb{R}^2$, ou seja, a posição de um neurônio i na grade é dada pelas coordenadas (x_i, y_i) em relação a uma origem pré-fixada. Neste caso, um neurônio pode ter vizinhos à esquerda, à direita, acima, abaixo e diagonalmente (ver Figura 3.2).

Seja $\mathbf{r}_i$ a localização (coordenadas) do neurônio i em um arranjo uni-dimensional, por exemplo. Então, assume-se que $h(i^*, i; t)$ é uma função unimodal da distância lateral $\|\mathbf{r}_{i^*} - \mathbf{r}_i\|$ entre o neurônio vencedor i^* e seu vizinho i no mapa. A função vizinhança deve ainda satisfazer os seguintes requisitos:

- alcançar seu valor máximo para o neurônio vencedor i^*, visto que, para este a distância lateral $\|\mathbf{r}_{i^*} - \mathbf{r}_i\|$ é nula;
- ser simétrica em relação ao neurônio vencedor;
- a amplitude de $h(i^*, i; t)$ decai monotonicamente com o aumento da distância lateral $\|\mathbf{r}_{i^*} - \mathbf{r}_i\|$, ou seja:

$$\text{Se} \quad \|\mathbf{r}_{i^*} - \mathbf{r}_i\| \longrightarrow \infty \quad \Longrightarrow \quad h(i^*, i; t) \longrightarrow 0. \tag{3.7}$$

Esta equação é uma condição necessária para convergência.

Uma escolha comum para $h(i^*, i; t)$ que satisfaz os requisitos anteriores é a função gaussiana, ou seja,

$$h(i^*, i; t) = \exp\left(-\frac{\|\mathbf{r}_{i^*} - \mathbf{r}_i\|^2}{2\sigma^2(t)}\right), \tag{3.8}$$

em que $\mathbf{r}_i(t)$ e $\mathbf{r}_{i^*}(t)$ são, respectivamente, as posições dos neurônios i e i^* na grade de saída predefinida, e o parâmetro $\sigma(t) > 0$ define a "largura efetiva" da vizinhança topológica, isto é, ele define como os demais neurônios participam no processo de aprendizagem juntamente com o neurônio vencedor no instante atual. É importante não confundir o parâmetro σ usado pela rede SOM com o seu equivalente das redes RBF e GRNN, pois,

este usado se refere ao raio de ação do neurônio definido pela largura da função de ativação gaussiana do mesmo.

Originalmente, o algoritmo SOM descrito em Kohonen (1982) utiliza uma função vizinhança de amplitude constante, chamada de retangular e dada por

$$h(i^*, i; t) = \begin{cases} 1, \text{ se } i \in N_{i^*}(t) \\ 0, \text{ caso contrário,} \end{cases} \tag{3.9}$$

em que $N_{i^*}(t)$, chamado de *conjunto vizinhança*, contém os neurônios vizinhos de $i^*(t)$. Embora esta função de vizinhança retangular apresente um custo computacional consideravelmente menor do que a gaussiana (KOHONEN, 1998), esta última é a utilizada em todas as implementações deste trabalho.

Tanto para a função vizinhança gaussiana quanto para a retangular, a largura da vizinhança decresce monotonicamente com o passar do processo de aprendizagem. A diminuição da largura desta função contribui para o ordenamento e convergência da rede.

3.1.3 Adaptação - Ajuste dos Pesos

Dentro do processo cíclico que conduz a formação de um mapa auto-organizável, o ajuste dos pesos sinápticos pode ser visto como a última etapa, antes que se obtenha o mapa desejado ou, então, que se apresente um novo padrão à rede. A questão que se coloca é como realizar a alteração dos pesos. Como a rede SOM é não-supervisionada, o vetor de pesos de um dado neurônio i deve ser modificado como função do estímulo de entrada apenas. Como pôde ser visto, as etapas de competição e cooperação atuam conjuntamente durante a adaptação sináptica com o objetivo de extrair algum tipo de regularidade presente no espaço de entrada $\mathcal{X}$. A regra de aprendizagem para o algoritmo SOM baseia-se em suposições feitas por Hebb (1949), em uma tentativa de relacionar a alteração estrutural de sinapses reais com a memória e, conseqüentemente, com aprendizagem ou experiência.

Neste sentido, uma abstração matemática destas suposições foi proposta por Kohonen como uma regra recursiva para ajuste dos pesos (KOHONEN, 1998)

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)h(i^*, i; t)[\mathbf{x}(t) - \mathbf{w}_i(t)], \tag{3.10}$$

onde o parâmetro $0 < \alpha(t) < 1$, chamado de *taxa de aprendizagem*, controla a intensidade com que os pesos sinápticos são modificados. Assim como no caso da largura da vizinhança topológica, a taxa de aprendizagem pode diminuir com o transcorrer do treinamento de modo a garantir convergência e, principalmente, estabilidade do mapa[1].

[1]Neste contexto, estabilidade se refere à manutenção de memória previamente aprendida quando novos

3.1.4 Ordenamento e Convergência

A partir de uma configuração inicial em que os pesos guardam valores atribuídos aleatoriamente, o algoritmo SOM modifica maximamente os valores dos pesos em direção a uma representação que reflita a estrutura (topologia) do espaço de entrada. Para que uma configuração "organizada" e estável do mapa seja atingida, faz-se necessária uma seleção criteriosa dos parâmetros α_0 e σ_0. Quando uma configuração organizada final é alcançada, diz-se que o algoritmo convergiu ou atingiu um estado final.

O primeiro ponto fundamental de importância para a convergência e estabilidade do mapa é a diminuição gradual da amplitude da taxa de aprendizagem $\alpha(t)$. Em particular, este parâmetro deve começar com um valor inicial α_0 para $t = t_0$, e então diminuir monotonicamente até atingir um valor final α_T após um número fixo de iterações T. Este requisito pode ser satisfeito escolhendo-se um decaimento exponencial (RITTER et al., 1992) como mostrado na Eq. (3.11).

$$\alpha(t) = \alpha_0 \left(\frac{\alpha_T}{\alpha_0}\right)^{\frac{t}{T}} \quad \text{e} \quad \sigma(t) = \sigma_0 \left(\frac{\sigma_T}{\sigma_0}\right)^{\frac{t}{T}}, \tag{3.11}$$

em que (α_0 , σ_0) e (α_T , σ_T) são os valores inicial e final de ($\alpha(t)$, $\sigma(t)$), respectivamente.

Outro ponto de importância capital para o processo de convergência da rede é a diminuição da largura da vizinhança topológica. Para o caso de uma função vizinhança gaussiana, esta largura é refletida no valor do parâmetro $\sigma(t)$. A largura deve ser inicialmente alta para promover um rápido ordenamento dos pesos e decrescer de modo a garantir convergência dos mesmos. Uma escolha comum para a dependência temporal de $\sigma(t)$ tem a forma apresentada na (Eq. 3.11).

O propósito de uma largura inicial grande para $h(i^*, i; t)$ é correlacionar as direções de ajuste dos pesos de um grande número de neurônios da rede. À medida que a largura de $h(i^*, i; t)$ decresce, também decresce o número de neurônios, cujas direções de ajuste estejam correlacionadas.

3.1.5 Preservação de Topologia

De modo geral, pode-se expressar a propriedade de preservação de topologia da rede SOM da seguinte forma (HERTZ et al., 1991): sejam $\mathbf{x}_1$ e $\mathbf{x}_2$ dois vetores no espaço de entrada $\mathcal{X}$, $\mathbf{r}_{i_1^*}$ e $\mathbf{r}_{i_2^*}$ as coordenadas dos neurônios vencedores para $\mathbf{x}_1$ e $\mathbf{x}_2$, respectiva-

dados são apresentados à rede SOM.

mente. Desta forma, a rede SOM preserva, de modo aproximado[2], a topologia do espaço de entrada se a seguinte relação for observada

$$\|\mathbf{x}_1 - \mathbf{x}_2\| \longrightarrow 0 \quad \Longrightarrow \quad \|\mathbf{r}_{i_1^*} - \mathbf{r}_{i_2^*}\| \longrightarrow 0, \tag{3.12}$$

ou seja, se quaisquer dois vetores estão fisicamente próximos no espaço de entrada, então eles terão neurônios vencedores espacialmente próximos na rede. Dá-se a essa característica, o nome de *Propriedade de Preservação de Topologia*. Com base nesta propriedade, pode-se afirmar que o espaço contínuo $\mathcal{X}$ é mapeado em um espaço discreto de saída $\mathcal{A}$ por um processo de competição-cooperação entre as unidades da rede, de forma tal que a sua topologia é preservada.

Devido à propriedade de preservação de topologia, a rede SOM é capaz de construir uma **aproximação do espaço de entrada**, ou seja, ela constrói uma aproximação discreta do espaço de entrada, na qual cada neurônio da rede representa uma determinada região do espaço de entrada que define sua **região de atração** ou **campo receptivo**. Esta região é conhecida também como **célula de Voronoi**. Assim, uma das principais aplicações da rede SOM é a categorização de dados não-rotulados em agrupamentos (*clusters*) e sua posterior utilização na classificação de vetores de características que não estavam presentes durante o treinamento.

Com o uso das propriedades da rede SOM, tais como competição e cooperação, esta rede neural é capaz de implementar uma projeção Φ que preserve relações de proximidade espacial entre os dados de entrada. Em outras palavras, o mapeamento preserva a topologia do espaço de entrada no espaço de saída (HAYKIN, 1994), conforme ilustrado na Figura 3.1, na qual $dim(\mathcal{X}) = p = 2$ e $dim(\mathcal{A}) = 1$, e os pontos pretos correspondem às coordenadas dos vetores de pesos do i-ésimo neurônio. Neurônios que são vizinhos na grade unidimensional são conectados por linhas tracejadas.

3.2 Mapas auto-organizáveis de Kohonen para classificação

Visto que a rede SOM é formada através de um processo não-supervisionado, que constrói uma representação compacta da distribuição estatística dos dados, este algoritmo é particularmente adequado às tarefas de reconhecimento de padrões não-supervisionadas, tais como análise de agrupamentos ou visualização dos dados.

O questionamento que pode surgir em decorrência do exposto nas seções anteriores

[2] A preservação de topologia levada a cabo pela rede SOM é apenas uma aproximação das relações topológicas entre dois espaços, visto que ela se dá em um único sentido (entrada → saída).

é "de que modo passar informações sobre as classes presentes nos dados para que a rede SOM as discrimine satisfatoriamente e possa generalizar para padrões novos?" ou, posto de outra forma, "como usar esta rede para classificar padrões?".

Em resposta a estas indagações, será mostrado nas subseções seguintes que a arquitetura SOM é flexível o bastante para permitir aos seus usuários aplicá-la também às tarefas de classificação supervisionada. Com este propósito, diversas abordagens foram propostas através dos anos, e quatro destas, serão utilizadas nesta dissertação.

Conforme será visto, a classificação baseada na rede SOM é realizada com base no vetor de pesos do neurônio vencedor, desse modo, todos os seus classificadores resultantes pertencem a família de classificadores baseados em protótipos (LAHA; PAL, 2001).

3.2.1 Definições Preliminares

Antes de prosseguir com a descrição dos classificadores convém instanciar alguns termos que serão utilizados no decorrer desta dissertação.

- **Iterações de Treinamento** - Divide-se o conjunto de faces em 02 grupos: um para treinar o classificador e o outro para testá-lo. Ao processo de apresentar um vetor de entrada ao mapa e atualizar seus pesos dá-se o nome de *iteração de treinamento*.
- **Épocas de Treinamento (E)** - Quando todos os vetores de entrada são apresentados para a rede diz-se que uma *época de treinamento* foi concluída.
- **Realizações (R)** - Quando concluem-se todas as *épocas de treinamento* e testa-se a rede com os vetores de teste, extraindo as medidas estatísticas que se queira, diz-se que uma *realização* foi executada.

3.2.2 Rotulação por Voto Majoritário (SOM-C1)

Após a determinação os conjuntos de treino e teste, treina-se no modo não-supervisionado usual uma rede auto-organizável de Kohonen utilizando o conjunto de vetores obtidos das imagens de treino, isto é, aquelas que não foram selecionadas para o teste. Feito isto, passa-se a uma fase de *rotulação pós-treinamento* que deve ser realizada para todos os neurônios da rede SOM antes da fase de teste. Para este fim, utiliza-se o mesmo conjunto de vetores de treinamento, porém nesta etapa os pesos dos neurônios são "congelados", ou seja, não são mais modificados.

O processo de rotulação pode ser dividido nos seguintes passos:

- apresenta-se novamente o conjunto de treinamento à rede e, para cada um dos padrões de entrada, encontra-se o neurônio vencedor correspondente;
- cada neurônio vencedor armazena os rótulos dos padrões de entrada que são nele mapeados;
- ao final da apresentação dos padrões de entrada, o rótulo atribuído a um dado neurônio é aquele de maior ocorrência dentre os padrões nele mapeados.

Visto que o mapeamento se dá dos dados de entrada para os neurônio da rede $\mathcal{X} \rightarrow \mathcal{A}$, algumas situações indesejáveis podem decorrer do processo de rotulação ora descrito. A Figura 3.3 auxilia no entendimento destas situações.

Situação 1 - Um dado neurônio pode ser selecionado como o vencedor para vetores de entrada pertencentes a classes distintas (neurônios em azul). Em caso de empate escolhe-se aleatoriamente a classe vencedora entre as mais representativas.

Situação 2 - Um ou mais neurônios podem não receber rótulos por não terem sido selecionados vencedores para quaisquer dos padrões de entrada (neurônios em vermelho). Neste trabalho, optou-se por descartar neurônios não-rotulados, ou seja, não usá-los na etapa de teste, muito embora outras alternativas[3] fossem possíveis.

Ao término dos processos de treinamento não-supervisionado e de rotulação dos neurônios por voto majoritário, os vetores de teste são apresentados aos neurônios da rede. Para cada um destes padrões de entrada, encontra-se o neurônio vencedor de acordo com a Eq. (3.5). A rede então indica como classe do padrão de entrada o rótulo do neurônio

[3]Na ocorrência de neurônios não-rotulados, estes podem herdar o rótulo do seu neurônio vizinho ou ser etiquetado com um rótulo de classe "desconhecida".

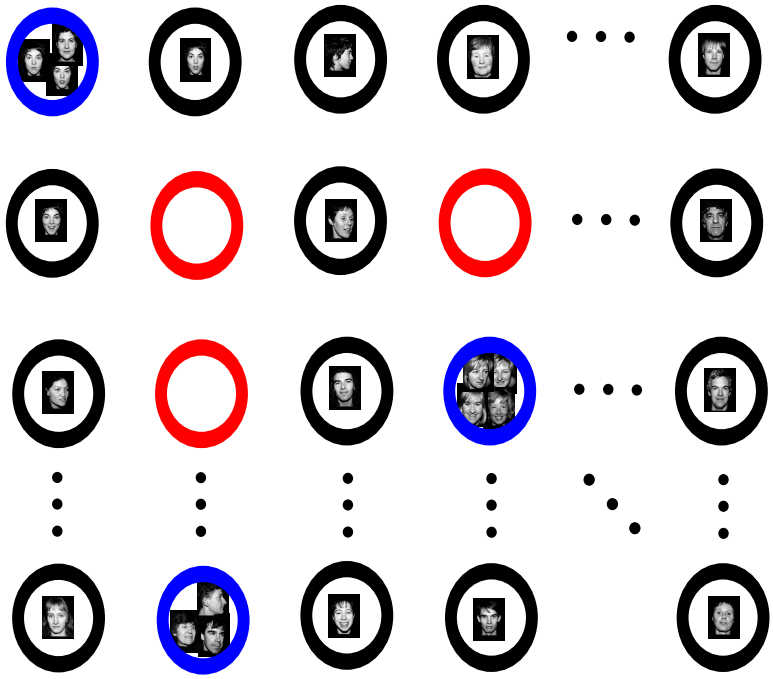


Figura 3.3 : ilustração de problemas encontrados na rotulação por voto majoritário.

vencedor. Neste momento, o rótulo do vetor de entrada é comparado ao rótulo do neurônio vencedor. Caso sejam iguais, um acerto é computado. Este processo é repetido para todos os vetores de teste disponíveis.

Classificadores baseados na rede SOM projetados via rotulação por voto majoritário são descritos, por exemplo, nos trabalhos de Wyns et al. (2004), Christodoulou et al. (2003), Laha & Pal (2001) e Suganthan (1999).

3.2.3 Rotulação por Mapas Individuais (SOM-C2)

A segunda abordagem usa uma rede SOM para cada classe disponível. No presente trabalho, isto corresponde a ter uma rede SOM para cada indivíduo a ser identificado. Matematicamente, tem-se tantos mapeamentos quantas forem as classes presentes no conjunto de vetores de faces, ou seja

$$\Phi_j : \mathcal{X}_j \rightarrow \mathcal{A}_j. \tag{3.13}$$

Em que j indica o j-ésimo subconjunto de vetores de faces de um indivíduo.

Por exemplo, se três classes de dados estão disponíveis, três redes SOM serão treinadas, uma para cada classe. Esta é, por exemplo, a abordagem utilizada por Biebelmann et al. (1996) no reconhecimento de texturas. As diversas redes SOM, contudo, são treinadas independentemente, usando somente os vetores de dados da classe a qual cada uma

representa. Neste trabalho, todas os mapas formados possuíam a mesma quantidade de neurônios, por questões de simplicidade.

Finalizadas as etapas de treinamento individuais, todos os mapas obtidos ($\Phi_1, \ldots,$ $\Phi_j, \ldots, \Phi_c$) já estão na verdade *pré-rotulados* e o processo de teste pode ser realizado da seguinte forma:

- um neurônio vencedor é procurado entre os neurônios de todas as c redes SOM disponíveis;
- a rede SOM à qual pertence o neurônio vencedor indica como classe do padrão de entrada o rótulo deste neurônio;
- caso o rótulo da imagem de teste seja igual ao rótulo do neurônio vencedor, o contador de acertos é incrementado;
- este processo se repete pra todos os padrões de teste disponíveis.

Neste caso, é importante destacar as principais diferenças entre os classificadores SOM-C1 e SOM-C2. No classificador SOM-C2 o número de redes é igual ao número de classes, sendo que a rotulação dos neurônios ocorre *antes* do treinamento das redes. Já no classificador SOM-C1, há apenas uma rede SOM, na qual os neurônios são rotulados *após* o treinamento da rede.

3.2.4 Rotulação Auto-Supervisionada (SOM-C3)

Na terceira abordagem, a rede SOM é "tornada" supervisionada pelo acréscimo da informação da classe para cada vetor de dados de entrada. Assim como no classificador SOM-C1, apenas uma rede SOM é utilizada. Esta abordagem é descrita pelo próprio Kohonen em seu livro (KOHONEN, 1997) e mais recentemente em Xiao et al. (2005). De acordo com esta abordagem, cada padrão de entrada é agora expresso como

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}^p(t) \\ \cdots \\ \mathbf{x}^l(t) \end{pmatrix} = \begin{pmatrix} x_1^p(t) \\ x_2^p(t) \\ \vdots \\ x_m^p(t) \\ \cdots \\ x_1^l(t) \\ x_2^l(t) \\ \vdots \\ x_c^l(t) \end{pmatrix}. \tag{3.14}$$

Em palavras, os vetores de entrada $\mathbf{x}(t) \in \mathbb{R}^{m+c}$ são agora formados de duas partes, denotadas por $\mathbf{x}^p(t) \in \mathbb{R}^m$ e $\mathbf{x}^l(t) \in \mathbb{R}^c$, em que $\mathbf{x}^p(t)$ é o próprio padrão de treinamento, enquanto $\mathbf{x}^l(t)$ é o rótulo da classe correspondente de $\mathbf{x}^p(t)$. Durante o treinamento, estes vetores são concatenados para construir vetores expandidos $\mathbf{x}(t) = [\mathbf{x}^p(t) \;\; \mathbf{x}^l(t)]^T$ que são usados como entradas para rede SOM. Os respectivos vetores de pesos expandidos, $\mathbf{w}_i(t) = [\mathbf{w}_i^p(t) \;\; \mathbf{w}_i^l(t)]^T$, são ajustados como no procedimento usual de treinamento da rede SOM.

Geralmente, o vetor de rótulo $\mathbf{x}^l(t)$ é representado como um vetor binário (ou bipolar), em que somente um de seus componentes é igual a "+1", enquanto os outros são feitos iguais a "0" (ou "-1"). O índice da posição "+1" indica a classe do vetor padrão $\mathbf{x}^p(t)$. Por exemplo, se três classes estão disponíveis, então três rótulos são possíveis

$$\begin{aligned} \mathbf{x}^{l=1} &= [+1 \;\; -1 \;\; -1]^T, \\ \mathbf{x}^{l=2} &= [-1 \;\; +1 \;\; -1]^T, \\ \mathbf{x}^{l=3} &= [-1 \;\; -1 \;\; +1]^T. \end{aligned} \tag{3.15}$$

Uma vez treinada, o procedimento de teste é realizado como se segue

- ao apresentar um vetor de rótulo desconhecido à rede, a sua parte $\mathbf{x}^l(t)$ não é considerada durante a busca pelo neurônio vencedor: isto significa que somente a

parte $\mathbf{x}^p(t)$ é comparada com a parte correspondente aos vetores de pesos,

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{x}^p(t) - \mathbf{w}_i^p(t)\|; \tag{3.16}$$

- o rótulo da classe do padrão desconhecido é comparado com a parte $\mathbf{w}_i^l(t)$ do vetor de pesos do neurônio vencedor $\mathbf{w}_{i^*}(t)$, os índices da componentes de $\mathbf{w}_{i^*}^l(t)$ e $\mathbf{x}^l(t)$ com maior valor são comparados e, caso haja coincidência, a rede é dita como tendo classificado corretamente o padrão desconhecido $\mathbf{x}^p(t)$ e o contador de acertos é incrementado.

Assim, como ocorre para o classificador SOM-C2, entende-se o processo de rotulação do classificador SOM-C3 como uma rotulação do tipo pré-treinamento.

3.2.5 Rotulação pelo Centróide mais Próximo (SOM-C4)

A quarta abordagem é uma proposta desta dissertação (MONTEIRO et al., 2006; QUEIROZ et al., 2006), que teve como motivação principal eliminar os problemas encontrados no processo de rotulação dos neurônios do classificador SOM-C1 (neurônios sem rótulo algum e neurônios com vários rótulos). A característica principal da proposta é usar os centróides das c classes disponíveis, calculados usando o conjunto de vetores de treinamento, para rotular os neurônios da rede SOM treinada de modo não-supervisionado. O procedimento de rotulação é detalhado a seguir.

Com os vetores de treinamento devidamente selecionados e supondo que, de antemão, todos os seus rótulos sejam conhecidos, treina-se um mapa auto-organizável com estes vetores no modo convencional (i.e. não-supervisionado). Após esta etapa, inicia-se o processo de rotulação o qual pode ser dividido, grosso modo, nos seguintes passos:

- supondo que existam N_j vetores de treinamento para uma dada classe j, o seu centróide é calculado da seguinte maneira

$$\mathbf{c}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in C_j} \mathbf{x}; \tag{3.17}$$

- cada um dos neurônios obtidos após o treinamento herda, então, o rótulo do vetor centróide mais próximo, ou seja,

$$Rotulo(\mathbf{w}_i) = \arg\min_{\forall j} \|\mathbf{w}_i - \mathbf{c}_j\|. \tag{3.18}$$

Com os neurônios do mapa devidamente rotulados, a etapa de teste do classificador ocorre do mesmo modo que para o classificador SOM-C1. Uma vantagem do método de

rotulação proposto sobre o classificador SOM-C1 é que cada neurônio da rede é levado em um, e somente um, dos centróides disponíveis. Isto implica que cada neurônio é associado a uma classe apenas, o que elimina situações indesejáveis na rotulação usada pelo classificador SOM-C1, tais como empate no processo de voto majoritário e a ocorrência de neurônios sem rótulos.

3.2.6 Resultados de Classificação

De antemão, algumas condições para realizar o treinamento foram seguidas. A primeira delas é que nesta dissertação optou-se por fixar em dez o número de épocas de treinamento para todos os classificadores. Embora este número de épocas seja considerado pequeno, o objetivo é avaliar o desempenho em situações que se exige um rápido treinamento das redes. Este tipo de treinamento rápido pode ser útil, por exemplo, em projetos de classificadores para sistemas embarcados de reconhecimento de voz (VARGA; KISS, 2008).

As Tabelas 3.1 a 3.5 mostram os valores adotados para os parâmetros dos classificadores quando aplicados aos diferentes bancos de faces apresentados no Capítulo 2. Algumas convenções, comuns ao treinamento dos classificadores baseados na rede SOM, são feitas a seguir.

- Todos os mapas utilizados foram do tipo bidimensionais com M_x neurônios em uma dimensão e M_y na outra. Assim, o produto $M_x \times M_y$ indica o número de neurônios da rede. Optou-se por utilizar mapas quadrados, ou seja, $M_x = M_y$ e os valores mostrados nas tabelas resultam de uma busca partindo do valor inicial $M_x = 2$ e incrementando-se este valor de 1 até um limite superior admissível de $2 \cdot \sqrt{\text{nº total de vetores de faces}}$. Assim, os valores destes parâmetros foram escolhidos com base na taxa de acerto média e no tempo de treinamento do classificador, nesta seqüência de prioridade.

- O parâmetro α_0 foi escolhido de modo que a rede convergisse rapidamente para o espaço de características apresentado pelos vetores de faces e mantivesse uma elevada taxa de acerto média. Este valor foi inicialmente feito igual a $0,1$ e incrementado de $0,05$ até um limite superior de $0,8$. Dentre estes valores testados, os apresentados nas tabelas foram os escolhidos com base no critério acima citado.

- O parâmetro σ_0 é tomado com relação às dimensões iniciais do mapa da seguinte forma:

$$\sigma_0 = \frac{max(M_x, M_y)}{2} \tag{3.19}$$

caso este valor não seja inteiro, esta variável recebe o valor inteiro imediatamente superior.

Tabela 3.1 : parâmetros de treinamento para o banco de faces BARTLETT.

Parâmetros Ajustáveis	**Classificadores**			
	$SOM - C1$	$SOM - C2$	$SOM - C3$	$SOM - C4$
M_x	12	3	13	13
M_y	12	3	13	13
α_0	0,5	0,5	0,5	0,5
α_T	0,001	0,001	0,001	0,001
σ_0	6	2	7	7
σ_T	0,001	0,001	0,001	0,001
R	300	300	300	300

Tabela 3.2 : parâmetros de treinamento para o banco de faces CMU.

Parâmetros Ajustáveis	**Classificadores**			
	$SOM - C1$	$SOM - C2$	$SOM - C3$	$SOM - C4$
M_x	16	3	17	17
M_y	16	3	17	17
α_0	0,6	0,6	0,5	0,4
α_T	0,001	0,001	0,001	0,001
σ_0	8	2	9	9
σ_T	0,001	0,001	0,001	0,001
R	300	300	300	300

Tabela 3.3 : parâmetros de treinamento para o banco de faces ORL.

Parâmetros Ajustáveis	**Classificadores**			
	$SOM - C1$	$SOM - C2$	$SOM - C3$	$SOM - C4$
M_x	13	3	14	12
M_y	13	3	14	12
α_0	0,3	0,6	0,3	0,2
α_T	0,001	0,001	0,001	0,001
σ_0	7	2	7	6
σ_T	0,001	0,001	0,001	0,001
R	300	300	300	300

Tabela 3.4 : parâmetros de treinamento para o banco de faces STIRLING.

Parâmetros Ajustáveis	**Classificadores**			
	$SOM - C1$	$SOM - C2$	$SOM - C3$	$SOM - C4$
M_x	16	5	13	14
M_y	16	5	13	14
α_0	0,5	0,4	0,7	0,2
α_T	0,001	0,001	0,001	0,001
σ_0	8	3	7	7
σ_T	0,001	0,001	0,001	0,001
R	300	300	300	300

Tabela 3.5 : parâmetros de treinamento para o banco de faces YALE-1.

Parâmetros Ajustáveis	**Classificadores**			
	$SOM - C1$	$SOM - C2$	$SOM - C3$	$SOM - C4$
M_x	12	3	14	15
M_y	12	3	14	15
α_0	0,3	0,5	0,3	0,3
α_T	0,001	0,001	0,001	0,001
σ_0	6	2	7	8
σ_T	0,001	0,001	0,001	0,001
R	300	300	300	300

Expostas as condições de treinamento adotadas, as Tabelas 3.6 a 3.10 apresentam alguns indicadores de desempenho obtidos para os classificadores descritos neste capítulo. O objetivo primeiro é avaliar a capacidade discriminatória dos classificadores baseados na rede SOM em reconhecer os indivíduos, quando suas novas imagens de faces são apresentadas na fase de teste. Nestas Tabelas, encontram-se as taxas de acerto média, mínima e máxima e o coeficiente de variação (cv), todos expressos em percentagem.

O coeficiente de variação é uma medida de variabilidade relativa (BOSLAUGH; WATTERS, 2008). Este é calculado dividindo-se o desvio-padrão (sd) da taxa de acerto pelo seu valor médio (md), ou seja

$$cv = 100 \times \frac{sd}{md} \ (\%). \tag{3.20}$$

Numa comparação entre classificadores que apresentam desempenhos médios equivalentes, quanto menor o valor de cv, melhor é o classificador. Neste sentido, o coeficiente de

variação contribui para indicar o quanto varia a capacidade discriminativa do classificador sob análise, visto que as condições de treinamento mudam de uma realização para outra ou, em outras palavras, a confiança de que classificador apresente uma taxa de acerto próxima do seu valor médio.

Tabela 3.6 : desempenho obtido para as faces BARTLETT.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
SOM-C1	81,58	55,00	100,0	0,60	9,49
SOM-C2	93,71	80,00	100,0	0,34	6,22
SOM-C3	92,53	70,00	100,0	0,42	7,00
SOM-C4	78,03	55,00	85,00	0,39	8,00

Com relação aos resultados mostrados na Tabela 3.6, vale lembrar que o conjunto de faces BARTLETT apresenta poucos exemplos por classe. Mesmo sob esta condição, duas arquiteturas apresentaram elevadas taxas de acerto e pode-se dizer que, dentre estas, o classificador SOM-C2 apresentou melhor desempenho se comparado aos demais.

Tabela 3.7 : desempenho obtido para as faces CMU.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
SOM-C1	97,43	90,00	100,0	0,07	2,72
SOM-C2	97,70	90,00	100,0	0,09	3,07
SOM-C3	98,10	90,00	100,0	0,06	2,50
SOM-C4	96,13	85,00	100,0	0,11	3,45

Pelos resultados mostrados na Tabela 3.7, pode-se dizer que o conjunto de faces CMU não apresenta dificuldades para os classificadores baseados na rede SOM, visto que todos atingem facilmente taxas de acerto acima de 96%. Embora os desempenhos sejam muito próximos, pode-se afirmar que o classificador SOM-C2 apresenta melhor desempenho se comparado aos demais.

Tabela 3.8 : desempenho obtido para as faces ORL.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
SOM-C1	87,60	72,50	100,0	0,34	6,66
SOM-C2	91,98	75,00	100,0	0,25	5,44
SOM-C3	93,54	80,00	100,0	0,18	4,54
SOM-C4	78,39	60,00	95,00	0,39	7,97

Analisando os resultados da Tabela 3.8, o classificador SOM-C3 se destaca como o melhor para reconhecer os indivíduos no conjunto de faces ORL.

Tabela 3.9 : desempenho obtido para as faces STIRLING.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
SOM-C1	75,86	58,33	91,16	0,43	8,64
SOM-C2	83,56	69,44	94,44	0,27	6,22
SOM-C3	86,36	69,44	100,0	0,26	5,90
SOM-C4	82,33	59,11	100,0	0,55	9,00

De acordo com os resultados mostrados na Tabela 3.9 percebe-se claramente que o banco de faces STIRLING foi um dos mais desafiadores, visto que nenhum dos classificadores obteve desempenho médio acima de 90%, muito embora as taxas obtidas possam ainda ser consideradas elevadas. Os classificadores com melhores desempenhos foram os classificadores SOM-C2 e SOM-C3, com destaque ao desempenho deste último.

Tabela 3.10 : desempenho obtido para as faces YALE-1.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
SOM-C1	90,15	60,00	100,0	0,99	11,04
SOM-C2	96,37	86,66	100,0	0,12	3,60
SOM-C3	96,15	80,00	100,0	0,17	4,30
SOM-C4	87,46	40,00	100,0	1,77	15,21

Para o banco de faces YALE-1, os melhores classificadores foram SOM-C2 e SOM-C3, conforme mostrado na Tabela 3.10. Os coeficientes de variação destes dois classificadores demonstram quão mais estáveis são suas capacidades discriminatórias se comparadas às demais.

Um meio bastante útil de visualizar os resultados das tabelas de desempenho dos classificadores baseados na rede SOM consiste no uso de diagramas de caixa (*boxplots*). A Figura 3.4 mostra os vários diagramas para cada um dos classificadores para cada banco de faces (items de a a e). Neles pode-se observar de pronto os desempenhos atípicos (i.e. *outliers*) e o intervalo entre o primeiro e terceiro quartis[4] desta distribuição de acertos. Os diagramas de caixa foram gerados no *Matlab*© através do comando *boxplot* da toolbox de Estatística.

[4]Assim como média, mediana e moda, os quartis são medidas de posição para uma dada distribuição de dados. Estas medidas dividem o conjunto em quatro partes iguais. O primeiro quartil (Q_1) significa que 25% dos dados estão à esquerda de Q_1, ou que 75% dos dados estão a sua direita. O terceiro quartil (Q_3) significa que 25% dos dados estão à direita de Q_3, ou que 75% dos dados estão a sua esquerda.

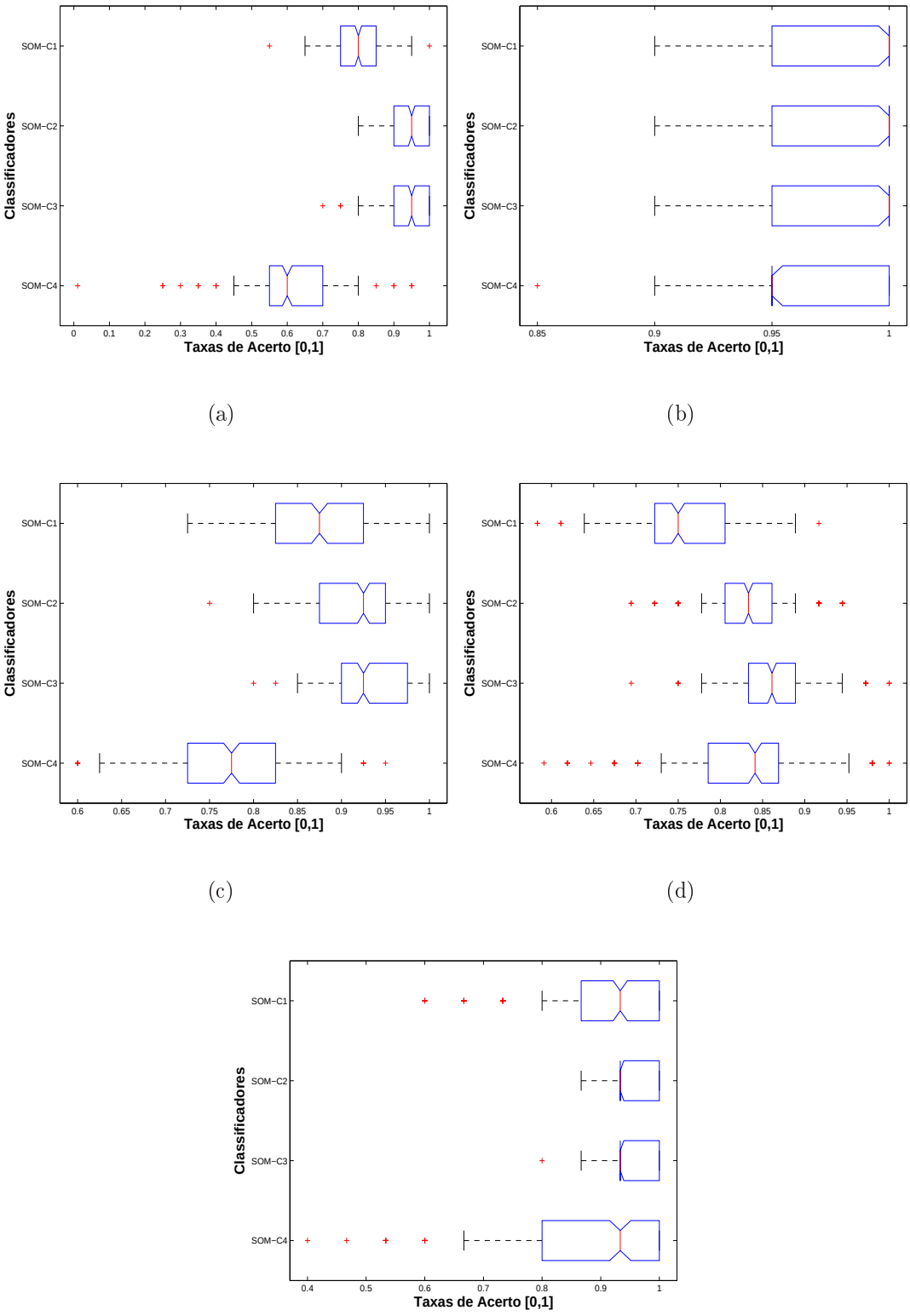


Figura 3.4 : diagramas de caixa correspondentes aos desempenhos dos classificadores baseados na rede SOM para os bancos de faces: (a) BARTLETT, (b) CMU, (c) ORL, (d) STIRLING e (e) YALE-1.

Analisando os gráficos da Figura 3.4, pode-se chegar às seguintes conclusões gerais.

Conjunto BARTLETT

- Exceto pelo classificador SOM-C4, todos os demais conseguiram em algum momento acertar todas as faces novas apresentadas durante a fase de teste.
- Apenas o classificador SOM-C4 apresentou uma assimetria interquartil em torno de sua tendência central ou mediana.
- Todos os classificadores apresentaram resultados tidos como *outliers*, exceto o classificador SOM-C2.
- Deste modo o classificador SOM-C2 foi o que apresentou melhor desempenho para este conjunto de faces.

Conjunto CMU

- Para este conjunto de faces todos os classificadores obtiveram o mesmo intervalo interquartil entre 95% e 100% de acerto. Disto pode-se inferir que discriminar estes indivíduos não é uma tarefa complexa para estas arquiteturas.
- Apenas o classificador SOM-C4 apresentou resultados tidos como *outliers*.
- Com base nestes resultados estes classificadores não apresentaram diferença significativa de desempenhos para este conjunto de faces.

Conjunto ORL

- Os três primeiros classificadores conseguiram, em algum momento acertar todas as faces apresentadas durante a fase de teste, fato que não foi conseguido pelo classificador SOM-C4.
- Exceto pelo classificador SOM-C1, todas as demais arquiteturas apresentaram taxas de classificação média pouco frequentes ou *outliers*.
- Os classificadores SOM-C2 e SOM-C3 apresentaram uma assimetria interquartil. O primeiro com enviesamento para a esquerda e o segundo para a direita.
- Os classificadores SOM-C1 e SOM-C4 apresentaram simetria interquartil. Entretanto, ambos apresentaram maiores dispersões que os demais.
- Pode-se concluir que o classificador SOM-C3 é superior aos demais pra discriminar os indivíduos deste conjunto de faces.

Conjunto STIRLING

- Apenas os classificadores SOM-C3 e SOM-C4 conseguiram, em algum momento acertar todas as faces apresentadas durante a fase de teste, fato que não foi conseguido pelos demais classificadores.
- Todos os classificadores apresentaram taxas de classificação médias tidas como *outliers*.
- Os classificadores SOM-C2 e SOM-C3 apresentaram uma simetria interquartil. Entretanto, ambos apresentaram menores dispersões que os demais.
- Os classificadores SOM-C1 e SOM-C4 apresentaram assimetria interquartil. O primeiro com enviesamento para a direita e o segundo para a esquerda.
- O classificador SOM-C3 apresenta, em termos médios, um desempenho superior aos demais.

Conjunto YALE-1

- Todos os classificadores conseguiram, em algum momento acertar todas as faces apresentadas durante a fase de teste.
- Apenas o classificador SOM-C2 não apresentou resultados de classificação médios tidos como *outliers*.
- Os resultados de classificação média apresentados por todos os classificadores obtiveram a mesma mediana, contudo, menores dispersões foram obtidas com os classificadores SOM-C2 e SOM-C3.
- Para este conjunto de faces, o classificador SOM-C2 é superior às demais arquiteturas.

3.3 Resumo do Capítulo

Neste Capítulo arquiteturas de classificadores baseadas na rede SOM foram apresentadas e suas capacidades de discriminar faces humanas sob diferentes configuração foram avaliadas. Muito embora, a rede SOM seja aplicada costumeiramente a problemas de natureza não-supervisionada, tais como visualização de dados, quantização vetorial e formação de agrupamentos, ela pode ser também utilizada como um classificador de padrões *per se*.

Tendo isso em mente, foram apresentadas quatro técnicas de projeto de classificadores baseados na rede SOM, que diferenciam-se entre si basicamente pela forma como os neurônios da rede são rotulados. Pode-se adiantar que, com os resultados a serem mostrados nos próximos capítulos, será verificado que classificadores baseados na rede SOM podem ser uma alternativa viável aos classificadores tradicionais (e.g. MLP, RBF e SVM).

No próximo Capítulo arquiteturas de classificação baseadas na rede Perceptron serão descritas e os classificadores apresentados serão avaliados quanto à tarefa de classificação de faces humanas para os mesmos conjuntos de faces apresentados no Capítulo 2.

4 *Redes Perceptron*

"O ato de classificar fez do macaco um homem."

Ernest Cushing Richardson

Neste Capítulo, descrevem-se de modo sucinto os classificadores baseados em redes do tipo *Perceptron* (HAYKIN, 1994). Em função da superfície de decisão, estes classificadores podem ser divididos em classificadores lineares (e.g. perceptron simples, madaline e perceptron logístico) e classificadores não-lineares (e.g. perceptron multicamadas, com uma e duas camadas escondidas).

Embora desde a proposição da regra de aprendizado perceptron em 1960 até a presente época, novas e importantes técnicas tenham surgido e, de certo modo, tenham ganho a preferência dos usuários quando se busca resolver problemas de classificação, ver-se-á neste Capítulo que, ao menos para os cenários de classificação utilizados, classificadores lineares, tal como o perceptron logístico (PL), podem alcançar uma capacidade de discriminação entre classes equivalente às de classificadores mais poderosos, tais como perceptron multicamadas (MLP), funções de base radial (RBF) e até máquinas de vetor-suporte (SVM).

Ao término da descrição das arquiteturas mencionadas no parágrafo anterior, seus desempenhos frente à tarefa de classificar faces humanas serão apresentados e discutidos. Com isto busca-se auxiliar o usuário na escolha de uma ou outra destas arquiteturas para resolver um problema similar de classificação.

4.1 Classificadores Lineares

4.1.1 Rede Perceptron Simples

A rede perceptron simples (PS) (ROSENBLATT, 1958) é formada por $p+1$ unidades de entrada $\{x_0, x_1, ..., x_p\}$ conectadas a uma camada com c neurônios artificiais, conforme representado na Figura 4.1. Embora não seja indicado nesta figura para não sobrecarregá-la, estas conexões são ponderadas através de um conjunto ajustável de parâmetros, conhecidos como pesos sinápticos. Ao k-ésimo neurônio da rede PS está associado o vetor de pesos $\mathbf{w}_k = [w_{k0}\ w_{k1}\ \cdots\ w_{kp}]^T$, $k = 1, ..., c$. Os pesos, juntamente com o bias do k-ésimo neurônio compõem o vetor $\mathbf{w}_k(t) \in \mathbb{R}^p$. O sobrescrito T indica o transposto de um dado vetor ou matriz e t indica a iteração corrente do algoritmo. A unidade de entrada x_0 tem valor constante igual a -1, logo o peso sináptico w_{k0} corresponde ao limiar de disparo do k-ésimo neurônio. O vetor de entrada $\mathbf{x} = [x_0\ x_1\ \cdots\ x_p]^T$, aplicado na iteração t corresponde a um padrão de entrada da rede.

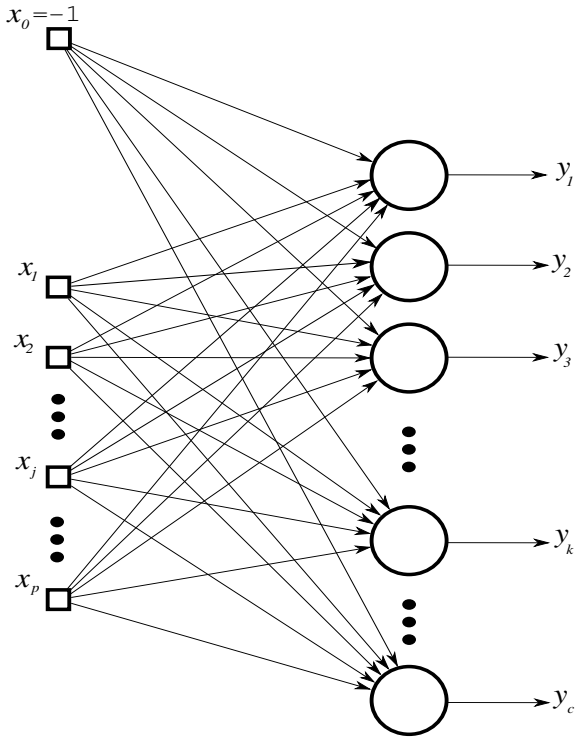


Figura 4.1 : arquitetura genérica de uma rede perceptron simples.

O treinamento da rede PS é feito de modo supervisionado. Durante esta etapa os rótulos numéricos, associados aos padrões de entrada, guiam o aprendizado da rede no sentido de ajustar os pesos da rede e fazer com que a mesma discrimine as classes apresentadas. Se as classes são linearmente separáveis, a rede converge para uma solução ótima

em um número finito de épocas (MINSKY; PAPERT, 1969). A Figura 4.2 apresenta com mais detalhes um neurônio artificial da rede PS. Com base nesta Figura, a saída e a regra de aprendizagem para o k-ésimo neurônio são dados por

$$y_k(t) = \text{sign}\left(\mathbf{w}_k^T(t)\mathbf{x}(t)\right) = \text{sign}\left(\sum_{j=1}^{p} w_{kj}(t)x_j(t) - w_{k0}(t)\right) \tag{4.1}$$

e

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \eta e_k(t)\mathbf{x}(t), \tag{4.2}$$

em que sign$(\cdot)$ é a função sinal, $x_j(t)$ é a j-ésima componente do vetor de entrada, $w_{kj}(t)$ é o peso que conecta a j-ésima entrada ao k-ésimo neurônio de saída, w_{k0} é o limiar de ativação do k-ésimo neurônio de saída e $0 < \eta < 1$ é a taxa de aprendizagem. Note que

$$e_k(t) = d_k(t) - y_k(t) \tag{4.3}$$

é o erro do k-ésimo neurônio para uma resposta (ou saída) desejada $d_k(t)$.

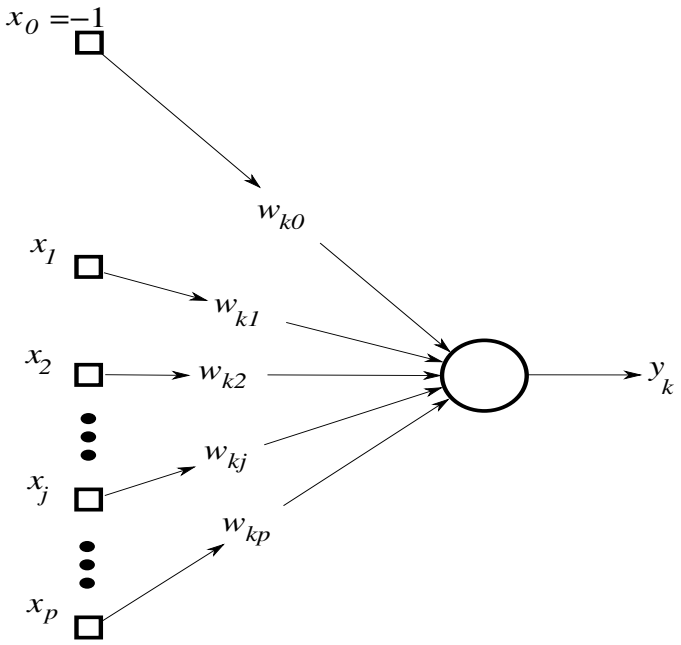


Figura 4.2 : diagrama de um neurônio da rede perceptron simples.

4.1.2 Rede Madaline

A rede MADALINE, termo este que significa *Multiple ADAptive LINear Element*, é composta por múltiplos neurônios do tipo ADALINE (WIDROW; LEHR, 1990). Esta rede possui a mesma topologia da rede perceptron simples anteriormente descrita. A única diferença está no fato de que a rede PS utiliza a função sinal como função de ativação de saída, enquanto a rede MADALINE usa a função identidade. Em termos genéricos, a saída e a regra de aprendizagem do k-ésimo elemento (ou neurônio) de saída deste classificador são dadas por

$$y_k(t) = \mathbf{w}_k^T(t)\mathbf{x}(t) = \sum_{j=1}^{p} w_{kj}(t)x_j(t) - w_{k0}(t) \tag{4.4}$$

e

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \eta e_k(t)\mathbf{x}(t). \tag{4.5}$$

A regra de aprendizagem da rede MADALINE, mostrada na (Eq. 4.5), é também conhecida como regra de Widrow-Hoff, regra delta, ou ainda regra LMS (*least-mean squares*). A principal diferença entre os classificadores PS e MADALINE reside no erro de saída $e_k(t)$ obtido, visto que na estrutura do primeiro este erro assume apenas três valores discretos (-2, 0 ou +2), enquanto que para a rede MADALINE esta variável é contínua, uma vez que esta rede usa função de ativação linear nas saídas dos neurônios.

4.1.3 Perceptron Logístico

O classificador Perceptron Logístico pode ser entendido como uma Rede Perceptron Simples em que a função sinal (não-linearidade *hard*) é substituída por uma função de ativação sigmoidal (não-linearidade $soft$). Neste trabalho, a função tangente hiperbólica[1] foi a adotada. Isto posto, a saída e a regra de aprendizagem do k-ésimo neurônio deste classificador são dadas por

$$y_k(t) = \tanh\left(\mathbf{w}_k^T(t)\mathbf{x}(t)\right) = \tanh\left(\sum_{j=1}^{p} w_{kj}(t)x_j(t) - w_{k0}(t)\right) \tag{4.6}$$

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \eta e_k(t)\left(1 - y_k^2(t)\right)\mathbf{x}(t), \tag{4.7}$$

em que $\tanh(\cdot)$ é a função tangente hiperbólica.

[1] Esta função é dada por:

$$y_k(t) \quad = \quad \frac{1 - \exp[-u_k(t)]}{1 + \exp[-u_k(t)]}$$

em que $u_k(t) = \mathbf{w}_k^T(t)\mathbf{x}(t)$.

Nota-se que a regra de aprendizagem da rede PL, como mostrada na (Eq.4.7), difere das demais já apresentadas pela inclusão do fator $1 - y_k^2(t)$. Este fator é a derivada da saída y_k em relação à ativação u_k no instante t. Um efeito negativo desta derivada é tornar o aprendizado mais lento, principalmente quando a saída do neurônio está nas regiões de saturação (i.e. quando $y_k(t) \approx \pm 1$). Um efeito positivo é tornar o aprendizado mais estável do ponto de vista da convergência para uma solução subótima.

4.1.4 Perceptron Logístico + MEKA

Uma alternativa para se treinar arquiteturas do tipo perceptron é o algoritmo de Kalman estendido múltiplo (*Multiple Extended Kalman Algorithm* - MEKA), proposto por Palmieri & Shah (1989).

Considerando-se o vetor de pesos $\mathbf{w}$ de todos os neurônios destas redes, a função custo que deve ser minimizada durante a etapa de treinamento é definida como

$$E_{av}(\mathbf{w}) = \frac{1}{2N} \sum_{t=1}^{N} \sum_{j \in C} [d_j(t) - y_j(t)]^2 \tag{4.8}$$

onde d_j é a saída desejada para o neurônio j, y_j a sua saída obtida, C inclui todas as saídas dos neurônios da rede e N é o número de vetores de treinamento.

Esta função custo, como pode ser vista, é dependente do vetor de pesos $\mathbf{w}$. Aplicando-se o algoritmo de Kalman estendido múltiplo a este problema, lineariza-se esta função custo em torno de cada ponto de operação. Com isto particiona-se o problema global em sub-problemas, um para cada neurônio da rede.

De modo genérico, quando aplicado ao classificador Perceptron Logístico, a saída e a regra de aprendizado para o k-ésimo neurônio desta rede são expressas por

$$\begin{aligned} y_k(t) &= \tanh\left(\mathbf{w}_k^T(t)\mathbf{x}(t)\right), & (4.9)\\ \mathbf{w}_k(t+1) &= \mathbf{w}_k(t) + \eta e_k(t)\mathbf{k}_k(t), & (4.10) \end{aligned}$$

em que o fator $\mathbf{k}_k(t)$ é chamado de ganho de Kalman, cujo o cálculo recursivo envolve as seguintes equações:

$$\begin{aligned} \mathbf{k}_k(t) &= \mathbf{r}_k(t)\left[1 + \mathbf{r}_k^T(t)\mathbf{q}_k(t)\right]^{-1}, & (4.11)\\ \mathbf{r}_k(t) &= \lambda^{-1}\mathbf{P}_k(t)\mathbf{q}_k(t), & (4.12)\\ \mathbf{q}_k(t) &= \left(1 - y_k^2(t)\right)\mathbf{x}_k(t), & (4.13)\\ \mathbf{P}_k(t+1) &= \lambda^{-1}\mathbf{P}_k(t) - \mathbf{k}_k(t)\mathbf{r}_k^T(t), & (4.14) \end{aligned}$$

tal que $\mathbf{P}_k(t)$ é uma estimativa da inversa da matriz de covariância de $\mathbf{q}_k(t)$, e λ ($0 <$

$\lambda \leq 1$) é chamado de fator de esquecimento. A (Eq. 4.14) é chamada de *equação de diferença de Riccatti*. De acordo com a regra MEKA, cada neurônio, ao possuir um conjunto próprio de variáveis $\mathbf{P}_k(t)$, $\mathbf{q}_k(t)$, $\mathbf{r}_k(t)$ e $\mathbf{k}_k(t)$, opera como um filtro de Kalman estendido local (SHAH; PALMIERI, 1990) que busca minimizar o seu erro quadrático médio, independente dos demais.

A principal vantagem do algoritmo MEKA sobre a regra padrão, expressa na Eq. (4.7) para o classificador (PL), é o aumento da velocidade de convergência. Contudo, vale ressaltar que isto ocorre às custas de um aumento considerável do custo computacional e de espaço em memória para seu treinamento (PALMIERI et al., 1991).

4.2 Classificadores Não-Lineares

Com o redescobrimento do algoritmo de retropropagação do erro (*error backpropagation*) em meados da década de 80, com destaque ao trabalho seminal de Rumelhart et al. (1986), foi possível comprovar empiricamente que redes perceptron multicamadas (MLP, sigla em inglês) são capazes de resolver problemas não-linearmente separáveis. Análises teóricas do poder discriminante de tais arquiteturas não custaram a aparecer (e.g. Hornik et al. (1989)), contribuindo para que a rede MLP passasse a figurar entre as arquiteturas neurais mais utilizadas em reconhecimento de padrões.

A Figura 4.3 ilustra a estrutura de uma rede perceptron multicamadas genérica que, como se observa, é uma extensão dos classificadores perceptron descritos anteriormente com a inclusão de camadas escondidas. Considerando-se as regras de atualização dos pesos descritos até então, um dos problemas encontrados no projeto destas redes foi o chamado *credit assignment problem* (MINSKY; PAPERT, 1969). Grosso modo, o problema consistia em como ajustar os pesos dos neurônios escondidos em função dos erros dos neurônios de saída. O algoritmo de retropropagação contornou esta dificuldade generalizando a regra de Widrow-Hoff para arquiteturas multicamadas (HAYKIN, 1994).

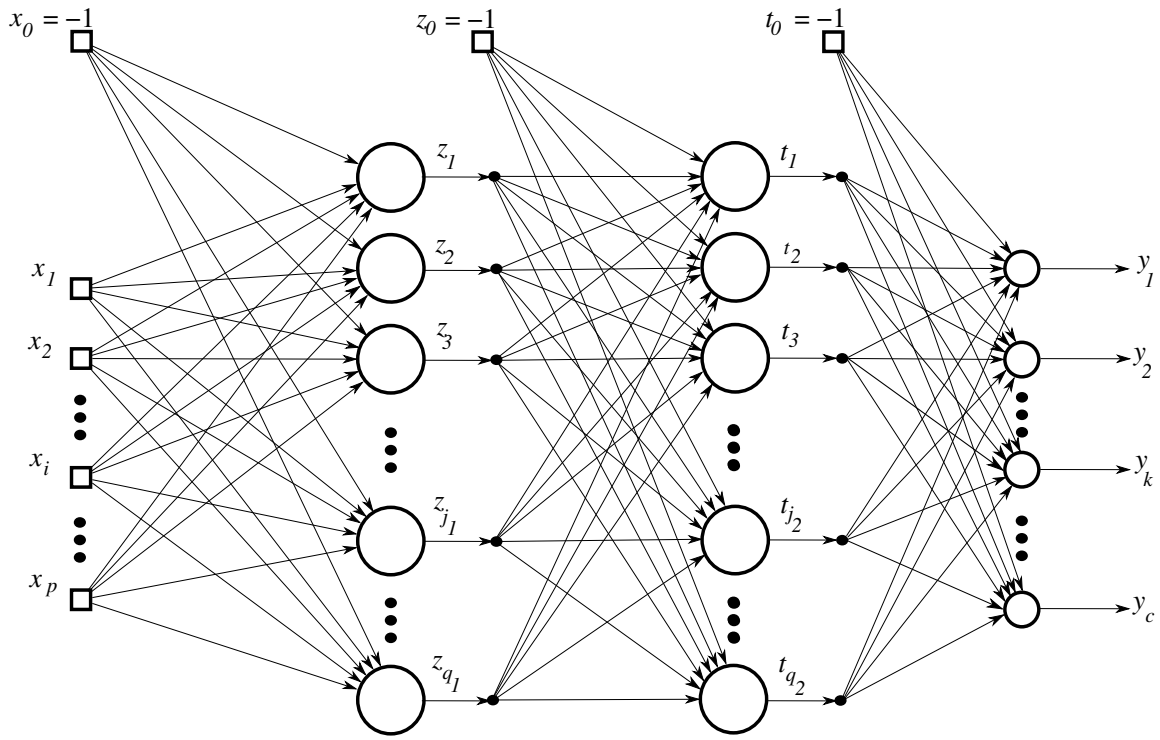


Figura 4.3 : arquitetura genérica de uma rede perceptron multicamadas.

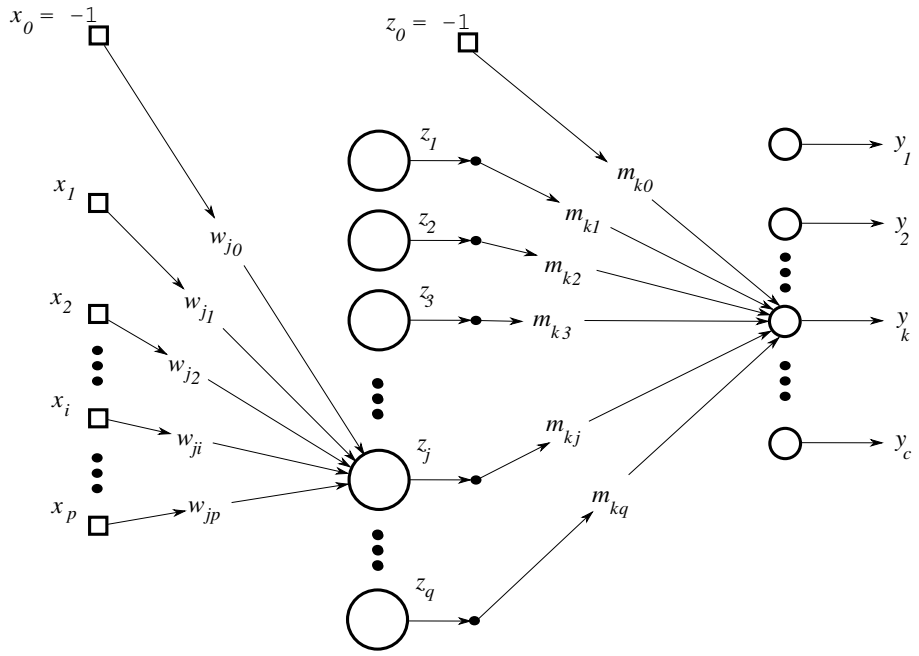
A Figura apresentada mostra uma rede totalmente conectada. É possível, entretanto, que muitas das conexões sejam desnecessárias ou redundantes. Neste sentido, é comum eliminar tais conexões durante ou após o treinamento da rede através de técnicas de poda de conexões (MEDEIROS; BARRETO, 2007; KARNIN, 1990), a fim de evitar problemas de sobreajuste (*overfitting*) da rede aos dados. Neste trabalho, optou-se por não utilizar tais técnicas e utilizar a rede MLP na sua forma mais convencional.

4.2.1 Rede MLP e o Algoritmo de Retropropagação de Erros

A rede perceptron multicamadas de uma camada escondida (MLP-1C) consiste simplesmente de um número p de unidades de entrada, que é igual ao número de atributos que compõem o vetor de entrada, uma camada escondida e outra de saída com q e c neurônios, respectivamente. Os neurônios escondidos e de saída fazem uso de funções de ativação sigmoidais, mais especificamente, da função tangente hiperbólica.

Pode-se dividir o treinamento desta rede em duas etapas. Na primeira, chamada de fase direta, o fluxo de informação decorrente da apresentação de um vetor de entrada é ponderado pelos pesos propagando-se da entrada para a saída. Na segunda etapa, chamada de fase inversa, os erros gerados pelos neurônios de saída da rede são retropropagados em direção à(s) camada(s) escondida(s) e às unidades de entrada do classificador.

As equações que definem as etapas direta e inversa do treinamento de redes MLP são apresentadas a seguir.



(a)

Figura 4.4 : fluxo direto para atualização dos pesos.

Sentido Direto

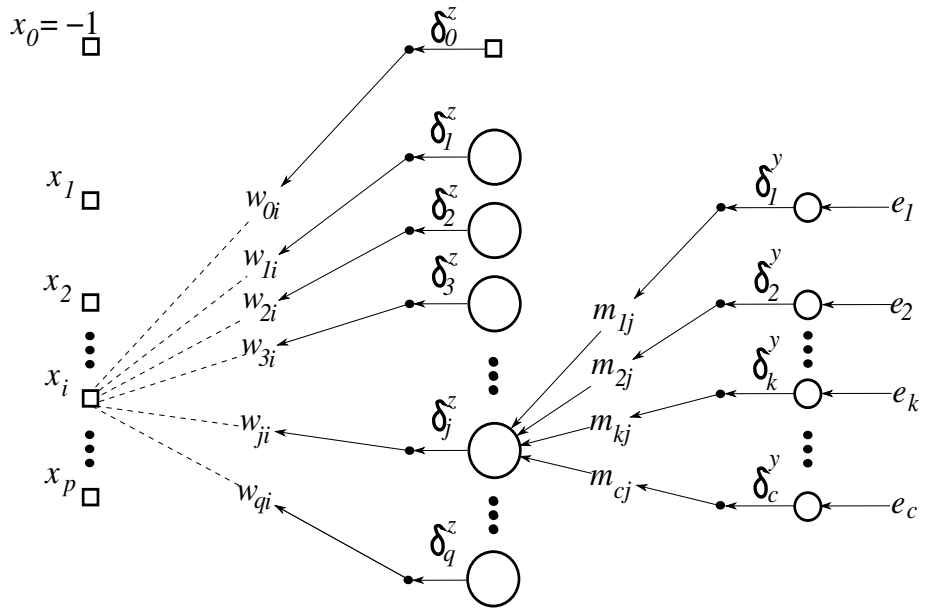
Considerando a Figura 4.4, tem-se que as saídas do j-ésimo neurônio escondido $z_j(t)$ e do k-ésimo neurônio de saída $y_k(t)$, $j = 1, \ldots, q$ e $k = 1, \ldots, c$, são dadas por

$$z_j(t) = \phi(u_j(t)) = \phi\left[\sum_{i=1}^{p} w_{ji}(t)x_i(t) - w_{j0}(t)\right], \tag{4.15}$$

$$y_k(t) = \phi(u_k(t)) = \phi\left[\sum_{j=1}^{q} m_{kj}(t)z_j(t) - m_{k0}(t)\right], \tag{4.16}$$

em que $x_i(t)$ é o i-ésimo componente do padrão de entrada atual, w_{ji} é o peso que conecta a i-ésima entrada ao j-ésimo neurônio escondido, m_{kj} é o peso que conecta o j-ésimo neurônio escondido ao k-ésimo neurônio de saída. Os parâmetros w_{j0} e m_{k0} são os limiares de ativação do j-ésimo neurônio escondido e do k-ésimo neurônio de saída, respectivamente. Os erros gerados pelos neurônios de saída da rede são então computados como

$$e_k(t) = d_k(t) - y_k(t), \qquad k = 1, \ldots, c. \tag{4.17}$$



(a)

Figura 4.5 : fluxo inverso para atualização dos pesos.

Sentido Inverso

De posse dos erros de saída no instante t, calculam-se os gradientes locais de todos os neurônios da rede, escondidos e de saída. A Figura 4.5 ajuda a visualizar o processo. Assim, para os neurônios de saída tem-se

$$\delta_k^y(t) = e_k(t)\phi'(u_k(t)), \qquad k = 1, \ldots, c. \tag{4.18}$$

O erro $e_k(t)$ é calculado na (Eq. 4.17) e a derivada $\phi'(u_k(t))$, para o caso em que $\phi(u_k(t))$ é a tangente hiperbólica, é dada pela seguinte expressão:

$$\phi'(u_k(t)) = \frac{d\phi(u_k(t))}{du_k(t)} = 1 - y_k^2(t). \tag{4.19}$$

Em seguida, determinam-se os gradientes locais dos neurônios escondidos, sendo estes determinados por

$$\delta_j^z(t) = \phi'(u_j(t)) \sum_{k=1}^{c} m_{kj}(t)\delta_k^y(t), \qquad j = 0, \ldots, q. \tag{4.20}$$

Após determinar todos os gradientes locais da rede MLP em questão, o passo seguinte é o ajuste dos pesos da mesma. Assim, para os pesos que conectam as unidades de entrada

com os neurônios escondidos, a regra de aprendizagem é dada por

$$\begin{aligned} w_{ji}(t+1) &= w_{ji}(t) + \Delta w_{ji}(t) \\ &= w_{ji}(t) + \alpha \delta_j^z(t) x_i(t) \end{aligned} \tag{4.21}$$

e para os pesos que conectam a camada escondida a de saída, a regra de aprendizagem é dada por

$$\begin{aligned} m_{kj}(t+1) &= m_{kj}(t) + \Delta m_{kj}(t) \\ &= m_{kj}(t) + \alpha \delta_k^y(t) z_j(t). \end{aligned} \tag{4.22}$$

Com isto, conclui-se uma iteração de treinamento. Note que este ciclo é, então, repetido para todos os padrões de entrada decorrentes do conjunto de faces até que se complete uma época de treinamento. Pode-se optar por treinar a rede MLP por um número fixo de épocas ou até que o erro quadrático médio, por época, atinja um valor mínimo pré-especificado. Durante a fase de teste, vetores de entrada representando faces não utilizadas durante o treinamento são apresentados à rede. Nesta fase, os pesos não são modificados, sendo apenas registradas as suas taxas de reconhecimento.

Certos cuidados, principalmente com relação à especificação do número de neurônios escondidos, devem ser tomados de modo a evitar que a rede "memorize" os dados de treinamento e tenha desempenho pífio durante o teste. Esta situação caracteriza sobreajustamento da rede aos dados de treino.

Neste trabalho, o número de neurônios escondidos foi determinado por experimentação, usando o número de neurônios sugeridos pela regra de Baum-Haussler (BAUM; HAUSSLER, 1989; HAYKIN, 1994) como sugestão inicial. De acordo com essa regra, o número de vetores de treinamento necessários para que a rede alcance uma boa generalização deve ser

$$N \geq \frac{32W}{\epsilon} \ln\left(\frac{32M}{\epsilon}\right) \tag{4.23}$$

em que $\ln(\cdot)$ representa o logaritmo natural, W é a quantidade de pesos sinápticos da rede, M é o número de neurônios ocultos e ϵ é o erro tolerado no teste.

Se o fator logarítmico for ignorado na (Eq. 4.23) vê-se que o número apropriado de vetores de treinamento é, em uma aproximação de primeira ordem, diretamente proporcional ao número de pesos da rede e inversamente proporcional à ϵ. Na prática, para uma boa generalização a condição adotada é

$$N > \frac{W}{\epsilon}. \tag{4.24}$$

Como exemplo, um erro aceitável de 10% no teste implica em um número de vetores

de treinamento de aproximadamente dez vezes o número de pesos sinápticos da rede.

O treinamento da rede MLP com duas camadas escondidas (MLP-2C) segue o mesmo princípio de retropropagar os erros dos neurônios de saída em direção à entrada, camada a camada, calculando-se os gradientes locais e atualizando os pesos correspondentes. Os procedimentos de treinamento e teste descritos para a rede MLP-1C valem também para a rede MLP-2C.

4.3 Resultados de Classificação

Nesta seção, os desempenhos dos classificadores descritos neste Capítulo são apresentados. Os conjuntos de faces foram utilizados sob as mesmas condições adotadas para os classificadores baseados na rede SOM. Assim, apenas dez épocas foram utilizadas para se treinar as redes PS, Madaline, PL, PL+MEKA e MLP-1C. A única exceção ocorreu para o classificador baseado na rede MLP-2C, para o qual foram utilizadas cinqüenta épocas, dada a convergência mais lenta deste em virtude de um número maior de parâmetros a ser estimado.

A fim de melhorar o desempenho dos classificadores foram adotadas taxas de aprendizado decrescentes durante o treinamento. O valor apresentado nas Tabelas é o valor inicial o qual, antes de ser utilizado nas equações de atualização dos pesos, era dividido pelo valor corrente da época. Deste modo, ao término do treinamento, o valor de α é um décimo de valor de α_0.

Para os classificadores multicamadas, os parâmetros q_1 e q_2 representam a quantidade de neurônios na primeira e segunda camada oculta, respectivamente. O parâmetro q_1, nestes casos, foi inicialmente escolhido como sendo $\lceil\sqrt{c \cdot p}\rceil$, em que $\lceil u \rceil$ retorna o menor inteiro maior que u, c é dimensão de saída do classificador e p é a sua dimensão de entrada. O valor escolhido foi obtido após avaliar o treino e teste de classificadores com q_1 próximo a este valor.

Para o classificador com duas camadas escondidas, os valores adotados de q_1 e q_2 foram encontrados após treino e teste com valores a começar por $q_1 = 20$ e $q_2 = 10$ e incrementando estes de uma unidade até os limites superiores de 65 e 40. Aqueles valores com os quais o classificador obteve as taxas de acerto mais elevadas foram os adotados.

O valor da taxa de aprendizado α_0 foi escolhido de modo que o aprendizado do classificador se desse de modo rápido, tendo em vista a escolha de dez épocas de treinamento apenas. Foram testados valores entre $0,1$ e $0,8$ para todas as redes exceto o classificador multicamadas MLP-2C para o qual foram testados valores entre $0,01$ e $0,1$ já que este demonstrou melhor aprendizado com um treinamento mais lento. O compromisso

entre poucas épocas e aprendizado fez com que para este classificador fossem adotados cinqüenta épocas.

Para os classificadores PS, Madaline, PL e PL+MEKA, entretanto, o parâmetro q_1 indica o número de neurônios de saída, ou seja, a quantidade de classes. Outro aspecto a se observar é que o classificador PL+MEKA, em particular, quando aplicado ao conjuntos STIRLING e YALE-1 apresentou uma alta variância e uma baixa taxa de acerto média para valores do parâmetro fator de esquecimento λ entre $0,96$ e $0,99$. Ao adotar valores entre $1,01$ e $1,05$ este comportamento não foi mais verificado.

Nas Tabelas 4.1 à 4.5 são mostrados os valores finais adotados para os parâmetros de treinamento dos classificadores.

Tabela 4.1 : parâmetros de treinamento para o banco de faces BARTLETT.

Parâmetros	**Classificadores**					
Ajustáveis	PS	$Madaline$	PL	$PL-MEKA$	$MLP-1C$	$MLP-2C$
q_1	20	20	20	20	25	23
q_2	–	–	–	–	–	15
α_0	0,4	0,4	0,4	0,4	0,2	0,05
λ	–	–	–	$0,99$	–	–
R	300	300	300	300	300	300

Tabela 4.2 : parâmetros de treinamento para o banco de faces CMU.

Parâmetros	**Classificadores**					
Ajustáveis	PS	$Madaline$	PL	$PL-MEKA$	$MLP-1C$	$MLP-2C$
q_1	20	20	20	20	40	35
q_2	–	–	–	–	–	20
α_0	0,4	0,4	0,4	0,4	0,2	0,05
λ	–	–	–	$0,99$	–	–
R	300	300	300	300	300	300

Tabela 4.3 : parâmetros de treinamento para o banco de faces ORL.

Parâmetros Ajustáveis	**Classificadores**					
	PS	$Madaline$	PL	$PL-MEKA$	$MLP-1C$	$MLP-2C$
q_1	40	40	40	40	60	45
q_2	–	–	–	–	–	25
α_0	0,4	0,4	0,5	0,4	0,2	0,02
λ	–	–	–	0,97	–	–
R	300	300	300	300	300	300

Tabela 4.4 : parâmetros de treinamento para o banco de faces STIRLING.

Parâmetros Ajustáveis	**Classificadores**					
	PS	$Madaline$	PL	$PL-MEKA$	$MLP-1C$	$MLP-2C$
q_1	36	36	36	36	65	50
q_2	–	–	–	–	–	30
α_0	0,4	0,4	0,3	0,3	0,2	0,01
λ	–	–	–	1,03	–	–
R	300	300	300	300	300	300

Tabela 4.5 : parâmetros de treinamento para o banco de faces YALE-1.

Parâmetros Ajustáveis	**Classificadores**					
	PS	$Madaline$	PL	$PL-MEKA$	$MLP-1C$	$MLP-2C$
q_1	15	15	15	15	25	23
q_2	–	–	–	–	–	15
α_0	0,4	0,4	0,4	0,4	0,2	0,03
λ	–	–	–	1,01	–	–
R	300	300	300	300	300	300

Definidas as condições de treinamento adotadas, as Tabelas 4.6 à 4.10 apresentam as taxas de desempenho obtidos para os classificadores descritos neste capítulo.

Tabela 4.6 : desempenho obtido para as faces BARTLETT.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
PS	65,50	30,00	100,0	1,91	21,10
MADALINE	83,90	70,00	95,00	0,52	8,60
PL	90,27	65,00	100,0	0,41	7,09
PL-MEKA	83,88	65,00	100,0	0,76	10,40
MLP-1C	96,08	80,00	100,0	0,16	4,16
MLP-2C	88,73	55,00	100,0	1,01	11,33

Nota-se na Tabela 4.6 que os classificadores PS, Madaline e Perceptrons Logísticos (PL e PL+MEKA) apresentaram taxas médias de acertos distintas mesmo sendo arquiteturalmente iguais. Este comportamento deve-se às funções de ativação que cada um utiliza, bem como às sutis diferenças entre suas regras de atualização dos pesos. Assim, para o conjunto de faces BARTLETT, o classificador PL é superior às demais redes lineares. Digno de nota é o fato do treinamento da rede PL com a técnica MEKA não ter produzido melhoras na capacidade discriminante desta arquitetura.

Com relação às redes MLP-1C e MLP-2C, aquela apresentou melhor desempenho do que esta. O desempenho do classificador MLP-2C foi equivalente ao do classificador PL. Dentre todos os classificadores testados, o classificador MLP-1C foi o que obteve a mais alta taxa média de acerto, bem como o menor coeficiente de variação.

Tabela 4.7 : desempenho obtido para as faces CMU.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
PS	96,76	80,00	100,0	0,16	4,13
MADALINE	98,97	92,50	100,0	0,03	1,75
PL	99,13	90,00	100,0	0,04	2,02
PL-MEKA	99,02	95,00	100,0	0,04	2,02
MLP-1C	99,45	95,00	100,0	0,02	1,42
MLP-2C	98,60	90,00	100,0	0,06	2,48

Com os resultados apresentados na na Tabela 4.7 para o conjunto de faces CMU não se pode identificar diferenças significativas de desempenho entre os classificadores sob análise (taxas médias de acerto). Situação similar foi observada no Capítulo 3, confirmando o fato de que este banco de faces não é muito desafiador. Entretanto, mesmo assim, pode-se ver que o desempenho da rede MLP-1C foi novamente o melhor (maior taxa média de

acerto e menor coeficiente de variação), seguido bem de perto pelos desempenhos dos classificadores PL e PL+MEKA.

Tabela 4.8 : desempenho obtido para as faces ORL.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
PS	75,90	57,50	95,00	0,56	9.86
MADALINE	93,64	80,00	100,0	0,24	5,23
PL	92,42	77,50	100,0	0,22	5,07
PL-MEKA	93,19	82,50	100,0	0,22	5,03
MLP-1C	95,90	85,00	100,0	0,10	3,30
MLP-2C	86,57	70,00	100,0	0,42	7,48

Para o banco de dados ORL, a Tabela 4.8 mostra que quatro classificadores obtiveram taxas médias de acerto superiores a 90% e, exceto para o classificador PS, todos os demais conseguiram, em algum momento, classificar todas as faces de teste. Embora as quatro redes de maior destaque (MADALINE, PL, PL-MEKA e MLP-1C) apresentem taxas médias de acerto próximas, o classificador MLP-1C pode ser considerado o melhor por apresentar a maior taxa de acerto e o menor coeficiente de variação.

Tabela 4.9 : desempenho obtido para as faces STIRLING.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
PS	65,12	41,67	86,11	0,70	12,85
MADALINE	83,95	63,88	97,22	0,35	7,05
PL	75,00	50,00	91,17	0,65	10,75
PL-MEKA	84,24	69,44	94,44	0,31	6,61
MLP-1C	88,63	77,78	97,22	0,21	5,17
MLP-2C	70,05	44,44	91,67	0,60	11,06

Para o conjunto de faces STIRLING, os resultados na Tabela 4.9 mostram que nenhum dos classificadores conseguiu, em alguma das realizações, discriminar todas as faces de teste apresentadas. Este fato demonstra que tal banco de faces é bastante desafiador. Novamente, a rede MLP-1C pode ser considerada o classificador de melhor desempenho. Um fato interessante de ser comentado é de que, para este banco de faces, a diferença entre os desempenhos dos classificadores PL e PL+MEKA é bem significativa em favor do último.

Tabela 4.10 : desempenho obtido para as faces YALE-1.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	*máxima*	*variância*	*cv*
PS	85,51	33,33	100,0	2,65	19,04
MADALINE	96,15	80,00	100,0	0,22	4,88
PL	96,88	73,33	100,0	0,25	5,16
PL-MEKA	94,35	53,33	100,0	0,79	9,42
MLP-1C	96,37	86,67	100,0	0,22	4,87
MLP-2C	89,71	46,67	100,0	2,32	16,98

Para o conjunto de faces YALE-1, os resultados da Tabela 4.10 mostram fatos curiosos. Primeiro que dois classificadores lineares, Madaline e PL, apresentaram desempenhos equivalentes ao desempenho da rede MLP-1C, tanto no quesito taxa média de acerto quanto no coeficiente de variação. Segundo que todos os classificadores, sem exceção, conseguiram em alguma realização classificar todos os vetores representantes das faces de teste.

Para complementar os resultados numéricos apresentados nas tabelas, a Figura 4.6 apresenta os diagramas de caixa das taxas médias de acerto dos classificadores da família perceptron.

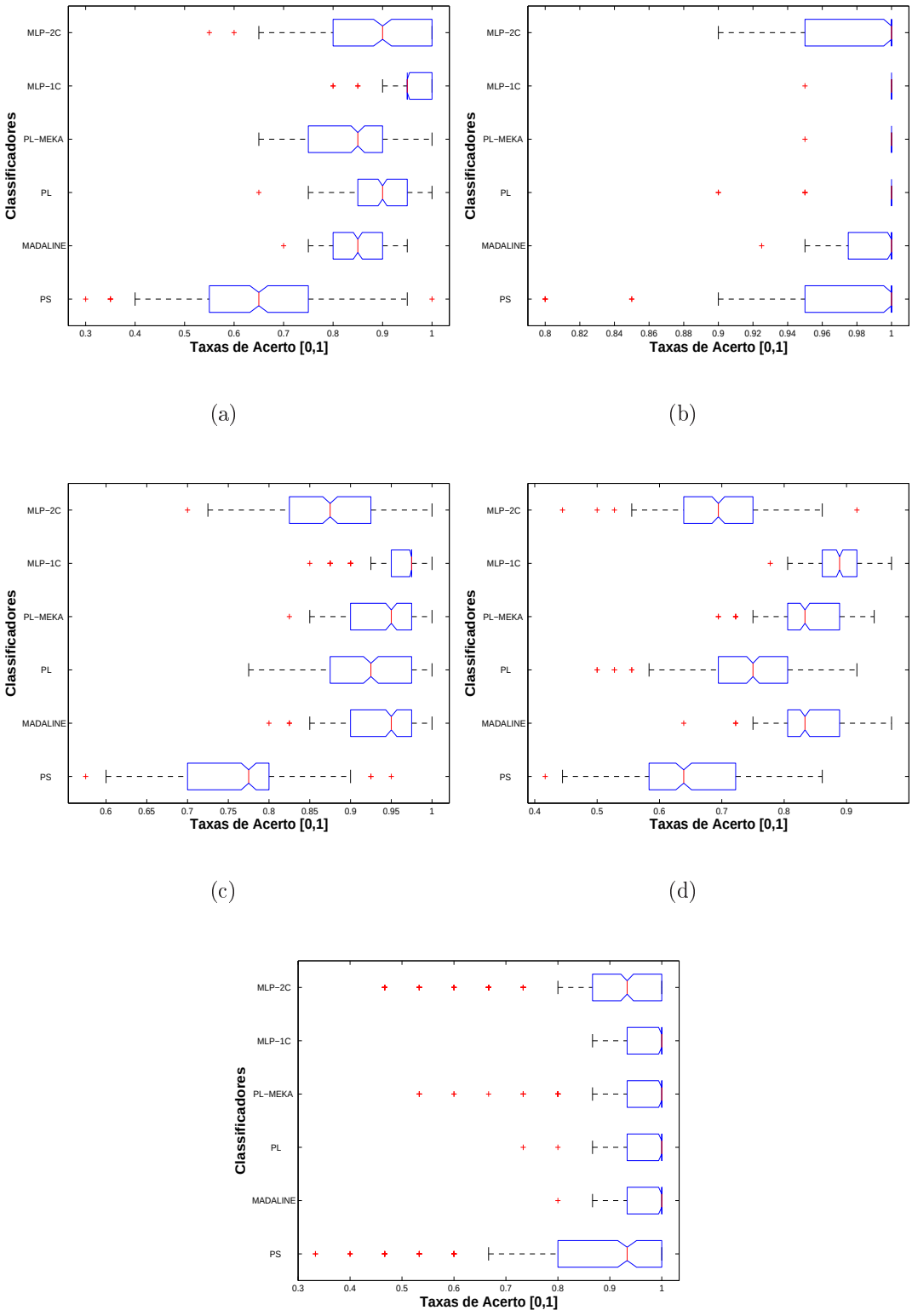


Figura 4.6 : diagramas de caixa correspondentes aos desempenhos dos classificadores da família perceptron para os bancos de faces: (a) BARTLETT, (b) CMU, (c) ORL, (d) STIRLING e (e) YALE-1.

Analisando os gráficos da Figura 4.6, pode-se chegar às seguintes conclusões gerais.

Conjunto BARTLETT

- Exceto pelo classificador Madaline, todos os demais conseguiram em algum momento acertar todos os vetores de faces novas apresentadas durante a fase de teste.
- Apenas os classificadores PL-MEKA e MLP-1C apresentaram uma assimetria interquartil. O primeiro com viés para a esquerda, ou seja, concentração de taxas de acerto inferiores à mediana e o segundo, em condição oposta ao primeiro, com viés para a direita.
- Apenas o classificador PL-MEKA não apresentou resultados tidos como *outliers* (i.e. valores excessivamente baixos e pouco freqüentes da taxa de acerto).
- Para este conjunto de faces, portanto, o classificador MLP-1C pode ser considerado o melhor. Sua distribuição interquartil foi a menor dentre as demais e apresentou apenas valores acima de 95%.

Conjunto CMU

- O classificador PS apresentou a taxa de acerto mais baixa se comparado com demais.
- Apenas o classificador MLP-2C não apresentou outliers.
- Pelos gráficos, os classificadores MLP-1C e PL-MEKA são equivalentes e são os melhores dentre os demais.

Conjunto ORL

- Apenas o classificador PS não conseguiu, em alguma realização treino-teste, acertar todos os vetores de faces apresentadas durante a fase de teste.
- Apenas o classificador PL não apresentou taxas de acerto de classificação atipicamente baixas (i.e. outliers).
- Somente os classificadores PL e MLP-2C apresentaram uma simetria interquartil.
- A assimetria interquartil dos demais foram todas com viés para a esquerda.
- Conclui-se que o classificador MLP-1C é superior aos demais por apresentar o menor intervalo interquartil e a maior mediana.

Conjunto STIRLING

- Todos os classificadores apresentaram taxas de classificação tidas como outliers.
- O classificador PS apresentou a menor taxa de acertos tida como mediana e a maior intervalo interquartil. É, portanto, a rede menos indicada para esta tarefa.
- Apenas os classificadores PS, Madaline e PL-MEKA apresentaram assimetria interquartil. Todos com viés à direita.
- O classificador MLP-1C é o melhor dentre os demais, pois apresenta o intervalo interquartil mais à direita no gráfico e com menor dispersão.

Conjunto YALE-1

- Todos os classificadores conseguiram, em alguma realização treino-teste, acertar todos os vetores de faces apresentadas durante a fase de teste.
- Apenas o classificador MLP-1C não apresentou taxas atipicamente baixas de classificação.
- Ao analisar-se apenas o intervalo interquartil, quatro classificadores são equivalentes: Madaline, PL, PL-MEKA, MLP-1C.
- Levando em conta principalmente a ausência de outliers, o classificador MLP-1C pode ser considerado superior aos demais para este banco de faces.

4.4 Resumo do Capítulo

Neste capítulo foram apresentados os classificadores baseados na arquitetura Perceptron. De um modo em geral, o classificador MLP com uma camada escondida apresentou um desempenho superior aos demais classificadores utilizados na tarefa de reconhecimento de faces mesmo com as restrições de projeto adotadas (poucas épocas de treinamento e elevada taxa de aprendizado). Já a rede MLP com duas camadas escondidas não apresentou desempenho equivalente. Isto pode ser justificado pelo número superior de parâmetros a estimar que este classificador possui, o que pode ter causado sobreajustamento.

A partir da análise dos resultados obtidos também foi possível constatar que mesmo os classificadores lineares podem atingir desempenhos equiparáveis aos dos não-lineares, não devendo ser preteridos de imediato pelo usuário para aplicações desta natureza. Em particular, se a aplicação envolver o embarque do classificador em dispositivos com limitada capacidade de processamento e memória, a escolha de arquiteturas lineares é sugerida pelo seu menor custo computacional.

No Capítulo 5 são descritas classificadores baseados em funções de kernel (núcleo), a saber, GRNN, RBF e SVM. Avaliam-se também os desempenho destes perante a tarefa de classificar faces humanas sob diferentes condições.

5 *Classificadores Baseados em Kernel*

"Penso que não cegamos, penso que estamos cegos.
Cegos que veem, cegos que vendo, não veem."

José Saramago

Neste Capítulo, os classificadores baseados em funções-núcleo (kernel), tais como GRNN, RBF e SVM são descritos e avaliados em relação às taxas de acerto obtidas para classificar faces humanas sob diferentes configurações.

As redes neurais RBF são também aproximadores universais de funções (PARK; SANDBERG, 1991), apresentando-se como alternativa aos classificadores do tipo Perceptron Multicamadas. O desempenho de redes RBF em discriminar faces humanas tem sido investigada em vários trabalhos (FEITOSA et al., 2000; HOWELL, 2001; ER et al., 2002). Na presente dissertação, busca-se ampliar estes estudos, tanto pelo uso de uma maior quantidade de conjuntos de faces, quanto pela comparação com outras arquiteturas baseadas em kernel, a saber, classificadores GRNN (*Generalized Regression Neural Network*) e SVM (*Support Vector Machines*).

O classificador SVM, mais recentemente, tem recebido considerável atenção por parte da comunidade de Reconhecimento de Padrões e Aprendizado de Máquinas, e vem sendo aplicado com sucesso a um número cada vez maior de problemas (ABE, 2005; SOTO, 2005). Isto se deve principalmente a uma filosofia de projeto que leva em consideração a minimização do erro estrutural e não apenas a minimização do erro empírico, como ocorre para as redes MLP e RBF. Uma breve introdução à teoria de SVM é feita neste Capítulo, juntamente com algumas metodologias para aplicá-las a problemas de classificação multiclasses.

5.1 Rede de Funções de Base Radial

No decorrer desta primeira seção algumas das características estruturais e funcionais da rede RBF são apresentadas. Busca-se com isto obter um grau de detalhamento suficiente para que o seu desempenho frente à tarefa de reconhecer faces humanas, sob diferentes condições, seja bem entendido.

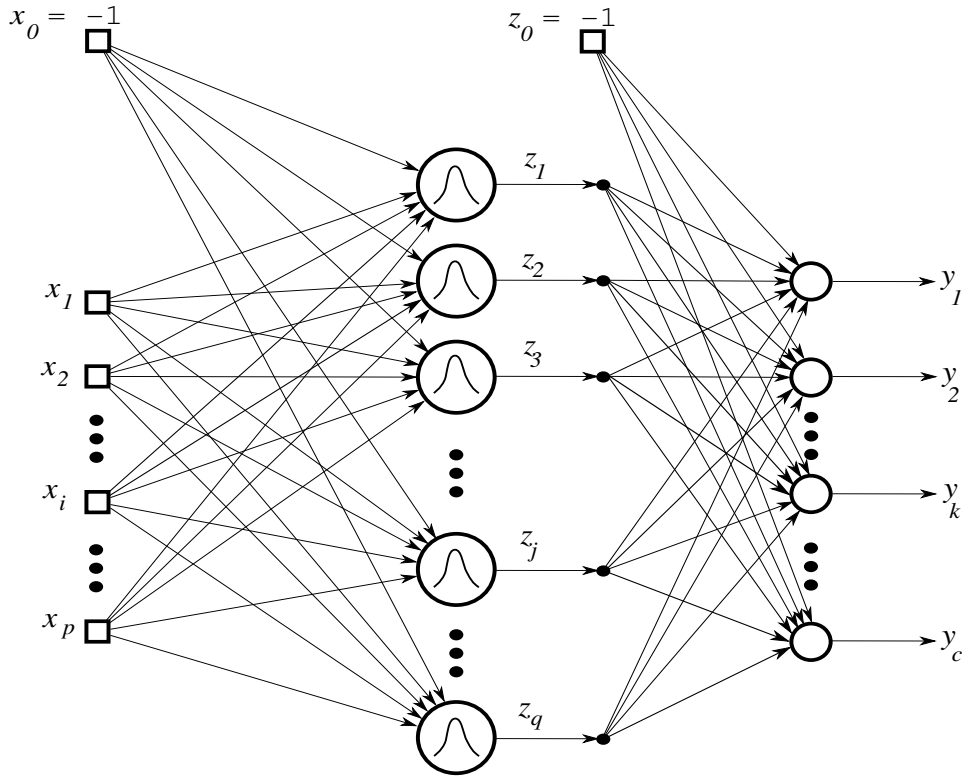


Figura 5.1 : arquitetura genérica da rede RBF.

Diferentemente da rede MLP, que pode apresentar uma ou duas camadas de neurônios ocultos, a rede RBF apresenta apenas uma camada oculta, além de uma camada de saída. A Figura 5.1 ilustra a arquitetura genérica de uma rede RBF. Os neurônios da camada oculta, doravante chamados de *funções de base radial*, possuem funções de ativação não-lineares, porém estas são bem diferentes das funções sigmoidais usadas na rede MLP. Já os neurônios da camada de saída possuem, em geral, função de ativação linear.

Para utilizar a rede RBF é preciso ter em mãos um número finito de N exemplos representados na forma de pares de vetores entrada-saída $(\mathbf{x}, \mathbf{d})$, tal que $\mathbf{x} \in \mathbb{R}^p$ e $\mathbf{d} \in \mathbb{R}^c$:

$$\begin{aligned} &\mathbf{x}(1), \quad \mathbf{d}(1) \\ &\mathbf{x}(2), \quad \mathbf{d}(2) \\ &\quad \vdots \quad\quad \vdots \\ &\mathbf{x}(N), \quad \mathbf{d}(N) \quad . \end{aligned} \tag{5.1}$$

Em geral, assume-se que estes vetores estão relacionados matematicamente por um mapeamento $\mathbf{F}(\cdot)$ desconhecido:

$$\mathbf{d}(t) = \mathbf{F}[\mathbf{x}(t)], \qquad t = 1, \ldots, N. \tag{5.2}$$

É justamente o mapeamento $\mathbf{F} : \mathbb{R}^p \rightarrow \mathbb{R}^c$ que se deseja conhecer. Uma maneira de se adquirir conhecimento sobre $\mathbf{F}(\cdot)$ é através dos dados disponíveis. Para isto, pode-se utilizar a rede RBF (assim como a rede MLP) para gerar uma *aproximação* de $\mathbf{F}(\cdot)$, denotada por $\hat{\mathbf{F}}(\cdot)$, tal que

$$\mathbf{y}(t) = \hat{\mathbf{F}}[\mathbf{x}(t)], \tag{5.3}$$

em que $\mathbf{y}(t)$ é a saída gerada pela rede que, espera-se, seja muito próxima da saída real $\mathbf{d}(t)$. Cada vetor de entrada é representado como

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_i(t) \\ \vdots \\ x_p(t) \end{pmatrix}, \tag{5.4}$$

em que uma interação $t = 1, 2, \cdots$, serve para indicar o instante de apresentação de um vetor de entrada à rede. O vetor de saída, representado como

$$\mathbf{d}(t) = \begin{pmatrix} d_1(t) \\ d_2(t) \\ \vdots \\ d_k(t) \\ \vdots \\ d_c(t) \end{pmatrix}, \tag{5.5}$$

é constituído pelas saídas desejadas associadas ao vetor de entrada atual. Ainda, x_i denota uma componente qualquer do vetor de entrada $\mathbf{x}$ e d_k denota uma componente qualquer

do vetor de saídas desejadas $\mathbf{d}$.

O vetor de pesos associado a cada função de base, também chamado de *centro* da j-ésima função de base, é representado como

$$\mathbf{c}_j^* = \begin{pmatrix} c_{j0}^* \\ c_{j1}^* \\ \vdots \\ c_{ji}^* \\ \vdots \\ c_{jp}^* \end{pmatrix} \tag{5.6}$$

em que c_{ji}^* é o peso que conecta a i-ésima unidade de entrada à j-ésima função de base. A Figura 5.2 ilustra as conexões de entrada da rede RBF.

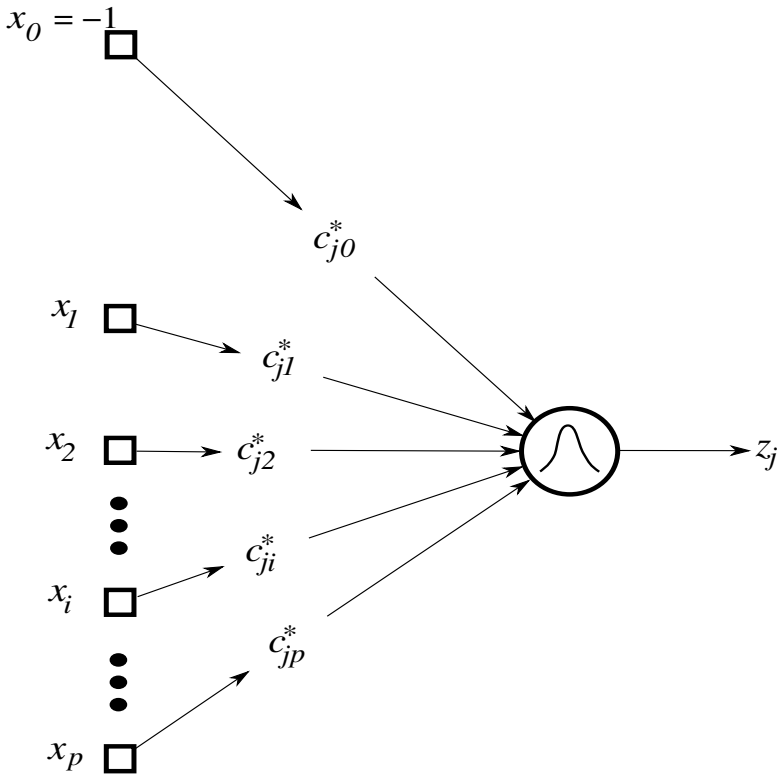


Figura 5.2 : detalhe das conexões de entrada da j-ésima função de base radial.

Assim como os neurônios ocultos da rede MLP, as funções de base não têm acesso direto à saída da rede RBF, ponto onde são calculados os erros de aproximação. Conecta-se a camada oculta com a de saída por um conjunto de pesos. Genericamente, o vetor de

pesos associado ao k-ésimo neurônio da camada de saída é representado como

$$\mathbf{m}_k = \begin{pmatrix} m_{k0} \\ m_{k1} \\ \vdots \\ m_{kj} \\ \vdots \\ m_{kq} \end{pmatrix} \tag{5.7}$$

A Figura 5.3, por sua vez, ilustra as conexões de saída da rede RBF.

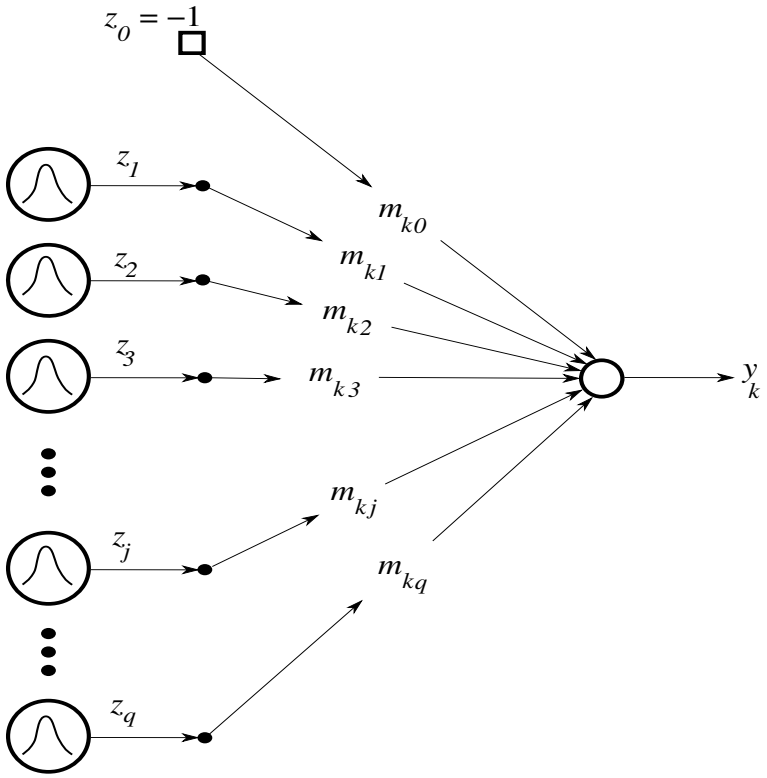


Figura 5.3 : detalhe de saída da rede RBF.

5.2 Projeto de uma Rede RBF

O projeto da rede RBF envolve basicamente a especificação do número de funções de base, a determinação dos centros e dos raios destes e determinação dos pesos dos neurônios de saída. Existem diversas maneiras de se projetar a rede RBF (TARASSENKO; ROBERTS, 1994; BAYKAL; ERKMEN, 2000; REDONDO et al., 2006), porém descreve-se neste trabalho o procedimento proposto por Moody e Darken (MOODY; DARKEN, 1989). Estes autores separam o treinamento de uma rede RBF em três etapas executadas sequencialmente.

Durante a primeira etapa, usa-se um algoritmo de formação de agrupamentos para encontrar os **centros** das funções de base. A segunda etapa trata do uso de métodos heurísticos para determinar o raio ou abertura (*spread*) de cada função de base. Por último, uma vez determinados os centros e os raios das diversas funções de base, pode-se computar os pesos dos neurônios de saída através da regra de aprendizagem do Perceptron Simples ou pelo método dos Mínimos Quadrados.

5.2.1 Descrição da Camada Oculta

Após a apresentação de um vetor de entrada $\mathbf{x}$ na iteração t, calcula-se a ativação da j-ésima função de base por meio da seguinte expressão

$$u_j(t) = \|\mathbf{x}(t) - \mathbf{c}_j^*(t)\|, \quad i = 1, \ldots, q, \tag{5.8}$$

em que q é o número de funções de base desta camada, e o vetor $\mathbf{c}_j^*$, mantido constante para o neurônio j, define o centro da j-ésima função de base.

A saída da j-ésima função de base é calculada por

$$z_j(t) = \phi(u_j(t)) = \exp\left\{-\frac{u_j^2(t)}{2\sigma_j^2}\right\}, \quad j = 1, \ldots, q, \tag{5.9}$$

em que σ_j denota o raio da j-ésima função de base, pois define a largura (abertura) da função de ativação gaussiana deste neurônio.

Note que, de acordo com a (Eq. 5.9), o neurônio j fornece resposta máxima, i.e. $z_j(t) \approx 1$, para vetores de entrada próximos do seu centro $\mathbf{c}_j^*$. Desta forma, diz-se que cada neurônio da camada escondida tem seu próprio **campo receptivo** no espaço de entrada, que é uma região centrada em $\mathbf{c}_j^*$ com tamanho proporcional a σ_j.

É comum normalizar a saída dos neurônios da camada escondida, tal que a soma de todas as saídas seja igual a 1, ou seja, $\sum_{j=1}^{q} z_j(t) = 1$. Para implementar esta normalização basta reescrever a (Eq. 5.9) como

$$z_j(t) = \frac{\phi_j(u_j(t))}{\sum_{l=1}^{q} \phi_l(u_l(t))} = \frac{\exp\left\{-\frac{u_j^2(t)}{2\sigma_j^2}\right\}}{\sum_{l=1}^{q} \exp\left\{-\frac{u_l^2(t)}{2\sigma_l^2}\right\}}. \tag{5.10}$$

5.2.2 Determinação dos Centros da Rede RBF

A forma mais comum de se determinar os centros $\mathbf{c}_i$ das funções de base é através de um algoritmo de formação de agrupamentos (*clustering*), sejam eles de origem não-neural, tais como K-médias ou LBG, ou de origem neural, tais como as redes SOM e Neural-Gas.

As redes RBF desta dissertação usam a rede SOM para estimar os centros das funções de base. O algoritmo da rede SOM aplicado à determinação dos centros da rede RBF é resumido a seguir.

Passo 1 - Definir o número de funções de base radial (q) e a taxa de aprendizagem ($0 < \eta < 1$).

Passo 2 - Iniciar com valores aleatórios os centros $\mathbf{c}_j^*$, $j = 1, \ldots, q$.

Passo 3 - Fazer $t = 1$.

Passo 4 - Enquanto $t \leq N$,

Passo 4.1 - Selecionar aleatoriamente (sem reposição) o vetor de entrada $\mathbf{x}(t)$.

Passo 4.2 - Determinar o índice do centro mais próximo de $\mathbf{x}(t)$, ou seja

$$j^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{c}_j^*(t)\|. \tag{5.11}$$

Passo 4.3 - Aplicar a seguinte regra de aprendizagem não-supervisionada:

$$\mathbf{c}_j^*(t+1) = \mathbf{c}_j^*(t) + \eta h(j^*, j; t)[\mathbf{x}(t) - \mathbf{c}_j^*(t)]. \tag{5.12}$$

Passo 4.4 - Fazer $t = t + 1$.

Passo 5 - Verificar se critério de parada do treinamento é atendido. Caso afirmativo, então finalizar treinamento. Caso contrário, retornar ao Passo 3.

5.2.3 Determinação do Raio da Função de Base

Uma vez que os centros das funções de base tenham sido determinados, o passo seguinte é determinar os raios (σ_j) das várias funções de base. Este parâmetro é de fundamental importância para o projeto da rede RBF. Se este raio for muito alto, existe um elevado grau de superposição entre os campos receptivos das funções de base, aumentando a suavidade da resposta da rede, porém diminuindo a sua precisão. Isto equivale a dizer que a rede generaliza demais (*underfitting*). Por outro lado, seste raio for muito pequeno, a superposição deixa de existir, porém precisão é elevada apenas para os casos em que $\mathbf{x}(t) \approx \mathbf{c}_j^*$. Neste caso, a rede não generaliza bem.

Dentre as várias técnicas para determinar σ_j, optou-se nesta dissertação por utilizar o seguinte procedimento:

- cada neurônio usa um raio próprio, σ_j, que tem seu valor definido como metade da distância entre o centro do neurônio j e o centro mais próximo, ou seja,

$$\sigma_j = \frac{d_{min}(\mathbf{c}_j^*, \mathbf{c}_i^*)}{2}, \quad \forall j \neq i, \tag{5.13}$$

em que $d_{min}(\mathbf{c}_j^*, \mathbf{c}_i^*) = \min_{\forall i \neq j}\{\|\mathbf{c}_j^* - \mathbf{c}_i^*\|\}$.

Outros procedimentos para a determinação de σ_j foram experimentados, porém aquele descrito na (Eq. 5.13) foi o que resultou no melhor desempenho para a rede RBF.

5.2.4 Descrição da Camada de Saída

Após a determinação dos centros e dos raios das funções de base, a terceira e última etapa consiste na determinação dos pesos dos neurônios de saída. Durante esta terceira etapa, os centros e os raios calculados nas duas etapas anteriores não tem seus valores alterados.

As abordagens mais comuns utilizam as regras de aprendizagem usadas por redes de uma camada, tais como a regra de aprendizagem do Perceptron Simples, a regra LMS (Adaline) ou ainda a regra Delta Generalizada (Perceptron Logístico). Todas estas regras de ajuste são recursivas, contudo é possível também optar por técnicas não-recursivas de estimação de parâmetros, e.g. método dos Mínimos Quadrados. A escolha de qual regra utilizar fica a cargo do projetista.

Para as redes implementadas nesta dissertação foram utilizados neurônios de saída com função de ativação linear. Assim, a saída do k-ésimo neurônio de saída é dada por

$$y_k(t) = \sum_{j=0}^{q} m_{kj}(t) z_j(t), \qquad k = 1, \ldots, c \tag{5.14}$$

em que c é o número de neurônios de saída. Note que as saídas das funções de base, $z_j(t)$, fazem o papel de entrada para os neurônios da camada de saída. Em seguida, os pesos do k-ésimo neurônio de saída são ajustados via regra LMS

$$m_{kj}(t+1) = m_{kj}(t+1) + \eta e_k(t) z_j(t), \qquad j = 0, ..., q, \tag{5.15}$$

em que $e_k(t) = d_k(t) - y_k(t)$ é o erro na saída do k-ésimo neurônio. Note que, para treinar a camada de saída, os vetores de entrada devem ser novamente apresentados à rede RBF por um certo número de épocas. A cada apresentação de um vetor de entrada, as saídas das funções de base (z_j) são calculadas pela (Eq. 5.10) e seus valores usados na (Eq. 5.15) para atualizar os pesos de saída (m_{kj}).

5.2.5 Rede GRNN

Outro classificador que foi utilizado nesta dissertação é o GRNN (SPECHT, 1991) (*Generalized Regression Neural Network*). Este classificador é um tipo de rede RBF que, no entanto, não necessita de uma etapa de treinamento. Ao se determinar o conjunto de vetores de treinamento, todos estes padrões são tomados como centros das q funções de base, ou seja

$$\mathbf{c}_j^* = \mathbf{x}(j), \qquad \forall j = 1, ..., N. \tag{5.16}$$

em que $\mathbf{x}(j)$ denota o j-ésimo vetor de treinamento e N é o número de padrões de treinamento disponíveis. Note que, nesta rede, tem-se $q = N$, em que q é o número de funções de base radial.

Por sua vez, os pesos de saída associados com a j-ésima função de base são definidos como as saídas desejadas associadas ao centro desta função de base, ou seja

$$m_{jk} = d_k(j), \qquad \forall j = 1, ..., N, \tag{5.17}$$

em que $d_k(j)$ é a k-ésima saída desejada associada ao j-ésimo padrão de entrada (que corresponde ao centro da j-ésima função de base.

Uma vez determinados os centros e os pesos dos neurônios de saída, o restante do projeto da rede GRNN envolve a especificação do tipo de função de base radial a ser utilizada. Para esta rede também foram utilizadas funções gaussianas como funções de base radial. Assim, a saída da j-ésima função de base da rede GRNN é calculada como na (Eq. 5.10). As larguras das funções de base, σ_j, $j = 1, ..., N$, foram determinadas pelo procedimento já descrito na Subseção 5.2.3.

5.3 Máquinas de Vetor Suporte

Máquinas de vetor suporte (*support vector machines - SVM*) são classificadores de padrões que se baseiam na teoria de aprendizado estatístico (VAPNIK, 1995, 1998) que, grosso modo, consiste na filosofia de projeto que leva em consideração a minimização do erro estrutural e não apenas a minimização do erro empírico[1], como ocorre para as redes MLP e RBF.

Antes de prosseguir, no intuito de melhor compreender classificadores SVM, deve-se ter em mente algumas definições importantes, a saber: generalização, dimensão VC e risco estrutural.

[1] Erro quadrático médio calculado para os vetores de treinamento.

- **Generalização** - Este termo, emprestado da psicologia, é usado aqui para qualificar o aprendizado do classificador. Sua capacidade de generalização é dita tão melhor quanto maior for a taxa de acerto para dados de teste (HAYKIN, 1994).

- **Dimensão VC** - É uma medida da capacidade ou poder de discriminação da família de funções que o algoritmo de aprendizado gerou após a etapa de treinamento. Em outras palavras, se um conjunto genérico contendo N vetores puder ser rotulado de 2^N modos diferentes e, para cada uma destas possibilidades, existir uma função, dentre as geradas pelo algoritmo, que possa discriminá-las corretamente diz-se, então, que este conjunto de vetores pode ser separado por este algoritmo e que a dimensão VC é, portanto, N.

- **Minimização do Risco Estrutural** - Esta é uma indução baseada no fato de a taxa de erro do algoritmo de aprendizagem sobre os dados de teste ser limitada pela soma da taxa de erro de treinamento e um termo que depende da dimensão VC (SHAWE-TAYLOR et al., 1998).

Definidos estes conceitos passa-se, nas linhas seguintes, a uma breve descrição da teoria básica que caracteriza as máquinas de vetor suporte.

5.3.1 Teoria Básica para SVM

Chama-se o hiperplano

$$\mathbf{w}_o \cdot \mathbf{x} + b_o = 0 \tag{5.18}$$

de ótimo se ele separa o conjunto de treinamento $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$ e se a margem entre o hiperplano e o vetor de treinamento mais próximo é máxima. Isto significa que o hiperplano ótimo tem que satisfazer as desigualdades

$$\mathbf{y}_i(\mathbf{w}_o \cdot \mathbf{x} + b_o) \geq 1, \qquad i = 1, \ldots, m. \tag{5.19}$$

e minimizar o funcional

$$R(\mathbf{w}) = \mathbf{w} \cdot \mathbf{w}. \tag{5.20}$$

Este problema de otimização quadrática pode ser resolvido no espaço dual dos multiplicadores de Lagrange (BAZARAA et al., 1993). Assim, constrói-se o lagrangiano da seguinte forma

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \tag{5.21}$$

e busca-se minimizá-lo com relação a $\mathbf{w}$ e b e maximizá-lo com relação aos multiplicadores

$$\alpha_i \geq 0, \qquad i = 1, \ldots, m. \tag{5.22}$$

Desse modo, ao se minimizar a (Eq. 5.21) com relação a $\mathbf{w}$ a b obtém-se, respectivamente, as equações

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{5.23}$$

e

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{5.24}$$

Substituindo-se a (Eq. 5.23) no Lagrangiano (Eq. 5.21) e considerando (Eq. 5.24), obtém-se o funcional

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \tag{5.25}$$

Ao se maximizar esta equação (Eq. 5.25) com relação ao parâmetro α e respeitando-se as restrições (Eq. 5.22) e (Eq. 5.24) obtém-se a solução ótima $\alpha^o = (\alpha_1^o, \alpha_2^o, \ldots, \alpha_m^o)$ a qual, por conseguinte, especifica os coeficientes para o hiperplano ótimo desejado

$$\mathbf{w}_o = \sum_{i=1}^{m} \alpha_i^o y_i \mathbf{x}_i \tag{5.26}$$

e

$$\sum_{i=1}^{m} \alpha_i^o y_i \mathbf{x}_i \cdot \mathbf{x} + b_o = 0, \tag{5.27}$$

em que b_o é escolhido de modo a maximizar a margem de separação hiperplano-vetor mais próximo. É importante ressaltar que a solução ótima satisfaz as condições de Kuhn-Tucker (BAZARAA et al., 1993)

$$\alpha_i^o [y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) - 1] = 0, \tag{5.28}$$

e supondo que $\alpha_i^o \neq 0$, tem-se que

$$y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) = 1. \tag{5.29}$$

Os vetores $\mathbf{x}_i$ que satisfazem (5.29) denominam-se, então, *vetores suporte*. E a norma do vetor $\mathbf{w}_o$ define a margem ρ entre o hiperplano de separação ótima e os vetores suporte

$$\rho = \frac{1}{\|\mathbf{w}_o\|}. \tag{5.30}$$

Portanto, levando-se em conta as equações (5.24) e (5.28), obtém-se

$$\frac{1}{\rho^2} = \mathbf{w}_o \cdot \mathbf{w}_o = \sum_{i=1}^{m} y_i \alpha_i^o \mathbf{w}_o \cdot \mathbf{x}_i = \sum_{i=1}^{m} y_i \alpha_i^o (\mathbf{w}_o \cdot \mathbf{x}_i + b_o) = \sum_{i=1}^{m} \alpha_i^o. \tag{5.31}$$

Por outro lado, para os casos de não-separabilidade do conjunto de treinamento, uma alternativa é a introdução de variáveis de flexibilização ξ_i, de modo que o funcional (5.20) assume a forma

$$R(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i, \tag{5.32}$$

em que C é um parâmetro de regularização. Sujeito às restrições

$$y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) \geq 1 - \xi_i, \tag{5.33}$$

$$\text{e} \quad \xi_i \geq 0, \tag{5.34}$$

o Lagrangiano deste problema assume a seguinte forma

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \nu_i \xi_i. \tag{5.35}$$

Nestas condições, deve-se minimizar a Eq. (5.35) com relação a $\mathbf{w}$, b e ξ_i e maximizá-lo com relação aos multiplicadores $\alpha_i \geq 0$ e $\nu_i \geq 0$.

Verifica-se que o resultado da minimização com relação a $\mathbf{w}$ e b conduz às restrições (5.23) e (5.24) e o resultado da minimização com relação a ξ_i implica na nova restrição

$$\alpha_i + \nu_i = C. \tag{5.36}$$

Considerando que $\nu_i \geq 0$, obtém-se

$$0 \leq \alpha_i \leq C. \tag{5.37}$$

Quando se utiliza (5.33) e (5.34) no Lagrangiano (5.35) tem-se que, para determinar o hiperplano ótimo, a maximização do funcional (5.25) deve respeitar as restrições (5.24) e (5.37).

Para o caso de não-separabilidade, as condições de Kuhn-Tucker

$$\alpha_i^o [y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) - 1 + \xi_i] = 0 \quad \text{e} \quad \nu_i \xi_i = 0 \tag{5.38}$$

devem ser satisfeitas. E, assim como ocorre para o caso anterior, os vetores $\mathbf{x}_i$, que correspondem aos α_i^o não nulos, são denominados vetores suporte. Neste caso, decorre que

$$y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) = 1 - \xi_i, \tag{5.39}$$

e, portanto, pelas condições (5.36) e (5.38) segue que se $\xi_i > 0$, então, $\nu = 0$ e $\alpha_i = C$. Neste ponto, pode-se distinguir entre dois tipos de vetores suporte: os vetores para os quais $0 < \alpha_i^o < C$ e aqueles para os quais $\alpha_i^o = C$.

Ao se projetar um classificador SVM, usualmente os vetores de entrada $\mathbf{x} \in \mathcal{X}$ são mapeados em um espaço aumentado ou espaço de características, $\phi(\mathbf{x}) \in \mathcal{F}$, com elevada dimensão onde se constroem os hiperplanos de separação ótima. O produto de dois vetores quaisquer neste espaço pode, então, assumir a forma generalizada

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j), \tag{5.40}$$

em que $k(\mathbf{x}_i, \mathbf{x}_j)$ é conhecida como função núcleo, ou simplesmente (*kernel*), que atende as condições de Mercer[2]. Na Tabela 5.1 algumas opções para a função de kernel são mostradas.

Tabela 5.1 : funções de kernel típicas.

$k(\mathbf{x}-\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2)$	Gaussiana RBF
$k(\mathbf{x}-\mathbf{y}) = (\|\mathbf{x}-\mathbf{y}\| + c^2)^{\frac{1}{2}}$	Multiquadrática
$k(\mathbf{x}-\mathbf{y}) = (\|\mathbf{x}-\mathbf{y}\| + c^2)^{-\frac{1}{2}}$	Multiquadrática Inversa
$k(\mathbf{x}-\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^{2n+1}$ $k(\mathbf{x}-\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^{2n} ln(\|\mathbf{x}-\mathbf{y}\|)$	Splines
$k(\mathbf{x},\mathbf{y}) = \tanh(\|\mathbf{x}\cdot\mathbf{y}\| - \theta)$	Tangente Hiperbólica
$k(\mathbf{x},\mathbf{y}) = (1 + \|\mathbf{x}\cdot\mathbf{y}\|)^d$	Polinomial de grau d
$k(x,y) = B_{2n+1}(x-y)$	B - splines
$k(x,y) = \frac{\sin(d+\frac{1}{2})(x-y)}{\sin\frac{(x-y)}{2}}$	Polinomial trigonométrico de grau d

5.3.2 Projeto de Classificadores SVM

A construção de um classificador SVM exige, em geral, que se resolvam problemas de otimização quadrática, os quais são fortemente dependentes do número e da dimensão dos vetores de treinamento (SMOLA et al., 2000). Contudo, como os conjuntos de imagens usados nesta dissertação são compostos por não mais que 400 exemplares cada e, após aplicar PCA, a dimensão dos vetores resultantes também se restringiu a pouco mais de 100 componentes (no máximo), nenhuma técnica adicional de aceleração da resolução de problemas de otimização foi usada. Aos interessados neste tipo de técnica de aceleração recomenda-se a leitura das referências Stitson & Weston (1996) e Osuna et al. (1997). Todos os problemas de otimização foram resolvidos com base nas implementações propostas por Gunn (1998) com o uso do método das restrições ativas (BAZARAA et al., 1993).

Outro aspecto importante referente à aplicação de SVM aos problemas de classificação aqui encontrados diz respeito ao fato deste classificador ter sido originalmente projetado

[2]Ser uma função definida positiva e simétrica (MERCER, 1909; AIZERMAN et al., 1964; BOSER et al., 1992).

para tarefas de classificação binária (CORTES; VAPNIK, 1995). Surge, então, mais um problema a superar: "como adequá-lo a problemas multiclasses?". Para contornar esta situação, no entanto, desenvolveram-se formulações que possibilitam aplicá-lo a problemas multiclasses (FERRIS; MUNSON, 2003; ABE, 2005) e, dentre estas, duas abordagens foram utilizadas nesta dissertação e serão descritas a seguir.

Abordagem Um Contra Todos (*One Against All* - OA)

Esta abordagem foi uma dos primeiras a tratar o problema de classificação multiclasses (BOTTOU et al., 1994). Ela caracteriza-se por construir c modelos SVM, em que c é o número de classes dos dados. No treinamento do i-ésimo classificador SVM, os exemplos da i-ésima classe recebem rótulos positivos ($+1$) enquanto que o restante dos dados recebem rótulo negativos (-1).

Deste modo, dado $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_m, y_m)\}$, onde $\mathbf{x}_i \in \mathcal{R}^n, i = 1, 2, \ldots, m$ e $y_i \in \{1, 2, \ldots, c\}$, o i-ésimo classificador SVM resolve o seguinte problema

$$\begin{aligned} \min_{\mathbf{w}^i, b^i, \xi^i} \quad & \frac{1}{2}(\mathbf{w}^i)^T\mathbf{w}^i + C\sum_{j=1}^{m}\xi_j^i, \quad \text{sujeito às seguintes restrições:} \\ & (\mathbf{w}^i)^T\phi(\mathbf{x}_j) + b^i \geq 1 - \xi_j^i, \qquad \text{se } y_j = i, \\ & (\mathbf{w}^i)^T\phi(\mathbf{x}_j) + b^i \leq \xi_j^i - 1, \qquad \text{se } y_j \neq i, \\ & \xi_j^i \geq 0, \qquad j = 1, 2, \ldots, m. \end{aligned} \tag{5.41}$$

Minimizar $\frac{1}{2}(\mathbf{w}^i)^T\mathbf{w}^i$ significa maximizar $2/\|\mathbf{w}^i\|$. Quando os dados não são linearmente separáveis, o termo $C\sum_{j=1}^{m}\xi_j^i$ busca reduzir o número de erros durante o treinamento.

Após resolver (5.41), dispõem-se de c funções de decisão

$$\begin{pmatrix} (\mathbf{w}^1)^T\phi(\mathbf{x}) + b^1 \\ (\mathbf{w}^2)^T\phi(\mathbf{x}) + b^2 \\ \vdots \\ (\mathbf{w}^i)^T\phi(\mathbf{x}) + b^i \\ \vdots \\ (\mathbf{w}^c)^T\phi(\mathbf{x}) + b^c \end{pmatrix}.$$

Diz-se então que $\mathbf{x}$ pertence a classe de cuja função de decisão apresente o maior valor, ou seja

$$\text{classe}(\mathbf{x}) \equiv \arg\max_{i=1,2,\ldots,c}((\mathbf{w}^i)^T\phi(\mathbf{x}) + b^i). \tag{5.42}$$

Abordagem Um Contra Um (*One Against One* - OO)

Esta abordagem foi proposta inicialmente por Knerr et al. (1990). Nela são construídos $c(c-1)/2$ classificadores em que cada um é treinado sobre duas classes. O problema que cada SVM resolve é formulado como

$$\begin{aligned} min_{\mathbf{w}^{ij},b^{ij},\xi^{ij}} \quad & \frac{1}{2}(\mathbf{w}^{ij})^T\mathbf{w}^{ij} + C\sum_{t}^{m}\xi_t^{ij} \\ (\mathbf{w}^{ij})^T\phi(\mathbf{x}_t) + b^{ij} &\geq 1 - \xi_t^{ij}, \qquad se \;\; y_t = i, \\ (\mathbf{w}^{ij})^T\phi(\mathbf{x}_t) + b^{ij} &\leq \xi_t^{ij} - 1, \qquad se \;\; y_t = j, \\ \xi_t^{ij} &\geq 0. \end{aligned} \tag{5.43}$$

Durante a etapa de teste, após os $c(c-1)/2$ classificadores serem obtidos, para cada vetor existe um contador que armazena o número de vezes que este foi classificado como pertencente a cada uma das classes existentes. Em outras palavras, se a função de decisão $(\mathbf{w}^{ij})^T\phi(\mathbf{x}) + b^{ij}$ indicar o vetor $\mathbf{x}$ como pertencente à i-ésima classe, o número de vitórias desta classe é incrementado, do contrário, incrementa-se o da j-ésima classe. Ao término deste processo, a estratégia de voto majoritário (Capítulo 3, Seção 3.2.2) é adotada e o vetor de teste é classificado como pertencendo a classe que apresentar maior número de vitória. Observe que caso ocorra empate entre classes, escolhe-se aleatoriamente uma destas para representar o vetor de teste.

Em termos comparativos, ao levar-se em conta que nesta dissertação cada classe de treinamento apresenta m/c vetores, esta abordagem exige que se resolvam $c(c-1)/2$ problemas de otimização quadrática, cada um com $2m/c$ variáveis. Na primeira abordagem, no entanto, tem-se c problemas de programação quadrática para se resolver com m variáveis cada.

5.4 Resultados de Classificação

Nesta seção, apresentam-se os desempenhos obtidos com os classificadores baseados em kernel descritos anteriormente. Antes, porém, de apresentar os resultados, algumas considerações de ajuste dos parâmetros de projeto desses classificadores são feitas a seguir.

Redes RBF

1. O treinamento das redes SOM para obtenção dos centros das redes RBF foi realizado utilizando-se apenas 10 épocas.

2. Em todos os experimentos utilizaram-se redes SOM-2D com mesmo número de neurônios em cada dimensão. A quantidade de centros para cada conjunto de faces foi o

seguinte: $q_{BARTLETT} = 25$, $q_{CMU} = 36$, $q_{ORL} = 49$, $q_{STIRLING} = 49$ e $q_{YALE-1} = 36$. Estes valores foram escolhidos após treino e teste com *grids* entre 2×2 e 10×10.

3. A taxa de aprendizado da rede SOM foi feita decrescente no tempo. Para todos os bancos de faces, o valor inicial foi fixado em $\eta_i = 0{,}2$ e o final em $\eta_f = 0{,}01$. O valor inicial foi escolhido após treino e teste realizados com valores deste parâmetro entre $0,1$ e $0,5$. Para o valor final não foram testados outros valores.

4. A largura da função vizinhança seguiu a mesma regra adotada para as redes SOM descritas no Capítulo 3.

5. A largura das funções de base gaussianas da rede GRNN foi, em todos os experimentos realizados, fixada em $\sigma = 3$, após alguma experimentação com os dados.

Redes SVM

1. Apenas duas funções kernel foram utilizadas, a saber, Polinomial de grau 1 e Gaussiana RBF (consultar Tabela 5.1 para outras possibilidades).

2. A nomenclatura adotada para cada uma das configurações é do tipo SVM-ftc, em que "f" indica a função kernel utilizada, ou seja, L (quando Polinomial) e G (quando Gaussiana), e "tc" indica o método de classificação adotado, isto é, OA (Um contra Todos) e OO (Um contra Um).

3. Em todas as implementações adotou-se $C = 10.000$.

4. Foram utilizados as implementações propostas por Gunn (1998).

A seguir, o desempenho dos classificadores, para cada um dos conjuntos de faces é apresentado e comentado. A fins de completude tem-se, ao lado do nome de cada classificador SVM, a quantidade de vetores suporte médios encontrados para as 300 realizações.

Tabela 5.2 : desempenho obtido para as faces BARTLETT.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
GRNN	77,92	55,00	90,00	1,46	15,51
RBF-SOM	94,80	90,00	100,0	0,21	4,83
SVM-LOO (47)	96,10	90,00	100,0	0,23	5,00
SVM-LOA (42)	91,15	80,00	100,0	0,54	8,06
SVM-GOO (45)	96,07	85,00	100,0	0,34	6,07
SVM-GOA (40)	94,82	90,00	100,0	0,09	3,16

Percebe-se na Tabela 5.2 que o classificador GRNN obteve a menor taxa de reconhecimento e o maior coeficiente de variação para o banco de faces BARTLETT, apresentando desempenho bem inferior aos demais. Todas os outros classificadores obtiveram taxas superiores a 90%. Com destaque para as redes RBF-SOM e SVM-GOA, consideradas as melhores neste caso, levando em conta os valores dos seus coeficientes de variação.

Tabela 5.3 : desempenho obtido para as faces CMU.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
GRNN	97,13	90,00	100,0	0,09	3,09
RBF-SOM	99,05	95,00	100,0	0,04	2,02
SVM-LOO (35)	98,07	90,00	100,0	0,07	2,70
SVM-LOA (34)	98,85	95,00	100,0	0,04	2,02
SVM-GOO (33)	98,55	95,00	100,0	0,05	2,27
SVM-GOA (31)	99,05	95,00	100,0	0,04	2,02

Ao analisar a Tabela 5.3, percebe-se facilmente que todos os classificadores obtiveram taxas de classificação superiores a 97% para o banco de faces CMU. Os classificadores RBF-SOM, SVM-LOA e SVM-GOA obtiveram resultados idênticos em termos do coeficiente de variação. Isto equivale, grosso modo, a dizer que seus desempenhos são equivalentes do ponto de visto estatístico. Neste caso, a escolha do usuário por um ou outro classificador pode ser balizada pelo custo implementacional de cada classificador.

Tabela 5.4 : desempenho obtido para as faces ORL.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
GRNN	68,32	60,00	77,50	0,30	8,02
RBF-SOM	92,31	82,50	100,0	0,37	6,59
SVM-LOO (61)	94,88	90,00	100,0	0,15	4,08
SVM-LOA (60)	95,83	87,50	100,0	0,17	4,30
SVM-GOO (58)	91,93	85,00	100,0	0,30	5,96
SVM-GOA (56)	97,11	92,50	100,0	0,07	2,72

Pelos resultados apresentados na Tabela 5.4 pode-se dizer que, para o conjunto de faces ORL, o método de classificação *Um contra Todos* confere às arquiteturas SVM superioridade quando comparadas ao método *Um contra Um*. Em particular, o classificador SVM-GOA apresentou o melhor para esta tarefa. Também digno de nota foi o fraco

desempenho do classificador GRNN, que apresentou taxas de acerto (média, mínima e máxima) bem inferiores aos demais.

Tabela 5.5 : desempenho obtido para as faces STIRLING.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
GRNN	75,95	66,67	83,33	0,23	6,31
RBF-SOM	89,10	83,33	94,44	0,19	4,89
SVM-LOO (63)	89,67	80,55	94,44	0,18	4,73
SVM-LOA (63)	87,43	80,55	94,44	0,18	4,85
SVM-GOO (60)	87,54	83,33	94,44	0,16	4,57
SVM-GOA (57)	91,41	86,11	94,44	0,08	3,09

A análise dos números mostrados na Tabela 5.5 permite inferir que reconhecer as faces do conjunto STIRLING é uma tarefa de complexidade maior que reconhecer as faces dos demais conjuntos discutidos até o momento. Apenas o classificador SVM-GOA conseguiu apresentar taxa média de acerto acima de 91%. Este resultado, conjugado a uma menor variância desta taxa, implicou no menor coeficiente de variação para o classificador SVM-GOA.

Tabela 5.6 : desempenho obtido para as faces YALE-1.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
GRNN	92,07	80,00	100,0	0,63	8,62
RBF-SOM	91,84	60,00	100,0	1,72	14,28
SVM-LOO (44)	94,91	80,00	100,0	0,48	7,30
SVM-LOA (40)	90,84	66,67	100,0	1,27	12,41
SVM-GOO (41)	94,53	80,00	100,0	0,43	6,94
SVM-GOA (37)	94,47	73,33	100,0	0,71	8,92

Conforme mostrado na Tabela 5.6, para o banco de faces YALE-1, todos os classificadores conseguiram apresentar desempenho satisfatório (i.e. taxa média acima de 90%). O melhor desempenho para este conjunto de faces foi obtido pelo classificador SVM-GOO. Vale ressaltar que o classificador GRNN superou algumas das outras arquiteturas, e.g. RBF-SOM e SVM-LOA, analisando apenas a taxa média de acerto. Considerando-se o coeficiente de variação, o classificador GRNN supera também o classificador SVM-GOA.

A título de melhor visualizar graficamente os desempenhos dos classificadores avalia-

dos neste capítulo, a Figura 5.4 apresenta os diagramas de caixa dos resultados apresentados por cada classificador ao longo das realizações de teste executadas. Comentam-se brevemente os resultados em seguida.

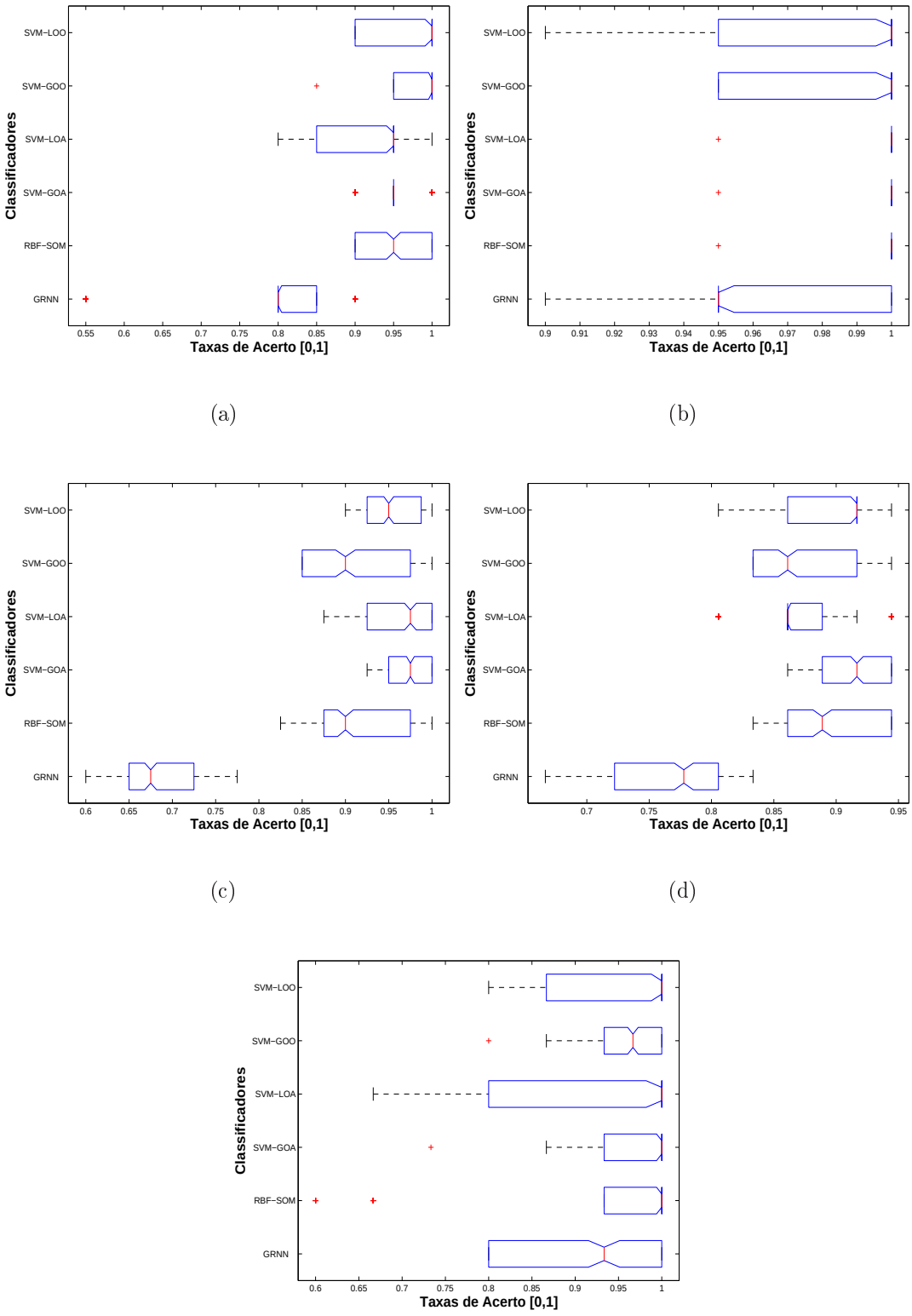


Figura 5.4 : diagramas de caixa correspondentes aos desempenhos dos classificadores baseados em kernel para os bancos de faces: (a) BARTLETT, (b) CMU, (c) ORL, (d) STIRLING e (e) YALE-1.

Conjunto BARTLETT

- Exceto os classificadores SVM lineares, todos os demais apresentaram taxas de acerto tidas como pouco freqüentes (i.e. *outliers*).
- Somente os classficadores RBF-SOM e SVM-GOA apresentaram simetria interquartil. Sendo que o último apresentou uma dispersão bem inferior à do primeiro.
- Para este conjunto de faces, o classificador SVM-GOO pode ser considerado o melhor em comparação aos demais, visto que sua distribuição interquartil foi a menor dentre as demais e com a mediana mais à direita.

Conjunto CMU

- O classificador GRNN apresentou desempenho bem inferior aos demais, mesmo para este conjunto, que é considerado pouco desafiador.
- Os classificadores SVM-LOA, SVM-GOA e RBF-SOM são os que apresentam melhores desempenhos, sendo equivalentes entre si.

Conjunto ORL

- Nenhum classificador apresentou taxas de classificação atípicas.
- O classificador GRNN é, com destaque, inferior aos demais quando aplicado a este banco de faces.
- Apenas o classificador SVM-GOA apresentou uma simetria interquartil.
- Por estar mais à direita no gráfico e apresentar a menor dispersão de taxas de acerto, o classificador SVM-GOA pode ser considerado o melhor para discriminar este banco de faces.

Conjunto STIRLING

- Mais uma vez o classificador GRNN demonstra um poder discriminante inferior aos demais.
- Somente o classificador SVM-GOA apresentou uma simetria interquartil.
- Apenas o classificador SVM-LOA apresentou taxas de acerto atípicas.
- O classificador SVM-GOA pode ser considerado o melhor para este conjunto de faces STIRLING, visto que possui uma baixa dispersão de taxas de acerto com valores mais à direita no gráfico.

Conjunto YALE-1

- Todos os classificadores conseguiram, em alguma realização treino-teste, acertar todos os vetores de faces apresentadas durante a fase de teste.
- As maiores dispersões interquartis foram características dos classificadores GRNN e SVM-LOA, enquanto que as menores foram dos classificadores RBF-SOM e SVM com kernels não-lineares.
- Por apresentar menor dispersão interquartil concentrada no extremo direito do gráfico, pode-se considerar o classificador SVM-GOO o mais adequado para classificar este conjunto de dados.

5.5 Resumo do Capítulo

Neste Capítulo apresentaram-se os classificadores baseados em kernel, da família RBF e SVM. O primeiro foi implementado utilizando a rede SOM para obtenção dos seus centros. Da mesma família, foi também apresentado o classificador GRNN. Com relação aos classificadores SVM, seus conceitos básicos foram descritos juntamente com discussões das técnicas para implementá-los. Todas estas arquiteturas foram aplicadas a problemas de classificação de faces e seus resultados comentados.

O classificador GRNN obteve desempenho geral inferior aos demais testados neste Capítulo. Este fato pode ser justificado devido a inexistência de treinamento neste algoritmo e elevada quantidade de funções de base. Em contrapartida, a rede RBF apresentou superiores taxas de acerto média para uma quantidade de kernels inferior em comparação ao GRNN. O método de cálculo do raio das funções de base pode também ter influenciado no mau desempenho do classificador GRNN.

Como conclusão geral a partir dos resultados obtidos, os classificadores SVM demonstraram poder discriminante relativamente superior às redes RBF. Mesmo suas versões com kernels lineares obtiveram elevadas taxas de acerto. A quantidade de vetores suporte encontrados com funções de base lineares foi, na maioria das vezes, superior aos encontrados com kernels gaussianos. Recomenda-se, no entanto, utilizar métodos de programação quadrática que acelerem a determinação dos vetores suporte quando a dimensionalidade dos dados for elevada, fato não característico deste problema após o uso do PCA.

No Capítulo 6, testes estatísticos de hipóteses serão realizados para comparar os melhores classificadores obtidos nesta dissertação. Serão comparados também a matriz de confusão de cada um deles.

6 *Resultados Adicionais*

"A sutileza do pensamento consiste em descobrir a semelhança das coisas diferentes e a diferença das coisas semelhantes."

Montesquieu

Neste Capítulo, estende-se a investigação da capacidade discriminante dos classificadores utilizados nesta dissertação. Isto se dá pela aplicação destas arquiteturas à imagens de teste com um grau de ruído associado.

Além disso, para cada um dos conjuntos de faces utilizados, os classificadores que apresentarem os melhores desempenhos serão comparados utilizando-se teste estatístico de significância. Esta comparação visa determinar o grau de similaridade estatística entre estes classificadores, tomados dois à dois. Analisa-se também as matrizes de confusão dos classificadores, a fim de julgar o grau de similaridade entre os desempenhos destes por classe.

Para fins de completude, foram ainda avaliados dois outros classificadores descritos no Apêndice A, a saber, Memória Associativa Linear Ótima (OLAM, sigla em inglês) e Classificador Gaussiano de Máxima Verossimilhança (MLG, sigla em inglês). Detalhes sobre os algoritmos destes classificadores e os resultados obtidos estão mostrados no referido Apêndice A.

Por fim, embora não seja este o foco principal desta dissertação, apresentam-se medidas preliminares acerca do custo computacional de todos os classificadores implementados.

6.1 Teste com Imagens Ruidosas

Nesta seção são apresentados a descrição do tipo de ruído adicionado às imagens de teste e os resultados de classificação para todas as arquiteturas empregadas neste trabalho.

6.1.1 Ruído Sal-e-Pimenta

Diferentes formas de ruído podem degradar a qualidade de uma dada imagem durante sua captura, transmissão ou processamento (BONCELET, 2005; GONZALEZ; WOODS, 2007). Optou-se, neste experimento, pelo uso do ruído sal-e-pimenta, também conhecido como ruído *spike*, e que pertence à classe dos ruídos impulsivos (CHAN et al., 2005).

As imagens com este tipo de ruído caracterizam-se por apresentarem uma certa quantidade de seus pixels brancos e pretos. Sua aplicação à imagens permite modelar condições defeituosas em sensores de câmeras ou erros de transmissão de bits por um canal ruidoso, por exemplo (SONKA et al., 1998).

Supondo-se que a probabilidade de que um pixel esteja corrompido é dada por

$$\rho_r \ , \qquad \text{tal que } \ 0 \leq \rho_r \leq 1,$$

introduz-se o ruído **sal e pimenta** em uma imagem com N_p pixels ajustando-se, aleatoriamente, duas frações

$$\frac{\rho_r}{2} \times N_p$$

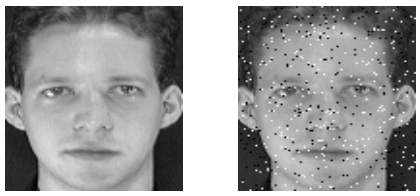
de seus pixels, uma para preto e outra para branco.

Com isto, nos experimentos a serem apresentados na Seção 6.1.2 foi adotado $\rho_r = 0,05$, ou seja, $2,5\%$ dos pixels de cada imagem de teste foram aleatoriamente ajustados para preto (recebendo o valor 255) e outros $2,5\%$ para branco (recebendo o valor 0).

A Figura 6.1 exemplifica o efeito da adição de ruído sal-e-pimenta às imagens presentes nos conjuntos de faces avaliados. É importante ressaltar que a escolha do ruído sal-e-pimenta em vez de, por exemplo, ruído gaussiano, deu-se porque a utilização de PCA usado na redução de dimensionalidade dos vetores pode *filtrar* ruído gaussiano uma vez que este tipo de ruído tem natureza aditiva.

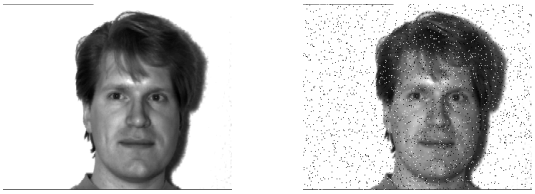
(a) BARTLETT

(b) CMU



(c) ORL

(d) STIRLING



(e) YALE-1

Figura 6.1 : exemplos de imagens sem ruído (à esquerda) e com ruído sal-e-pimenta.

6.1.2 Resultado de Classificação para Imagens Ruidosas

Nas tabelas seguintes são apresentados os resultados de classificação para imagens de teste com ruído. Destaca-se, em cada uma delas, o classificador com melhor desempenho em cada grupo. Vale ressaltar que, exceto pela adição de ruído às imagens de teste, tanto a metodologia descrita no Capítulo 2 quanto os ajustes dos classificadores se mantém inalterados para estes experimentos.

Tabela 6.1 : desempenho para as faces Bartlett com ruído.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
SOM-C1	47,20	30,00	65,00	0,57	16,04
SOM-C2	53,58	35,00	65,00	0,70	15,62
SOM-C3	53,55	20,00	75,00	1,00	18,67
SOM-C4	44,60	20,00	70,00	1,32	25,76
PS	27,40	0,76	50,00	0,81	32,85
MADALINE	50,71	25,00	70,00	1,27	22,22
PL	50,17	20,00	70,00	1,25	22,28
PL-MEKA	41,77	15,00	70,00	1,50	29,32
MLP-1C	75,00	54,48	95,00	0,62	10,50
MLP-2C	43,43	20,00	80,00	1,36	26,85
GRNN	48,12	25,00	60,00	1,68	26,94
RBF-SOM	49,02	25,00	65,00	1,80	27,37
SVM-LOO	57,07	50,00	65,00	0,47	12,01
SVM-LOA	53,28	35,00	65,00	1,57	23,52
SVM-GOO	59,22	50,00	65,00	0,23	8,10
SVM-GOA	58,73	40,00	70,00	1,21	18,73
OLAM	7,25	0,00	20,00	0,38	85,07
MLG	50,40	30,00	60,00	0,97	19,54

Para o conjunto de faces BARTLETT, os resultados mostrados na Tabela 6.1 indicam os classificadores SOM-C2, MLP-1C, SVM-GOO e MLG como os melhores em seus respectivos grupos de classificadores. No âmbito geral, pode-se dizer que a rede MLP-1C é a melhor em termos de taxa média de acerto mesmo apresentando um coeficiente de variação superior ao do classificador SVM-GOO.

Tabela 6.2 : desempenho para as faces CMU com ruído.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
SOM-C1	86,33	65,00	100,0	0,51	8,27
SOM-C2	83,33	55,00	100,0	0,57	9,06
SOM-C3	84,37	65,00	100,0	0,60	9,18
SOM-C4	83,91	60,00	100,0	0,58	9,12
PS	50,15	1,18	80,00	1,32	22,91
MADALINE	81,16	70,00	90,00	0,17	5,08
PL	79,11	55,00	95,00	0,47	8,67
PL-MEKA	78,33	65,00	95,00	0,51	9,12
MLP-1C	82,41	65,00	95,00	0,49	8,50
MLP-2C	56,80	20,00	90,00	1,30	20,07
GRNN	69,57	55,00	80,00	0,50	10,16
RBF-SOM	83,32	70,00	95,00	0,42	7,78
SVM-LOO	80,92	70,00	90,00	0,36	7,41
SVM-LOA	75,65	65,00	90,00	0,28	7,00
SVM-GOO	81,90	70,00	95,00	0,38	7,53
SVM-GOA	84,78	75,00	95,00	0,39	7,37
OLAM	3,15	0,00	15,00	0,15	122,95
MLG	80,35	65,00	90,00	0,34	7,27

Para o conjunto de faces CMU, os resultados mostrados na Tabela 6.2 indicam os classificadores SOM-C1, MLP-1C, SVM-GOA e MLG como os melhores em seus respectivos grupos de classificadores. Observa-se que apenas os classificadores baseados na rede SOM conseguem em algum momento discriminar todos os dados de teste. Com base nos valores da taxa média de acerto, pode-se dizer que a rede SOM-C1 é o classificador de melhor desempenho. Contudo, analisando os valores do coeficiente de variação, observa-se que o classificador MLG obteve o menor valor desta métrica entre todos os classificadores, muito embora a sua taxa média não seja uma das melhores. Um ponto a favor do classificador MLG é o seu baixo custo computacional em relação ao custo de implementação dos classificadores SOM, MLP, RBF e SVM. Deste modo, em muitas aplicações, talvez o classificador MLG fosse a escolha do usuário.

Tabela 6.3 : desempenho para as faces ORL com ruído.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
SOM-C1	50,33	35,00	67,50	0,53	14,46
SOM-C2	56,31	40,00	72,50	0,36	10,65
SOM-C3	55,20	40,00	70,00	0,44	12,02
SOM-C4	42,95	32,50	57,50	0,31	12,96
PS	14,00	0,26	30,00	0,28	37,80
MADALINE	49,51	32,50	62,50	0,45	13,55
PL	49,84	25,00	67,50	0,48	13,90
PL-MEKA	51,48	37,50	70,00	0,48	13,45
MLP-1C	62,44	45,00	80,00	0,45	10,74
MLP-2C	27,47	10,00	50,00	0,55	27,00
GRNN	26,60	20,00	32,50	0,11	12,47
RBF-SOM	44,89	25,00	70,00	1,59	28,09
SVM-LOO	62,25	55,00	75,00	0,50	11,36
SVM-LOA	55,02	42,50	67,50	0,62	14,31
SVM-GOO	32,73	20,00	50,00	0,55	22,66
SVM-GOA	64,53	52,50	70,00	0,34	9,04
OLAM	0,50	0,00	2,50	0,01	200,00
MLG	59,48	52,50	67,50	0,25	8,41

Para o conjunto de faces ORL, os resultados mostrados na Tabela 6.3 indicam os classificadores SOM-C2, MLP-1C, SVM-GOA e MLG como os melhores em seus respectivos grupos de classificadores, muito embora houve muitos resultados de classificação inferiores a 50%. Este fato indica o quanto o ruído adicionado prejudicou a classificação para este conjunto de vetores de faces. No contexto geral da tarefa, pode-se dizer que o classificador SVM-GOA é o melhor em termos de taxa média de acerto. Com relação ao coeficiente de variação, as mesmas observações feitas para o desempenho do classificador MLG para o banco de faces CMU com ruído valem para o banco de faces ORL.

Tabela 6.4 : desempenho para as faces STIRLING com ruído.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
SOM-C1	62,50	47,22	77,78	0,42	10,37
SOM-C2	61,01	41,67	77,78	0,41	10,50
SOM-C3	61,83	52,77	75,00	0,27	8,40
SOM-C4	53,50	35,00	68,33	0,60	14,48
PS	23,71	0,46	44,44	0,50	29,82
MADALINE	55,51	1,14	75,00	1,27	20,30
PL	47,26	22,22	66,67	0,52	15,26
PL-MEKA	58,38	38,88	72,22	0,49	12,00
MLP-1C	58,66	41,67	72,22	0,38	10,51
MLP-2C	20,92	2,78	44,44	0,42	30,98
GRNN	38,67	33,33	47,22	0,16	10,34
RBF-SOM	72,61	66,67	77,78	0,13	4,96
SVM-LOO	62,53	50,00	72,22	0,38	9,86
SVM-LOA	62,39	52,78	77,78	0,55	11,90
SVM-GOO	58,31	50,00	72,22	0,66	13,92
SVM-GOA	75,24	66,67	86,11	0,34	7,75
OLAM	10,15	8,33	13,89	0,05	22,00
MLG	54,20	47,22	66,67	0,29	9,94

Para o conjunto de faces STIRLING, os resultados mostrados na Tabela 6.4 indicam os classificadores SOM-C1, MLP-1C, SVM-GOA e MLG como as melhores em seus respectivos grupos de classificadores. No âmbito geral, pode-se dizer que o classificador SVM-GOA é o melhor em relação à taxa média de acerto, tendo este classificador também apresentado o segundo menor coeficiente de variação.

Tabela 6.5 : desempenho para as faces YALE-1 com ruído.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
SOM-C1	89,55	53,33	100,0	1,38	13,15
SOM-C2	92,13	73,33	100,0	0,50	7,67
SOM-C3	92,67	73,33	100,0	0,40	6,82
SOM-C4	84,80	46,67	100,0	2,18	17,42
PS	74,66	3,28	100,0	3,54	25,20
MADALINE	90,00	1,43	100,0	1,73	14,61
PL	92,17	60,00	100,0	0,54	7,97
PL-MEKA	87,11	33,33	100,0	2,24	17,18
MLP-1C	91,57	60,00	100,0	0,71	9,20
MLP-2C	82,17	26,67	100,0	2,66	19,85
GRNN	85,86	60,00	100,0	1,39	13,83
RBF-SOM	84,40	53,33	100,0	2,50	18,73
SVM-LOO	90,53	60,00	100,0	1,10	11,58
SVM-LOA	85,27	60,00	100,0	1,29	13,32
SVM-GOO	92,78	66,67	100,0	0,82	9,76
SVM-GOA	92,31	73,33	100,0	0,61	8,46
OLAM	55,00	0,00	80,00	8,26	52,25
MLG	90,33	66,67	100,0	1,04	11,30

Para o conjunto de faces YALE-1, os resultados mostrados na Tabela 6.5 indicam os classificadores SOM-C3, PL, SVM-GOA e MLG como os melhores em seus respectivos grupos de classificadores. No contexto geral, pode-se dizer que as redes PL e SVM-GOA são os melhores em termos de taxa média de acerto e coeficiente de variação. Outro aspecto relevante é o fato deste ruído, diferentemente do que ocorreu para os conjuntos anteriores, ter reduzido pouco a capacidade discriminante da maioria dos classificadores para este banco de faces.

Foram feitos outras combinações para treino e teste com ruído (treino com ruído/teste com ruído, treino com ruído/teste sem ruído) e seus resultados não serão apresentados. A tendência das taxas de acerto para estes casos foi

6.2 Testes de Hipóteses e Matrizes de Confusão

Nesta seção são comparados os melhores classificadores de cada grupo, tomados dois a dois, através de teste estatístico de hipótese baseado na distribuição t de *Student* e, para os casos em que os classificadores foram tidos como iguais, suas matrizes de confusão foram também analisadas no intuito de verificar o grau de similaridade entre as dificuldades de classificação por classe. Antes, porém, é feita uma breve descrição desta estatística de avaliação de classificadores.

6.2.1 O Teste de Hipótese Adotado

Os testes de hipótese adotados aqui utilizam a distribuição de probabilidade t de *Student*, desenvolvida por Willian Gosset (BOSLAUGH; WATTERS, 2008). Esta estatística, quando utilizada na comparação de dois conjuntos amostrais, é chamada de *Teste-T Pareado*. No âmbito desta dissertação, o teste-t pareado é aplicado aos melhores classificadores com o objetivo de determinar o grau de similaridade estatística entre eles. Para isto, duas hipóteses são consideradas, a saber:

- **Hipótese Nula (H_0)**: Não existe diferença significativa entre os desempenhos dos dois classificadores comparados. Neste caso, a escolha de uma destas arquiteturas de classificação pelo projetista deve ser feita com base em outros critérios de desempenho, tais como o tempo de computação e exigências de memória.
- **Hipótese Alternativa (H_A)**: Existe diferença significativa entre os classificadores. Neste caso o projetista deve supostamente dar preferência à de maior poder discriminante para a tarefa em questão[1].

Observe que estas hipóteses são mutuamente exclusivas e a aceitação de uma delas se dá com base na determinação da estatística t_d. Esta estatística é dada por:

$$t_d = \frac{m_1 - m_2}{\sqrt{\frac{\sigma_1^2}{R_1 - 1} + \frac{\sigma_2^2}{R_2 - 1}}}, \quad \text{se } R_1 \neq R_2, \tag{6.1}$$

ou

$$t_d = \frac{m_d}{\sqrt{\frac{\sigma_d^2}{R - 1}}}, \quad \text{se } R_1 = R_2, \tag{6.2}$$

em que m_d, σ_d^2 e R_i são, respectivamente, a diferença entre as taxas médias de acerto de cada classificador, a soma de suas variâncias e a quantidade de realizações executadas para a i-ésima amostra.

[1] Contudo, se questões como o custo computacional e espaço em memória são críticas, o projetista deve assumir o ônus pela escolha do classificador mais "leve", em detrimento do de melhor desempenho.

Após determinar-se t_d, compara-se este valor com o valor correspondente tabelado $t_{\frac{\alpha}{2},R-1}$ (ver Tabela B.1 no Apêndice B). Assim, tem-se as seguintes situações:

- Se $-t_{\frac{\alpha}{2},R-1} \leq t_d \leq +t_{\frac{\alpha}{2},R-1}$, aceita-se H_0.
- Caso contrário, rejeita-se H_0.

A Tabela 6.6 mostra as possíveis situações que podem advir do uso do teste-t pareado.

Tabela 6.6 : erros do tipo I e II.

	$\boldsymbol{H_0}$ **é verdadeira**	$\boldsymbol{H_1}$ **é verdadeira**
Aceita-se $\boldsymbol{H_0}$	Decisão Correta	Erro do Tipo II (β)
Rejeita-se $\boldsymbol{H_0}$	Erro do Tipo I (α)	Decisão Correta

Como se pode notar, existe a possibilidade de se tomar decisões erradas, haja vista a condição finita das amostras sob análise. Estas condições são as seguintes:

- Erro do tipo I (α) ou falso negativo - Quando se rejeita H_0 erroneamente.
- Erro do tipo II (β) ou Falso Positivo - Quando o oposto ocorre, ou seja, aceita-se H_0 erroneamente.

Utilizar tais indicadores de erro pode contribuir para a análise do problema de interesse (TESAURO et al., 1996; LOCHNER et al., 2001; GIOVAGNOLI et al., 2002; SINGH et al., 2008). Contudo, nesta dissertação, apenas $\frac{\alpha}{2}$ é tomado relevante. Restringindo-se apenas à representar o grau de significância deste teste, ou seja, a probabilidade de se cometer erros do tipo I. Posto de outra forma, se $\frac{\alpha}{2} = 0,025$, por exemplo, isto significa aceitar que há 5% de chance de se cometer erros deste tipo.

6.2.2 Teste-*t* Pareado Aplicado aos Melhores Classificadores

Nesta seção, são apresentados os resultados da comparação, por teste de hipótese, dos melhores classificadores obtidos para os cinco conjuntos de faces utilizados.

Foram adotados dois graus de significância para este estudo, a saber, $\frac{\alpha}{2} = 0,05$ e $\frac{\alpha}{2} = 0,01$, o que implica em $t_{0,05;299} = 1,645$ e $t_{0,01;299} = 2,326$ de acordo com a Tabela B.1. As comparações são feitas apenas para os resultados obtidos sem adição de ruído ao conjunto de teste.

Da Tabela 6.7 à 6.19, são apresentados os resultados comparativos por teste-t e matriz de confusão dos melhores classificadores para cada conjunto de faces. As matrizes de

confusão são empregadas para avaliar os resultados de "igualdade" estatística indicador pelo teste-t. Para não sobecarregar o texto, optou-se por apresentá-las de forma reduzida, i.e. mostrando apenas as classes nas quais houve algum erro de classificação.

Tabela 6.7 : teste-t para o conjunto de faces BARTLETT.

	SOM-C2	**MLP-1C**	**SVM-LOO**
MLP-1C	*-5,7960*		
SVM-LOO	*-5,4740*	*-0,0554*	
MLG	*3,0166*	*10,0188*	*9,3094*

Para o conjunto de faces BARTLETT, apenas os classificadores MLP-1C e SVM-LOO podem ser considerados iguais do ponto de vista estatístico, pois

$$-1,645 < t_d = -0,0554 < 1,645 \quad e \quad -2,326 < t_d = -0,0554 < 2,326.$$

Nas Tabelas 6.8 e 6.9 são mostradas as matrizes de confusão dos classificadores SVM-LOO e MLP-1C, respectivamente, para o banco de faces BARTLETT. Os valores nas tabelas são valores médios das realizações e estão em unidades de porcentagem (%).

Tabela 6.8 : matriz de confusão reduzida (SVM-LOO/BARTLETT).

Classes	Classes Desejadas			
Obtidas	1	8	13	20
1	80,0			
3				20,0
8		81,0		
11			19,0	
13			81,0	
17	20,0	19,0		
20				80,0

Tabela 6.9 : matriz de confusão reduzida (MLP-1C/BARTLETT).

Classes	Classes Desejadas			
Obtidas	1	5	8	18
1	83,0			
5		84,0		
8			94,0	
9				1,0
14		16,0		
17	17,0		6,0	
18				99,0

Reportando-se à Figura 2.1, que mostra os indivíduos do banco de faces BARTLETT, das Tabelas 6.8 e 6.9 pode-se inferir que os classificadores SVM-LOO e MLP-1C são diferentes sob a avaliação por matriz de confusão. Apenas 50% das confusões que ambos apresentam são equivalentes (confundem os indivíduos 01 e 08 com o indivíduo 17).

Tabela 6.10 : teste-t para o conjunto de faces CMU.

	SOM-C3	**MLP-1C**	**SVM-GOA**
MLP-1C	*-8,2532*		
SVM-GOA	*-5,1947*	*2,8237*	
MLG	*-1,8486*	*5,1947*	*2,7454*

Para o conjunto de faces CMU, os classificadores SOM-C3 e MLG podem ser considerados iguais estatisticamente para $\frac{\alpha}{2} = 0,01$, pois

$$-2,326 < t_d = -1,8486 < 2,326.$$

Nas Tabelas 6.11 e 6.12 são mostradas as matrizes de confusão dos classificadores SOM-C3 e MLG, respectivamente, para o banco de faces CMU.

Tabela 6.11 : matriz de confusão reduzida (SOM-C3/CMU).

Classes	Classes Desejadas											
Obtidas	2	7	8	9	10	11	12	13	14	17	19	20
2	98,3											
3									0,3		0,3	
4	1,7											
7		98,0				0,3					0,3	
8			97,7							0,3	0,3	
9		3,3		98,0								
10		6,7	0,7	1,0	93,0	1,00			6,0	0,3	0,7	2,3
11					0,7	97,4						
12							99,0				0,7	
13								97,7				
14		1,0	1,3	1,0	6,3	1,3			93,7	1,3	2,3	1,7
16							0,3					
17										98,1		
18								2,3				
19			0,3				0,7				95,4	
20												96,0

Tabela 6.12 : matriz de confusão reduzida (MLG/CMU).

Classes	Classes Desejadas				
Obtidas	2	10	11	13	14
1					2,67
2	81,00				
4	19,00				
7			2,00		
10		97,33			
11		2,67	98,00		
13				98,00	
14					97,33
18				2,00	

Reportando-se à Figura 2.3, que mostra os indivíduos do conjunto de faces CMU, pode-se inferir da análise das Tabelas 6.11 e 6.12 que as classes (indivíduos) para as quais o classificador MLG apresenta dificuldades de classificar constituem apenas um pequeno subconjunto das classes que também representam problemas para a rede SOM-C3 e,

na maioria dos casos, as confusões são as mesmas. Com isto, pode-se dizer que estes classificadores são diferentes sob o critério de avaliação da matriz de confusão e que, caso o critério de escolha entre eles seja a quantidade de classes corretamente classificadas, o MLG deve ser escolhido, contudo, caso o critério seja o erro cometido por classe, o SOM-C3 deve ser escolhido visto que o MLG apresentou um erro grande para a classe 2.

Tabela 6.13 : teste-t para o conjunto de faces ORL.

	SOM-C3	**MLP-1C**	**SVM-GOA**
MLP-1C	*-7,7120*		
SVM-GOA	*-12,3462*	*-5,0745*	
MLG	*-4,4934*	*3,1414*	*7,9240*

Pelos resultados apresentados na Tabela 6.13 para o conjunto de faces ORL, todos os classificadores avaliados podem ser considerados diferentes para os níveis de significância adotados.

Tabela 6.14 : teste-t para o conjunto de faces STIRLING.

	SOM-C3	**MLP-1C**	**SVM-GOA**
MLP-1C	*-5,7255*		
SVM-GOA	*-14,9757*	*-8,9265*	
MLG	*0,0791*	*6,4517*	*17,5683*

Para o conjunto de faces STIRLING (ver Tabela 6.14), apenas os classificadores SOM-C3 e MLG, podem ser considerados iguais para os dois níveis de significância adotados.

Tabela 6.15 : matriz de confusão reduzida (SOM-C3/STIRLING).

Classes	Classes Desejadas																														
Obtidas	1	2	3	4	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21	23	24	25	26	27	28	29	30	31	32	34	35	36
1	84,7				0,3		2,0										1,3		0,3				0,3									
2		74,3		2,0			1,7				0,7				0,7		0,3						4,3	6,3		1,0	0,3	0,3				
3		0,3	91,3			0,3		0,7		0,3			0,3		0,7			0,3					0,7	1,0	0,7				2,7			
4	7,7	3,0	4,7	83,3		0,3	5,7		1,3	3,7	1,7	3,0		2,3	0,7	7,3	0,3		0,3		0,3			9,7	0,3		0,3			1,0		0,3
5								1,0						1,0	0,3										0,3	0,3					0,3	
6	6,0	1,3			96,3		0,7		0,3	0,3	18,7		0,7		3,3		0,3				0,7		0,3					0,7	0,3			
7			0,3			1,0	2,3				0,3				0,7								0,3	0,3				0,3				
8		1,0		0,3		93,7					0,3	0,7				0,3													3,0			
9	0,7	0,7					66,3		1,3	0,7	0,7	4,3		0,3	3,0		3,0		6,7		1,7		0,3				0,7					
10				0,3				88,3				1,3			2,0						0,3							1,3				
11				0,7				1,3	92,7		1,3	2,0			0,3						2,0		0,7				1,7					
12		0,3	0,3	1,0			0,3		1,3	72,3	1,7			0,3	0,3	0,3		1,3			2,7		1,0	0,7								
13		0,3			2,0		0,3		0,7		67,0				1,0	0,3					6,7			0,3	0,3			3,0	0,3			
14								0,7			0,3	85,0			0,7													0,3				
15			0,3					0,3					96,0	0,3											0,3							
16				3,3	0,7		1,0	0,7	0,3		1,3		2,3	87,7	0,3						0,7			10,3	1,0		3,0		0,3			
17		0,3			0,3		0,7								61,3													0,7				
18	0,3	0,7		0,7			0,7			1,7	1,0				2,0	88,0	2,0	0,7			0,7		0,3						0,7			
19		0,3					0,7										89,7	0,3	0,7					0,3						0,3		
20		6,7		0,3	0,3		3,3	2,3		6,7	0,3			0,3	0,3	2,0	0,3	94,3			0,3		4,0	0,3		0,7	1,3		1,7	0,3		0,3
21							6,0										0,3		92,0								0,3					
22			0,3								0,3					0,3	1,7															0,3
23						1,3									1,3					99,7			0,7	0,3	0,3	0,3			1,0	0,7	
24		0,7	0,7	3,0		0,3	1,0			9,3						0,3					71,7			4,7	1,7	1,7	2,7		1,7			4,7
25			0,3			0,3	0,3	1,3			0,3				1,0							99,7		6,0	1,0							0,7
26	0,7	0,7	0,3	0,3			1,7		0,7	0,7					3,3	0,3					0,7		78,3	6,3	0,3	1,3	1,0		2,3	4,7		
27		7,0		2,3			1,0			2,3				6,7	2,0			1,7			4,0			46,0		0,7	0,7					0,3
28			0,7								1,3	1,0	0,3	0,3	5,7						0,7	0,3	0,3	0,3	87,3		0,3	0,3	0,3		2,3	0,7
29		0,7						0,3		0,3					0,7										0,7	92,0	0,3				0,3	
30							2,7		0,7	0,3		1,7			4,0						2,0		2,0			0,7	77,7		0,7	0,7		
31		0,3				0,3			0,3	0,7	2,3	0,7		0,3	2,0								0,3	0,3	0,3				92,3	1,3		
32		0,7				2,3	1,0	2,0					0,3		0,3		0,7				0,3		1,3						0,7	81,0		0,3	
33		0,3						0,3			0,3			0,3							0,3											
34				2,0			0,3		0,3						1,7	0,7				0,3			4,3		2,3	0,7	9,7			91,0		0,7
35							0,3			0,3					0,3						0,3		0,3		0,3	0,3			3,7		96,0	
36		0,3	0,7	0,3				0,7		0,3		0,3						1,3			4,0			6,7	2,7	0,3				1,0		92,0

Tabela 6.16 : matriz de confusão reduzida (MLG/STIRLING).

Classes	Classes Desejadas																							
Obtidas	1	2	3	4	9	10	11	12	13	14	15	16	17	18	19	21	24	27	29	30	31	32	34	36
1	78,0																							
2		64,7																						
3			92,0																					
4	22,0	12,7	8,0	88,0	8,0		11,7			11,7		12,0	13,0	10,7				12,0						
6									19,7															
9					70,3		10,7			13,0			8,0			9,0								
10						76,0																		
11							77,7																	
12								80,3																
13									80,3								13,0				10,7			
14										75,3														
15											88,3													
16											11,7	88,0						13,0		10,7				
17													47,7											
18														89,3										
19															89,3									
20		12,0			9,0	11,7		8,0																
21																91,0								
24				12,0				11,7									87,0	12,3	12,7			9,0		10,7
25						12,3												9,0						
26					12,7								9,0					10,7					12,0	
27		10,7											10,7					43,0		8,0				
28													11,7											
29																			87,3					
30																				69,3				
31																					89,3			
32															10,7							91,0		
34																				12,0			88,0	
36																								89,3

Da análise das Tabelas 6.15 e 6.16 pode-se concluir que o classificador MLG consegue acertar as classes 05,06,07,08,20,22,23,25,26,28,33 e 35 sempre. Enquanto que o SOM-C3

consegue acertar apenas as classes 05,07,22 e 33. Além disto, a matriz de confusão de classificador MLG é mais esparsa que a do SOM-C3, ou seja, este último confunde uma dada classe com um maior número de outras classes. Assim, sob esta avaliação, pode-se concluir que estes classificadores são, na verdade, bem diferentes. Considerando-se as mesmas taxas de acerto o classificador MLG deve ser o escolhido pois a possibilidade de confusão de classificação é reduzida a uma menor quantidade de outras classes que o SOM-C3.

Tabela 6.17 : teste-t para o conjunto de faces YALE-1.

	SOM-C2	**MLP-1C**	**SVM-GOO**
MLP-1C	*zero*		
SVM-GOO	*4,2901*	*3,9464*	
MLG	*4,7239*	*4,3262*	*0,2467*

Finalmente, para o conjunto de faces YALE-1 (ver Tabela 6.17), os classificadores SOM-C2 e MLP-1C podem ser considerados estatisticamente iguais para os dois níveis de significância adotados assim como os classificadores MLG e SVM-GOO. Nas Tabelas 6.18 e 6.19 encontram-se as matrizes de confusão dos dois primeiros classificadores.

Tabela 6.18 : matriz de confusão reduzida (SOM-C2/YALE-1).

Classes	Classes Desejadas						
Obtidas	1	2	4	7	8	9	15
1	88,7	0,3					1,0
2	11,3	99,7					
3			0,3		5,0		10,0
4			93,0				
7				93,4			0,7
8			1,7	3,3	88,0	5,3	
9					6,7	94,7	
10			0,7				
15			4,3	3,3	0,3		88,3

Tabela 6.19 : matriz de confusão reduzida (MLP-1C/YALE-1).

Classes	Classes Desejadas						
Obtidas	1	3	4	6	8	9	15
1	77,0						
2	23,0		1,0				
3		88,0					10,0
4			95,0				
6				95,0			
7				5,0			
8					88,0	6,0	
9		12,0			11,0	94,0	
10			4,0				
13					1,0		
15							90,0

Da análise das Tabelas 6.18 e 6.19 pode-se concluir que ambos os classificadores cometem erros de classificação para sete das quinze classes presentes no conjunto YALE-1. Destas sete classes, cinco são comuns às duas redes. E nestas cinco Dos erros cometidos são também similares. Deste modo pode-se concluir que, sob esta análise, estes classificadores são, de fato, equivalentes.

Nas Tabelas 6.20 e 6.21 são apresentados as matrizes de confusão reduzidas para os classificadores MLG e SVM-GOO, respectivamente.

Tabela 6.20 : matriz de confusão reduzida (MLG/YALE-1).

Classes	Classes Desejadas					
Obtidas	1	8	9	10	13	15
1	88,33			8,00		
2	11,67					
3			8,00			
7					8,33	17,33
8		91,67	12,00			
9			70,67			
10				92,00		
13		8,33			91,67	
15			9,33			82,67

Tabela 6.21 : matriz de confusão reduzida (SVM-GOO/YALE-1).

Classes	Classes Desejadas							
Obtidas	1	3	4	5	8	9	14	15
1	89,33							
2	10,67							
3		83,67					7,33	7,33
4			91,67					
5				91,67				
8		8,33	8,33		83,67	7,33		
9		8,00			16,33	92,67		
14							92,67	
15				8,33				92,67

Da análise das Tabelas 6.20 e 6.21 vê-se que estes classificadores apresentam erros comuns apenas para as classes 1, 8, 9 e 15. O classificador MLG tem dificuldade de classificação com menos classes que o SVM-GOO e, portanto, é melhor que este. Considerando-se o desempenho médio para estas classes o classificador SVM-GOO deve ser escolhido em relação ao MLG pois este apresenta uma maior confusão que aquele. Exemplo disto é a classe 9.

Na Figura 6.2 são ilustrados os erros de classificação cometidos pelo classificador SVM-GOO para o conjunto de faces YALE-1.

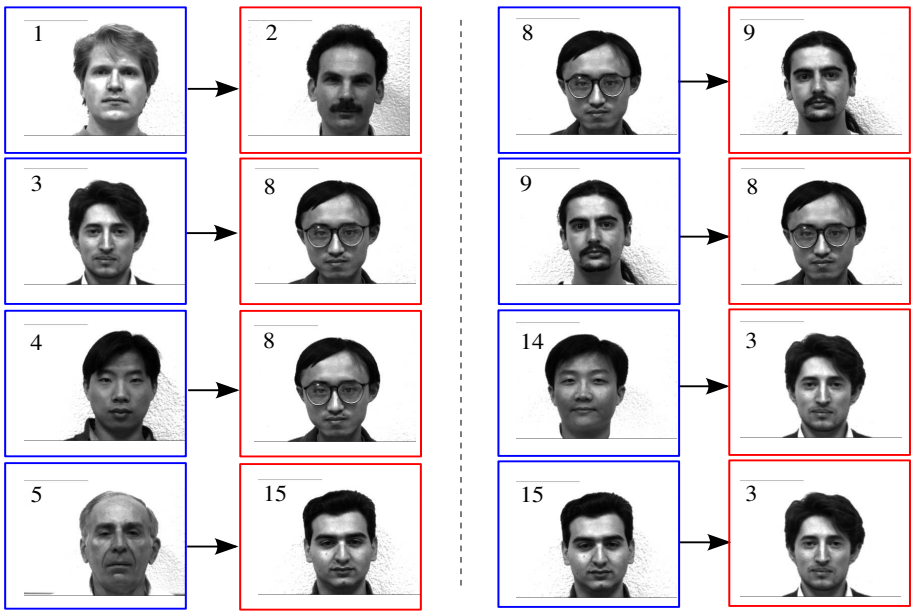


Figura 6.2 : exemplos de imagens classificadas erroneamente pelo classificador (classe desejada azul - classe obtida vermelha).

6.3 Análise do Custo Computacional

Embora esta dissertação não tenha como um de seus objetivos principais avaliar o custo computacional dos classificadores implementados[2], na Tabela 6.22 são apresentados os tempos gastos por estes para cada conjunto de faces utilizado. Estes tempos correspondem a média dos tempos de todas as 300 realizações treino-teste feitas para cada classificador. Na Tabela, os melhores resultados por grupo de classificadores são incluídos em sobrescrito ao lado do tempo total gasto pelo classificador.

Todos os tempos foram obtidos para implementações em linguagem script do ambiente Matlab© e realizadas em sistema operacional de 32 bits com processador Intel Core^{TM}2 Duo de 1,83GHz e 2MB de memória RAM DDR2 667MHz. Foram usados as funções nativas *tic*() e *toc*() do Matlab© para registrar os tempos.

Tabela 6.22 : tempo computacional estimado para cada classificador (em segundos).

Modelos Testados	**Conjuntos de Faces**				
	BARTLETT	CMU	ORL	STIRLING	YALE-1
SOM-C1	7,85	60,32	37,83	30,17	14,22
SOM-C2	$3{,}81^{(93,71)}$	4,64	3,43	2,87	$1{,}67^{(96,37)}$
SOM-C3	6,37	$47{,}05^{(98,10)}$	$31{,}03^{(93,54)}$	$24{,}68^{(86,36)}$	11,12
SOM-C4	6,59	45,93	29,91	23,73	11,05
PS	0,72	4,15	3,49	2,68	1,09
MADALINE	0,65	4,56	3,53	2,80	1,13
PL	6,96	44,24	51,70	37,43	$8{,}72^{(96,88)}$
PL-MEKA	7,33	59,59	79,91	56,88	10,31
MLP-1C	$0{,}93^{(96,08)}$	$7{,}46^{(99,45)}$	$6{,}07^{(95,90)}$	$4{,}89^{(88,63)}$	1,73
MLP-2C	3,10	27,75	19,95	15,60	5,54
GRNN	0,59	0,53	0,65	0,59	0,54
RBF-SOM	5,16	$9{,}57^{(99,05)}$	7,73	6,42	3,40
SVM-LOO	$2{,}60^{(96,10)}$	4,32	21,17	14,63	$1{,}82^{(94,91)}$
SVM-LOA	0,93	4,47	3,65	2,79	0,96
SVM-GOO	2,95	6,39	25,19	17,11	2,14
SVM-GOA	1,15	$16{,}21^{(99,05)}$	$21{,}64^{(97,11)}$	$16{,}46^{(91,41)}$	1,07
OLAM	0,03	0,03	0,04	0,03	0,02
MLG	$0{,}09^{(92,37)}$	$0{,}10^{(98,50)}$	$0{,}10^{(95,01)}$	$0{,}10^{(86,33)}$	$0{,}08^{(94,40)}$

[2]Os leitores interessados neste tópico específico podem buscar mais informações em Stan & Kamen (1997), Kusumoto & Takefuji (2006) e Wickramasinghe et al. (2007).

É importante enfatizar que os valores apresentados na Tabela 6.22 podem variar de acordo com a expertise do usuário em programação e as configurações do equipamento que o mesmo utilize. Mesmo assim, ao reportá-la, busca-se transmitir noções preliminares acerca do custo computacional, em termos de tempo de processamento, que cada classificador demanda sob as condições impostas neste trabalho. A análise inicial dos tempos gastos pode ajudar o usuário na escolha do classificador que melhor se adequa à tarefa de interesse, principalmente se esta envolver recursos limitados de processador e memória.

Ao analisar-se estes resultados é possível concluir que se o critério tempo de treinamento e teste for mandatório para a escolha do classificador o usuário deve optar pelo MLG já que a sua rapidez é superior aos demais de cada grupo em todos os conjuntos de imagens e apresentando taxa média de acerto elevado. Na Tabela 6.23 são apresentados os tempos médios de classificação para cada arquitetura.

Tabela 6.23 : tempo computacional médio de classificação (em segundos).

Modelos Testados	**Conjuntos de Faces**				
	BARTLETT	CMU	ORL	STIRLING	YALE-1
SOM-C1	0,014	0,016	0,015	0,015	0,014
SOM-C2	$0{,}012^{(93,71)}$	0,014	0,014	0,013	$0{,}012^{(96,37)}$
SOM-C3	0,015	$0{,}016^{(98,10)}$	$0{,}016^{(93,54)}$	$0{,}016^{(86,36)}$	0,016
SOM-C4	0,017	0,018	0,019	0,019	0,017
PS	0,001	0,001	0,001	0,001	0,001
MADALINE	0,001	0,001	0,001	0,001	0,001
PL	0,002	0,003	0,004	0,003	$0{,}002^{(96,88)}$
PL-MEKA	0,002	0,003	0,004	0,003	0,002
MLP-1C	$0{,}022^{(96,08)}$	$0{,}028^{(99,45)}$	$0{,}030^{(95,90)}$	$0{,}029^{(88,63)}$	0,021
MLP-2C	0,022	0,029	0,031	0,030	0,022
GRNN	0,002	0,004	0,004	0,003	0,002
RBF-SOM	0,002	$0{,}004^{(99,05)}$	0,004	0,003	0,002
SVM-LOO	$0{,}008^{(96,10)}$	0,010	0,022	0,019	$0{,}008^{(94,91)}$
SVM-LOA	0,005	0,006	0,007	0,006	0,005
SVM-GOO	0,010	0,018	0,091	0,068	0,007
SVM-GOA	0,005	$0{,}007^{(99,05)}$	$0{,}021^{(97,11)}$	$0{,}018^{(91,41)}$	0,005
OLAM	0,001	0,001	0,001	0,001	0,001
MLG	$0{,}004^{(92,37)}$	$0{,}004^{(98,50)}$	$0{,}005^{(95,01)}$	$0{,}005^{(86,33)}$	$0{,}004^{(94,40)}$

Por estes resultados (Tabela 6.23) nota-se que a etapa de treinamento é mais crítica em termos de tempo e que o classificador MLP-1C é o melhor para estas aplicações.

6.4 Resumo do Capítulo

Neste capítulo os classificadores foram testados com imagens ruidosas. O ruído sal-e-pimenta foi adotado para este fim e, a partir da análise do resultados dos experimentos, percebeu-se que a capacidade de todos os classificadores foi reduzida consideravelmente, exceto para o banco de faces YALE-1.

Em seguida, todos os classificadores foram avaliados através de teste de hipótese (Teste-T Pareado) e, para os casos em que os classificadores foram tidos como iguais, suas matrizes de confusão foram também analisadas no intuito de verificar o grau de similaridade entre as dificuldades de classificação por classe.

Por fim, e a título de complementar a descrição dos classificadores utilizados, foram apresentados os tempos computacionais gastos por cada classificador estudado.

No próximo capítulo são apresentadas as conclusões gerais desta dissertação, bem como sugeridas algumas indicações de trabalhos futuros.

7 *Conclusões e Perspectivas*

"Saber é uma palavra cujo significado é muito amplo.
Prefiro dizer que estamos capacitados a fazer algumas suposições."

Aldous Leonard Huxley

Neste capítulo, as observações finais e perspectivas de trabalhos futuros são apresentadas.

7.1 Conclusões deste Trabalho

Ao longo desta dissertação, dezoito classificadores foram aplicados à tarefa de reconhecer faces humanas. Procurou-se, com isto, fornecer aos usuários, que necessitem resolver problemas similares, informações qualitativas acerca da capacidade destas diferentes arquiteturas. Deste modo, os seguintes pontos merecem destaque:

- Um dos desafios inerentes ao problema de reconhecer faces é determinar, *a priori*, quais características apresentar para o classificador. Neste sentido, projeções lineares obtidas pela Análise de Componentes Principais - PCA se mostram capazes de contornar bem esta questão para os classificadores testados. A variância explicada fixa em 90% destacou os classificadores MLP-1C e SVM-GOA como os de melhor desempenho. Contudo a elevada taxa de compressão conseguida pode ser a responsável por tornar o conjunto STIRLING mais desafiador.

- Os classificadores baseados na rede auto-organizável de Kohonen conseguem desempenhos comparáveis aos tradicionais MLP, RBF e SVM.

- Para aplicações práticas, principalmente, o método SOM-C2 é o mais indicado. Suas características de modularidade e rapidez justificam esta escolha dentre as demais. A primeira destas características evita que o classificador treine todos os seus neurônios novamente caso se deseje inserir uma nova classe. Neste caso cria-se apenas um novo mapa dedicado a esta classe.

- Embora a rede MLP se destaque dentre os classificadores perceptron, não se deve preterir de imediato as versões lineares, pois, mesmo para condições ruidosas, estas podem apresentar um desempenho elevado (ver Tabela 6.5).

- Mesmo redes perceptron lógistico treinadas com o algoritmo MEKA podem apresentar elevadas taxas de acerto. Pelas condições de treinamento adotadas, pode-se comprovar a rapidez de convergência desta técnica, em termos do número de épocas de treinamento. Seu uso, no entanto, não é indicado para aplicações com pouca disponibilidade de memória e capacidade de processamento, pois, como visto, cada neurônio possui um conjunto particular de parâmetros (principalmente, matrizes) a ajustar, cuja ordem depende diretamente da dimensão dos vetores de entrada.

- Redes RBF que utilizam mapas auto-organizáveis de Kohonen para determinar seus kernels conseguem elevadas taxas de acerto para este tipo de problema de classificação, ocupando estas as melhores posições entre os classificadores.

- Pôde-se comprovar a elevada capacidade dos classificadores SVM em resolver problemas de classificação de faces dado que nenhuma técnica de otimização fora utilizada no intuito de acelerar seus processamentos. Estes classificadores podem ser considerados os melhores em termos de taxa de acerto e, em favor destes, têm-se os resultados obtidos para o conjunto de faces STIRLING, considerado o mais desafiador dos conjuntos utilizados. Apenas o algoritmo SVM-GOA consegue taxa média de acerto superior a noventa porcento para as condições impostas.

- Mesmo classificadores com taxas de acerto próximas e que, sob a perspectiva de teste de hipótese, sejam considerados iguais, numa análise por matriz de confusão podem apresentar comportamentos bem distintos. Esta última ferramenta de análise deve, portanto, ser utilizada pelo usuário para auxiliá-lo na escolha do classificador mais adequado ao seu problema.

- As elevadas taxas médias de acerto conseguidas pelo classificador MLG podem ser justificadas pela metodologia empregada que descorrelaciona (PCA) e normaliza os dados.

- O classificador que melhor combina desempenho e rapidez de classificação para resolver este tipo de problema é o classificador MLP-1C. Configurando-se numa boa escolha para resolver este tipo de problema.

7.2 Perspectivas

A seguir são feitas algumas sugestões de investigações futuras nesta área de reconhecimento de faces:

- O classificador SOM-C1 apresenta problemas de rotulação. Outras alternativas de solução para este problema, que não as que foram apresentadas no Capítulo 3, podem elevar as taxas de acerto deste classificador. Como exemplo, não descartar os neurônios sem rótulos e, sim, fazê-los herdar o rótulo do neurônio rotulado mais próximo.

- O classificador SOM-C4 apresentou, em muitos dos experimentos, taxas de classificação inferiores aos demais da mesma família. Contudo, estudos preliminares indicam que este método consegue melhorar seu desempenho se um maior número de épocas de treinamento for adotado.

- Aplicar técnicas não lineares para pré-processar os dados é um dos estudos que estão sendo conduzidos pelo autor. Combinar Análise por Componentes Independentes - ICA e PCA (BARTLETT, 2001) podem permitir a alguns classificadores superar deficiências impostas por condições ruidosas, por exemplo.

- Outro aspecto que está sob investigação é a capacidade de classificação dado que se disponha de uma imagem para treinamento apenas. Aliado à esta condição, desenvolve-se mecanismo de pré-processamento de imagens sem o uso de técnicas do tipo PCA ou similares. Com isto, pode-se avaliar o desempenho dos classificadores SVM para esta condição sem PCA e, por conseguinte, de alta dimensionalidade.

- A *novidade* é outro objeto importante para futuros estudos dentro desta área de reconhecimento de faces. Determinar a capacidade de detecção de novos indivíduos por estes classificadores pode contribuir para aplicações práticas na área da biometria (acesso à áreas restritas, bancos, empresas). Neste sentido, a questão temporal deve ser estudada em maior detalhes também, ou seja, uma face considerada *nova* pode ser na verdade de um indivíduo já classificado que, no entanto, sofreu alterações ao longo do tempo como, por exemplo, aquele indivíduo que era barbudo e agora não mais.

- Através da avaliação das matrizes de confusão de alguns classificadores com os melhores desempenhos, pôde-se constatar que suas dificuldades de classificação, por classe, eram muitas vezes distintas. Este fato motiva investigações no sentido de combinar estes classificadores em estruturas do tipo *comitês de máquinas de aprendizado* e, por conseguinte, estudar os ganhos que estas podem fornecer.

- Futuros estudos podem ser realizados com o objetivo de determinar as configurações de faces que facilitem o reconhecimento destas pelos classificadores. O conjunto de faces YALE-1, por exemplo, mesmo sob condições ruidosas não degradou como os demais os desempenhos obtidos pelos classificadores. Deste modo, as configurações deste conjunto se mostram como um bom ponto de partida para esta investigação.

APÊNDICE A – Outros Classificadores

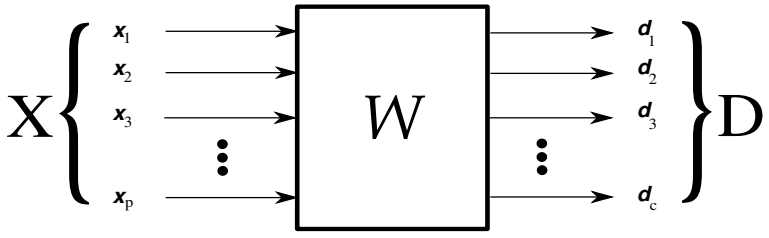
"Deveríamos tomar cuidado para tirar de uma experiência apenas a sabedoria que ela contém."

Mark Twain

Este apêndice apresenta uma descrição sucinta dos classificadores: Memória Associativa Linerar Ótima - OLAM e Classificador Gaussiano de Máxima Verossimilhança - MLG. Em seguida são apresentados os resultados destes classificadores quando aplicados aos conjuntos de faces.

A.1 Memórias Associativas Lineares

Memórias associativas podem ser entendidas como um sistema entrada-saída, assim como mostrado pela Figura A.1. Estas memórias podem ser classificadas de várias maneiras dependendo da arquitetura (com ou sem realimentação), da natureza das associações feitas (autoassociativas ou heteroassociativas), da capacidade e complexidade de armazenamento e recuperação da informação, por exemplo (KOHONEN, 1984). Neste modelo, o objetivo é determinar $\mathbf{W}$ capaz de armazenar todos os pares ou associações $(\mathbf{x}, \mathbf{d})$ utilizados pelo usuário.



(a)

Figura A.1 : memória heteroassociativa ($\mathbf{X} \neq \mathbf{D}$) genérica.

A.1.1 Memória Associativa Linear Ótima - OLAM

Uma das características do modelo de memória apresentado anteriormente é a presença de um termo de interferência cruzada ou *crosstalk* (HAYKIN, 1994) que compromete o processo de associação perfeita entre os padrões de entrada e saída. Para se contornar este problema, o usuário dispõe de um modelo conhecido como Memória Associativa Linear Ótima - OLAM (KOHONEN, 1972; KOHONEN; RUOHONEN, 1973). Em termos matemáticos, este modelo deve satisfazer a seguinte equação:

$$\mathbf{D}_{tr} = \mathbf{W}^{o}\mathbf{X}_{tr} \tag{A.1}$$

Note que o subescrito tr indica o conjunto selecionado para treino enquanto o sobrescrito o indica a condição ótima. Por conseguinte, se $\mathbf{X}_{tr}^{-1}$ existe, temos que:

$$\mathbf{W}^{o} = \mathbf{D}_{tr}\mathbf{X}_{tr}^{-1} \tag{A.2}$$

Caso contrário, calcula-se a pseudo-inversa de modo que esta Equação A.2 fica da forma:

$$\mathbf{W}^{o} = \mathbf{D}_{tr}\mathbf{X}_{tr}^{T}(\mathbf{X}_{tr}\mathbf{X}_{tr}^{T})^{-1} \tag{A.3}$$

onde T indica a transposta da matriz.

Deste modo, após determinar-se a matriz $\mathbf{W^o}$ pode-se aplicar o conjunto $\mathbf{X}_{te}$ de teste a este modelo e comparar as saídas obtidas $\mathbf{Y}_{te}$ com as desejadas $\mathbf{D}_{te}$.

A.2 Classificador Gaussiano de Máxima Verossimilhança - MLG

Durante a implementação de classificadores de padrões, principalmente quando se dispõem de poucas informações sobre os dados amostrais, algumas suposições sobre estes se fazem necessárias.

Neste sentido, o classificador MLG caracteriza-se pelas seguintes suposições:

1. Todas as classes são equiprováveis.
2. As características de entrada possuem distribuição gaussiana.
3. Características de entrada descorrelacionadas entre si e com a mesma variância, ou seja, $\sigma_i = \sigma, \forall i$.

Com isto em mente, este classificador pode ser implementado como um classificador de distância mínima ao centróide, ou seja, determinam-se todos os vetores $\overline{\mathbf{w}}_k$ onde cada um destes representa o vetor médio da k-ésima classe presente no conjunto de faces de treino,

$$\overline{\mathbf{w}}_k = \frac{1}{N_k} \sum_{\mathbf{x} \in k} \mathbf{x} \tag{A.4}$$

onde N_k é o número de vetores que compõe a classe de treino k.

E desta forma, após determinarem-se todos os centróides, cada um dos vetores de entrada $\mathbf{x}(t)$ é classificado como pertencente a classe k do centróide $\overline{\mathbf{w}}_k$ para o qual apresente a menor distância.

$$classe(\mathbf{x}(t)) = \arg\min_{\forall i} \{\|\mathbf{x}(t) - \mathbf{w}_i\|\} \tag{A.5}$$

onde $\|\cdot\|$ indica a distância euclidiana.

A.3 Resultados de Classificação

Tabela A.1 : desempenho para as faces Bartlett.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
OLAM	20,98	0,00	40,00	2,57	76,41
MLG	92,37	85,00	100,0	0,25	5,41

Para este conjunto BARTLETT, os classificadores OLAM e MLG apresentaram desempenhos distintos. Destes resultados conclui-se que apenas o segundo resolve o problema satisfatoriamente.

Tabela A.2 : desempenho para as faces CMU.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
OLAM	96,18	90,00	100,0	0,12	3,60
MLG	98,50	90,00	100,0	0,08	2,87

Diferente do acontece no caso anterior, para o CMU ambos os classificadores demonstram elevada capacidade de classificação com pequena superioridade do classificador MLG se comparado ao OLAM.

Tabela A.3 : desempenho para as faces ORL.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
OLAM	77,63	62,50	87,50	0,46	8,74
MLG	95,01	90,00	100,0	0,14	3,94

Para o conjunto ORL, o classificador MLG pode ser considerado melhor que o OLAM pois apresentou superior taxa de acerto média e inferior coeficiente de variação.

Tabela A.4 : desempenho para as faces STIRLING.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
OLAM	58,24	50,00	66,67	0,25	8,58
MLG	86,33	80,56	94,44	0,17	4,78

Comparando os resultados de classificação para este conjunto STIRLING, nota-se a superioridade da arquitetura MLG (taxa de acerto média maior e coeficiente de variação inferior) em relação a OLAM.

Tabela A.5 : desempenho para as faces Yale-1.

Modelos Testados	**Taxas de Reconhecimento (%)**				
	média	*mínima*	máxima	*variância*	*cv*
OLAM	68,13	0,00	93,33	9,67	45,64
MLG	94,40	80,00	100,0	0,40	6,70

Assim como para os conjuntos anteriores, o classificador OLAM obteve desempenho inferior que o MLG. Contudo, merece destaque o fato dele apresentar um elevado coeficiente de variação.

A seguir, tem-se a disposição os diagramas de caixa obtidos por estes classificadores para cada conjunto de faces.

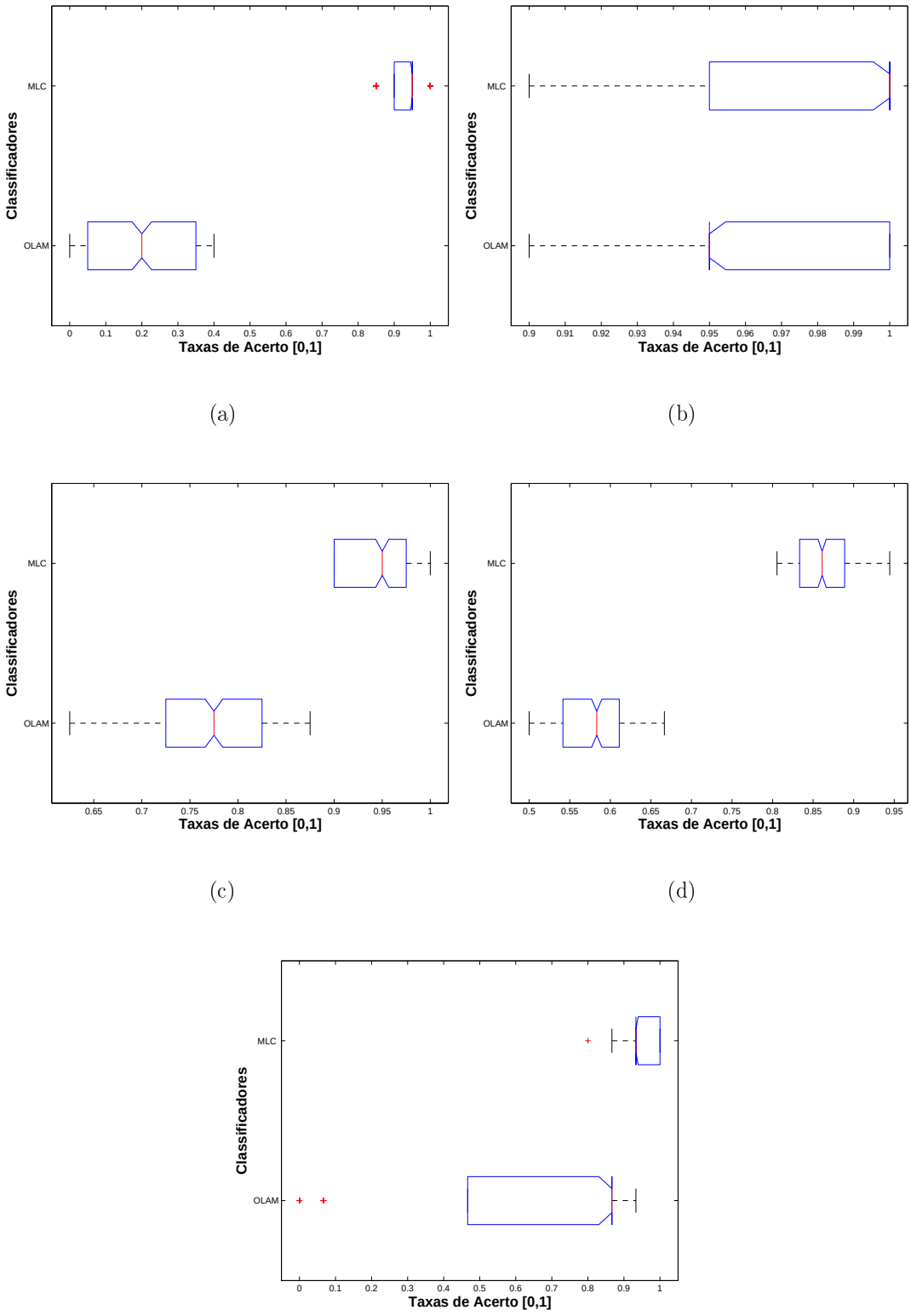


Figura A.2 : diagramas de caixa correspondentes aos desempenhos dos classificadores MLG e OLAM para os bancos de faces: (a) BARTLETT, (b) CMU, (c) ORL, (d) STIRLING e (e) YALE-1.

Conjunto BARTLETT:

- Apenas o classificador MLG apresentou resultados pouco frequentes, tidos como *outliers* e uma distribuição interquartil assimétrica.
- O classificador OLAM além de apresentar uma dispersão superior também obteve desempenho inferiores ao MLG. Assim, este pode ser considerado melhor que o primeiro.

Conjunto CMU:

- Como visto, ambos os classificadores apresentaram comportamentos similares. Ambos assimétricos, de igual dispersão e mesmo intervalo interquartil.
- O classificador MLG pode ser considerado superior em termos de mediana.

Conjunto ORL:

- Nenhum dos classificadores apresentou *outliers*.
- Apenas o classificador OLAM apresentou distribuição interquartil simétrica. Em contrapartida, este apresentou resultados inferiores aos obtidos pelo MLG. Sendo este último, portanto, o melhor.

Conjunto STIRLING:

- Nenhum dos classificadores apresentou *outliers*.
- O classificador OLAM apresentou distribuição interquartil assimétrica e desempenho inferior ao MLG.

Conjunto YALE-1:

- Ambos os classificadores apresentaram *outliers*.
- O classificador OLAM apresentou uma dispersão superior ao MLG.

Pelos resultados obtidos o classificador MLG é o mais indicado para classificar os vetores representantes das faces de acordo com a metodologia adotada.

APÊNDICE B – Distribuição t de Student

"Encontrei uma demonstração verdadeiramente admirável,
mas a margem é muito pequena para apresentá-la"

Pierre de Fermat

Tabela B.1 : distribuição t de Student.

Graus de Liberdade	**Valores Percentis para a Distribuição t de Student (α)**								
	0,50	*0,40*	*0,30*	*0,20*	*0,10*	*0,05*	*0,02*	*0,01*	*0,001*
1	1,000	1,376	1,963	3,078	6,314	12,706	31,821	63,657	636,619
2	0,816	1,061	1,386	1,886	2,920	4,303	6,965	9,925	31,598
3	0,765	0,978	1,250	1,638	2,353	3,182	4,541	5,841	12,929
4	0,741	0,941	1,190	1,533	2,132	2,776	3,747	4,604	8,610
5	0,727	0,920	1,156	1,476	2,015	2,571	3,365	4,032	6,869
6	0,718	0,906	1,134	1,440	1,943	2,447	3,143	3,707	5,959
7	0,711	0,896	1,119	1,415	1,895	2,365	2,998	3,499	5,408
8	0,706	0,889	1,108	1,397	1,860	2,306	2,896	3,355	5,041
9	0,703	0,883	1,100	1,383	1,833	2,262	2,821	3,250	4,781
10	0,700	0,879	1,093	1,372	1,812	2,228	2,764	3,169	4,587
11	0,697	0,876	1,088	1,363	1,796	2,201	2,718	3,106	4,437
12	0,695	0,873	1,083	1,356	1,782	2,179	2,681	3,055	4,318
13	0,694	0,870	1,079	1,350	1,771	2,160	2,650	3,012	4,221
14	0,692	0,868	1,076	1,345	1,761	2,145	2,624	2,977	4,140
15	0,691	0,866	1,074	1,341	1,753	2,131	2,602	2,947	4,073
16	0,690	0,865	1,071	1,337	1,746	2,120	2,583	2,921	4,015
17	0,689	0,863	1,069	1,333	1,740	2,110	2,567	2,898	3,965
18	0,688	0,862	1,067	1,330	1,734	2,101	2,552	2,878	3,922
19	0,688	0,861	1,066	1,328	1,729	2,093	2,539	2,861	3,883

continua na próxima página

continuação da página anterior									
20	0,687	0,860	1,064	1,325	1,725	2,086	2,528	2,845	3,850
21	0,686	0,859	1,063	1,323	1,721	2,080	2,518	2,831	3,819
22	0,686	0,858	1,061	1,321	1,717	2,074	2,508	2,819	3,792
23	0,685	0,858	1,060	1,319	1,714	2,069	2,500	2,807	3,767
24	0,685	0,857	1,059	1,318	1,711	2,064	2,492	2,797	3,745
25	0,684	0,856	1,058	1,316	1,708	2,060	2,485	2,787	3,725
26	0,684	0,856	1,058	1,315	1,706	2,056	2,479	2,779	3,707
27	0,684	0,855	1,057	1,314	1,703	2,052	2,473	2,771	3,690
28	0,683	0,855	1,056	1,313	1,701	2,048	2,467	2,763	3,674
29	0,683	0,854	1,055	1,311	1,699	2,045	2,462	2,756	3,659
30	0,683	0,854	1,055	1,310	1,697	2,042	2,457	2,750	3,646
35	0,682	0,852	1,052	1,306	1,690	2,030	2,438	2,724	3,592
40	0,681	0,851	1,050	1,303	1,684	2,021	2,423	2,705	3,551
45	0,680	0,850	1,049	1,301	1,679	2,014	2,412	2,690	3,521
50	0,679	0,849	1,047	1,299	1,676	2,009	2,403	2,678	3,497
60	0,679	0,848	1,046	1,296	1,671	2,000	2,390	2,660	3,461
80	0,678	0,846	1,043	1,292	1,664	1,990	2,374	2,639	3,417
100	0,677	0,845	1,042	1,290	1,660	1,984	2,364	2,626	3,391
∞	0,674	0,842	1,036	1,282	1,645	1,960	2,326	2,576	3,291

Referências

ABE, S. *Support vector machines for pattern classification*. London: Springer-Verlag, 2005. ISBN 1-85233-929-9.

AIZERMAN, M.; BRAVERMAN, E.; ROZONOER, L. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, v. 25, p. 821–837, 1964.

BARTLETT, M. S. *Face Image Analysis by Unsupervised Learning*. Norwell, Massachusetts: Kluwer Academic Publishers, 2001.

BAUM, E. B.; HAUSSLER, D. D. What size net gives valid generalization? *Neural Computation*, v. 1, n. 1, p. 151–160, 1989.

BAYKAL, N.; ERKMEN, A. M. Resilient backpropagation for rbf networks. In: *Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*. Brighton, UK: [s.n.], 2000. p. 624–627.

BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. *Nonlinear Programming Theory and Algorithms*. 2nd. ed. [S.l.]: Wiley, 1993.

BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 7, p. 711–720, 1997.

BELLMAN, R. *Adaptive Control Processes: A Guided Tour*. [S.l.]: Princeton University Press, 1961.

BIEBELMANN, E.; KÖPPEN, M.; NICKOLAY, B. Pratical aplications of neural networks in texture analysis. *Neurocomputing*, v. 13, n. 2–4, p. 261–279, 1996.

BONCELET, C. Handbook of image and video processing. In: _____. [S.l.]: Academic Press, 2005. cap. Image Noise Models.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. Pittsburgh, PA: ACM Press, 1992. p. 144–152.

BOSLAUGH, S.; WATTERS, P. A. *Statistics in a Nutshell*. [S.l.]: O'Reilly, 2008.

BOTTOU, L. et al. Comparison of classifiers methods: a case study in handwriting digit recognition. In: *International Conference on Pattern Recognition*. [S.l.]: IEEE Computer Society Press, 1994.

CHAN, R. H.; HO, C.-W.; NIKOLOVA, M. Salt-and-pepper noise removal by median-type noise detectors and edge-preserving regularization. *IEEE Transactions on Image Processing*, v. 14, p. 1479–1485, 2005.

CHELLAPPA, R.; WILSON, C. L.; SIROHEY, S. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, v. 83, n. 5, p. 705–741, 1995.

CHRISTODOULOU, C. I.; MICHAELIDES, S. C.; PATTICHIS, C. S. Multifeature texture analysis for the classification of clouds in satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, v. 41, n. 11, p. 2662–2668, 2003.

CORTES, C.; VAPNIK, V. Support vector network. *Machine Learning*, v. 20, p. 273–297, 1995.

D., B. Face recognition under varying pose. In: *Proceedings of the 1994 IEEE Computer Society on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 1994. p. 756–761.

DORIZZI, B. New trends in biometrics. In: *Telecommunications: Advances and Trends in Transmission, Networking and Applications*. Ceará, Brazil: Edson Queiroz Foundation, 2006. p. 157–171.

ER, M. J. et al. Face recognition with radial basis function (RBF) neural networks. *IEEE Transactions on Neural Networks*, v. 13, n. 3, p. 697–710, 2002.

FEITOSA, R. Q. et al. Facial expression classification using RBF and back-propagation neural networks. In: *Proceedings of the 6th International Conference on Information Systems Analysis and Synthesis (ISAS'2000)*. [S.l.: s.n.], 2000. p. 73–77.

FERRIS, M. C.; MUNSON, T. S. Interior-point methods for massive support vector machines. *Society for Industrial and Applied Mathematics*, v. 13, n. 3, p. 783–804, 2003.

FORT, J. C. Som's mathematics. *Neural Networks*, n. 19, p. 812–816, 2006.

FUKUNAGA, K.; HAYES, R. Effects of sample size in classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 11, n. 8, p. 873–885, 1989.

GASPAR, T. L. *Reconhecimento de faces humanas usando redes neurais MLP*. Dissertação (Mestrado) — Universidade de São Paulo, Departamento de Engenharia Elétrica-EESC, São Carlos, São Paulo, fevereiro 2006.

GIOVAGNOLI, M. R. et al. Cervical false negative cases detected by neural network-based technology: Critical review of cytologic errors. *Journal of Clinical Citology and Cytopathology - Acta Cytol*, v. 46, p. 1105–1109, 2002.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Pearson Prentice Hall, 2007.

GOOD, P. I. *Resampling Methods: A Practical Guide to Data Analysis*. 3rd edition. ed. [S.l.]: Birkhaüser Boston, c/o Springer Science+Business Media Inc., 2006. ISBN 0-8176-4386-9.

GUNN, S. R. *Support Vector Machines for Classification and Regression*. [S.l.], 1998.

GUO, G.; LI, S. Z.; CHAN, K. L. Support vector machines for face recognition. *Image and Vision Computing*, v. 19, n. 9-10, p. 631–638, 2001.

HASTIE, T.; STUETZLE, W. Principal curves. *Journal of the American Statistical Association*, v. 84, n. 406, p. 502–516, 1989.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Macmillan Publishing Company, 1994.

HEBB, D. O. *The Organization of Behavior*. New York: Wiley, 1949.

HERTZ, J.; KROGH, A.; PALMER, R. G. *Introduction to the theory of neural computation*. Redwood City, CA: Addison-Wesley, 1991.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, p. 359–366, 1989.

HOWELL, A. J. Face recognition using rbf networks. In: HOWLETT, R. J.; JAIN, L. C. (Ed.). *Radial Basis Function Networks 2: New Advances in Design*. [S.l.]: Physica-Verlag, 2001. p. 103–142.

JIMENEZ, L. O.; LANDGREBE, D. A. Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions onSystems, Man and Cybernetics, Part C*, v. 28, n. 1, p. 39–54, 1998.

JOLLIFFE, I. T. *Principal Components Analysis*. New York: Springer-Verlag, 1986.

KARNIN, E. D. A simple procedure for pruning back-propagation trained neural networks. *IEEE Transactions on Neural Networks*, v. 1, n. 2, p. 239–242, 1990.

KNERR, S.; PERSONNAZ, L.; DREYFUZ, G. *Single-layer learning revisited: a stepwise procedure for building and training a neural network*. [S.l.]: Springer-Verlag, 1990.

KOHONEN, T. Correlation matrix memories. *IEEE Transactions on Computers*, C-21, p. 353–359, 1972.

KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, v. 43, p. 59–69, 1982.

KOHONEN, T. *Self-Organization and Associative Memory*. 3rd. ed. Berlin, Heidelberg: Springer-Verlag, 1984.

KOHONEN, T. *Self-Organizing Maps*. 2nd extended. ed. Berlin, Heidelberg: Springer Series in Information Sciences, Vol. 30, 1997.

KOHONEN, T.; RUOHONEN, M. Representation of associated data by matrix operators. *IEEE Transactions on Computers*, v. 22, p. 701–702, 1973.

KOHONEN, T. K. The self-organizing map. *Neurocomputing*, v. 21, p. 1–6, 1998.

KROL, M.; FLOREK, A. Comparison of statistical classifiers as applied to the face recognition system based on active shape models. In: *Computer Recognition Systems*. [S.l.]: Springer, 2005. p. 791–797.

KUSUMOTO, H.; TAKEFUJI, Y. O(log_2 M) Self-Organizing Map algorithm without learning of neighborhood vectors. *IEEE Transactions on Neural Networks*, v. 17, n. 6, p. 1656–1661, 2006.

LAHA, A.; PAL, N. R. Some novel classifiers designed using prototypes extracted by a new scheme based on self-organizing feature map. *IEEE Transactions on Systems, Man, and Cybernetics*, B-31, n. 6, p. 881–890, 2001.

LOCHNER, H. V.; BHANDARI, M.; P.TORNETTA. Type-II error rates (beta errors) of randomized trials in orthopaedic trauma. *Journal of Bone and Joint Surgery*, v. 83, p. 1650–1655, 2001.

MEDEIROS, C. M. S.; BARRETO, G. A. An efficient method for pruning the multilayer perceptron based on the correlation of errors. In: *International Conference on Artificial Neural Networks (ICANN'07)*. [S.l.: s.n.], 2007.

MERCER, J. Functions of positive and negative type and their connection with the theory of integral equations. *A Philos. Trans. Roy. Soc.*, v. 209, p. 425–446, 1909.

MERKWIRTH, C.; WICHARD, J.; OGORZAŁEK, M. A software toolbox for constructing ensembles of heterogenous linear and nonlinear models. In: *Proceedings of the 2005 European Conference on Circuit Theory and Design*. [S.l.: s.n.], 2005. v. 3, p. 197–200.

MINSKY, M.; PAPERT, S. *Perceptrons*. Cambridge, Mass.: MIT Press, 1969.

MITCHELL, T. *Machine Learning*. [S.l.]: McGraw Hill, 1997.

MONTEIRO, I. Q. et al. Face recognition independent of facial expression through SOM-based classifiers. In: *Proceedings of the 2006 IEEE/SBrT International Telecommunications Symposium (ITS'06)*. [S.l.: s.n.], 2006. p. 13–18.

MOODY, J. E.; DARKEN, C. Fast learning in networks of locally-tuned processing units. *Neural Computation*, v. 1, n. 1, p. 281–294, 1989.

OJA, E. Neural networks, principal components and subspaces. *International Journal of Neural Systems*, v. 1, n. 1, p. 61–68, 1989.

OJA, M.; KASKI, S.; KOHONEN, T. Bibliography of self-organizing map SOM papers: 1998-2001 addendum. *Neural Computing Surveys*, v. 3, p. 140–156, 2003.

OSUNA, E.; FREUND, R.; GIROSI, F. An improved training algorithm for support vector machines. In: PRINCIPE, J. et al. (Ed.). *Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing VII*. New York: IEEE, 1997. p. 276–285.

PALMIERI, F. et al. Sound localization with a neural network trained with the multiple extended Kalman algorithm. In: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)'91*. [S.l.: s.n.], 1991. p. 125–131.

PALMIERI, F.; SHAH, S. A new algorithm to train Multilayer Perceptrons. In: *Proceedings of the IEEE International Conference on Systems, Man Cybernetics*. [S.l.: s.n.], 1989. p. 14–17.

PARK, J.; SANDBERG, I. W. Universal approximation using radial-basis-function networks. *Neural Computation*, MIT Press, Cambridge, MA, USA, v. 3, n. 2, p. 246–257, 1991.

PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. *Neural Adaptive Systems: Fundamentals Through Simulations*. [S.l.]: John Willey and Sons, 2000.

QUEIROZ, S. D. et al. Classificação robusta de faces usando a rede de Kohonen. In: *Anais do XVI Congresso Brasileiro de Automática (CBA'06)*. [S.l.: s.n.], 2006. p. 2826–2831.

REDONDO, M. F.; SOSPEDRA, J. T.; ESPINOSA, C. H. Training radial basis functions by gradient descent. In: *International Joint Conference on Neural Networks*. [S.l.: s.n.], 2006. p. 756–762.

RITTER, H. J.; MARTINETZ, T. M.; SCHULTEN, K. J. *Neural Computation and Self-Organizing Maps: An Introduction*. Reading, MA: Addison-Wesley, 1992.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958.

RUMELHART, D. E.; MCCLELLAND, J. L.; EDITORS the P. R. G. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986.

SAMARIA, F.; HARTER, A. Parameterisation of a stochastic model for human face identification. In: *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*. [S.l.: s.n.], 1994.

SANTOS, A. R. *Identificação de faces humanas através de PCA-LDA e redes neurais SOM*. Dissertação (Mestrado) — Universidade de São Paulo, Departamento de Engenharia Elétrica-EESC, São Carlos, São Paulo, Setembro 2005.

SHAH, S.; PALMIERI, F. Meka - a fast, local algorithm for training feedforward neural networks. In: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)'90*. [S.l.: s.n.], 1990. v. 3, p. 41–46.

SHAWE-TAYLOR, J. et al. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, v. 44, n. 5, p. 1926–1940, 1998.

SINGH, A. K.; KELLEY, K.; AGARWAL, R. Interpreting results of clinical trials: A conceptual framework. *Clinical Journal of the American Society of Nephrology*, v. 3, p. 1246–1252, 2008.

SMOLA, A. J. et al. *Advances in Large Margin Classifiers*. Cambridge, Massachusetts: The MIT Press, 2000.

SONKA, M.; HLAVAC, V.; BOYLE, R. *Image Processing: Analysis and Machine Vision*. [S.l.]: CL-Engineering, 1998.

SOTO, M. T. *Face verification based on Support Vector Machines*. Dissertação (Mestrado) — Swiss Federal Institute of Technology EPFL, Lausanne, Sitzerland, Junho 2005.

SPECHT, D. F. A general regression neural network. *IEEE Transactions on Neural Networks*, v. 2, n. 6, p. 568–576, 1991.

STAN, O.; KAMEN, E. W. A new local linearized least squares algorithm for training feedforward neural networks. *IEEE Transactions on Neural Networks*, v. 11, n. 2, p. 487–495, 1997.

STIRLING. 2008. Disponível em http://pics.psych.stir.ac.uk/.

STITSON, M. O.; WESTON, J. A. E. *Implementational issues of support vector machines*. [S.l.], 1996.

SUGANTHAN, P. N. Hierarchical overlapped som's for pattern classification. *IEEE Transactions on Neural Networks*, v. 10, n. 1, p. 193–196, 1999.

TAN, A.-H. Adaptive resonance associative map. *Neural Networks*, v. 8, n. 3, p. 437–446, 1995.

TARASSENKO, L.; ROBERTS, S. Supervised and unsupervised learning in radial basis function classifiers. *IEE Proceedings — Vision, Image and Signal Processing*, v. 141, n. 4, p. 210–216, 1994.

TESAURO, G.; KEPHART, J. O.; SORKIN, G. B. Neural networks for computer virus recognition. *IEEE Expert*, v. 11, n. 4, p. 5–6, 1996.

TOH, K. A.; YAU, W. Y. Fingerprint and speaker verification decisions fusion using a functional link network. *IEEE Transactions on Systems, Man and Cybernetics*, C-35, n. 3, p. 357–370, 2005.

VAPNIK, V. *The Nature of Statistical Learning Theory*. [S.l.]: Springer, 1995.

VAPNIK, V. *Statistical Learning Theory*. [S.l.]: Wiley-Interscience, 1998.

VARGA, I.; KISS, I. Speech recognition in mobile phones. In: TAN, Z.-H.; LINDBERG, B. (Ed.). *Automatic Speech Recognition on Mobile Devices and over Communication Networks*. [S.l.]: Springer, 2008. p. 301–325.

VERLEYSEN, M.; FRANÇOIS, D. The curse of dimensionality in data mining and time series prediction. In: CABESTANY, J.; PRIETO, A.; SANDOVAL, F. (Ed.). *Computational Intelligence and Bioinspired Systems*. [S.l.]: Springer-Verlag, 2005, (Lecture Notes in Computer Science vol. 3512). p. 758–770.

WICKRAMASINGHE, L. K.; ALAHAKOON, L. D.; SMITH-MILES, K. A novel Episodic Associative Memory model for enhanced classification accuracy. *Pattern Recognition Letters*, v. 28, p. 1193–1202, 2007.

WIDROW, B.; LEHR, M. 30 years of adaptive neural networks: Perceptron, Madaline and backpropagation. *Proceedings of the IEEE*, v. 78, n. 9, p. 1415–1442, 1990.

WYNS, B. et al. Prediction of diagnosis in patients with early arthritis using a combined Kohonen mapping and instance-based evaluation criterion. *Artificial Intelligence in Medicine*, v. 31, n. 1, p. 45–55, 2004.

XIAO, Y.-D. et al. Supervised self-organizing maps in drug discovery. 1. robust behavior with overdetermined data sets. *Journal of Chemical Information and Modeling*, v. 45, n. 6, p. 1749–1758, 2005.

XU, X.; AHMADI, M. A human face recognition system using neural classifiers. In: *Proceedings of the 4th International Conference on Computer Graphics, Imaging and Visualisation (CGIV)'07*. [S.l.: s.n.], 2007. p. 354–357.

ZHAO, W. et al. Face recognition: A literature survey. *ACM Computing Surveys*, v. 35, n. 4, p. 399–458, 2003.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo