



Universidade Federal do Ceará (UFC)  
Centro de Tecnologia  
Departamento de Engenharia de Teleinformática (DETI)  
Curso de Graduação de Engenharia de Teleinformática

RECONHECIMENTO DE FACES USANDO A REDE NEURAL  
FUZZY ART: UMA COMPARAÇÃO DE DESEMPENHO

Autor: THIAGO ROMCY ACCIOLY NOGUEIRA

Orientador: Prof. Dr. GUILHERME DE ALENCAR BARRETO

FORTALEZA, CEARÁ

2009

Autor: **Thiago Romcy Accioly Nogueira**

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

RECONHECIMENTO DE FACES USANDO A REDE NEURAL  
FUZZY ART: UMA COMPARAÇÃO DE DESEMPENHO

Monografia de Conclusão de Curso apresentada à Coordenação do Curso de Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como parte dos requisitos exigidos para obtenção do grau de **Engenheiro de Teleinformática**.

Área de Concentração: **Inteligência Computacional Aplicada**

FORTALEZA, CEARÁ

2009

*Dedico este trabalho a minha Família  
Alberto, Fernanda, Renata e Érica  
pelo amor e apoio por toda a vida.*

# *Agradecimentos*

Aos meus pais Alberto e Fernanda, pela dedicação e amor por toda a vida.

Ao meu orientador, Guilherme de Alencar Barreto, pelo ensinamento durante minha vida acadêmica e principalmente neste trabalho.

À minha família.

À Deus, por todas as conquistas.

Aos colegas do *Grupo de Aprendizagem de Máquinas*(GRAMA) e *Samsung Movement Tracking and Research Group*(SMTRG), que muito contribuíram com esta Monografia.

Aos colegas do LATIN, em especial ao Leonardo Aguayo, Isaque Monteiro e Magnus Cruz, por ajudarem na escrita desta monografia.

À Samsung, pela infraestrutura disponibilizada.

*“Faça o que puder, com o que tiver, onde estiver.”*

**Theodore Roosevelt**

# *Resumo*

Esta monografia tem como principal objetivo propor um método de reconhecimento de faces baseado na rede neural Fuzzy ART, rede competitiva auto-organizável, com o qual se busca tempos de treinamento e reconhecimento reduzidos, com baixa exigência de recursos de memória, ao mesmo tempo que provê taxas de reconhecimento aceitáveis. O desempenho do método proposto foi testado em dois bancos de faces, YALE-1 e CMU, que contêm faces sob diferentes condições de iluminação, posições e expressões faciais. Redimensionamento da imagem e Análise das Componentes Principais (PCA) foram usadas para reduzir a dimensionalidade dos padrões de entrada gerados a partir da vetorização das imagens digitais das faces.

Nove classificadores foram gerados a partir do método proposto, onde o desempenho de cada um foi avaliado nos bancos de faces supracitados. Os resultados obtidos foram comparados com aqueles produzidos pela rede Perceptron Multicamadas (MLP). Os tempos de execução de cada rede durante as etapas de treinamento e reconhecimento foram também determinados para fins de comparação. Dentre as nove redes baseadas em Fuzzy ART, estudou-se ainda o número de categorias geradas, a fim de mensurar o grau de economia de recursos de memória.

Os resultados experimentais indicam que as redes propostas fazem o reconhecimento a uma velocidade bastante superior quando comparada com a rede MLP para os dois bancos de faces utilizados, com taxas de acerto inferiores, porém aceitáveis. Pode-se constatar também uma grande economia nos recursos de memória em certas redes da família Fuzzy ART.

**Palavras-Chave:** Redes Neurais Artificiais, Teoria da Ressonância Adaptativa, Redes Auto-Organizáveis, Perceptron Multicamadas, Reconhecimento de Faces.

# *Abstract*

In this dissertation we introduce a face recognition method based on the Fuzzy ART neural network, which is a self-organizing competitive neural model, aiming at developing a model with reduced training and testing times, low memory use, and acceptable recognition rates.

The performance of the proposed approach was evaluated on two well-known face image datasets, YALE-1 and CMU, which contain face images in different poses, expressions, configurations and illumination conditions. Image resizing and transformation through Principal Components Analysis (PCA) were performed to reduce the dimensionality of the pattern vectors obtained by vectorizing the available face images.

Nine classifiers were designed based on the proposed methodology, and the performance of each one was evaluated on the aforementioned face image datasets. The obtained results were compared to those produced by the Multilayer Perceptron (MLP) network. The execution times of each evaluated network during the training and recognition phases were also computed for the purpose of comparison. In addition, for the nine Fuzzy ART based classifiers, a systematic study of the number of categories generated by each one was also carried out in order to evaluate the economy in memory use.

The experimental results obtained for the two face image datasets indicate that the proposed classifiers have recognition rates smaller than those obtained by the MLP classifier. However, the main conclusion is that the proposed Fuzzy ART classifiers are able to perform face recognition in a execution time much smaller that of a MLP network, with still acceptable recognition rates. Furthermore, it was possible to verify that some of the Fuzzy ART based classifiers require much less memory than others as a function of the vigilance parameter.

**Keywords:** Artificial Neural Networks, Adaptive Resonance Theory, Self-Organized Networks, Multilayer Perceptron, Face Recognition.

## *Lista de Figuras*

2.1	Indivíduos que compõem o banco de imagens CMU. . . . .	6
2.2	Variações faciais existentes no banco de imagens CMU para um dado indivíduo. . . . .	7
2.3	Indivíduos que compõem o banco de imagens YALE-1. . . . .	8
2.4	Variações faciais existentes no banco de dados YALE-1 para um dado indivíduo. . . . .	9
2.5	Metodologia de projeto e validação de classificadores usada nesta monografia.	10
2.6	Importância relativa das primeiras 50 componentes principais associadas às imagens do conjunto YALE-1 selecionadas para treinamento: (a) Importância individual (b) Importância Acumulada . . . . .	13
2.7	Importância relativa das primeiras 80 componentes principais associadas às imagens do conjunto CMU selecionadas para treinamento: (a) Importância individual (b) Importância Acumulada . . . . .	13
3.1	Rede ART-1. (a): esquema das conexões entre $\mathbf{A}$ e $\mathbf{R}$ , explicitando o fato de as camadas serem totalmente interconectadas. (b): detalhe das conexões $\mathbf{B}$ e $\mathbf{T}$ . (c): representação completa da rede. . . . .	20
3.2	Interpretação geométrica da evolução dos pesos da rede Fuzzy ART: (a) O padrão $\mathbf{P}$ nunca foi ativado, sendo representado por um ponto. O padrão de entrada $\mathbf{f}^M$ expande a abrangência de $\mathbf{P}$ em (b). . . . .	27
3.3	Categorização de uma região anular pela rede Fuzzy ART. . . . .	28
4.1	Taxa de acerto médio $\times$ parâmetro $\rho$ . . . . .	35
4.2	Número de categorias $\times$ Tempo de treinamento . . . . .	38
4.3	Número de categorias $\times$ Tempo de reconhecimento . . . . .	38
4.4	Tempos de Treinamento médios das diversas redes avaliadas . . . . .	39

---

4.5	Tempos de reconhecimento médios das diversas redes avaliadas . . . . .	40
4.6	Taxa de acerto médio $\times$ parâmetro $\rho$ . . . . .	42
4.7	Número de categorias $\times$ Tempo de treinamento para a rede Fuzzy ART . .	44
4.8	Número de categorias $\times$ Tempo de reconhecimento para a rede Fuzzy ART	44
4.9	Tempos de Treinamento médios das diversas redes . . . . .	45
4.10	Tempos de reconhecimento médios das diversas redes . . . . .	46
A.1	Aplicativo em Simulink usando Fuzzy ART . . . . .	49
A.2	Tela de captura . . . . .	50
B.1	Aplicativo em GUIDE usando Fuzzy ART . . . . .	52
B.2	Aplicativo em GUIDE iniciado . . . . .	53
B.3	Reconhecimento da primeira face treinada . . . . .	54
B.4	Reconhecimento da segunda face treinada . . . . .	54
B.5	Reconhecimento da terceira face treinada . . . . .	54

## *Lista de Tabelas*

2.1	Dimensões das imagens vetorizadas antes e depois da aplicação de PCA. . . . .	14
3.1	Algoritmo Fuzzy ART. . . . .	29
4.1	Parâmetros utilizados na rede neural Fuzzy ART . . . . .	33
4.2	Parâmetros utilizados na rede neural MLP . . . . .	33
4.3	Desempenho obtido para as faces YALE. . . . .	34
4.4	Número de categorias geradas para banco YALE-1 . . . . .	36
4.5	Tempo de treinamento para o banco de faces YALE . . . . .	37
4.6	Tempo de Reconhecimento para o banco de faces YALE . . . . .	37
4.7	Desempenho obtido pela rede MLP para o banco de faces YALE-1. . . . .	39
4.8	Desempenho obtido para as faces CMU. . . . .	41
4.9	Número de categorias geradas para banco CMU . . . . .	42
4.10	Tempo de treinamento para o banco de faces CMU . . . . .	43
4.11	Tempo de Reconhecimento para o banco de faces CMU . . . . .	43
4.12	Desempenho obtido pela rede MLP. . . . .	45

## *Lista de Abreviaturas e Siglas*

MLP	Perceptron Multicamadas ( <i>Multi-layer Perceptron</i> )
PCA	Análise das Componentes Principais ( <i>Principal Component Analysis</i> )
ART	Teoria da Ressonância Adaptativa ( <i>Adaptive Resonance Theory</i> )
PGM	Portable Graymap
GIF	Graphics Interchange Format
SOM	Self Organizing Maps

## *Lista de Símbolos*

$t$	Tempo discreto
$\mathcal{C}_b$	Conjunto de Faces $b$
$\mathcal{X}$	Espaço contínuo dos dados de entrada
$q$	Número de neurônios existentes: ou na camada escondida do classificador MLP
$\alpha_m$	Taxa de aprendizagem usada pela rede MLP
$R$	Épocas de treinamento usada pela rede MLP
$\gamma_1, \gamma_2$	Parâmetros de controle nas redes das Famílias ART-1 e ART-2
$\mathbf{I}$	Imagem digital com dimensão $M \times N$
$\mathbf{x}$	Imagem convertida em vetor com dimensão $M \cdot N$
$\mathbf{C}_x$	Matriz de autocovariância
$E[\cdot]$	Operador valor esperado
$\mathcal{X}$	Dados antes da aplicação do PCA
$\mathcal{Y}$	Dados transformados por PCA
$y_{il}$	Componente da imagem vetorizada
$\rho$	Parâmetro de vigilância usado na rede Fuzzy ART
$\alpha$	Parâmetro de escolha usado na ativação dos pesos na rede Fuzzy ART
$\lambda$	Parâmetro de memória usado na rede Fuzzy ART
$\mathbf{c}$	Vetor de ativação dos neurônios da rede Fuzzy ART
$\mathbf{W}$	Matriz de pesos da rede Fuzzy ART
$\mathbf{w}^k$	Vetor peso associado a um neurônio da rede Fuzzy ART
$\mathbf{e}^M, \bar{\mathbf{e}}^M$	Vetor de entrada e seu complementar no uso de <i>Complement Coding</i>

# *Sumário*

<b>Lista de Figuras</b>	<b>vii</b>
<b>Lista de Tabelas</b>	<b>viii</b>
<b>Lista de Siglas</b>	<b>ix</b>
<b>Lista de Símbolos</b>	<b>x</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Introdução . . . . .	1
1.2 Objetivo Geral . . . . .	3
1.2.1 Objetivos Específicos . . . . .	3
1.3 Organização da Monografia . . . . .	4
<b>2 Descrição dos Bancos de Dados e Metodologia Utilizados</b>	<b>5</b>
2.1 Bancos de Dados de Faces Utilizados . . . . .	5
2.1.1 Banco de Faces CMU . . . . .	6
2.1.2 Banco de Faces YALE-1 . . . . .	7
2.2 Metodologia . . . . .	10
2.3 Resumo do Capítulo . . . . .	15
<b>3 A REDE NEURAL FUZZY ART</b>	<b>17</b>
3.1 INTRODUÇÃO . . . . .	17
3.2 REDES ART ( <i>Adaptive Resonance Theory</i> ) . . . . .	18
3.2.1 Descrição Geral do Funcionamento e Propriedades . . . . .	19

3.2.1.1	Etapa 1: Ativação de <b>A</b> . . . . .	21
3.2.1.2	Etapa 2: Escolha do Vencedor em <b>R</b> . . . . .	22
3.2.1.3	Etapa 3: Atualização dos Pesos. . . . .	23
3.2.1.4	Reação à Presença de Novidades: o Papel do Parâmetro de Vigilância . . . . .	23
3.2.1.5	O Papel do Parâmetro $\gamma_2$ . . . . .	24
3.2.2	Algoritmo Fuzzy ART . . . . .	24
3.2.2.1	Resumo do Algoritmo . . . . .	25
3.2.2.2	Uso de <i>Complement Coding</i> (Codificação Complementar) . . . . .	26
3.3	CLASSIFICADOR BASEADO NA REDE FUZZY ART: UMA NOVA PROPOSTA . . . . .	30
3.3.1	Rotulação dos Neurônios . . . . .	30
3.3.2	Critério de vigilância . . . . .	30
3.3.3	Reconhecimento . . . . .	30
3.4	RESUMO DO CAPÍTULO . . . . .	31
<b>4</b>	<b>RESULTADOS</b> . . . . .	<b>32</b>
4.1	INTRODUÇÃO . . . . .	32
4.2	BANCO DE DADOS YALE-1 . . . . .	34
4.2.1	Desempenho da Rede Fuzzy ART . . . . .	34
4.2.2	Número de Neurônios Gerados . . . . .	35
4.2.3	Tempos de Treinamento e Reconhecimento . . . . .	36
4.2.4	Desempenho da Rede MLP . . . . .	39
4.3	BANCO DE FACES CMU . . . . .	41
4.3.1	Taxas de Acerto . . . . .	41
4.3.2	Numero de Neurônios Gerados . . . . .	42
4.3.3	Tempos de Treinamento e Reconhecimento . . . . .	43
4.3.4	Desempenho da Rede MLP . . . . .	45

---

4.4 RESUMO DO CAPÍTULO . . . . .	46
<b>5 CONCLUSÃO</b>	<b>47</b>
5.1 CONCLUSÕES FINAIS . . . . .	47
5.2 PERSPECTIVAS . . . . .	48
<b>Apêndice A - CLASSIFICADOR NO SIMULINK</b>	<b>49</b>
A.1 BLOCOS E SUAS FUNÇÕES . . . . .	49
<b>Apêndice B - CLASSIFICADOR NO GUIDE</b>	<b>52</b>
B.1 GUIDE . . . . .	52
<b>Referências</b>	<b>55</b>

# 1 *Introdução*

*“Nas grandes batalhas da vida, o primeiro passo  
para a vitória é o desejo de vencer!”*

**Mahatma Gandhi**

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

## 1.1 *Introdução*

O cérebro tem maneiras especiais de reconhecer objetos que têm importância especial para ele, como faces. As diferenças físicas entre faces não são grandes, mas são distinguidas facilmente (AAMODT; WANG, 2009).

Tudo começa com a visão que se forma no olho, projetado como uma camera. Uma lente na face dos olhos foca a luz para uma camada delgada de neurônios situada na parte de trás, chamada retina. Os neurônios da retina estão dispostos como uma camada de *pixels*, e cada *pixel* detecta a intensidade de luz em determinada região do campo visual.

Três tipos de células em cone existentes na retina detectam as cores vermelha, verde e azul; esses neurônios enviam sinais cada vez mais fortes à medida que a intensidade de luz detectada por eles aumenta. Outras cores são formadas por diferentes níveis de atividade com combinações entre estes três tipos de célula. O processo é semelhante ao usado para preparar diversas cores de tintas misturando-se cores primárias, mas as cores primárias são diferentes porque a luz se mistura de forma diferente da tinta. Um quarto tipo de célula, chamado bastonete, é sensível à luz de baixa intensidade, mas não contribui para a visão em cores. É por isso que não é possível visualizar as cores tão bem em ambientes pouco iluminados.

Os neurônios estão dispostos na forma de um “mapas” do mundo visual, de modo que

as informações provenientes de pontos próximos à imagem visualizada são representadas pelo padrão dos potenciais de ação dos neurônios vizinhos em cada área visual do cérebro, da mesma forma que pontos próximos em uma imagem também estão juntos em uma fotografia desta imagem. Esta organização facilita a comunicação entre os neurônios que representam áreas vizinhas do mundo visual quando eles estão tentando compreender a que região da imagem pertencem.

Um dos primeiros problemas que o cérebro tem para resolver é determinar a luminosidade de cada parte do objeto que produziu a imagem visual. Provavelmente esta tarefa seja considerada simples, porém não é, pois a atividade neural depende da quantidade real de luz que chega ao olho, que varia de acordo com as características do objeto e com o padrão de iluminação e de sombras em uma imagem. O mesmo objeto parece bem diferente sob a luz do sol e sobre a escrivaninha, e também dependendo da parte deste objeto que está na sombra.

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 da 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 deste problema de reconhecimento impõe desafios ainda em aberto para os métodos de aprendizado de máquina. Dentre eles pode-se citar: a taxa de acerto, o tempo gasto no treinamento, o tempo gasto no reconhecimento, o número de exemplos necessários para aprender a reconhecer um rosto, a dimensionalidade dos dados necessários para o reconhecimento.

Estes desafios tornam-se ainda maiores quando se faz necessário desenvolver um método para se utilizar em sistemas embarcados. Onde velocidade de processamento e memória disponível são limitados.

## 1.2 Objetivo Geral

Construir um sistema de reconhecimento de faces baseado na rede Fuzzy ART, visando uma boa taxa de reconhecimento, baixa complexidade computacional e necessidade reduzida de memória.

### 1.2.1 Objetivos Específicos

1. Propor modificações na rede Fuzzy ART para que a mesma possa ser usada em problemas de reconhecimento de faces
2. Comparação de desempenho entre a rede Fuzzy ART e a rede Perceptron Multicamadas.
3. Desenvolver um aplicativo baseado no Simulink para reconhecimento de faces usando a rede Fuzzy ART modificada.
4. Desenvolver um aplicativo gráfico baseado no GUIDE do Matlab para reconhecimento de faces usando a rede Fuzzy ART modificada.

Este trabalho busca apresentar um método de reconhecimento de faces, baseado na rede Fuzzy ART, onde se busca uma boa taxa de acerto, baixo custo computacional e necessidade reduzida de memória. O trabalho também busca fazer a comparação do método proposto com os resultados obtidos pela rede neural Perceptron Multicamadas. Por fim, desenvolver um aplicativo gráfico baseado no GUIDE do Matlab para visualização do processo de cadastramento, treinamento e reconhecimento de faces usando a rede Fuzzy ART.

## 1.3 Organização da Monografia

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 da rede auto-organizável Fuzzy ART e em seguida os classificadores dela decorrente.
- No Capítulo 4 os classificadores baseados na rede Fuzzy ART e suas variantes são avaliados para a tarefa de classificar faces, obtendo as taxas de acerto para cada rede. Em seguida, o custo computacional destas redes, comparando-se o tempo de treinamento e reconhecimento e o custo de recursos de memória. Por fim, os bancos de faces usados são classificados pela rede Perceptron Multicamadas a fim de comparar as taxas de acerto e o tempo de treinamento e reconhecimento com a rede baseada em Fuzzy ART.
- No Capítulo 5 são feitas as conclusões e recomendações finais desta monografia.
- No Apêndice A um aplicativo gráfico baseado no SIMULINK do Matlab para visualização do processo de cadastramento, treinamento e reconhecimento de faces usando a rede Fuzzy ART é apresentado.
- No Apêndice B um aplicativo gráfico baseado no GUIDE do Matlab para visualização do processo de cadastramento, treinamento e reconhecimento de faces usando a rede Fuzzy ART é apresentado.

## 2 *Descrição dos Bancos de Dados e Metodologia Utilizados*

*“A inteligência artificial é o estudo de como fazer os computadores realizarem tarefas em que, no momento, as pessoas são melhores.”*

**Elaine Rich**

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

### 2.1 Bancos de Dados de Faces Utilizados

Nesta monografia, dois bancos de faces humanas foram utilizados. Eles são aqui denominados de CMU e YALE-1 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. Os dois bancos de faces serão descritos com mais detalhes nas seções seguintes.

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 + YALE).

Como ponto de partida para as descrições que se seguem, as seguintes definições serão usadas ao longo desta monografia:

- **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 dois conjuntos neste trabalho, a saber,  $b = \{CMU, YALE\}$ .
- **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 CMU

Este conjunto de imagens, disponibilizado por Tom Mitchell da Carnegie Mellon University (MITCHELL, 1997), é composto de 20 subconjuntos contendo o mesmo número de imagens. Cada subconjunto corresponde a diferentes imagens de um mesmo indivíduo, sendo que cada indivíduo deve ser entendido como uma classe (ver Figura 2.1). Todas as 640 imagens estão em escala de cinza no formato PGM.

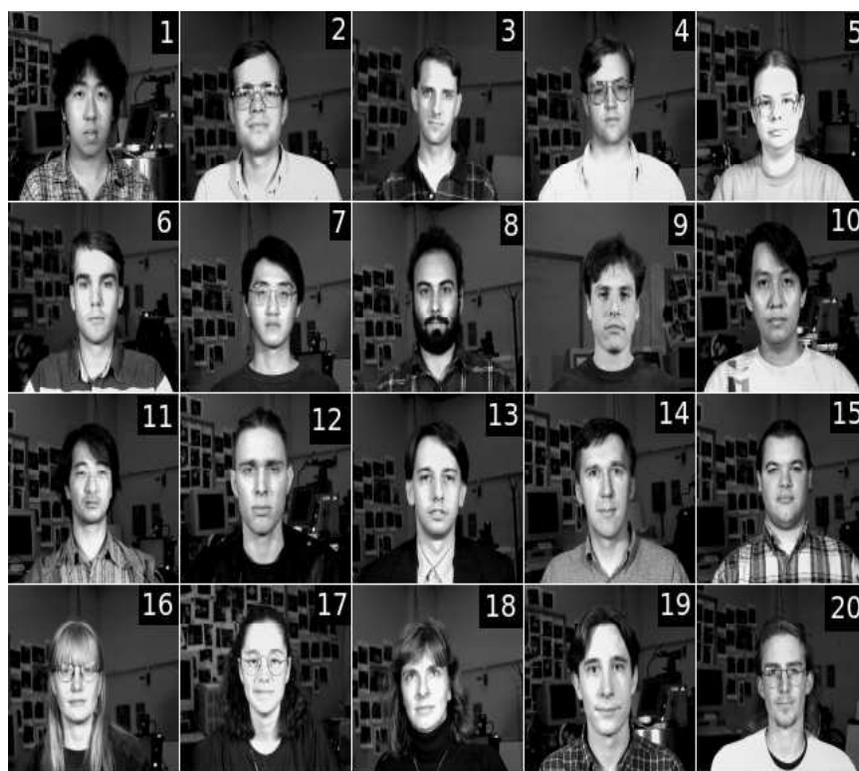


Figura 2.1 Indivíduos que compõem o banco de imagens CMU.

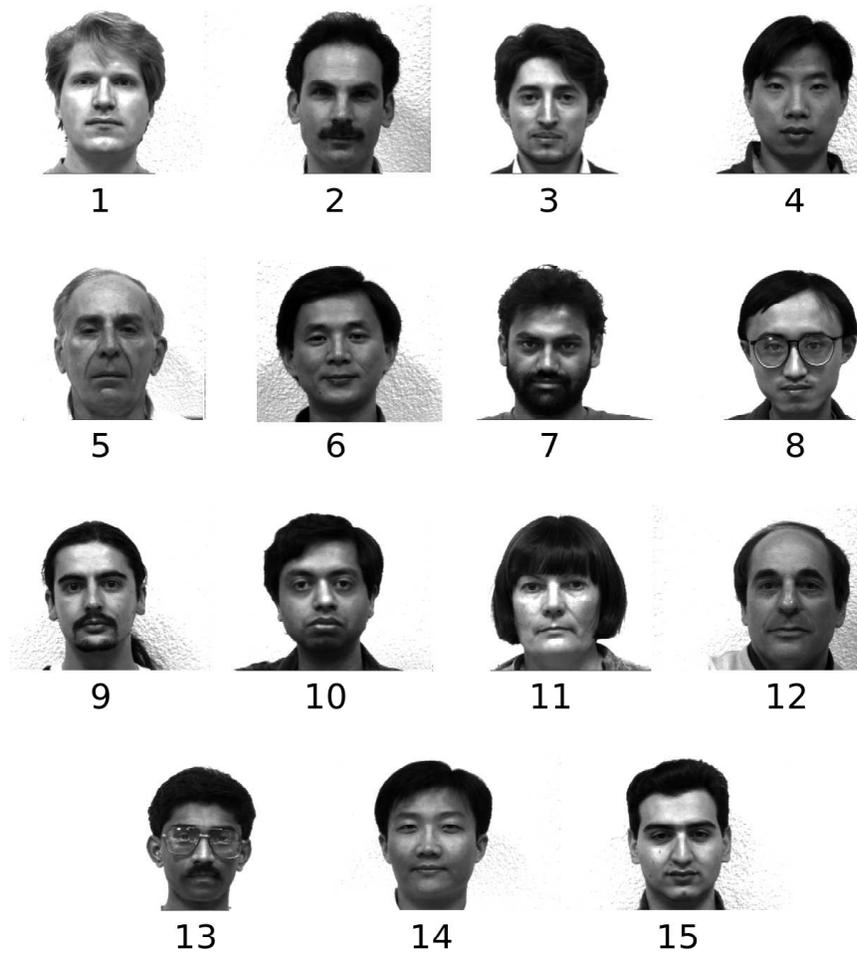
Cada subconjunto é resultado da combinação de poses (rosto em vista frontal, lateral direita, lateral esquerda, e inclinado para cima), expressões (neutralidade, alegria, tristeza e raiva) e acessórios (com e sem óculos de sol). A figura 2.2 exemplifica a variedade de cada subconjunto. O tamanho das imagens utilizadas é de  $128 \times 120$  pixels embora seja disponível também outros dois conjuntos com tamanhos inferiores ( $64 \times 60$  e  $32 \times 30$  pixels, respectivamente).



**Figura 2.2** Variações faciais existentes no banco de imagens CMU para um dado indivíduo.

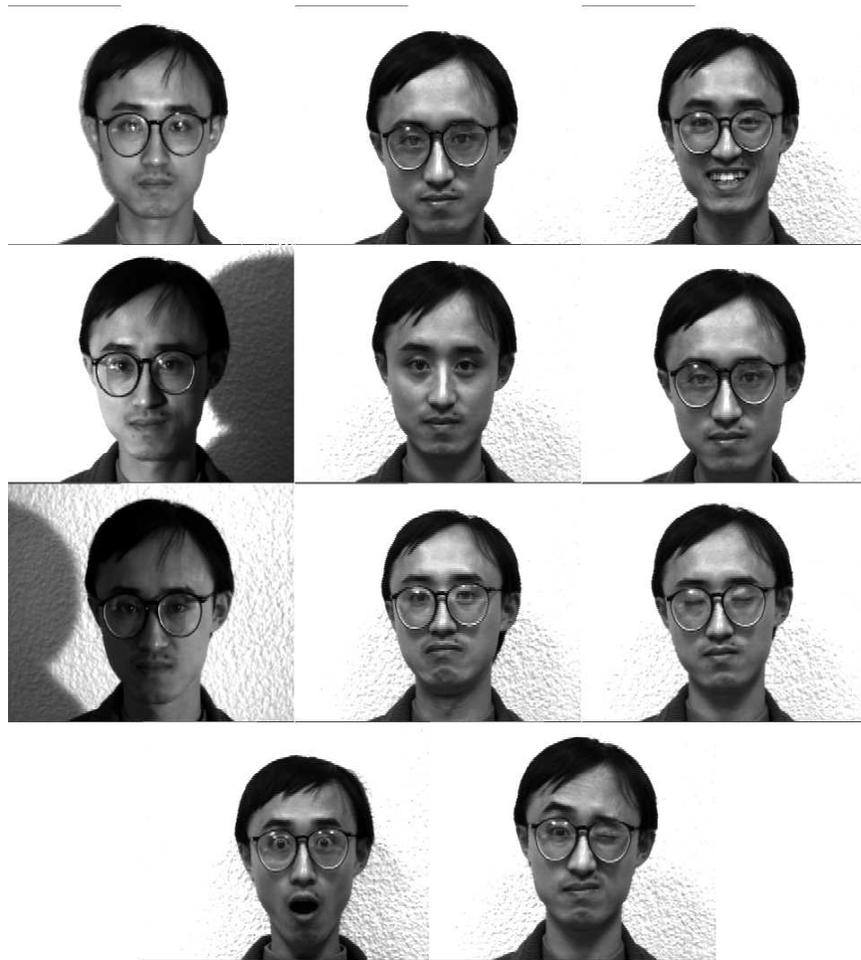
### 2.1.2 Banco de Faces YALE-1

O segundo 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, de um total de 15, possui 11 imagens em diferentes situações (ver Figura 2.3).



**Figura 2.3** Indivíduos que compõem o banco de imagens YALE-1.

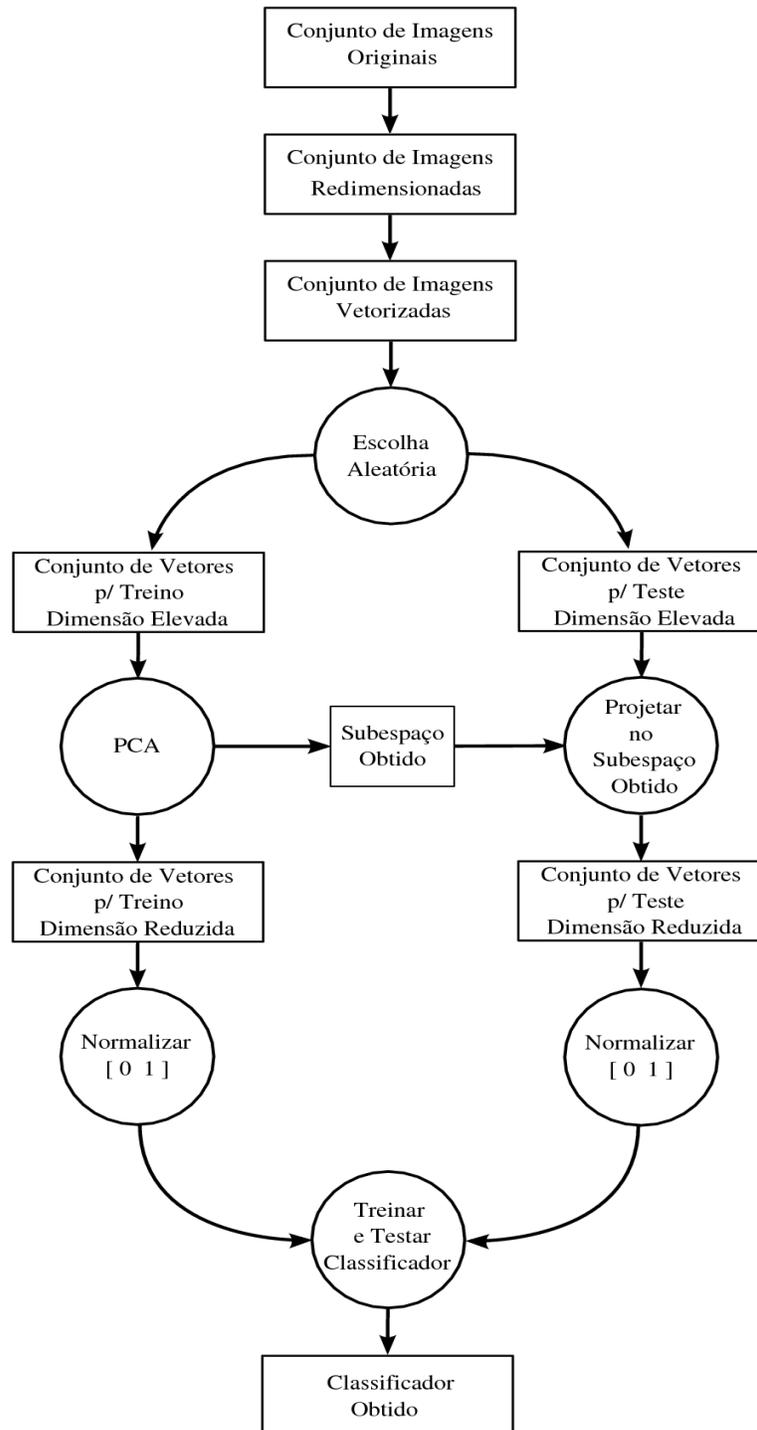
As onze situações em que um indivíduo do conjunto YALE-1 envolvem expressões faciais: tristeza, sonolência, surpresa, piscar um olho e alegria; bem como configurações diferentes: iluminação frontal, expressão normal sem óculos, expressão normal com óculos, iluminação lateral à esquerda e iluminação lateral à direita. A figura 2.4 exemplifica as variações encontradas. O tamanho de todas as imagens é  $243 \times 320$  pixels.



**Figura 2.4** Variações faciais existentes no banco de dados YALE-1 para um dado indivíduo.

## 2.2 Metodologia

Os experimentos realizados nesta monografia obedeceram a metodologia de projeto e validação ilustrada na figura 2.5, metodologia semelhante usada por Monteiro (2009). Os passos que a compõe são melhor detalhados a seguir.



**Figura 2.5** Metodologia de projeto e validação de classificadores usada nesta monografia.

## Redução das Imagens

Como foi visto, os bancos de faces utilizados têm imagens de dimensões diferentes. O banco de faces CMU utiliza imagens de  $128 \times 120$  pixels, O YALE-1,  $243 \times 320$ . O primeiro passo no processamento destas imagens foi fazer a redução de cada uma delas para imagens de  $25 \times 25$  pixels com o objetivo de diminuir o custo computacional dos passos futuros.

## 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

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

Este mapeamento se faz necessário porque classificadores de padrões manipulam vetores na entrada, e não matrizes. Neste trabalho, 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.  $25 \times 25$ ), tais como as obtidas após a redução dos dois bancos, as respectivas imagens vetorizadas terão dimensão 625, que é um número elevado. Por isso, faz-se também necessário reduzir a dimensão dos vetores a serem apresentados aos classificadores.

## Redução de Dimensionalidade dos Vetores

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 das 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. primeira componente principal ou hiperplano) é alinhada com a

direção de maior variância dos dados, a segunda componente principal é alinhada com a direção de maior variância dentre todas as direções ortogonais à primeira direção obtida, e assim por diante. Estas componentes podem ser, então, usadas para descrever os dados originais e a quantidade destas a ser utilizada é uma escolha de projeto que resulta da quantidade de variância que se deseja explicar dos dados originais.

Assim, aplicou-se PCA às imagens vetorizadas, doravante chamadas de vetores ou padrões de entrada, selecionados para treino dos classificadores e foram utilizados tantas componentes principais quantos fossem necessárias para explicar 97.5% da variância do conjunto original.

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 aqui 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. Conforme será visto adiante, esta abordagem mostrou-se satisfatória para os bancos de imagens utilizadas, visto que as taxas de reconhecimento produzidas pelos classificadores foram bastante elevadas, mesmo para imagens adquiridas sob condições adversas de iluminação, *background* e expressões faciais.

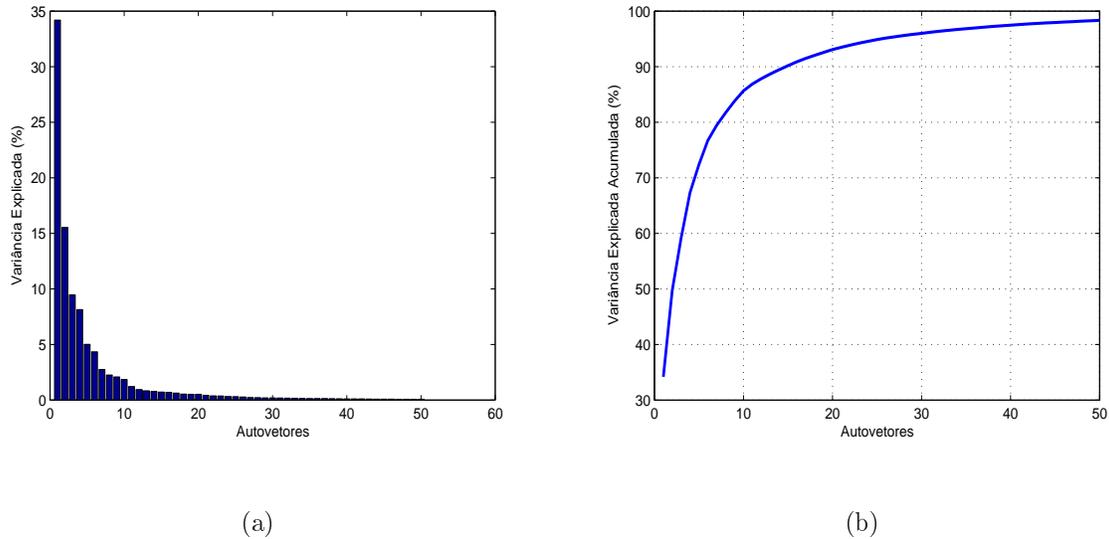
Antes de aplicar PCA, todas as imagens vetorizadas de um certo conjunto foram organizadas como colunas de uma matriz de dados  $\mathcal{X} \in \mathbb{R}^{MN} \times \mathbb{R}^P$ , em que  $P$  é o número de exemplos de treinamento. Ao aplicar PCA, faz-se necessário calcular  $M \cdot N$  autovalores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

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

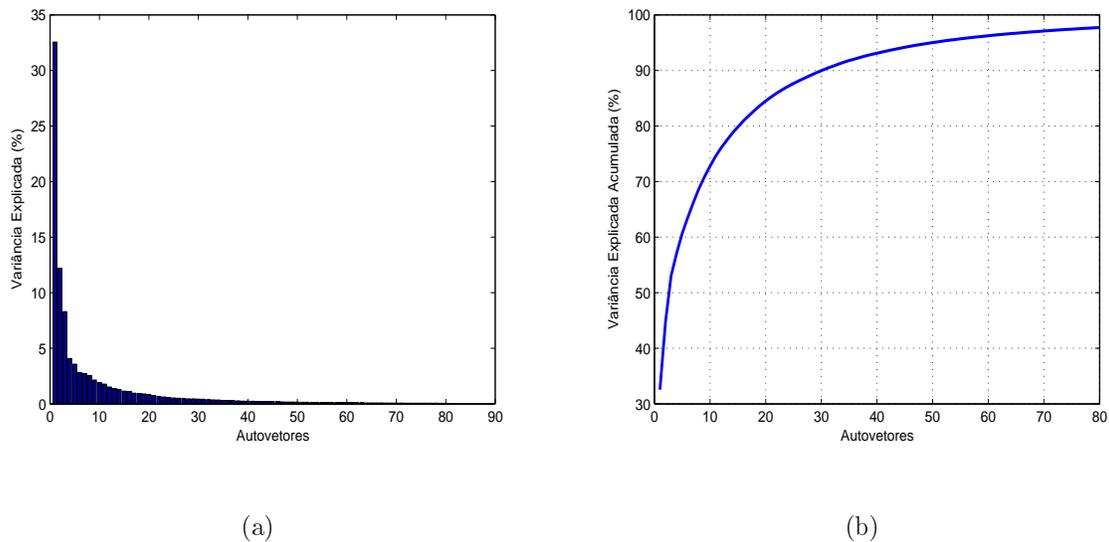
em que  $E[\cdot]$  é o operador valor esperado e o sobrescrito  $T$  denota a matriz transposta. O número de componentes principais  $L$  ( $L \ll MN$ ) a serem escolhidas definem a nova dimensão dos vetores de entrada a serem usados para treinar ou testar os classificadores avaliados neste trabalho.

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  $\mathbf{x}_i \in \mathbb{R}^{MN}$ ,  $i = 1, \dots, P$ , 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  $\mathcal{Y}$ .

Da Figura 2.6 à 2.7 são mostradas a “importância” de cada componente principal (autovetor) para uma dada matriz de covariância; ou seja, quanto uma dada componente principal explica da variância original para um certo conjunto de imagens.



**Figura 2.6** Importância relativa das primeiras 50 componentes principais associadas às imagens do conjunto YALE-1 selecionadas para treinamento: (a) Importância individual (b) Importância Acumulada



**Figura 2.7** Importância relativa das primeiras 80 componentes principais associadas às imagens do conjunto CMU selecionadas para treinamento: (a) Importância individual (b) Importância Acumulada

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 pequeno de componentes principais para explicar 97.5% 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 a seguir apresenta os valores das dimensões dos vetores antes e depois da aplicação de PCA para redução de dimensionalidade das imagens vetorizadas para os todos os bancos de faces avaliados.

**Tabela 2.1** Dimensões das imagens vetorizadas antes e depois da aplicação de PCA.

Database	#vetores	imagem original	imagem reduzida	pós PCA
CMU	640	15.360	625	77
YALE-1	165	77.760	625	41

A última coluna da tabela 2.1 representa, na prática, a quantidade de componentes principais (combinação linear de pixels das imagens) que foram obtidas das imagens selecionadas para treino. Assim, após dividir-se as imagens do conjunto, uma parte para treino e outra para teste, estas últimas foram projetadas sobre o espaço encontrado com as primeiras.

### Etapas de Aplicação do PCA em Matlab

No software Matlab a aplicação do PCA pode ser feita seguindo os seguintes comandos, onde X guarda a matriz de imagens vetorizadas em colunas como descrito nesta monografia.

```

» Cx          = cov(X);           % matriz de autocovariancia
» [aVet aVal] = eig(Ce);          % autovetores e autovalores
» aVal        = flipud(fliplr(aVal)); % ordem decrescente autovalores
» aVet        = fliplr(aVet);     % ordena autovetores
» di          = (diag(aVal));     % obtem os autovalores
» den         = sum(di);
» for i=1:length(di),             % acumula a relevancias dos autovalores
»     ve(i)    = sum(di(1:i))/den;
» end
» ve          = ve*100;
» L           = find(ve>=97.5); % configura o local de corte

```

```

» Lcut      = L(1);           % local de corte
» aVet      = aVet(:,1:Lcut); % obtem autovetores relevantes
» v        = aVet'*entrada'; % obtem dados redimensionados

```

### Normalização

Uma vez obtida a matriz de dados transformados  $\mathcal{Y}$ , a etapa seguinte consiste em efetuar uma normalização isotrópica<sup>1</sup> das componentes dos vetores  $\mathbf{y}_i$ ,  $i = 1, \dots, P$ , condicionando ao intervalo  $[0 \ 1]$ . Com isto busca-se basicamente evitar que uma dimensão se sobreponha em relação às outras, fato que pode comprometer sobremaneira o desempenho de certos tipos de classificadores de padrões.

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

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

em que

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

## 2.3 Resumo do Capítulo

Neste capítulo, foram brevemente descritos os dois conjuntos de faces utilizados neste trabalho e a metodologia prévia que tratou previamente estes dados para uso dos classificadores de padrões que serão descritos nos capítulos seguintes. Esta etapa de pré-processamento pode ser, a grosso modo, resumida em três sub-etapas:

1. Redução das imagens.
2. Vetorizar as imagens.
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

---

<sup>1</sup>Isotropia no espaço: não existem direções no espaço privilegiadas, ou, equivalentemente, identificáveis. Qualquer direção do espaço é equivalente às outras direções.

seja descrita. No capítulo seguinte serão apresentados os classificadores baseados na rede auto-organizável Fuzzy ART.

## 3 A REDE NEURAL FUZZY ART

*“A experiência é aquilo que lhe permite reconhecer um erro quando você o comete de novo.”*

**Earl Wilson**

Neste terceiro Capítulo, discute-se em detalhes o algoritmo da rede neural auto-organizável Fuzzy ART.

### 3.1 INTRODUÇÃO

Redes neurais competitivas não-supervisionadas constituem uma classe de redes neurais artificiais usada para construir uma representação estatística compacta de um conjunto de dados de entrada não-rotulados. Os primeiros modelos de redes neurais competitivas surgiram a partir de estudos que investigavam hipóteses sobre a organização dos neurônios em determinadas áreas do córtex cerebral - por exemplo, o córtex visual - a partir de estímulos sensoriais e como esta organização neuronal está relacionada ao aprendizado (GROSSBERG, 1976; RUMELHART; ZIPSER, 1985; GROSSBERG, 1987).

De acordo com Muszkat (2006), uma dessas hipóteses está fundamentada no paradigma cognitivo que entende o cérebro como um ecossistema em que os próprios neurônios vivem em situação de competição e organização ditadas pelos estímulos sensoriais provenientes do ambiente em que o portador do cérebro está inserido.

Com base na metáfora do cérebro como um sistema auto-organizável, modelos de redes neurais competitivas tentam reproduzir dois níveis de auto-organização: (i) a formação de padrões de conexões e (ii) a formação de padrões de atividade. Para implementá-los, há basicamente duas abordagens. A primeira constrói modelos no domínio do tempo contínuo, em que os modelos são formulados por meio de equações diferenciais (KELSO, 1995), sendo a teoria da sinérgica (HAKEN, 2004) e a teoria do campo de redes neurais auto-

organizáveis (AMARI, 1983) duas das principais representantes desta linha de pesquisa. A segunda abordagem formula modelos no domínio do tempo discreto, tendo experimentado grande avanço nos últimos 20 anos em função da popularização do computador digital. Os principais representantes desta linha de pesquisa são os modelos competitivos propostos por Kohonen (KOHONEN, 1982, 1990). Há ainda modelos que foram inicialmente propostos em tempo contínuo, mas que tiveram versões posteriormente adaptadas para tempo discreto, tais como os modelos da família ART (*Adaptive Resonance Theory*) (CARPENTER; GROSSBERG, 2003).

Qualquer que seja a abordagem adotada para construir uma rede neural competitiva, a idéia básica da competição entre neurônios é a seguinte: neurônios da camada de saída competem entre si pelo direito de responder, ou seja, de permanecerem ativos para um determinado estímulo de entrada. Ao final da competição, apenas um neurônio (ou um pequeno grupo de neurônios) estará ativo em resposta àquela informação de entrada. Em geral, ao longo do processo de ajuste das conexões sinápticas, um neurônio atuará como um detector de características (*feature detector*) (RUMELHART; ZIPSER, 1985), ou seja, ele passará a indicar, por meio de seu disparo, a ocorrência de um padrão de entrada que possua um determinado conjunto de características próprias do grupo a que o padrão de entrada pertence. Isto é possível por que os neurônios de uma rede competitiva extraem propriedades estatísticas (médias) do conjunto de padrões de entrada.

Redes neurais auto-organizáveis da família ART são modelos computacionais que tem como características, além da sua capacidade de formar agrupamentos, o rápido aprendizado, o uso econômico dos recursos de memória e a estabilidade temporal do conhecimento armazenado.

## 3.2 REDES ART (*Adaptive Resonance Theory*)

Ao final dos anos 80 e início da década de 90, Grossberg, Carpenter *et al.* (CARPENTER; GROSSBERG, 1987b, 1987a, 1988; CARPENTER *et al.*, 1991b, 1991a, 1991c, 1992) propuseram as primeiras arquiteturas da família ART e suas principais variantes: ART-1, ART-2, ART-2a, ARTMAP, Fuzzy-ART e Fuzzy-ARTMAP. Tais arquiteturas fornecem uma solução a um problema enfrentado por sistemas adaptativos de reconhecimento de padrões: o dilema estabilidade-plasticidade (CARPENTER; GROSSBERG, 1987b): como lidar com o compromisso entre duas tarefas antagônicas - (*i*) adaptar os pesos da rede em resposta a novos padrões e (*ii*) manter a informação referente aos padrões antigos. As

próximas subseções têm por objetivo expor o funcionamento básico deste tipo de arquitetura de RNAs e mostrar como redes ART podem ser úteis como sistemas (ou parte de sistemas) detectores de novidades.

### 3.2.1 Descrição Geral do Funcionamento e Propriedades

Grossberg propôs (CARPENTER; GROSSBERG, 1987b) na arquitetura funcional da rede ART uma idéia simples: caso o padrão de entrada seja suficientemente diferente dos padrões já armazenados na memória de longo prazo da rede, associe a este padrão diferente um neurônio ainda não utilizado na rede.

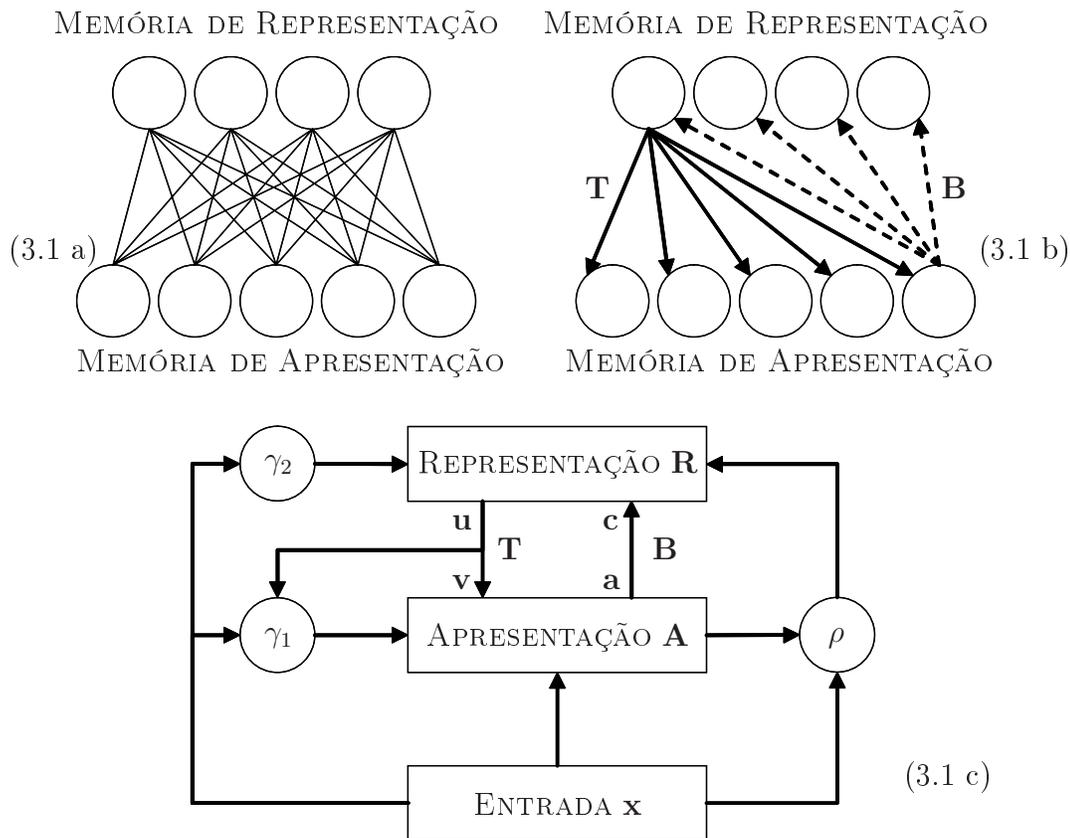
No funcionamento das redes ART existe, portanto, um limiar que é usado para decidir sobre o grau de similaridade entre informação armazenada e informação apresentada. Tal limiar é definido por um parâmetro chamado **parâmetro de vigilância**, denotado por  $\rho$ . Note-se que, por construção, a rede ART se torna especialmente adequada para a detecção de novidades.

Como em qualquer projeto de especificação de uma rede neural, há o ponto delicado de se especificar os parâmetros externos à rede (i.e., não auto-ajustáveis) de modo a prover os melhores resultados. Considere-se, por exemplo, as redes de Kohonen. Existem alguns parâmetros que devem ser definidos de acordo com o problema de identificação:

1. Número de neurônios da grade;
2. Forma da função vizinhança  $\mathcal{H}_n^{i,j}$  e, no caso de ser gaussiana, sua variância  $\sigma_n^2$ ;
3. Valores iniciais e finais para a taxa de aprendizagem  $\eta_n$ .

Existem, portanto, ao menos três parâmetros externos que podem influenciar o desempenho da rede SOM. Aqui surge uma vantagem das redes ART do ponto de vista de configuração: especificar adequadamente um único valor, o do parâmetro de vigilância.

As redes da família ART têm em comum um ciclo básico de funcionamento: inicialmente, apresenta-se à rede um padrão de entrada  $\mathbf{x}$ , o qual gera um vetor de saída ou ativação  $\mathbf{a}$  em uma primeira camada de neurônios, chamada aqui de **camada de apresentação**. As ativações  $\mathbf{a}$  são ponderadas por uma matriz de transmissão  $\mathbf{B}$ , resultando em um padrão  $\mathbf{c}$  apresentado a uma segunda camada de neurônios  $\mathbf{R}$ , uma **camada de recuperação**. Nela ocorre um processo de competição que determina um conjunto de ativações  $\mathbf{y}$ , tal que apenas um componente de  $\mathbf{y}$  não é nulo. Quando  $\mathbf{y}$  é ativado, gera



**Figura 3.1** Rede ART-1. (a): esquema das conexões entre **A** e **R**, explicitando o fato de as camadas serem totalmente interconectadas. (b): detalhe das conexões **B** e **T**. (c): representação completa da rede.

um sinal  $\mathbf{u}$  a ser ponderado por outra matriz de transmissão  $\mathbf{T}$ , resultando em um vetor  $\mathbf{v}$ . De volta à camada de apresentação, compara-se  $\mathbf{v}$  com o padrão de entrada  $\mathbf{x}$ . Caso  $\mathbf{v}$  seja suficientemente similar a  $\mathbf{x}$ , diz-se que ocorre um estado de **ressonância** na rede. O termo ressonância **adaptativa** pode ser justificado pelo fato de as matrizes de transmissão **B** e **T** terem seus valores ajustados dinamicamente.

A arquitetura da rede ART-1, projetada para lidar com sinais de entrada binários, está apresentada na Figura 3.1. A rede é composta por quatro bancos de memória: **A**, **R**, **B** e **T**; e de mecanismos de controle, indicados por  $\gamma_1$ ,  $\gamma_2$  e  $\rho$ . Cada elemento da rede será descrito a seguir<sup>1</sup>.

**a) Banco de memória A** . Conjunto de neurônios que realiza a interface da rede com os padrões de entrada, denotado por  $\mathbf{x}$ . Caracteriza-se por ser uma memória de curta duração, também chamada de “campo de representação de característica” (*feature representation field*). A camada **A** representa internamente  $\mathbf{x}$  mantendo as

<sup>1</sup>A nomenclatura utilizada por Grossberg e seus colaboradores é bastante particular. Termos como “mecanismo atencional” e “subsistema orientacional” são comuns na descrição das redes.

dimensões deste último: cada componente do vetor  $\mathbf{x}$  corresponde à entrada de um único neurônio nesta camada.

- b) **Banco de memória  $\mathbf{R}$**  . Assim como  $\mathbf{A}$ , compõe-se também de um conjunto de neurônios, e armazena os protótipos (categorias) de padrões já existentes. Também classificada como memória de curta duração, caracteriza-se por manter uma **representação** interna das categorias armazenadas. Na literatura é chamada de “campo de representação de categoria” (*category representation field*). Na Figura 3.1, procurou-se mostrar que as camadas  $\mathbf{A}$  e  $\mathbf{R}$  estão totalmente interconectadas, i.e., todos os neurônios de  $\mathbf{A}$  têm conexão com os de  $\mathbf{R}$  e vice-versa. Note-se ainda, na mesma Figura, como cada saída de  $\mathbf{A}$  torna-se um elemento do vetor de pesos  $\mathbf{R}$  (linhas cheias). Neste campo de memória, há um processo de competição entre os neurônios.
- c) **Memórias de longa duração  $\mathbf{B}$  e  $\mathbf{T}$**  . São chamadas memórias de longo prazo: utilizou-se  $\mathbf{B}$ , devido ao termo em inglês “bottom-up” e  $\mathbf{T}$  para “top-down”. Atualizam-se as matrizes  $\mathbf{B}$  e  $\mathbf{T}$  durante o processo de treinamento e as mesmas são responsáveis pela interação entre as camadas  $\mathbf{A}$  e  $\mathbf{B}$ .
- d) **Controles  $\gamma_1$  e  $\gamma_2$**  . Usados como sinais de controle de ativação das camadas  $\mathbf{A}$  e  $\mathbf{R}$ .
- e) **Parâmetro de Vigilância  $\rho$**  . Parâmetro de medida de semelhança entre padrão de entrada apresentado e protótipos armazenados.

A simples exposição dos blocos não é suficiente para entender a dinâmica de funcionamento da rede. Para tanto, é necessário descrever o processo de funcionamento, que será baseado na descrição dada em Carpenter et al. (1991a) e dividido em três etapas.

### 3.2.1.1 Etapa 1: Ativação de $\mathbf{A}$ .

A camada  $\mathbf{A}$  compõe-se de  $M$  neurônios, sendo cada neurônio alimentado por três grupos de entradas. A primeira é o padrão a ser classificado, um vetor de entrada  $\mathbf{x}^M = [x^1 \ x^2 \ \dots \ x^M]^T$ . A segunda, o vetor de saída da camada  $\mathbf{R}$ , já ponderada pela memória  $\mathbf{T}$ ; a terceira é um sinal de controle  $\gamma_1$ . Um neurônio  $j$  desta camada é dito **ativo** se sua saída  $a^j$  tem valor 1; neurônios inativos produzem sinal de saída igual a 0. Note-se que a camada  $\mathbf{A}$ , como um todo, produz como saída um vetor  $\mathbf{a}^M$ . A regra para determinar se um neurônio está ativo é chamada de **regra dos 2/3**: o neurônio está ativo se duas

das suas três entradas têm valor maior do que um determinado limiar. Especificamente, tem-se:

$$a^j = \begin{cases} 1 & \text{se } x^j + v^j + \gamma_1 > 1 + \bar{z} \\ 0 & \text{caso contrário,} \end{cases}, \quad j = 1, 2, \dots, M. \quad (3.1)$$

onde  $0 < \bar{z} < 1$  e

$$v^j = \sum_{k=1}^N y^k T^{kj}, \quad (3.2)$$

ou, em forma matricial,

$$\mathbf{v} = \mathbf{T}^T \mathbf{y}. \quad (3.3)$$

em que o sobrescrito  $T$  denota a operação de transposição de vetores. O parâmetro  $\gamma_1$  é um sinal de controle, definido de forma a inibir o funcionamento de  $\mathbf{A}$  quando  $\mathbf{R}$  está ativa:

$$\gamma_1 = \begin{cases} 1 & \text{se a entrada } \mathbf{x} \text{ está ativa e } \mathbf{R} \text{ inativa} \\ 0 & \text{caso contrário.} \end{cases} \quad (3.4)$$

### 3.2.1.2 Etapa 2: Escolha do Vencedor em $\mathbf{R}$ .

Nesta etapa, há uma competição entre os neurônios de  $\mathbf{R}$ , com a abordagem “vencedor leva tudo” (*winner-take-all strategy*). O primeiro passo é calcular  $\mathbf{c}^N$ , a projeção de  $\mathbf{a}^M$  sobre cada conjunto de pesos  $\mathbf{B}$ :

$$c^k = \sum_{j=1}^M a^j B^{jk}, \quad (3.5)$$

ou, em forma matricial,

$$\mathbf{c} = \mathbf{B}^T \mathbf{a}. \quad (3.6)$$

Em seguida, escolhe-se o vencedor  $i^*$  por

$$i^* = \arg \max_k \{c^k\}, \quad k = 1, 2, \dots, N. \quad (3.7)$$

Neste ponto do processo, tem-se em mãos o índice  $i^*$  do neurônio vencedor, representante da classe à qual  $\mathbf{a}^M$  pertence. Finalmente, constrói-se o vetor de saída da camada  $\mathbf{R}$ ,  $\mathbf{y}^N = [y^1 \ y^2 \ \dots \ y^N]^T$ , obedecendo

$$y^k = \begin{cases} 1 & \text{se } k = i^* \\ 0 & \text{caso contrário} \end{cases}, \quad k = 1, 2, \dots, N. \quad (3.8)$$

### 3.2.1.3 Etapa 3: Atualização dos Pesos.

Os pesos  $\mathbf{B}$  e  $\mathbf{T}$  são atualizados por regras diferentes. Considere-se, inicialmente, o aprendizado “top-down”  $\mathbf{R} \rightarrow \mathbf{A}$ , onde os pesos  $\mathbf{T}$  estão envolvidos. A regra de atualização é simplesmente

$$T^{i^*j} = a^j, \quad j = 1, 2, \dots, M. \quad (3.9)$$

ou seja, os pesos  $\mathbf{T}^{i^*}$  armazenam  $\mathbf{a}$ , quando  $i^*$  está ativo. Os pesos restantes  $T^{ij}, i \neq i^*$  não são alterados. Lembrando que a rede ART-1 lida com sinais binários, é possível mostrar (CARPENTER et al., 1991a) que, quando o neurônio vencedor  $i^*$  da camada  $\mathbf{R}$  está ativo,

$$T_n^{i^*j} = x^j \cap T_{n-1}^{i^*j}, \quad j = 1, 2, \dots, M. \quad (3.10)$$

onde  $(\mathbf{a} \cap \mathbf{b})^j = 1 \Leftrightarrow a^j = 1$  e  $b^j = 1$ .

Todos os pesos  $\mathbf{T}$  têm como valor inicial 1. Na primeira atualização, a Equação (3.10) resulta em

$$T_n^{i^*j} = x^j \cap 1 = x^j, \quad j = 1, 2, \dots, M, \quad (3.11)$$

ou seja, na primeira vez que o padrão de entrada é apresentado, fica armazenado na memória de longo prazo.

O aprendizado “bottom-up”  $\mathbf{A} \rightarrow \mathbf{R}$  é similar a  $\mathbf{R} \rightarrow \mathbf{A}$ , porém utiliza de uma normalização

$$B_n^{ji^*} = \frac{x^j \cap B_{n-1}^{ji^*}}{\delta + |x^j \cap B_{n-1}^{ji^*}|}, \quad j = 1, 2, \dots, M, \quad (3.12)$$

onde (i)  $\delta$  é uma constante positiva tal que  $\delta \ll 1$  e (ii)  $|\mathbf{x}| = \sum_{j=1}^M x^j$ .

### 3.2.1.4 Reação à Presença de Novidades: o Papel do Parâmetro de Vigilância

O parâmetro de vigilância  $\rho$  tem um papel fundamental no funcionamento da rede. Suponha-se que uma entrada específica  $\mathbf{x}$  seja tal que o vetor  $\mathbf{a}$  produzido na camada de apresentação  $\mathbf{A}$  seja muito diferente da memória de longo prazo armazenada em  $\mathbf{T}$ . Note-se que o fato de  $\mathbf{x}$  ser diferente não impede que haja a escolha de um vencedor  $i^*$  em  $\mathbf{R}$ , dado pela Equação (3.7). Porém, a escolha do vencedor leva à atualização dos pesos  $T^{i^*j}$  pela Equação (3.10), os quais por hipótese não correspondem a um bom representante para a classe à qual  $\mathbf{x}$  deve pertencer.

Deve então haver um procedimento para garantir que  $T^{i^*j}$  não seja alterado por  $\mathbf{x}$ . O algoritmo trata a entrada considerada dissimilar da seguinte forma: um sinal de entrada

no campo  $\mathbf{R}$ , representado pela seta vindo pela direita na Figura 3.1, desabilita o vencedor  $i^*$ . Este sinal é chamado por Grossberg de *sinal de reajuste*. Com  $i^*$  desabilitado, faz-se  $\mathbf{a} = \mathbf{x}$ , o que corresponde a procura de um segundo neurônio em  $\mathbf{R}$  que tenha a maior semelhança com  $\mathbf{x}$ . O processo se repete até que (i) encontra-se um representante adequado em  $\mathbf{R}$  ou (ii) seleciona-se um neurônio que ainda não tenha sido ativado.

Esta última opção corresponde, na prática, à criação de um novo neurônio específico para representar  $\mathbf{x}$ . Desta forma, a informação prévia armazenada em  $\mathbf{R}$  e nos pesos  $\mathbf{B}$  e  $\mathbf{T}$  (i.e., a memória de longo prazo) **não** é alterada devido à presença de um padrão totalmente novo. O disparo do sinal de reajuste é realizado apenas se satisfizer

$$|\mathbf{x} \cap T^{i^*j}| < \rho |\mathbf{x}|, \quad j = 1, 2, \dots, M. \quad (3.13)$$

Nota-se que  $\rho$  define a sensibilidade da rede a entradas com característica de novidade ou sinais espúrios: valores altos para  $\rho$  significam o disparo mais freqüente de novas buscas em  $\mathbf{R}$ , o que corresponde à formação mais freqüente de novas classes.

### 3.2.1.5 O Papel do Parâmetro $\gamma_2$ .

Na rede ART-1,  $\gamma_2$  tem um papel muito simples: habilita o funcionamento da camada  $\mathbf{R}$  sempre que há a presença de uma entrada  $\mathbf{x}$ . O critério de ativação pode ser alterado em outras variantes da rede ART, tais como a ARTMAP (CARPENTER et al., 1991a).

Descrito o funcionamento básico da rede ART, na próxima Subseção encontra-se uma breve exposição sobre uma variação da rede ART, a rede Fuzzy ART. Justifica-se o uso desta rede pela simplicidade de implementação, quando comparada com a da rede ART-1.

## 3.2.2 Algoritmo Fuzzy ART

A rede ART-1 tem a limitação de tratar apenas com sinais de entrada binários. Carpenter, Grossberg e Rosen, em Carpenter et al. (1991c), apresentam uma generalização do algoritmo ART-1 para lidar com sinais analógicos, por intermédio da incorporação de operadores de lógica fuzzy<sup>2</sup>. O algoritmo resultante é mais simples do que o descrito na Subseção 3.2.1, por substituir as matrizes  $\mathbf{B}$  e  $\mathbf{T}$  por uma única matriz de pesos  $\mathbf{W}$ .

Além da simplificação, a inserção do pré-processamento do vetor de entrada por intermédio da técnica de codificação complementar (*complement coding*) e a adoção dos

---

<sup>2</sup>Em português, lógica nebulosa. Adotou-se no texto o termo original em inglês.

operadores fuzzy  $\text{MIN}(\wedge)$  e  $\text{MAX}(\vee)$  (ZADEH, 1965) proporcionam uma interpretação geométrica do funcionamento do algoritmo.

### 3.2.2.1 Resumo do Algoritmo

A arquitetura estrutural é similar à da rede ART-1, já apresentada na Figura 3.1. Comentam-se aqui apenas as alterações relevantes ao funcionamento do algoritmo.

**Sinal de entrada e matriz de pesos.** Assim como no algoritmo ART-1, a entrada é um vetor de dimensão  $M$ , porém com valores analógicos. A única restrição que se faz é que as componentes estejam dentro de uma faixa restrita de valores:  $x^j \in [0, 1], j = 1, 2, \dots, M$ . A matriz de pesos  $\mathbf{W} = [\mathbf{w}^1 \mathbf{w}^2 \dots \mathbf{w}^N]^T$  é tal que ao neurônio  $k$  da camada  $\mathbf{R}$  associa-se o vetor de pesos  $\mathbf{w}^k = [w^{k1} w^{k2} \dots w^{kM}]$ . A condição inicial para os pesos é  $\mathbf{W} = \mathbf{1}^{N \times M}$ , quando os neurônios são ditos não-dedicados (*uncommitted*).

**Parâmetros externos.** São três: uma **constante**  $\alpha > 0$ , a **taxa de aprendizado**  $\lambda \in [0, 1]$  e o **parâmetro de vigilância**  $\rho \in [0, 1]$ .

**Processo de competição.** Calcula-se o vetor  $\mathbf{c} = [\mathbf{c}^1 \mathbf{c}^2 \dots \mathbf{c}^N]^T$  na camada  $\mathbf{R}$

$$\mathbf{c} = \frac{|\mathbf{x} \wedge \mathbf{w}^k|}{\alpha + |\mathbf{w}^k|}, \quad k = 1, 2, \dots, N \quad (3.14)$$

onde

$$(\mathbf{x}^k \wedge \mathbf{w}^k)^j \equiv \min(x^j, w^{kj}), \quad (3.15)$$

$|\mathbf{x}| = \sum_{j=1}^M |x^j|$  e  $\delta \ll 1$ . Busca-se em seguida o índice do vencedor  $i^*$

$$i^* = \arg \max_k \{c^k\}, \quad k = 1, 2, \dots, N. \quad (3.16)$$

**Critério de vigilância.** Para todo  $\mathbf{x}$ , realiza-se o teste de similaridade entre a entrada e o peso do neurônio vencedor

$$\frac{|\mathbf{x} \wedge \mathbf{w}^{i^*}|}{|\mathbf{x}|} \geq \rho. \quad (3.17)$$

Caso a condição seja satisfeita, atualizam-se os pesos como descrito a seguir. Caso contrário, segue-se o mesmo procedimento descrito na Subseção 3.2.1: procura-se um novo vencedor dentre os já dedicados, ou escolhe-se um não-dedicado.

**Atualização dos pesos.** Finalizada a busca pelo neurônio vencedor, atualizam-se seus

pesos  $\mathbf{w}$ :

$$\mathbf{w}_n^{i^*} = (1 - \lambda) \mathbf{w}_{n-1}^{i^*} + \lambda(\mathbf{x} \wedge \mathbf{w}_{n-1}^{i^*}), \quad (3.18)$$

onde  $0 < \lambda \leq 1$ . Caso  $i^*$  seja um neurônio não-dedicado, escolhe-se  $\lambda = 1$ , de modo que, na primeira vez que se torna ativo,  $\mathbf{w}_n^{i^*} = \mathbf{x}$ . Esta situação corresponde à criação de uma nova classe com  $\mathbf{x}$  como representante inicial.

### 3.2.2.2 Uso de *Complement Coding* (Codificação Complementar)

A técnica de *complement coding* transforma o vetor de entrada  $\mathbf{e}^M$  em um vetor  $\mathbf{x}$  de dimensão  $2M$ :

$$\mathbf{x} = \begin{bmatrix} \mathbf{e}^M \\ \bar{\mathbf{e}}^M \end{bmatrix} = \begin{bmatrix} \mathbf{e}^M \\ \mathbf{1}^{M \times 1} - \mathbf{e}^M \end{bmatrix}. \quad (3.19)$$

Com intuito de apresentar uma interpretação geométrica para o processo de atualização dos pesos utilizando-se *complement coding*, suponha-se agora que  $\mathbf{e}^M$  seja de dimensão 2, com componentes  $[e^1, e^2]$ . O vetor  $\mathbf{x}$  resultante é dado por

$$\mathbf{x} = [e^1 \quad e^2 \quad 1 - e^1 \quad 1 - e^2]^T. \quad (3.20)$$

De maneira geral, dado que agora a entrada da rede compõe-se de um vetor e de seu complemento, o vetor de pesos  $\mathbf{w}_n^j$  pode ser escrito como

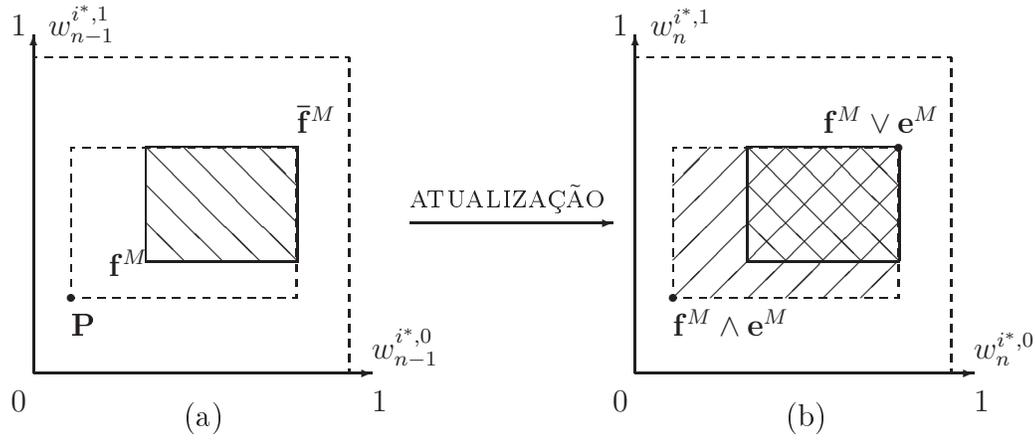
$$\mathbf{w}^j = \begin{bmatrix} \mathbf{p}^j \\ \bar{\mathbf{q}}^j \end{bmatrix}. \quad (3.21)$$

O vetor  $\mathbf{w}^j$  define um retângulo  $\Omega$  no espaço bidimensional  $\mathbb{R}^2$ , se tomarmos as coordenadas de  $\mathbf{p}^j$  e  $\mathbf{q}^j$  como sendo as coordenadas de vértices opostos do retângulo.

Considere-se agora o processo de atualização dos pesos do neurônio vencedor  $\mathbf{w}_n^{i^*}$ , dada pela Equação (3.18). Por simplicidade, tome-se  $\lambda = 1$ . Na primeira vez que o vencedor é ativado, todos os pesos têm valor 1, portanto

$$\mathbf{w}_n^{i^*} = \mathbf{x} \wedge \mathbf{1}^{2M \times 1} = \mathbf{x} = \begin{bmatrix} \mathbf{e}^M \\ \bar{\mathbf{e}}^M \end{bmatrix}. \quad (3.22)$$

As coordenadas no  $\mathbb{R}^2$  serão, portanto,  $(\mathbf{e}^M, \bar{\mathbf{e}}^M) = (\mathbf{e}^M, \mathbf{e}^M)$ , definindo um retângulo com diagonais opostas idênticas, correspondente ao ponto  $\mathbf{P}$  da Figura 3.2. Este exemplo é ilustrativo de como um padrão novo é interpretado como uma região pontual no espaço de características.



**Figura 3.2** Interpretação geométrica da evolução dos pesos da rede Fuzzy ART: (a) O padrão  $\mathbf{P}$  nunca foi ativado, sendo representado por um ponto. O padrão de entrada  $\mathbf{f}^M$  expande a abrangência de  $\mathbf{P}$  em (b).

Na próxima vez que o mesmo neurônio for declarado vencedor, devido agora a uma segunda entrada  $\mathbf{f}^M$ , os pesos serão atualizados para

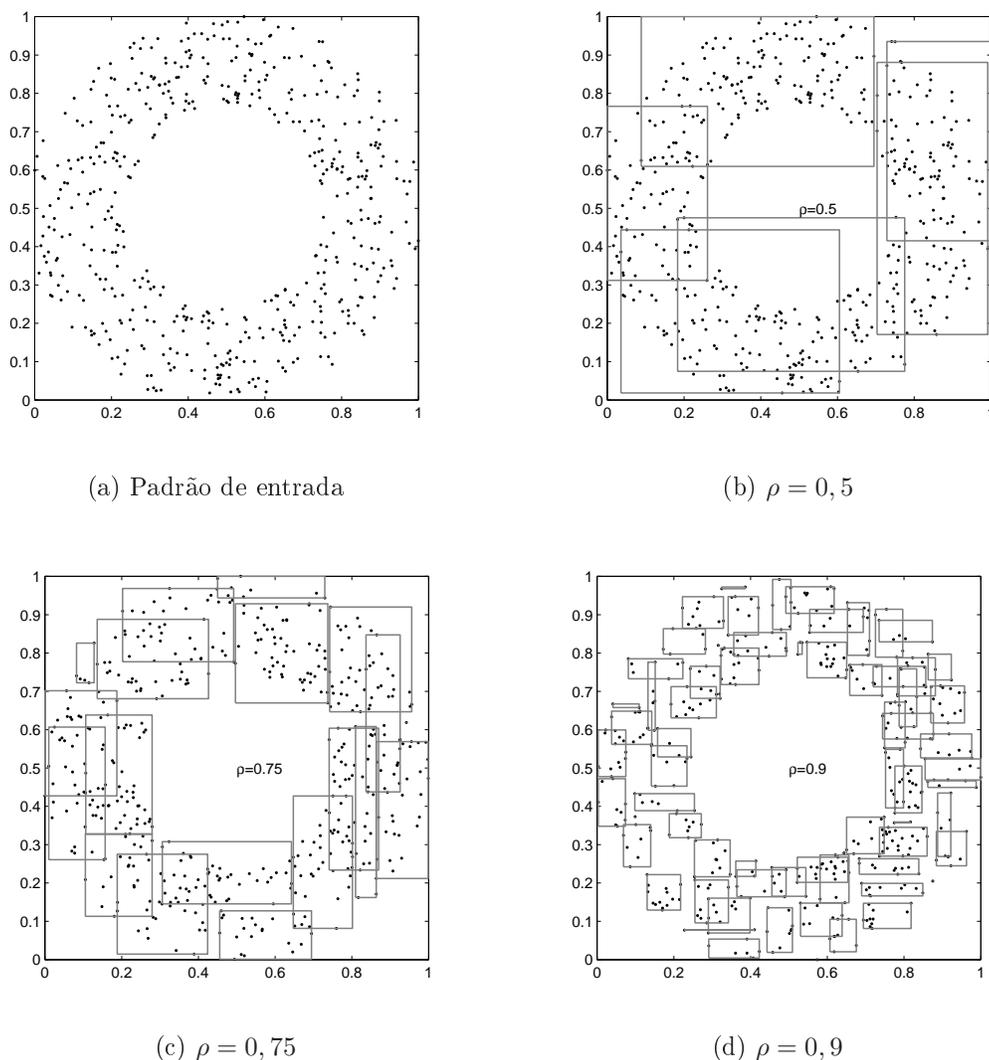
$$\mathbf{w}_n^{i^*} = \begin{bmatrix} \mathbf{f}^M \\ \bar{\mathbf{f}}^M \end{bmatrix} \wedge \mathbf{w}_{n-1}^{i^*} = \begin{bmatrix} \mathbf{f}^M \wedge \mathbf{e}^M \\ \bar{\mathbf{f}}^M \wedge \bar{\mathbf{e}}^M \end{bmatrix} = \begin{bmatrix} \mathbf{f}^M \wedge \mathbf{e}^M \\ \frac{\mathbf{f}^M \wedge \mathbf{e}^M}{\mathbf{f}^M \vee \mathbf{e}^M} \end{bmatrix}, \quad (3.23)$$

onde  $(\mathbf{e}^M \vee \mathbf{f}^M)^j \equiv \max(e^j, f^j)$ . Note-se que os vértices de  $\Omega$  definem o menor retângulo que contém as coordenadas do vetor de pesos anterior e a entrada atual: a região antes definida por apenas um ponto foi agora expandida de modo a acomodar os padrões  $\mathbf{e}^M$  e  $\mathbf{f}^M$ , ilustrando a plasticidade da rede. Qualquer outro padrão de entrada cujas coordenadas estiverem inscritas em  $\Omega$  ativarão o mesmo neurônio  $i^*$ . O limite para o crescimento de  $\Omega$  é definido pelo parâmetro de vigilância. Em Carpenter et al. (1991c) prova-se que a máxima região para um determinado padrão  $\Omega$  é dada por

$$|\Omega| \leq M(1 - \rho), \quad (3.24)$$

onde  $M$  é a dimensão do espaço de entrada. Por esta fórmula, compreende-se imediatamente a restrição imposta pelo parâmetro de vigilância à generalização que a rede ART pode realizar: valores altos para  $\rho$  levam a pequenas regiões para  $\Omega$  no espaço de entrada, enquanto um parâmetro de vigilância baixo permite uma generalização excessiva.

Para ilustrar simultaneamente qual o efeito do parâmetro de vigilância e como a rede Fuzzy ART cria novas categorias, tomou-se um exemplo bidimensional com uma distribuição de pontos no plano  $(x, y)$  dentro de uma região anular, mostrada na Figura 3.3(a). Utilizou-se a técnica de *complement coding* de modo a exemplificar a interpretação geométrica exposta, para um caso específico.



**Figura 3.3** Categorização de uma região anular pela rede Fuzzy ART.

Nota-se claramente que para valores de  $\rho$  relativamente baixos (por exemplo  $\rho < 0,5$ , Figura 3.3(b)) o número de categorias também é pequeno: notam-se apenas seis retângulos que cobrem a região anular, correspondentes às seis categorias formadas. Para  $\rho = 0,75$  e  $\rho = 0,9$ , o número de categorias resultante foi de 18 e 67, respectivamente mostrados nas Figuras 3.3(c) e 3.3(d). A granularidade das categorias aumenta consideravelmente, confirmando a restrição imposta pela Equação (3.24) para a região de abrangência dos protótipos armazenados nos pesos da rede.

Um resumo do algoritmo Fuzzy ART encontra-se na Tabela 3.1.

REDE FUZZY ART - ALGORITMO BÁSICO	
<b>Constantes</b>	
<b>Símbolo</b>	<b>Valor típico</b>
$L$ : número de neurônios da rede	Depende do problema
$\alpha$ : parâmetro de escolha, $\alpha \ll 1$	0.01
$\rho$ : parâmetro de vigilância, $0 < \rho \leq 1$	$0.1 < \rho < 1$
$\lambda$ : parâmetro de memória, $0 < \lambda \leq 1$	$0.5 < \lambda \leq 1$
<b>Entradas</b>	
$\mathbf{x}_n$ : vetor de entrada, dimensão $d$	
<b>Algoritmo</b>	
<b>1. Inicialização (<math>n = 0</math>)</b>	
Inicialize os pesos $\mathbf{w}_0^j = 1$ , $j = 1, 2, \dots, L$	
<b>2. Laço temporal (<math>n = 1, 2, \dots, n_{\text{MÁX}}</math>)</b>	
2.1 Tome $\mathbf{x}_n$ do conjunto de vetores de entrada	
2.2 Busca pelo índice do neurônio vencedor: calcule	
$\mathbf{c} = \frac{ \mathbf{x}_n \wedge \mathbf{w}_n^j }{\alpha +  \mathbf{w}_n^j }, \quad j = 1, 2, \dots, L$	
$i^* = \arg \max_j \{c^j\}$	
2.3 Teste de ressonância	
SE $ \mathbf{x}_n \wedge \mathbf{w}_n^{i^*}  > \rho  \mathbf{x}_n $ , vá para o passo 2.4	
SENÃO, volte para o passo 2.2 e busque um novo vencedor	
2.4 Atualização dos pesos	
SE $\mathbf{w}_n^{i^*} = \mathbf{1}$ (i.e., o vencedor nunca foi ativado antes)	
ENTÃO FAÇA	
$L = L + 1 \text{ e } \mathbf{w}^L = \mathbf{1} \text{ (i.e., crie um novo neurônio)}$	
$\mathbf{w}_n^{i^*} = \mathbf{x}_n \text{ (i.e., o vencedor armazena o novo padrão)}$	
SENÃO FAÇA	
$\mathbf{w}_n^{i^*} = (1 - \lambda) \mathbf{w}_{n-1}^{i^*} + \lambda (\mathbf{x} \wedge \mathbf{w}_{n-1}^{i^*})$	
<b>Saídas ou variáveis de interesse</b>	
$i^*$ : índice do vencedor	$\mathbf{w}_n^j$ : pesos dos neurônios, dimensão $d$
Após o treinamento, pode ser útil observar o erro de quantização $\mathbf{e}_n^{q,j} = (\mathbf{x}_n - \mathbf{w}_n^j)$	
<b>Observações</b>	
Tipicamente, usa-se a técnica de <i>complement coding</i> para pré-processar $\mathbf{x}_n$ .	

Tabela 3.1 Algoritmo Fuzzy ART.

## 3.3 CLASSIFICADOR BASEADO NA REDE FUZZY ART: UMA NOVA PROPOSTA

Por definição, o algoritmo Fuzzy ART é um algoritmo voltado para o *clustering*, ou seja, agrupa de modo não- supervisionado conjunto de dados em categorias diferentes, conforme suas semelhanças. O classificador baseado em Fuzzy ART utiliza esta característica para separar os dados em grupos distintos, representados por neurônios, rotulando cada neurônio com a classe que ele pertence. Um dado neurônio só terá seus pesos atualizados por uma entrada que pertença a mesma classe dele. Cada classe presente no banco de dados terá pelo menos um neurônio para representá-la. As modificações propostas são apresentadas a seguir.

### 3.3.1 Rotulação dos Neurônios

O vetor de entrada para treinamento deverá ter um rótulo, referente a classe a qual pertence. Quando um novo neurônio é considerado vencedor para um vetor de entrada, ele recebe o rótulo do vetor, passando a representar esta classe até o fim do treinamento. O neurônio só será considerado vencedor para um determinado vetor de entrada se ambos são representantes da mesma classe ou se o neurônio não representar classe alguma. Este controle é realizado pelo critério de vigilância, como será visto a seguir.

### 3.3.2 Critério de vigilância

Para que o neurônio seja considerado vencedor e este possa atualizar seus pesos com o novo vetor de entrada, ele não apenas terá que satisfazer a condição da Equação (3.17), como também uma das duas condições seguintes:

- a) o vetor de entrada deve pertencer a mesma classe do neurônio vencedor, previamente rotulado por um vetor de entrada anterior.
- b) o neurônio vencedor deve não ter rótulo algum, ou seja, um novo neurônio criado devido ao vetor de entrada não ser semelhante o suficiente com os vetores anteriores.

### 3.3.3 Reconhecimento

Após o treinamento, os pesos de cada neurônio deverão ser armazenados para se fazer o reconhecimento de indivíduos. O vetor de entrada que se quer classificar irá passar

novamente pelo processo de competição dado na Equação (3.14), onde será calculado a ativação de cada neurônio para o dado de entrada. Neste processo será achado o índice neurônio vencedor. A classe que este neurônio representa será considerada a classe do vetor de entrada.

### 3.4 RESUMO DO CAPÍTULO

Neste Capítulo, apresentaram-se definições e conceitos relacionados à Teoria da Resonância Adaptativa (ART, *Adaptive Resonance Theory*) e a modificações propostas para o projeto de um classificador de padrões baseado na rede Fuzzy ART.

## 4 RESULTADOS

*“Escolhas excluem, para cada sim existe um não.”*

**John Gardner**

Neste Capítulo, discutem-se os resultados obtidos através do algoritmo proposto em comparação com a rede neural Perceptron Multicamadas.

### 4.1 INTRODUÇÃO

Serão apresentados todos os resultados referentes as taxas de acerto para os dois bancos de faces avaliados. O número de categorias geradas pela rede Fuzzy ART é diretamente proporcional ao grau de generalização da mesma, representando assim a economia de recursos de memória, pois quanto menos protótipos (neurônios) existem, menos memória é necessária. Por fim, a comparação do tempo gasto para as tarefas de treinamento e reconhecimento das redes testadas. Importante ressaltar que os tempos das várias redes foram obtidos nas mesmas condições, em um mesmo computador. Os resultados referentes aos tempos servem apenas para base de comparação entre as várias redes, podendo divergir de um sistema para outro, porém mantendo sua proporcionalidade entre as redes, o que torna a comparação válida. A metodologia adotada segue critérios de validação estatísticos para quantificar o tempo gasto.

Cada rede terá 100 épocas de treinamento/teste independentes. Os pesos serão reiniciados e o banco de faces será embaralhado a cada nova época.

De antemão, alguns parâmetros foram configurados para se obter um estudo mais apurado do método proposto. Na Rede Fuzzy ART existem três parâmetros: o parâmetro de vigilância  $\rho$ , que define o grau de generalização da rede; o parâmetro de escolha  $\alpha$ , usado no cálculo de ativação dos neurônios; e o parâmetro de memória  $\lambda$ , usado na atualização

dos pesos.

Neste trabalho foram utilizados os valores mostrados na Tabela 4.1:

**Tabela 4.1** Parâmetros utilizados na rede neural Fuzzy ART

Redes Fuzzy ART	$\rho$	$\alpha$	$\lambda$
FART-C1	0,1	0,01	1,0
FART-C2	0,2	0,01	1,0
FART-C3	0,3	0,01	1,0
FART-C4	0,4	0,01	1,0
FART-C5	0,5	0,01	1,0
FART-C6	0,6	0,01	1,0
FART-C7	0,7	0,01	1,0
FART-C8	0,8	0,01	1,0
FART-C9	0,9	0,01	1,0

Pode-se notar que a diferença entre as redes será apenas o parâmetro  $\rho$ , pois este é o que causa maior impacto no desempenho da rede. O parâmetro  $\alpha$  deverá ser sempre um valor muito menor que um. O parâmetro  $\lambda$  deverá ser sempre 1,0, o que faz com que a Equação (3.18) passe a ser escrita apenas como:

$$\mathbf{w}_n^{i*} = \mathbf{x} \wedge \mathbf{w}_{n-1}^{i*} \quad (4.1)$$

Para avaliar os resultados obtidos pelo método proposto, este será comparado com a rede neural Perceptron Multicamadas (MLP, da sigla em inglês). Nesta rede também existem alguns parâmetros, que foram configurados conforme mostra a Tabela 4.2, onde  $q$  é o número de neurônios na camada oculta,  $\alpha_m$  é a taxa de aprendizagem e  $R$  é o número de épocas de treinamento da rede. Esta monografia usou o mesmo número de neurônios na camada oculta usado por Monteiro (2009) quando foi avaliado o banco de faces YALE-1.

**Tabela 4.2** Parâmetros utilizados na rede neural MLP

Redes MLP	$q$	$\alpha_m$	$R$
MLP-C1	25	0,3	15
MLP-C2	25	0,2	75
MLP-C3	25	0,1	150

Os parâmetros foram assim configurados para se ter uma rede MLP-C1 com poucas épocas de treinamento, porém com alta taxa (velocidade) de aprendizagem, fazendo com que esta rede tenha um treinamento rápido, além de uma segunda rede MLP-C2 com o número de épocas de treinamento e taxa de aprendizagem intermediárias, e por último, uma rede MLP-C3, mais lenta em seu treinamento, porém com uma taxa de acerto maior que as demais.

## 4.2 BANCO DE DADOS YALE-1

Nesta seção serão mostrados os resultados obtidos para o banco de faces YALE-1.

### 4.2.1 Desempenho da Rede Fuzzy ART

Analisando os resultados da Tabela 4.3, o classificador FART-C8 se destaca como o de melhor desempenho para reconhecer os indivíduos no conjunto de faces YALE-1.

**Tabela 4.3** Desempenho obtido para as faces YALE.

Redes Testadas	Taxas de Reconhecimento(%)			
	<i>média</i>	<i>mínima</i>	<i>máxima</i>	<i>desvio padrão</i>
FART-C1	80,42	60,61	93,94	7,16
FART-C2	79,58	63,64	96,97	7,09
FART-C3	81,09	63,64	93,94	6,27
FART-C4	80,21	63,64	93,94	6,92
FART-C5	81,00	51,52	96,97	7,41
FART-C6	82,70	57,58	96,97	7,53
FART-C7	85,36	69,70	100,00	6,32
FART-C8	86,30	57,58	96,97	6,60
FART-C9	85,79	66,67	100,00	3,13

Nota-se que as taxas de acerto média tendem a ser estatisticamente similares nas redes FART-C1, FART-C2, FART-C3, FART-C4 e FART-C5. Observa-se que a taxa de acerto apresenta uma tendência de crescimento com o aumento do parâmetro  $\rho$ , chegando ao seu valor máximo em  $\rho = 0,8$  como mostra a tabela 4.1.

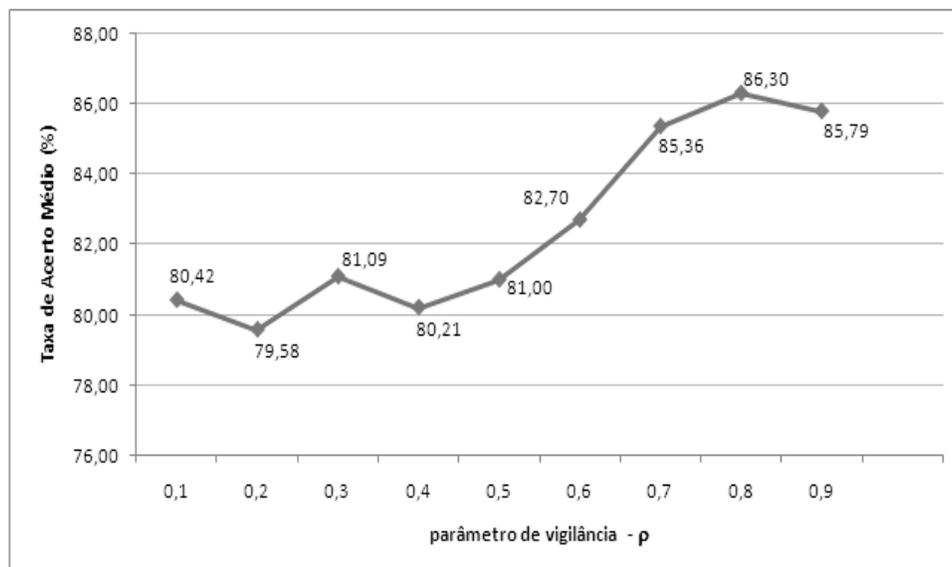


Figura 4.1 Taxa de acerto médio  $\times$  parâmetro  $\rho$

A taxa de acerto semelhante entre as redes FART-C1 a FART-C5 se deve ao número de categorias que elas geram, métrica que será discutida na seção seguinte.

#### 4.2.2 Número de Neurônios Gerados

O número de categorias geradas pela rede neural Fuzzy- ART é uma informação importante, tanto para se verificar o custo computacional relacionado ao tempo quanto o relacionado à economia de recursos de memória. Quanto maior o número de categorias, maior será o custo computacional como um todo. Cada categoria é representada por um neurônio. Na Tabela 4.4 podemos verificar o número de categorias geradas para cada rede Fuzzy ART.

**Tabela 4.4** Número de categorias geradas para banco YALE-1

<b>Redes Testadas</b>	Número de Categorias			
	<i>médio</i>	<i>mínimo</i>	<i>máximo</i>	<i>desvio padrão</i>
FART-C1	15,00	15,00	15,00	0,00
FART-C2	15,00	15,00	15,00	0,00
FART-C3	15,00	15,00	15,00	0,00
FART-C4	15,00	15,00	15,00	0,00
FART-C5	16,38	15,00	18,00	0,79
FART-C6	22,04	19,00	25,00	1,17
FART-C7	33,33	37,00	30,00	1,54
FART-C8	49,96	44,00	55,00	2,06
FART-C9	81,88	75,00	88,00	2,67

Pode-se portanto observar que as redes FART-C1 até FART-C4 sempre irão gerar apenas 15 categorias para o banco de faces YALE-1, conseguindo assim reduzir bastante seu custo computacional. Nota-se que para o banco YALE-1, 15 categorias é a quantidade mínima que ele pode gerar, pois existem 15 indivíduos, e cada um deve gerar ao menos uma categoria para representá-lo.

### 4.2.3 Tempos de Treinamento e Reconhecimento

Todos os tempos contidos nesta monografia foram obtidos com os comandos tic()-toc() do Matlab em cada rotina de treinamento e reconhecimento.

Os tempos são mostrados nas Tabelas 4.5 e 4.6 respectivamente.

**Tabela 4.5** Tempo de treinamento para o banco de faces YALE

<b>Redes Testadas</b>	Tempo de Treinamento (segundos)			
	<i>médio</i>	<i>mínimo</i>	<i>máximo</i>	<i>desvio padrão</i>
FART-C1	0,092675	0,089882	0,101573	0,001965
FART-C2	0,092051	0,089311	0,098474	0,001761
FART-C3	0,091704	0,088828	0,097410	0,001348
FART-C4	0,091593	0,088917	0,096668	0,001255
FART-C5	0,100495	0,188410	0,090891	0,014630
FART-C6	0,132810	0,107018	0,187473	0,026634
FART-C7	0,194426	0,152695	0,321068	0,045921
FART-C8	0,247844	0,205246	0,480937	0,046906
FART-C9	0,392744	0,352085	0,741316	0,046904

**Tabela 4.6** Tempo de Reconhecimento para o banco de faces YALE

<b>Redes Testadas</b>	Tempo de Reconhecimento (seg)			
	<i>médio</i>	<i>mínimo</i>	<i>máximo</i>	<i>desvio padrão</i>
FART-C1	0,000238	0,000208	0,000427	0,000040
FART-C2	0,000217	0,000208	0,000309	0,000019
FART-C3	0,000213	0,000207	0,000314	0,000010
FART-C4	0,000213	0,000208	0,000360	0,000017
FART-C5	0,000232	0,000210	0,000365	0,000017
FART-C6	0,000309	0,000265	0,000408	0,000021
FART-C7	0,000461	0,000413	0,000569	0,000024
FART-C8	0,000687	0,000603	0,000805	0,000032
FART-C9	0,001115	0,001010	0,001228	0,000042

Percebe-se que as redes que geraram um número elevado de categorias tiveram uma taxa de acerto maior, porém seu custo computacional também foi elevado devido ao maior número de protótipos.

A Figura 4.2 mostra a relação entre o número de categorias geradas e o tempo médio de treinamento.

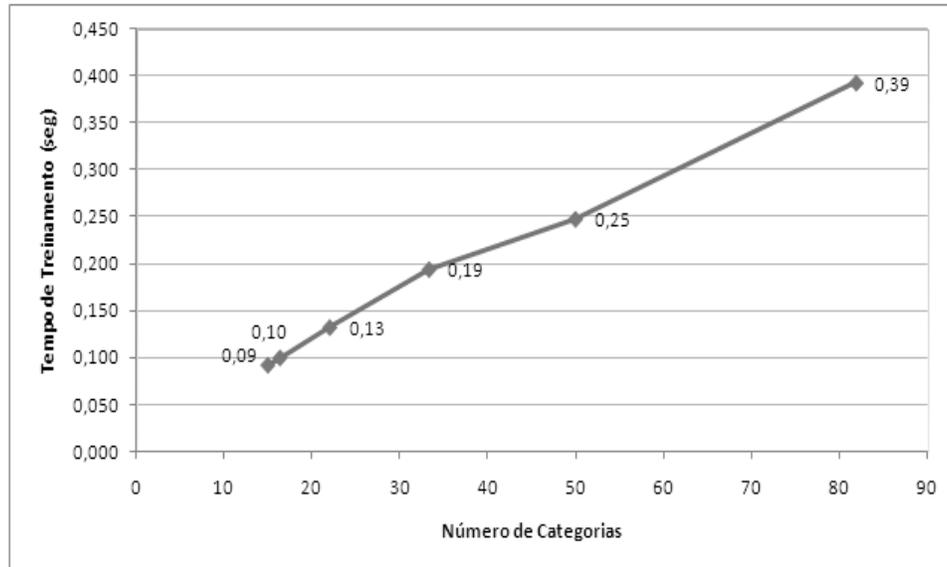


Figura 4.2 Número de categorias  $\times$  Tempo de treinamento

A Figura 4.3 mostra a relação entre o número de categorias geradas e o tempo médio de reconhecimento para o banco de dados YALE-1.

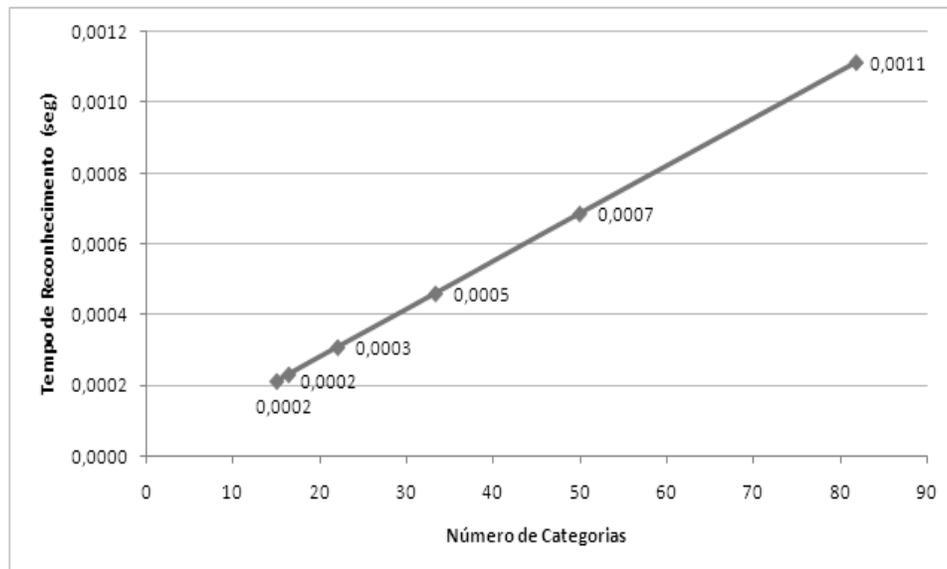


Figura 4.3 Número de categorias  $\times$  Tempo de reconhecimento

Percebe-se que o tempo de reconhecimento cresce linearmente com o número de categorias.

#### 4.2.4 Desempenho da Rede MLP

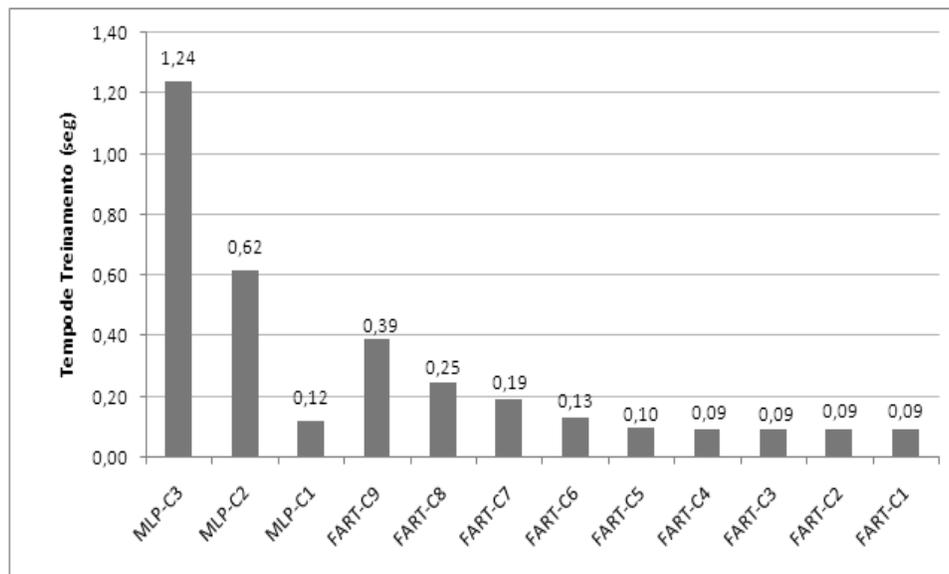
A rede MLP obteve os resultados mostrados na Tabela 4.7:

**Tabela 4.7** Desempenho obtido pela rede MLP para o banco de faces YALE-1.

<b>Redes Testadas</b>	<i>Taxa de Acerto (%)</i>	<i>Tempo para Treinamento (seg)</i>	<i>Tempo para Reconhecimento (seg)</i>
MLP-C1	92,97	0,123254	0,003536
MLP-C2	94.67	0,618593	0,003568
MLP-C3	94.70	1,240468	0,003577

Com estes resultados percebe-se que a rede neural MLP consegue uma taxa de acerto melhor quando comparada com a rede Fuzzy ART, porém as redes Fuzzy ART se destacaram das redes MLP pelo baixo tempo de treinamento e, principalmente, de reconhecimento.

Nas Figuras (4.4) e (4.5) são comparados os tempos de treinamento e reconhecimento das diversas redes testadas no banco de faces YALE-1.



**Figura 4.4** Tempos de Treinamento médios das diversas redes avaliadas

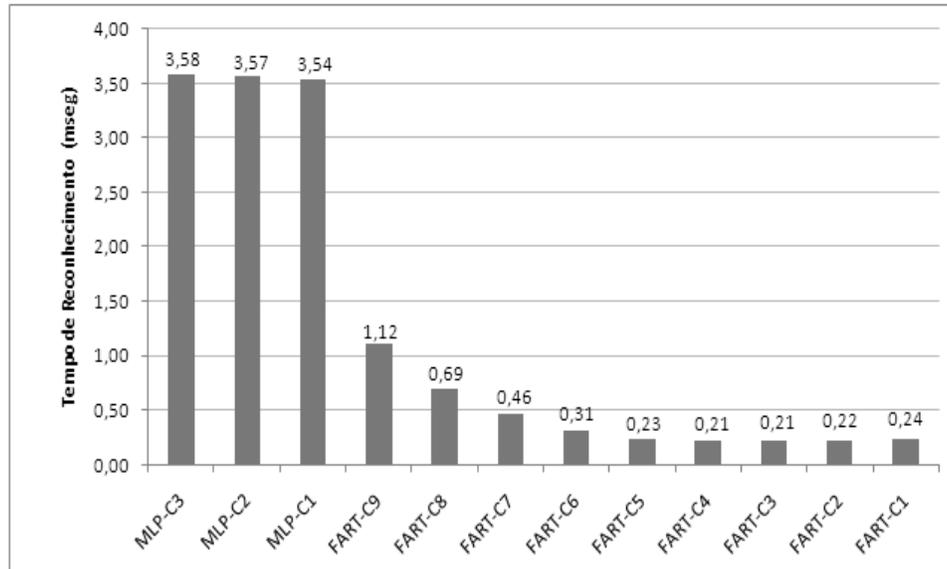


Figura 4.5 Tempos de reconhecimento médios das diversas redes avaliadas

## 4.3 BANCO DE FACES CMU

Nesta seção serão mostrados os resultados obtidos usando o banco de faces CMU.

### 4.3.1 Taxas de Acerto

Analisando os resultados da Tabela 4.8, o classificador FART-C7 foi melhor para reconhecer os indivíduos no conjunto de faces CMU, seguido de perto dos classificadores FART-C6 e FART-C8.

**Tabela 4.8** Desempenho obtido para as faces CMU.

<b>Redes</b> <b>Testadas</b>	Taxas de Reconhecimento(%)			
	<i>média</i>	<i>mínima</i>	<i>máxima</i>	<i>desvio padrão</i>
FART-C1	98,66	93,75	100,00	1,17
FART-C2	98,64	96,09	100,00	0,95
FART-C3	98,33	94,53	100,00	1,18
FART-C4	98,60	95,31	100,00	1,03
FART-C5	98,81	96,88	100,00	0,87
FART-C6	98,90	96,09	100,00	0,88
FART-C7	99,01	96,88	100,00	0,81
FART-C8	99,13	96,88	100,00	0,83
FART-C9	98,48	95,31	100,00	1,11

A Figura 4.6 mostra a evolução do desempenho da rede em função do valor do parâmetro  $\rho$ .

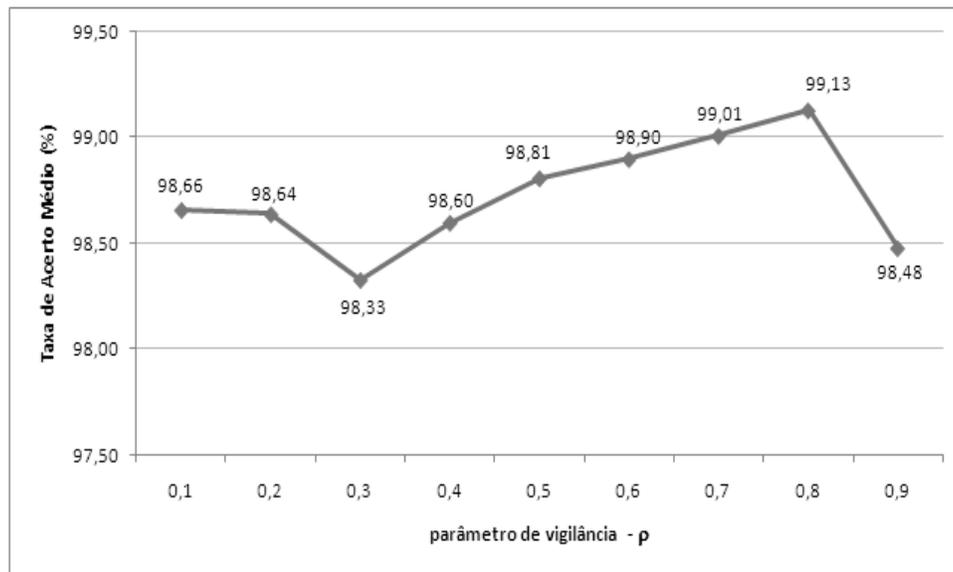


Figura 4.6 Taxa de acerto médio  $\times$  parâmetro  $\rho$

### 4.3.2 Numero de Neurônios Gerados

Na Tabela 4.9 pode-se verificar o número de categorias geradas para cada rede Fuzzy ART avaliada.

Tabela 4.9 Número de categorias geradas para banco CMU

Redes Testadas	Número de Categorias			
	<i>média</i>	<i>mínima</i>	<i>máxima</i>	<i>desvio padrão</i>
FART-C1	20,00	20,00	20,00	0,00
FART-C2	20,00	20,00	20,00	0,00
FART-C3	20,00	20,00	20,00	0,00
FART-C4	20,76	20,00	21,00	0,43
FART-C5	27,21	25,00	29,00	0,99
FART-C6	45,87	42,00	49,00	1,40
FART-C7	87,02	82,00	94,00	2,40
FART-C8	153,64	145,00	162,00	3,50
FART-C9	251,20	240,00	264,00	4,32

Pode-se portanto observar que as redes FART-C1 a FART-C3 irão gerar apenas 20 categorias, exatamente igual ao número de classes existentes, para o banco de dados CMU, conseguindo assim reduzir bastante seu custo computacional quando comparado com as

redes FART-C8 e FART-C9.

### 4.3.3 Tempos de Treinamento e Reconhecimento

Os tempos para treinamento e reconhecimento serão mostrados nas Tabelas 4.10 e 4.11 respectivamente.

**Tabela 4.10** Tempo de treinamento para o banco de faces CMU

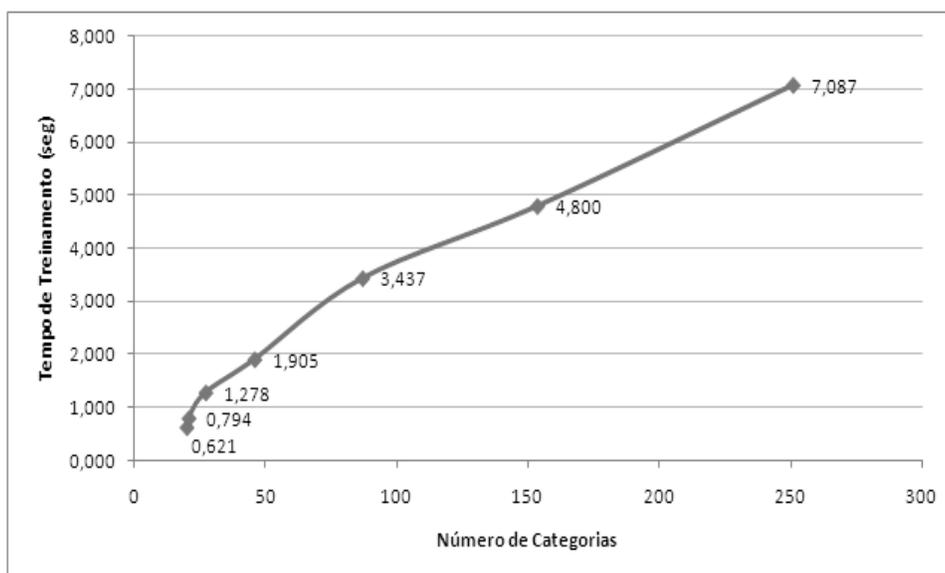
<b>Redes Testadas</b>	Tempo de Treinamento (seg)			
	<i>médio</i>	<i>mínimo</i>	<i>máximo</i>	<i>desvio padrão</i>
FART-C1	0,620850	0,614299	0,629362	0,002577
FART-C2	0,620407	0,615596	0,635810	0,003309
FART-C3	0,621898	0,617911	0,631588	0,002462
FART-C4	0,793829	0,611620	1,571216	0,276598
FART-C5	1,278137	0,724191	1,967219	0,276527
FART-C6	1,904542	1,166120	2,516147	0,300400
FART-C7	3,437082	2,076176	5,983904	0,592054
FART-C8	4,799727	3,762110	8,090744	1,128188
FART-C9	7,087001	6,169442	10,018410	1,290458

**Tabela 4.11** Tempo de Reconhecimento para o banco de faces CMU

<b>Redes Testadas</b>	Tempo de Reconhecimento (seg)			
	<i>médio</i>	<i>mínimo</i>	<i>máximo</i>	<i>desvio padrão</i>
FART-C1	0,000440	0,000437	0,000448	0,000002
FART-C2	0,000446	0,000437	0,000484	0,000006
FART-C3	0,000441	0,000436	0,000484	0,000006
FART-C4	0,000464	0,000440	0,000624	0,000020
FART-C5	0,000603	0,000547	0,000776	0,000032
FART-C6	0,001019	0,000933	0,001176	0,000041
FART-C7	0,001925	0,001795	0,002241	0,000069
FART-C8	0,003437	0,003240	0,003797	0,000091
FART-C9	0,005645	0,005412	0,005949	0,000117

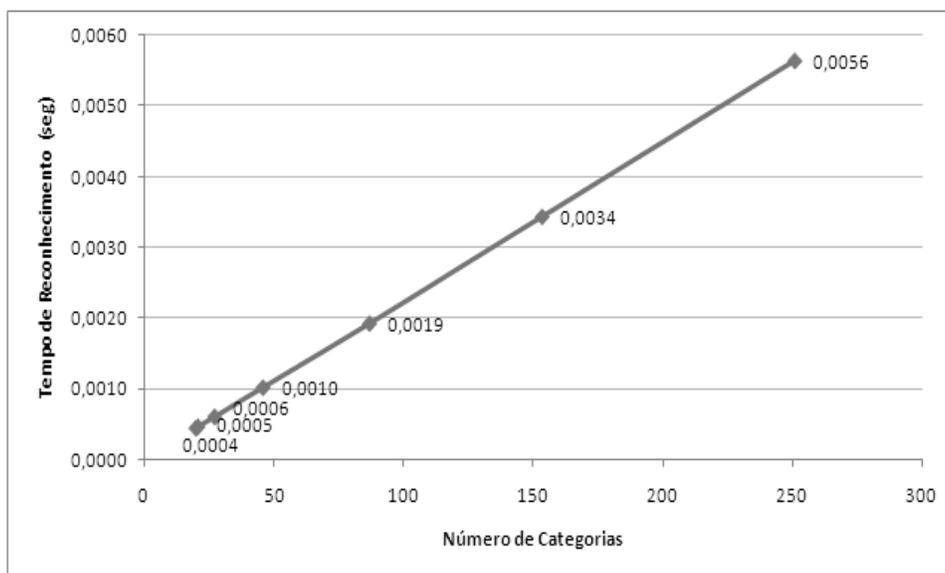
Percebe-se que as redes que geraram um número elevado de categorias tiveram uma taxa de acerto maior, porém seu custo computacional também foi elevado.

A Figura 4.7 mostra a relação entre o número de categorias geradas e o tempo médio de treinamento para o banco de dados CMU.



**Figura 4.7** Número de categorias  $\times$  Tempo de treinamento para a rede Fuzzy ART

A Figura 4.8 mostra a relação entre o número de categorias geradas e o tempo médio de reconhecimento para o banco de dados CMU.



**Figura 4.8** Número de categorias  $\times$  Tempo de reconhecimento para a rede Fuzzy ART

Novamente o tempo de reconhecimento tende a crescer linearmente com o número de categorias geradas.

### 4.3.4 Desempenho da Rede MLP

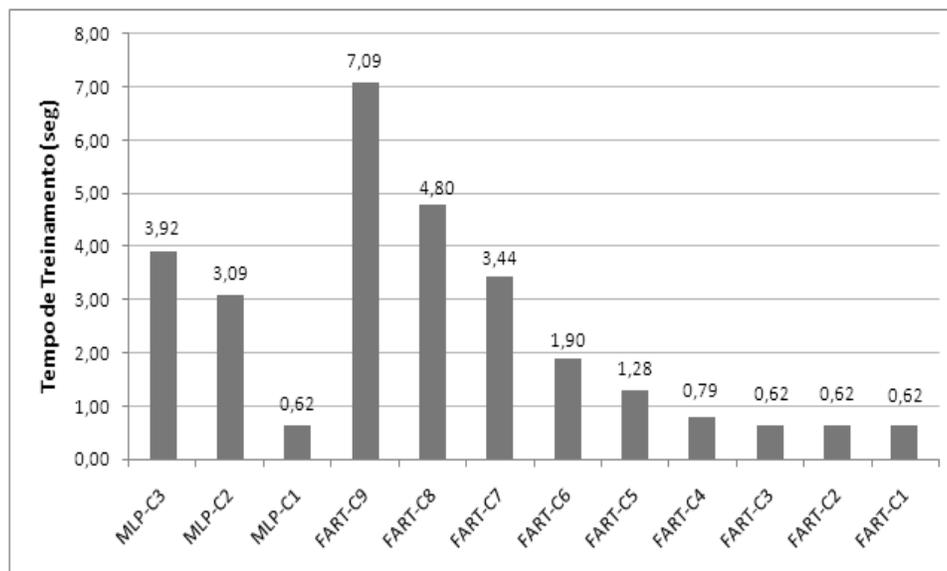
A rede MLP obteve os resultados mostrados na Tabela 4.12 para o banco de dados CMU.

**Tabela 4.12** Desempenho obtido pela rede MLP.

<b>Redes Testadas</b>	<i>Taxa de Acerto Médio (%)</i>	<i>Tempo Médio para Treinamento (seg)</i>	<i>Tempo Médio para Reconhecimento (seg)</i>
MLP-C1	99,62	0,615839	0,033264
MLP-C2	99,71	3,094871	0,033240
MLP-C3	99,77	3,916293	0,033468

Para este banco, a taxa de acerto para ambas as redes é bastante elevada, com a rede MLP obtendo melhores resultados gerais. Porém, esta é superada pela rede Fuzzy ART quando se analisa os tempos de reconhecimento, uma vez que a FART-C1 consegue determinar o neurônio vencedor usando aproximadamente 3% do tempo que a MLP-C1 usou para gerar as saídas. Os tempos de treinamento variam muito entre as redes, com os melhores tempos verificados nas redes MLP-C1 e FART-C1.

Nas Figuras 4.9 e 4.10 são comparados os tempos de treinamento e reconhecimento das diversas redes testadas.



**Figura 4.9** Tempos de Treinamento médios das diversas redes

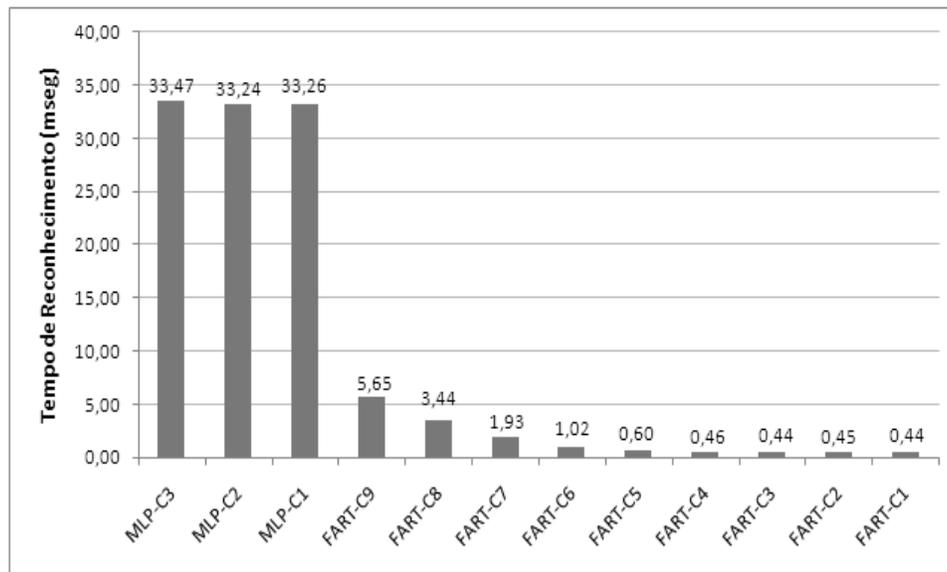


Figura 4.10 Tempos de reconhecimento médios das diversas redes

## 4.4 RESUMO DO CAPÍTULO

Neste Capítulo, apresentaram-se os resultados obtidos pelo algoritmo proposto baseado na rede Fuzzy ART. Assim como os resultados obtidos pela rede MLP. Os resultados mostraram uma velocidade de reconhecimento muito superior da rede neural Fuzzy ART, porém com taxas de acerto inferiores.

## 5 CONCLUSÃO

*“Inovação distingue um líder de um seguidor.”*

**Steve Jobs**

Neste Capítulo são apresentadas as conclusões e recomendações finais deste trabalho.

### 5.1 CONCLUSÕES FINAIS

Este trabalho propôs um método de classificação de padrões baseado na rede neural Fuzzy ART. Foi mostrado que é possível obter taxas de acerto razoáveis com um baixo custo computacional, além de conseguir economizar recursos de memória.

O classificador proposto obteve taxas de acerto de aproximadamente 83% para o banco de faces YALE-1 e 99% para o banco de faces CMU, resultados estes que podem ser considerados promissores.

A grande vantagem que o classificador Fuzzy mostrou ter foi seu custo computacional. Este classificador consegue calcular a saída de seu reconhecimento muito mais rápido do que, por exemplo, a Perceptron multicamadas. Estes resultados servem como incentivo para o uso da rede Fuzzy ART em sistemas que tenham recursos limitados, tais como sistemas embarcados.

O parâmetro de vigilância ( $\rho$ ) mostrou ser não apenas um parâmetro essencial para melhorar o desempenho referente aos tempos de treinamento e teste e à taxa de acerto, mas também referente a recursos de memória. Seu ajuste ideal é fundamental para o bom funcionamento na área de aplicação em que a rede será utilizada.

## 5.2 PERSPECTIVAS

As redes da família ART são ferramentas computacionais poderosas, e com muito espaço para pesquisa futura no campo de reconhecimento de padrões. Como recomendações de pesquisa temos:

- Introdução de um detector de novidades no classificador. A real função do parâmetro  $\rho$  é detectar um padrão considerado como novo. Durante o treinamento ele controla o número de categorias que serão criadas, sendo depois rotuladas. Contudo, durante a época de reconhecimento, este parâmetro não foi utilizado. Para um trabalho futuro, este parâmetro poderia ser utilizado na fase de reconhecimento, auxiliando na detecção de dados não treinados.
- Implementação do algoritmo proposto em um sistema embarcado, como celular ou plataformas robóticas. Como o objetivo desta rede é preferencialmente ser usada em dispositivos como estes, um possível trabalho futuro seria implementá-la em linguagem como C/C++ e embarcar para verificar seu desempenho em uma situação que realmente tenha recursos limitados.
- Verificar o impacto gerado no desempenho da rede quando o parâmetro  $\lambda$  for alterado. Na rede proposta, este parâmetro deve ser sempre igual a 1 (um), para que a rede tenha um treinamento rápido. Em aplicações em que o treinamento não precise ser rápido, o parâmetro  $\lambda$  pode vir a ser um aliado para o desempenho referente a taxa de acerto. Este trabalho não teve como objetivo testar este parâmetro, criando ficando este tópico como assunto para pesquisas futuras.

## APÊNDICE A – CLASSIFICADOR NO SIMULINK

Este apêndice apresenta uma descrição sucinta da aplicação feita em Simulink Matlab para cadastramento, treinamento e reconhecimento de faces usando a rede Fuzzy ART.

### A.1 BLOCOS E SUAS FUNÇÕES

Na figura (A.1) podemos verificar os blocos usados para a construção do aplicativo e o funcionamento de cada um.

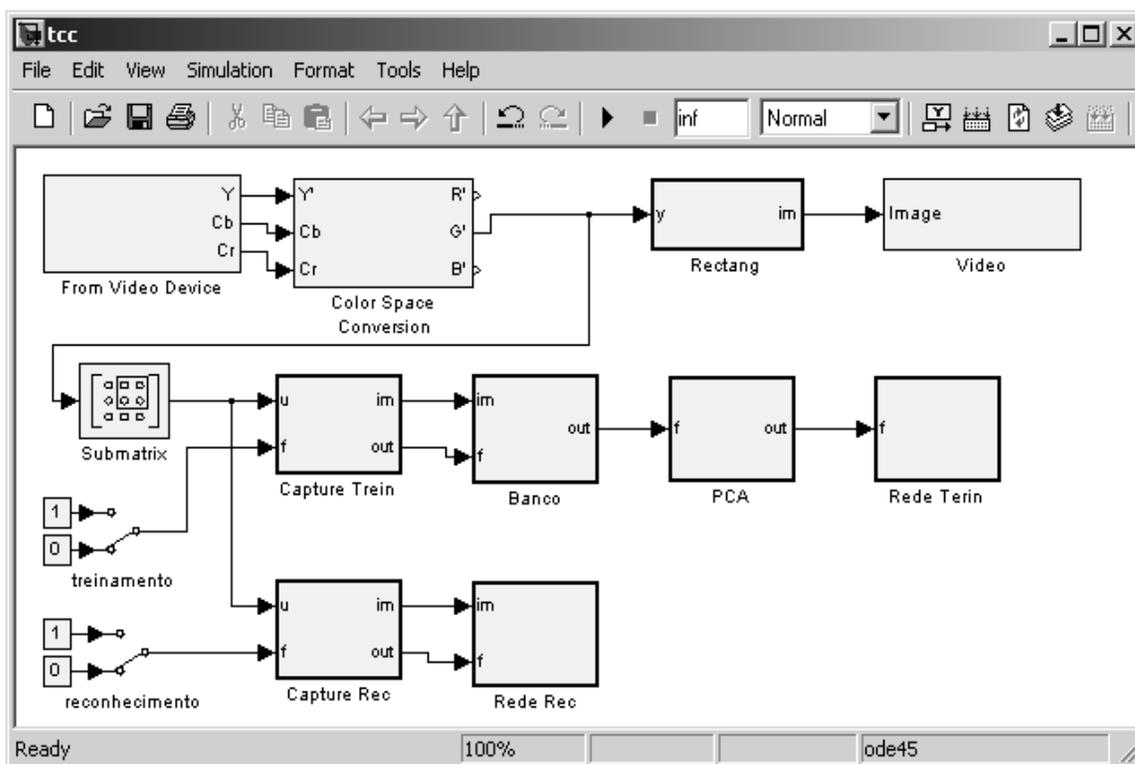


Figura A.1 Aplicativo em Simulink usando Fuzzy ART

**From Video Device** - captura frames do dispositivo de vídeo no formato  $YCbCr$

**Color Space Conversion** - converte o formato do vídeo de  $YCbCr$  para  $RGB$ , onde o aplicativo irá usar apenas o espectro verde, onde obtém maior contraste.

**Rectang** - desenha no frame o retângulo onde está a área que será capturada a face que será usada para fazer parte do banco de treinamento.

**Video** - saída de vídeo.

**Submatrix** - separa a área da face do restante do frame.

**Capture Train** - captura 15 imagens com intervalo de 1 segundo entre elas para compor o banco de faces do indivíduo.

**Banco** - recebe as imagens dos indivíduos e adiciona ao banco de treinamento.

**PCA** - faz o cálculo do PCA, para obter os componentes principais das imagens guardadas reduzindo assim a dimensão.

**Rede Trein** - recebe o banco de faces com seus devidos rótulos e faz o treinamento da rede.

**Capture Rec** - captura um frame para enviar para reconhecimento.

**Rede Rec** - recebe o frame capturado e faz o reconhecimento.

Ao iniciar a aplicação, o vídeo capturado é apresentado, e o programa espera a primeira face a ser capturada.



Figura A.2 Tela de captura

Para se capturar ou reconhecer uma face, é preciso mudar o switch *treinamento* ou *reconhecimento*, colocando a entrada para "1" e após isso, voltando novamente para "0". Para se ter um ambiente mais agradável ao usuário, se tornou necessário criar uma interface gráfica em GUIDE do Matlab, que será apresentada no APENDICE 2.

## APÊNDICE B – CLASSIFICADOR NO GUIDE

Este apêndice apresenta uma descrição sucinta da aplicação feita em GUIDE Matlab para cadastramento, treinamento e reconhecimento de faces usando a rede Fuzzy ART.

### B.1 GUIDE

A figura (B.1) mostra a interface criada.



**Figura B.1** Aplicativo em GUIDE usando Fuzzy ART

Pressionando o botão *iniciar*, a interface irá carregar a aplicação feita em Simulink apresentada no APENDICE 01, onde, a partir desta interface, ela irá controlar os comandos de captura, treinamento e reconhecimento.



Figura B.2 Aplicativo em GUIDE iniciado

Quando o botão *Guardar Faces* for pressionado, 15 faces serão guardadas de cada Indivíduo, onde serão redimensionadas pelo PCA e treinadas pela rede Fuzzy ART.

O botão *Reconhecer* será usado para classificar uma face que se queira fazer o reconhecimento. A classificação feita pela rede Fuzzy ART aparecerá no aplicativo. Três faces foram guardadas e treinadas. Após o treinamento o aplicativo tinha gerado três classes diferentes, uma para cada indivíduo. O aplicativo conseguiu reconhecer as três faces em todas as vezes que elas foram apresentadas.



**Figura B.3** Reconhecimento da primeira face treinada



**Figura B.4** Reconhecimento da segunda face treinada



**Figura B.5** Reconhecimento da terceira face treinada

## *Referências*

- AAMODT, S.; WANG, S. *Bem-Vindo ao seu Cérebro*. 1st. ed. [S.l.]: Cultrix, 2009.
- AMARI, S.-I. Field theory of self-organizing neural nets. *IEEE Transactions on Systems, Man and Cybernetics*, v. 13, n. 9-10, p. 741–748, 1983.
- 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.
- CARPENTER, G.; GROSSBERG, S. ART 2: Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optics*, v. 26, n. 23, p. 4919–4930, 1987.
- CARPENTER, G.; GROSSBERG, S. A massively parallel architecture of a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, v. 37, p. 54–115, 1987.
- CARPENTER, G.; GROSSBERG, S. Adaptive resonance theory. In: ARBIB, M. (Ed.). *The Handbook of Brain Theory and Neural Networks*. 2nd. ed. Cambridge, MA: MIT Press, 2003. p. 87–90.
- CARPENTER, G. et al. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, v. 3, n. 5, p. 698–713, 1992.
- CARPENTER, G.; GROSSBERG, S.; REYNOLDS, J. ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, v. 4, n. 5, p. 565–588, 1991.
- CARPENTER, G.; GROSSBERG, S.; ROSEN, D. ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, v. 4, n. 4, p. 493–504, 1991.
- CARPENTER, G.; GROSSBERG, S.; ROSEN, D. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, v. 4, n. 6, p. 759–771, 1991.
- CARPENTER, G. A.; GROSSBERG, S. The ART of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer Magazine*, v. 21, n. 3, p. 77–88, 1988.

- 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.
- 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.
- 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.
- GROSSBERG, S. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, v. 23, p. 121–134, 1976.
- GROSSBERG, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, v. 11, p. 23–63, 1987.
- HAKEN, H. *Synergetics: Introduction and Advanced Topics*. 1st. ed. [S.l.]: Springer Verlag, 2004.
- HASTIE, T.; STUETZLE, W. Principal curves. *Journal of the American Statistical Association*, v. 84, n. 406, p. 502–516, 1989.
- JIMENEZ, L. O.; LANDGREBE, D. A. Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, v. 28, n. 1, p. 39–54, 1998.
- JOLLIFFE, I. T. *Principal Components Analysis*. New York: Springer-Verlag, 1986.
- KELSO, J. S. *Dynamic Patterns: the self-organization of brain and behavior*. [S.l.]: MIT Press, 1995.
- KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, v. 43, p. 59–69, 1982.
- KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, v. 78, n. 9, p. 1464–1480, 1990.
- MITCHELL, T. *Machine Learning*. [S.l.]: McGraw Hill, 1997.
- MONTEIRO, I. Métodos de aprendizado de máquinas para reconhecimento de faces: Uma comparação de desempenho. In: . [S.l.: s.n.], 2009.
- MUSZKAT, M. Como o cérebro aprende. In: \_\_\_\_\_. [S.l.]: Revista Viver Mente & Cérebro - Scientific American, 2006. v. 8, cap. Dinâmica do Conhecimento, p. 41–47.
- OJA, E. Neural networks, principal components and subspaces. *International Journal of Neural Systems*, v. 1, n. 1, p. 61–68, 1989.
- RUMELHART, D. E.; ZIPSER, D. Feature discovery by competitive learning. *Cognitive Science*, v. 9, n. 1, p. 75–112, 1985.

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.

ZADEH, L. Fuzzy sets. *Information and Control*, v. 2, p. 338–353, 1965.

ZHAO, W. et al. Face recognition: A literature survey. *ACM Computing Surveys*, v. 35, n. 4, p. 399–458, 2003.