MARCELO NEPOMOCENO KAPP

Reconhecimento de Palavras Manuscritas Utilizando Redes Neurais Artificiais

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná, como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Curitiba, 20 de março de 2004

MARCELO NEPOMOCENO KAPP

Reconhecimento de Palavras Manuscritas Utilizando Redes Neurais Artificiais

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná, como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: Ciência da Imagem

Orientador: Prof. Dr.Robert Sabourin

Co-Orientador: Prof. Dra.Cinthia O. de A. Freitas

Curitiba, 20 de março de 2004

Kapp, Marcelo Nepomoceno

Reconhecimento de Palavras Manuscritas Utilizando Redes Neurais Artificiais. Curitiba:, 2003. 98 f.: il.

Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada, Curitiba, BR–PR, 2003. Orientador: Robert Sabourin; Co-Orientador: Cinthia O. de A. Freitas.

1.Reconhecimento. 2.Cheques Bancários. 3.Extração de Características. 4.Rejeição. 5.Redes Neurais Artificiais. 6.*Wrapper-Hill Climbing*. I.Pontifícia Universidade Católica do Paraná. II.Centro de Ciências Exatas e de Tecnologia. III.Programa de Pós-Graduação em Informática Aplicada.

Aos meus pais

Marlene e Carlos

e a toda minha família

com amor...

Agradecimentos

"Não mostre para Deus o tamanho do seu problema, mostre para o problema o tamanho do seu Deus" (autor desconhecido). Deus, obrigado pelo conforto e proteção.

Aos meus pais, Marlene e Carlos, que sempre me apóiam e incentivam a continuar lutando pelos meus ideais.

Gostaria de agradecer aos Professores Dr. Robert Sabourin e Dra. Cinthia Freitas pelas orientações precisas na fundamentação teórica do trabalho, paciência, e amizade desenvolvida. Ao Professor Dr. Júlio César Nievola pela ajuda, questionamentos e contribuições construtivas.

Aos Professores já citados e também ao Professor Dr. João Marques, pela participação na formação da banca examinadora.

Aos Professores Dr. Alceu de S. Britto Jr. pela oportunidade, Dr. Paulo Sérgio L. de Souza e MSc. Tatiana M. Celinski pelo incentivo.

À minha namorada Cinthia Rossa, pelo carinho, força e compreensão passados durante esses anos...

Aos colegas de mestrado e disciplinas, Aderly, Carlos Magno, Carlos Solana, César, Crysthiane, Fernanda, Luiz Felipe, Marcellus, Rafael, Rodrigo e Walter pelas constantes trocas de idéias e companheirismo.

Enfim, a todos que de algum modo contribuíram para a realização deste trabalho.

"If I have seen farther than others,
it is because I stood on the shoulders of giants."
— Sir Isaac Newton

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

RNA	Redes Neurais Artificiais
MLP	*Multi-Layer Perceptron*
K-NN	K-*Nearest Neighbor*
UNICAMP	Universidade de Campinas
UFCG	Universidade Federal de Campina Grande
ETS	*École de Technologie Supérieure*
PUC-PR	Pontifícia Universidade Católica do Paraná
LUCI	Laboratório Unificado de Ciências da Imagem
COMPE	Serviço de Compensação de Cheques e Outros Papéis
DOC	Documento de Crédito
MEM	Modelos Escondidos de Markov
HT	Histograma de Transição branco/preto
NLE	Número de laços contidos no lado esquerdo
NLD	Número de laços contidos no lado direito
NSCVE	Número de semicírculos côncavos no lado esquerdo
NSCVD	Número de semicírculos côncavos no lado direito
NSCXE	Número de semicírculos convexos no lado esquerdo
NSCXD	Número de semicírculos convexos no lado direito
NCPE	Número de *crossingpoints* no lado esquerdo
NCPD	Número de *crossingpoints* no lado direito
NBPE	Número de *branchpoints* no lado esquerdo
NBPD	Número de *branchpoints* no lado direito
NEPE	Número de *endpoints* no lado esquerdo
NEPD	Número de *endpoints* no lado direito
NCH	Número de cruzamentos horizontais
NAE	Número de ascendentes no lado esquerdo
NAD	Número de ascendentes no lado direito
NDE	Número de descendentes no lado esquerdo
NDD	Número de descendentes no lado direito
NPP	Proporção de pixels
NTV	Número de traços verticais

NTH	Número de traços horizontais
NLAE	Número de laços ascendentes no lado esquerdo
NLAD	Número de laços ascendentes no lado direito
NLDE	Número de laços descendentes no lado esquerdo
NLDD	Número de laços descendentes no lado direito
MIT	*Massachusetts Institute of Technology*
PE	Elemento de processamento
CRT	*Class-Related Thresholds*
MSE	*Mean Square Error*

Lista de Símbolos

O_i	Saída de RNA
w_i	classe do problema
M_{wi}	Módulo de uma classe w_i
Ω_0	Conjunto de dados de uma classe w_i
Ω_1	Conjunto de dados das classes restantes
$D(.,.)$	Vetor de saída
Z_{Ω_i}	Conjuntos de treinamento de dados
$P(.)$	Probabilidade
$\hat{P}(.)$	Probabilidade estimada
T_i	Limiar de rejeição
$A(.)$	Precisão
$R(.)$	Rejeição
R_{max}	Rejeição máxima
$E(.)$	Erro
E_{min}	Erro mínimo
$O(.)$	Ordem de complexidade
s_i	Valor de multiplicação entre informação *a priori* e a calculada
c_s	Constante de quanto utilizar da informação *a priori*
y_i	Saída de RNA
p_i	Probabilidade *a priori*
N_c	Número total de classes

Resumo

O estudo das palavras manuscritas está ligado ao desenvolvimento de métodos de reconhecimento voltados a aplicações do mundo real envolvendo palavras manuscritas, tais como: cheques bancários, envelopes postais, textos manuscritos, entre outros. Neste trabalho, utilizam-se palavras manuscritas do contexto de cheques bancários brasileiros, especificamente o conjunto de valores por extenso e o de meses do ano, sendo que não há restrições de tipos ou estilos de escrita e número de escritores. Um conjunto de características globais e duas arquiteturas de redes neurais artificiais (RNA) são testadas para a classificação das palavras. Os principais objetivos são de avaliar os desempenhos das arquiteturas de RNA MLP (*Multilayer Perceptron*) convencional e classe-modular, desenvolver um mecanismo de rejeição de múltiplos limiares e analisar o comportamento do conjunto de características proposto em ambas arquiteturas. O método desenvolvido extrai primitivas globais das palavras, tais como, número de laços, ascendentes e descendentes entre outras, gerando um vetor de 24 dimensões. Na etapa de reconhecimento, uma arquitetura RNA MLP convencional e uma classe-modular são treinadas e testadas separadamente. Um mecanismo de rejeição de múltiplos limiares é implementado para que padrões "desconhecidos"ou ambíguos sejam rejeitados e não classificados. Para a análise do conjunto de primitivas, utiliza-se o método *wrapper/hill climbing*, que permite uma seleção de características, apontando as mais relevantes para cada classe em determinada arquitetura. Resultados de experimentos com a base de imagens de palavras dos meses demonstram uma superioridade na utilização da arquitetura classe-modular em relação a RNA MLP convencional. O mecanismo de rejeição de múltiplos limiares também demonstrou desempenho favorável em ambas arquiteturas. As análises das características mostram a importância das primitivas estruturais como concavidades e convexidades e das primitivas perceptivas ascendentes e descendentes. Para a base de imagens de palavras manuscritas referentes aos meses do ano, obtem-se uma taxa de reconhecimento de 81,75% e admitindo uma taxa de rejeição de 25,33% atinge-se 91,52% de taxa de confiabilidade. Para a base de imagens de palavras referentes aos valores por extenso, obtem-se 52,35% de taxa de reconhecimento. Neste trabalho descreve-se uma metodologia para reconhecimento de palavras manuscritas e também análise do conjunto de características proposto, e assim busca-se contribuir com os estudos de reconhecimento de palavras manuscritas existentes.

Palavras-chave: 1. Cheques bancários. 2. Reconhecimento de palavras manuscritas. 3. Redes neurais artificiais. 4. Seleção de características.

Abstract

The study of handwritten words is tied up to the development of recognition methods for real world applications involving handwritten words, such as: bank checks, postal envelopes, handwritten texts, among others. In this work, handwritten words of the context of brazilian bank checks is used, specifically the sets of values for amount and the one of months of the year, and there are not restrictions of types or writing styles and number of writers. A global features set and two architectures of artificial neural networks (ANN) are tested for the classification of the words. The main objectives are of evaluating the performance of conventional and class-modular RNA MLP (Multilayer Perceptron) architectures, to develop a rejection mechanism of multiples thresholds and to analyze the behavior of the features set proposed in both architectures. The developed method extracts primitive global of the words, such as, number of loops, ascenders and descenders among other, generating a vector of 24 dimensions. In the recognition stage, a conventional and a class-modular RNA MLP is trained and tested separately. A rejection mechanism of multiple thresholds is implemented so that patterns "unknown"or ambiguous are rejected and not classified. For the analysis of the primitive set, the method wrapper/hill climbing is used that allows a feature selection, aiming the most important for each class in certain architecture. Results of experiments with the database of words of the months demonstrate a superiority in the use of the architecture to class-modular in relation to RNA MLP conventional. The rejection mechanism of multiple thresholds also demonstrated favorable performance in both architectures. The analyses of the features show the importance of the structural primitives as concavities and convexities and of the perceptual primitives ascenders and descenders. For the database of handwritten words referring to the months of the year, is obtained a recognition rate of 81,75% and admitting a rejection rate of 25,33% is reached 91,52% of reliability rate. For the database of handwritten words referring to the values for amount, is obtained 52,35% of recognition rate. In this work a methodology is described for recognition of handwritten words and also analysis of the proposed features set, and it is looked for like this to contribute with the studies of handwritten word recognition existent.

Keywords: 1. Bank checks. 2. Handwritten word recognition. 3. Artificial neural networks. 4. Features selection.

Capítulo 1

Introdução

De acordo com Plamondon e Srihari [PS00], a escrita manuscrita consiste de marcas gráficas em uma superfície, cujo propósito na maioria das vezes é a comunicação obtida em virtude da relação dos símbolos convencionais das linguagens. A escrita manuscrita é valorizada por ter contribuído muito para o desenvolvimento das culturas e civilizações.

Cada manuscrito é um conjunto de ícones, os quais são caracteres ou letras que possuem suas formas básicas definidas. Há regras para combinação de letras para formar unidades representativas lingüísticas de alto nível. Por exemplo, há regras para combinação de formas e letras individuais para formar palavras cursivas no alfabeto latino.

A razão da escrita manuscrita ter persistido ao longo dos anos na era do computador é a conveniência do papel e da caneta, comparada aos teclados, para as numerosas situações do dia a dia [PS00]. O estudo das palavras manuscritas está ligado ao desenvolvimento de métodos de reconhecimento voltados para aplicações do mundo real envolvendo palavras manuscritas, tais como: processamento automático de cheques bancários, envelopes postais, formulários, textos manuscritos, entre outros. As Figuras 1.1 e 1.2 caracterizam-se por contextos diferentes, destacando-se:

- Cheques bancários: léxico conhecido *a priori* e de pequena dimensão, ou seja, inferior a 100 palavras. [Côt97];
- Envelopes postais: léxico desconhecido e de grande dimensão, isto é, superior a 500 palavras. [Côt97].

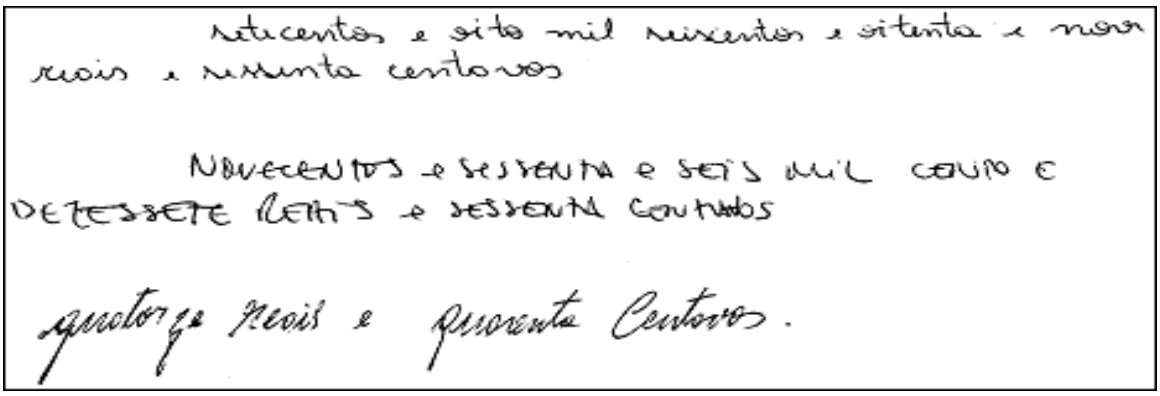

Figura 1.1: Amostras de palavras manuscritas e estilos de escrita no contexto de cheques bancários

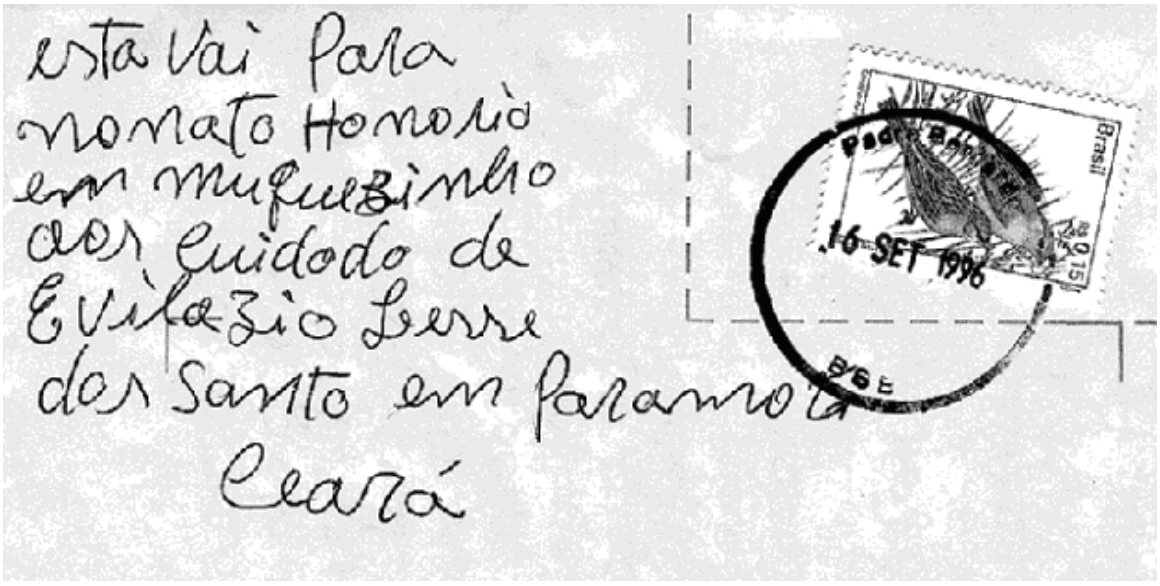

Figura 1.2: Amostra de endereço postal

A tarefa de leitura de manuscritos é um envolvimento especializado de habilidades humanas. Vários tipos de análise visando o reconhecimento, a interpretação e a identificação podem estar associadas com a escrita manuscrita. O reconhecimento de manuscritos é a transformação de uma linguagem representada inicialmente em forma espacial de marcas gráficas até sua representação simbólica. A interpretação de manuscritos é a tarefa de determinar o significado de uma palavra escrita, por exemplo, um endereço postal manuscrito. A identificação de manuscritos é o processo de determinar o autor de uma amostra de manuscrito dentre um conjunto de escritores, assumindo que cada manuscrito tem forma individualizada [PS00].

Entretanto, para a realização automática do reconhecimento, interpretação ou identificação, os dados manuscritos necessitam de uma conversão para a forma digital através do uso de "*scanners*" da escrita no papel, ou por um tipo especial de caneta ou superfície eletrônica tais como um digitalizador combinado com uma tela de cristal líquido. As duas abordagens são distinguidas como digitalização *off-line* e *on-line*, respectivamente. No caso *on-line*, as coordenadas bidimensionais de pontos sucessivos são descritos como uma função do tempo e armazenadas em ordem, isto é, a ordem dos segmentos de palavras traçados pelo escritor é prontamente disponibilizada. Já no *off-line* somente o manuscrito completo é disponível em uma imagem. A abordagem *on-line* fornece uma representação espaço-temporal da entrada, enquanto que o caso *off-line* envolve análise de espaço-luminosidade da imagem [PS00].

O reconhecimento de palavras manuscritas (*Handwritten Word Recognition*) *off-line* é a leitura de uma imagem de uma palavra manuscrita a partir de um léxico associado. As tarefas principais em reconhecimento de manuscritos *off-line* são o reconhecimento de caracteres, palavras e cadeias numéricas.

O tema central desta dissertação consiste no reconhecimento automático *off-line* de dois conjuntos de palavras manuscritas encontrados no contexto de cheques bancários brasileiros. O primeiro conjunto corresponde aos meses do ano, formando um léxico de doze palavras. O segundo é formado pelas palavras que compõem os valores por extenso, constituindo um léxico de trinta e nove palavras.

No presente trabalho, a metodologia de reconhecimento é composta das seguintes tarefas:

- Pré-processamento das imagens: para atenuar a variabilidade das palavras em relação à inclinação horizontal e vertical;
- Extração de primitivas: visa a obtenção de um conjunto de características das palavras manuscritas e a representação das mesmas;
- Classificação: efetua o reconhecimento dos padrões nas palavras;
- Mecanismo de rejeição: possibilita a rejeição de imagens que produzem um determinado grau de incerteza para o classificador.

Apesar de vários trabalhos tratarem do reconhecimento de palavras manuscritas em cheques bancários [Heu94], [Gui95], [AGHG95], [CDIP95], [FGK95], [Mon95], [Avi96], [KAB+96], [Côt97], [Oll99], [Fre01], [MYS+01], [JdCCFS02], [MSBS02], [MOS+02] e [KFNS03], tal tarefa torna-se ainda mais complexa, considerando a grande complexidade devido a variedade de estilos de escrita, números de escritores e erros de ortografia que compõem o contexto dos cheques bancários.

As aplicações de leitores automáticos de endereços postais, cheques bancários, e vários outros documentos têm sido alvo de estudos na área de reconhecimento de palavras manuscritas nos últimos anos, especialmente no Brasil, com pesquisadores da Universidade de Campinas - UNICAMP, Universidade Federal de Campina Grande - UFCG e Pontifícia Universidade Católica do Paraná - PUC-PR. As duas últimas juntamente com a *École de Technologie Supérieure* - ETS - Canadá, dedicam-se em conjunto a esta área de pesquisa.

Assim, o presente trabalho visa contribuir ao estudo de métodos de reconhecimento de palavras manuscritas, em especial à extração e representação de primitivas, classificação e rejeição em aplicações envolvendo conjuntos de palavras pequenos, ou seja, contendo de 1 a 100 palavras [Côt97].

1.1 Descrição do Problema

O problema abordado neste trabalho é o reconhecimento de dois conjuntos de palavras manuscritas no contexto de cheques bancários brasileiros. O primeiro é composto pelas palavras que representam os nomes dos meses do ano. Este é um problema importante, pois constitui um subproblema do reconhecimento de datas

em cheques bancários. Embora este conjunto seja limitado a 12 classes, há palavras muito semelhantes ou com mesma terminação, o que afeta o desempenho global da metodologia de reconhecimento, por exemplo: Jan**eiro**, Fever**eiro**, Ma**rço**, Abril, Ma**io**, Ju**nho**, Ju**lho**, Agosto, Set**embro**, Outu**bro**, Nov**embro** e Dez**embro**.

O segundo léxico está relacionado ao valor por extenso do cheque, que corresponde a um valor numérico, ao qual se aplica uma gramática conhecida no momento da grafia deste valor por extenso. Portanto a partir do valor numérico por extenso é possível definir duas características do problema: as palavras-chave e os blocos internos às palavras-chave formados por palavras que representam os valores numéricos, conforme a Figura 1.3.

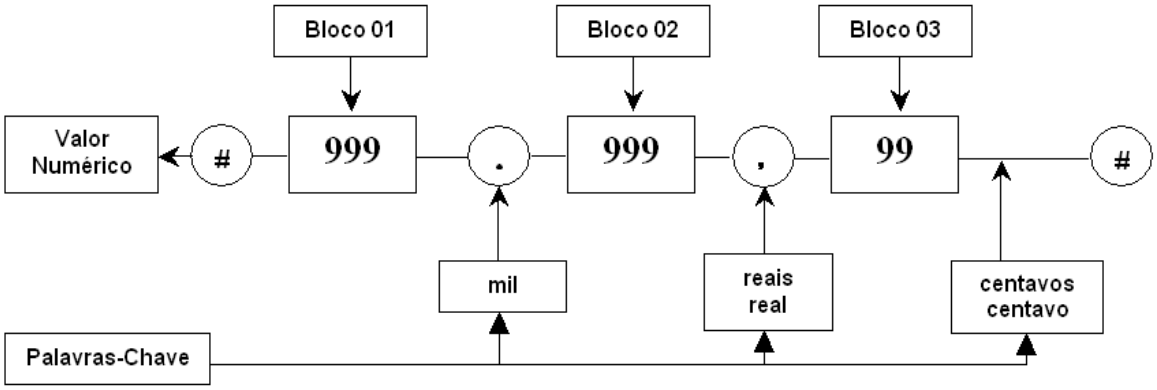

Figura 1.3: Valor numérico e palavras-chave [Fre01]

As palavras-chave são as três palavras que identificam três grandes blocos no extenso de um cheque. Considerando que as bases de dados disponíveis no Laboratório de Análise de Imagens de Documentos - LUCI (Laboratório Unificado de Ciências da Imagem) da PUC-PR foram formadas para atender valores entre R$ 0,01 (um centavo) e R$ 999.999,99 (novecentos e noventa e nove mil, novecentos e noventa e nove reais e noventa e nove centavos), as palavras-chave são justamente as palavras correspondentes no léxico a "ponto" da indicação do valor da **mil**har, a "vírgula" da indicação do valor da parte decimal em **centavos**/**centavo** [Fre01].

Entende-se por bloco o conjunto de palavras grafadas por extenso usadas para representar a quantia expressa em uma das partes do valor numérico do cheque. Deste modo, a análise dos blocos permite identificar a formação interna de cada um através de palavras do léxico. Os blocos 1 e 2, conforme a Figura 1.3, são

idênticos entre si. Esses blocos têm como terminais na gramática o mesmo conjunto de palavras. Por outro lado, o bloco 3, que representa o valor numérico referente a parte dos "centavos/centavo" tem a característica de não apresentar o conjunto de palavras terminadas por "entos". O léxico de 39 palavras pode, então, ser classificado de acordo com a ordem correspondente ao algarismo no valor numérico resultando em 5 classes de palavras, a saber [Fre01]:

- classe dos "entos": novecentos, oitocentos, setecentos, seiscentos, quinhentos, quatrocentos, trezentos, duzentos, cem/cento;
- classe dos "enta": noventa, oitenta, setenta, sessenta, cinquenta/cincoenta, quarenta, trinta e vinte;
- classe das dezenas: dezenove, dezoito, dezessete, dezesseis quinze, quatorze, treze, doze, onze e dez;
- classe das unidades: nove, oito, sete, seis, cinco, quatro, três, dois, um/hum;
- classe das palavras-chave: mil, reais/real, centavos/centavo.

Dois fatores aumentam a complexidade do problema de reconhecimento neste conjunto de palavras. Primeiramente, a grande similaridade de formas entre as palavras do léxico, como mostrado na Figura 1.4. Segundo, o fato da gramática portuguesa aceitar diferentes grafias para uma mesma palavra, por exemplo: "um" e "hum", "quatorze" e "catorze", "cinqüenta" e "cincoenta".

Para o reconhecimento de palavras manuscritas existem uma abundância de técnicas que são capazes de descrever a similaridade das formas que pertencem a uma mesma classe e que permitem ao mesmo tempo distinguir as representações de formas separadamente entre classes diferentes. Portanto, o objetivo do reconhecimento automático é o de desenvolver um método que se aproxime do ser humano em sua capacidade de ler qualquer palavra manuscrita.

Porém há fatores de complexidade que representam um desafio neste tipo de aplicação, os principais são:

1. **Tipos de escrita:** Para Tappert *et al.* [TSW90] existem 5 categorias de escrita, situadas entre as categorias caixa alta e cursiva pura, conforme mostrado na Figura 1.5

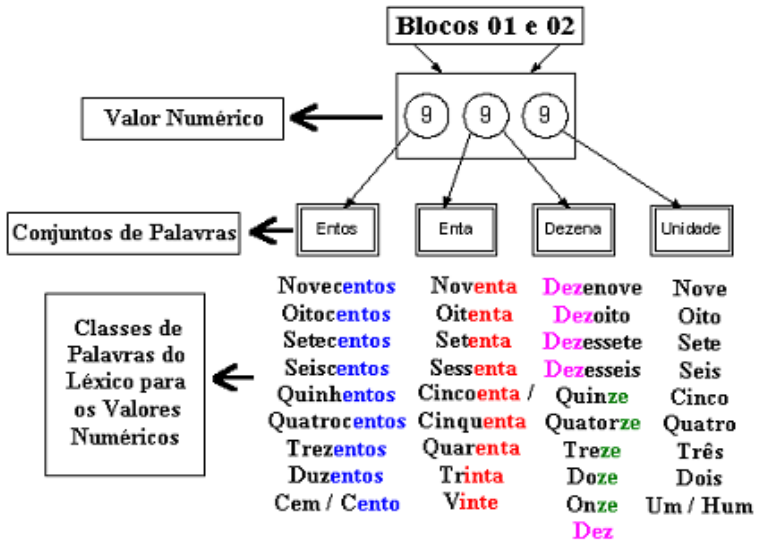

Figura 1.4: Classes constituintes dos valores por extenso de cheques [Fre01]

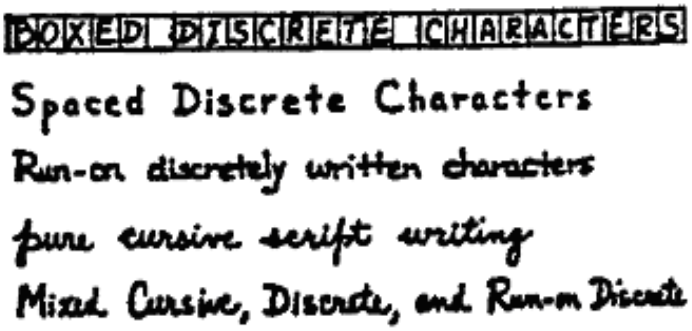

Figura 1.5: Tipos de escrita [TSW90]

2. **Número de escritores:** A variabilidade de formas a serem consideradas pelas metodologias de reconhecimento depende do número de escritores. Três categorias estão relacionadas [Côt97]:

   - Mono-escritor: um único escritor;
   - Multi-escritor: número limitado de escritores;
   - Omni-escritor: número grande de escritores.

3. **Tamanho do Léxico:** A maioria dos métodos de reconhecimento da escrita manuscrita tem como base um léxico de palavras associado. Para Côté [Côt97] os tamanhos a considerar são:

- Pequeno: contém de 1 a 100 palavras. A maioria das aplicações de léxicos pequenos são também omni-escritores;
- Médio ou intermediário: acima de 100 até 500 palavras;
- Grande: contém mais de 500 palavras.

4. **Qualidade da imagem:** É importante para um processo de reconhecimento que as imagens utilizadas sejam de qualidade. A falta de qualidade da imagem é determinada pela presença de ruído como os obtidos através de um processo de segmentação ou ainda pela má utilização do *scanner*, como mostrado na Figura 1.6.

a) b)

Figura 1.6: Fatores que prejudicam a qualidade da imagem [Kim96]: Imagem a) Ruído "salpicado" e b) Incorreta operação do *scanner*

5. **Fatores humanos:** As características de um manuscrito são determinadas na maioria das vezes pelo escritor [Fre01]. Pode-se citar ainda o estilo e adicionar um número de fatores relevantes tais como:

   - Inclinação vertical: É muito comum escrever inclinado, especialmente na escrita cursiva. A inclinação horizontal, ou seja, da linha da base, é denominada *skew* e a inclinação vertical é denominada *slant* [Fre01]. Ambas as inclinações estão exemplificadas na Figura 1.7 como α e β respectivamente.

Figura 1.7: Exemplo de inclinações *slant* e *skew*

6. **Traços contínuos ou descontínuos:** Exemplos da ocorrência destes fatores são mostrados na Figura 1.8.

a) b)

Figura 1.8: Exemplos de traços contínuos e descontínuos: a) Traço contínuo e b) Traço descontínuo

7. **Erros e variações ortográficas:** Alguns casos de erro são comumente encontrados, como "seissentos" ao invés do correto "seiscentos", ou ainda, palavras que permitem diferentes grafias, como: "cinquenta" e "cincoenta", "quatorze" e "catorze" e "um" e "hum".

1.2 Objetivos

Este trabalho visa apresentar uma metodologia desenvolvida no decorrer das atividades de pesquisa para a implementação de um sistema básico (*baseline system*) de reconhecimento de palavras manuscritas no contexto de cheques bancários. Duas arquiteturas de redes neurais artificiais MLP (*Multiple Layer Perceptron*) são testadas:

- Convencional: o conjunto de características proposto é utilizado para o treinamento e reconhecimento em apenas uma rede neural MLP;
- Classe-Modular: um comitê de redes neurais MLP é testado. Cada classe do problema possui uma rede específica, assim como descrito em [OS02] e [KFNS03].

Adicionalmente, um mecanismo de rejeição com múltiplos limiares e uma análise do conjunto de características proposto são realizados.

As técnicas empregadas em cada etapa do sistema, que vão desde o pré-processamento até o reconhecimento, passando também pela análise do conjunto de

características, são descritas. Para isto, são cumpridos os seguintes objetivos específicos:

- Visto que o contexto no qual o presente trabalho está inserido não restringe-se a um tipo de manuscrito, o conjunto de características proposto deve ser capaz de representar as palavras independentemente do estilo de escrita;
- Essas características devem possuir uma representação compatível com o classificador adotado, neste caso, as redes neurais artificiais;
- Uma divisão da base de imagens e um processo de treinamento, validação e teste do classificador devem ser realizados para ambas as arquiteturas;
- Após os treinamentos, os classificadores ajustados permitem a inclusão de um mecanismo de rejeição;
- O método *Wrapper-Hill Climbing* é usado para realizar uma análise do conjunto de características proposto.

Por fim, os resultados obtidos são apresentados e avaliados procurando deste modo tirar conclusões que possam contribuir com novos estudos na área em questão.

1.3 Justificativas

Hoje, o Serviço de Compensação de Cheques e Outros Papéis - Compe, ligado ao Banco Central do Brasil, que atualmente tem papel preponderante no sistema de pagamentos, informa que os principais documentos compensáveis são o cheque, o Documento de Crédito - DOC e o bloqueto de cobrança, sendo que, segundo a Compe, a média mensal do total de lançamentos compensados a débito da conta Reservas Bancárias oriundos da Compe é da ordem de R$ 54,4 bilhões, cerca de R$ 2,6 bilhões por dia [Com03].

Os valores a débito da rede bancária, antes de compensados com os créditos, apresentam a média mensal de R$ 305,5 bilhões, assim decompostos:

a) Cheques: R$ 144,6 bilhões;

b) DOC: R$ 112,7 bilhões;

c) Cobrança: R$ 41,4 bilhões;

d) Cartão de crédito: R$ 2,3 bilhões (valor estimado);

e) Outros: R$ 4,5 bilhões (valor estimado).

Portanto, os cheques e os DOC respondem por mais de 86% de todo o resultado da Compe [Com03]. Sendo assim, dado o volume de compensações e dado que na maioria dos casos essa tarefa é manual, tal processo é monótono e entediante, não sendo adequado para ser realizada por um ser humano. Além disto, o fator financeiro também deve ser considerado, pois com a automação obtem-se um aumento de produção.

Dessa forma, faz-se necessário criar um sistema de automação bancária que possa trabalhar com os cheques. Tal processo consiste em um sistema de visão que seja capaz de segmentar uma imagem de cheque bancário, extraindo as informações principais como o valor numérico, o valor por extenso, o local, a data completa, o nome do cliente e sua assinatura, interpretando-as e no caso das assinaturas verificando-as, possibilitando, dessa forma, a compensação direta do cheque.

Quanto à metodologia empregada no presente trabalho, o conjunto das características é composto de primitivas que foram selecionadas buscando formar um conjunto simples, de pequena dimensão, e que seja compatível com os diversos tipos e variabilidades de escrita.

A escolha do uso de redes neurais artificiais MLP como classificadores é devido ao fato de que além de poderem ser utilizados similarmente a métodos estatísticos clássicos, possuem características importantes como: capacidade de aprenderem relacionamentos complexos não lineares entre dados de entrada e saída, utilizarem procedimentos de treinamento seqüenciais e ainda se adaptarem aos dados [JDM00]. Além disto, o presente trabalho baseia-se no estudo de Oh e Suen [OS02], em que RNAs-MLP são utilizadas como classificadores organizados de forma classe-modular.

1.4 Contribuições

Nesta subseção apresentam-se as contribuições originais que este trabalho de pesquisa produziu:

- Avaliação de um conjunto de primitivas de pequena dimensão voltado a um mecanismo de zoneamento da palavra em apenas duas áreas, reduzindo a quantidade de informação extraída e mantendo performances aproximadas ou equivalentes aos estudos de Oliveira *et al.* [JdCCFS02] e Morita *et al.* [MSBS02].

- Avaliação das arquiteturas de redes neurais convencional e classe-modular *Multiple Layer Perceptron* conforme descrito em [KFNS03], para o reconhecimento de palavras manuscritas, possibilitando também sugerir como podem ser realizados os processos de treinamento e reconhecimento nas arquiteturas avaliadas;

- Mecanismo de rejeição de múltiplos limiares baseado em Fumera *et al.* [FRG00]. Tal mecanismo obtém limiares para cada classe individualmente, através de uma busca local, possibilitando estender a sua utilização para uma arquitetura classe-modular MLP;

- Análise do conjunto de primitivas utilizado através do método de seleção de características *Wrapper-Hill Climbing*;

- Metodologia como um todo para o reconhecimento de palavras manuscritas e seleção de características;

- Aplicação da metodologia em duas bases de dados diferentes.

1.5 Organização da Dissertação

Esta dissertação está organizada em cinco capítulos. No Capítulo 2 apresenta-se uma revisão sobre trabalhos anteriores. O Capítulo 3 descreve o método proposto para o reconhecimento de palavras manuscritas no contexto de cheques bancários.

Os experimentos realizados neste trabalho para validar o método proposto são apresentados no Capítulo 4, juntamente com uma análise do conjunto de características utilizado em relação aos resultados nas bases de imagens. No Capítulo 5 são apresentadas as conclusões e as propostas de trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Neste capítulo são apresentados trabalhos que se relacionam com o tema central da Dissertação, o reconhecimento de palavras manuscritas no contexto de cheques bancários. Embora o relacionamento de alguns trabalhos com o tema aqui proposto, não seja direto, eles são focados em diferentes aspectos do reconhecimento de palavras e caracteres manuscritos.

Heutte[Heu94] aplica uma abordagem para o reconhecimento de palavras e cadeias numéricas manuscritas em envelopes postais americanos e cadeias numéricas em cheques bancários franceses, baseada em oito famílias de características selecionadas *a priori* considerando três fatores: simplicidade de detecção, poder discriminante e a baixa correlação entre as características. De acordo com o autor, estas características permitem combinar a pesquisa de elementos estruturantes como: junções X e Y, extremos, arcos côncavos, laços, natureza de perfis e também uma análise, por medidas de ordem estatística, da distribuição de pontos que pertençam ou não ao caractere, como: momentos invariantes, projeções horizontais e verticais. Busca-se fazer a combinação entre análises global e local dos caracteres. Tais características servem de entrada para um processo de classificação estatística do tipo "separação linear" ou distância de Mahalanobis, que requer uma representação vetorial de dados. O reconhecimento de palavras cursivas realizou-se pelo que o autor denominou de fusão de "grafemas de solução", obtidos através de dois métodos de reconhecimento diferentes, separação linear e decisão neural, testados com aproximadamente 4000 palavras cursivas. Os resultados obtidos estão entre 80% e 90% no caso de léxicos de tamanho reduzido (10 palavras) e variaram em 50% para o contexto de

envelopes postais em geral.

Guillevic [Gui95], trabalha com palavras do contexto de cheques bancários ingleses e franceses. Neste estudo, Guillevic aplica uma abordagem global, analisando a palavra como um todo, e uma local, tratando os caracteres isoladamente, para o reconhecimento de palavras. Para a abordagem global, as características utilizadas são: posição relativa de ascendentes, descendentes e laços, a quantidade de ocorrência dos mesmos, a localização de traços verticais e horizontais, diagonais sul-leste e sul-oeste, e estimativa do comprimento da palavra. Na abordagem local, uma métrica de distância entre *pixels* é utilizada como característica. O processo resume-se em: extrair o vetor de características de uma determinada imagem; e para cada classe no léxico computar um novo vetor de características baseado no primeiro e calcular a probabilidade para a classe em questão; e por último obtêm-se as probabilidades ordenadas. Para classificação, Guillevic usa o que denominou de K vizinho mais próximos modificado, que combina distâncias e pesos que são otimizados através de algoritmos genéticos. O reconhecedor na abordagem global é treinado com um conjunto de 1.496 sentenças de cheques, totalizando 5.322 palavras. Para os testes experimentais é utilizado um conjunto diferente de 712 sentenças, totalizando 2515 palavras. O reconhecedor da abordagem local, é treinado com 5.615 caracteres e testado com 2.414 caracteres. As sentenças/palavras são correspondentes ao valor por extenso dos cheques. A metodologia combina os resultados de ambos os reconhecedores. Os resultados para o reconhecimento de sentenças inteiras no idioma Francês são de aproximadamente 65,7% e Inglês 44,4%.

Montoliu [Mon95] normaliza o tamanho das palavras em janelas de tamanho fixo de 16x32 *pixels* chamada de retina. Uma Arquitetura Multi-Agente é utilizada sobre esta janela. Um macro-agente denominado de *Mots*(Palavra) é subdividido em outros quatro agentes: um agente neuronal, dois agentes de K vizinhos mais próximos e um agente Markoviano. O agente neuronal, que de acordo com o autor é o mais efetivo, é apresentado primeiramente seguido pelos restantes. Cada agente possui suas próprias características e um mecanismo de subjanelas que tem seu tamanho variado em cada agente. Para o agente Markoviano as principais características utilizadas são: direções, ascendentes, descendentes e laços. A fusão dos resultados dos quatro agentes compõe o resultado final. Os resultados experimentais para palavras francesas no contexto de cheques bancários variaram em 79% e 87%.

Em [Avi96], o autor relata os diferentes estágios que permitem obter uma descrição global e analítica de palavras. A descrição global utiliza três primitivas: traços ascendentes, descendentes e espaços. Na descrição analítica, para toda palavra ou pedaço de palavra de uma oração, de uma descrição cronológica, propõe-se a extração de primitivas com base em um alfabeto de 12 traços, sendo que, as formas elementares são obtidas a partir dos pontos que passam sobre a linha média do esqueleto da palavra original. Modelos Escondidos de Markov são utilizados como classificador. A base de dados utilizada para a obtenção dos resultados experimentais é formada por sentenças/palavras de cheques bancários franceses. Os resultados variam de 40% a 60% na base de aprendizagem e em 56,5% na base de teste. O autor ressalta que em suas observações em relação ao desempenho dos métodos separadamente, a abordagem global foi superior a local.

O sistema A2iA para o reconhecimento de cheques manuscritos franceses é apresentado por Knerr em [KAB+96]. Este sistema combina uma abordagem global e local para o reconhecimento das palavras manuscritas. A extração de características é composta de um complexo conjunto de primitivas divididas em 3 tipos:

- Tipo 1: primitivas gerais que descrevem a imagem do caractere (18 primitivas) Exemplos: altura do caractere, tipo do perfil direito/esquerdo/superior/inferior do caracter, número de componentes conectados, entre outros;

- Tipo 2: primitivas que descrevem a posição relativa de partes do caractere em relação a imagem do caractere (9 primitivas). Exemplos: ascendentes, descendentes, traços horizontais, verticais com +45 graus ou -45 graus, laços, entre outros.

- Tipo 3: primitivas que dependem da posição do caractere em relação a linha de texto no cheque. São elas: posição vertical e horizontal do caractere em relação a linha de texto e número de conexões entre caracteres vizinhos de uma mesma linha de texto.

Os classificadores utilizados para a abordagem local das palavras, ou seja, a nível de caracteres foram:

- Classificador Bayes: utiliza primitivas do tipo 1 e 3;

- Classificador *Template-based*: usa as primitivas do tipo 2;
- Redes Neurais *Multi-Layer Perceptrons* (MLP): as matrizes formadas pelos caracteres isolados formam as características utilizadas por este classificador.

Para classificar as palavras na abordagem global, o autor utiliza os Modelos Escondidos de Markov com as primitivas tipo 2 entre outras. Os resultados deste estudo para as palavras no contexto dos valores por extenso variam em 64,5% para as diferentes combinações das abordagens. O desempenho do sistema para o reconhecimento do cheque por completo, ou seja, valor por extenso e a cadeia numérica, é entre 50% e 60%.

Côté [Côt97] trabalha com a abordagem global de características: ascendentes, descendentes, laços, concavidades e convexidades. A autora emprega e define letras-chave, ou sejam, os ascendentes (letras:"t","l","b"), descendentes (letras:"p", "g", "q"), os ascendentes-descendentes (letras:"f","gh") e os laços no corpo das palavras (letras: "o","e"). As demais primitivas selecionadas são denominadas condicionais, compreendendo as primitivas que estão associadas entre si através de uma condição, por exemplo, o laço da letra "d" com o ascendente desta mesma letra. Côté ainda divide tais características em três tipos: primárias, secundárias e de vales. As características primárias são utilizadas para detectar letras-chave no corpo da palavra. As letras-chave são os componentes conectados que possuem traços nas regiões ascendentes e descendentes. Os componentes conectados que possuem laços em seu corpo são também considerados como sendo letras-chave. Características secundárias (b-loops, d-loops, ou as barras T) são condicionais, porque são apenas detectadas na presença de características primárias. As características de vale com cavidade para cima e/ou para baixo são extraídas do fundo da imagem. Os vales de cavidade para cima e de cavidade para baixo são componentes conectados do fundo da imagem, extraídos entre os contornos superior e inferior da palavra. Para a classificação a autora utiliza as redes neurais artificiais. Os resultados obtidos variam em 73,6% para a base de teste e 76,1% para a base de aprendizagem.

No estudo [SR98], Senior e Robinson acreditam que o desempenho de um reconhecedor pode ser melhorado utilizando-se mais informações sobre as primitivas ditas *notáveis* das palavra. As primitivas são extraídas sobre o esqueleto codificado da imagem da palavra através de um *grid*. As primitivas *notáveis* selecionadas pelos

autores foram:

- Pontos sobre as letras "i" e "j", ou ainda traços isolados sobre ou acima da linha média são marcadores potenciais dos pontos nestas letras (*dots*);
- Junções entre dois traços ou cruzamentos (*junctions*);
- Pontos finalizadores de traços (*endpoints*);
- Pontos que caracterizam a mudança de direção do traçado (*turning points*);
- Laços (*loops*).

Utilizam também a combinação de dois classificadores: Redes Neurais Recorrentes e Modelos Escondidos de Markov (MEM). A Rede Neural Recorrente é usada para estimar as probabilidades para cada *frame* de dados na representação. As probabilidades são combinadas em um MEM que encontra a melhor escolha de palavra que se associe aos dados observados. Uma base de 2.360 palavras para o treinamento, 675 para validação e 1.016 para testes.Os resultados experimentais são de aproximadamente 60% para um léxico de 10 palavras.

Ollivier em [Oll99], utiliza uma estratégia e um conjunto de características globais para o reconhecimento de palavras semelhantes a de [Mon95], pois também representa o módulo de reconhecimento como um macro-agente, composto de outros dois: uma rede neural e um algoritmo K-vizinhos mais próximos. A performance do sistema tratando o reconhecimento das palavras isoladamente no contexto de cheques franceses é dada em 60,4%.

No estudo de Freitas [Fre01], são testados diferentes conjuntos de primitivas denominados: PF (primitivas perceptivas)[MG01], PFCC (primitivas perceptivas, concavidades e convexidades) e o PFCCD (primitivas perceptivas, concavidades e convexidades rotuladas). Para todos, segue-se uma abordagem global para o reconhecimento, obtendo-se diferentes performances para cada conjunto, sendo de 56,4% (PF), 64,8% (PFCC) e 70,6% (PFCCD). Nesta metodologia, os Modelos Escondidos de Markov são usados como classificador, e uma base formada de 11.948 imagens de palavras manuscritas referentes ao valor por extenso de cheques bancários brasileiros é utilizada para a obtenção dos resultados experimentais.

A idéia básica do sistema de reconhecimento de Gomes em [GL01], diz respeito a transformação de uma palavra manuscrita em uma seqüência ordenada de linhas curvas, linhas retas e laços a fim de reduzir a variação de estilos de escrita. O sistema de reconhecimento proposto é composto por duas fases: uma fase de treinamento e uma fase de reconhecimento. Tanto na fase de treinamento como na fase de reconhecimento a imagem de uma palavra manuscrita é pré-processada, segmentada e suas características são extraídas. Os procedimentos incluídos no pré-processamento são bem conhecidos na literatura, nominalmente, suavização, rotação e correção de inclinação [Gui95]. Após o pré-processamento a imagem de uma palavra é segmentada em caracteres. Os resultados finais desta operação são letras e/ou partes de letras, ambas denominadas genericamente neste artigo como segmentos de palavra. No procedimento de extração de características cada segmento da palavra é decomposto em segmentos de linha, para os quais são calculados valores de pertinência a conjuntos *fuzzy* representando diferentes tipos de segmentos de linha curva e de linha reta. Dessa forma, é possível representar uma palavra como uma seqüência ordenada de linhas, sendo que cada uma dessas linhas apresenta um valor específico de pertinência a cada conjunto *fuzzy*. A referida seqüência é processada durante o procedimento de classificação por Modelos Escondidos de Markov *Fuzzy* (MEMFs). A palavra a ser reconhecida é classificada na classe de palavras que apresentar o maior valor de similaridade conforme um algoritmo de Viterbi *Fuzzy*. A base de 2.416 imagens é formada de palavras isoladas referentes ao valor por extenso em cheques bancários brasileiros. O resultados são obtidos através de um processo de validação cruzada fator 10, sendo a média final de 50%.

Park [Par02], argumenta que falta em métodos anteriores a habilidade de se adaptarem a entrada: imagem e léxico. A idéia básica nesta abordagem está em considerar o problema de classificação como um processo de decisão feito recursivamente em que características são assumidas como sendo variáveis randômicas. Características são disponíveis seqüencialmente a cada passo. Um *decision-making* decide entre as duas escolhas: de atualização da observação ou término com uma resposta. A melhor seqüência de características ordenada é obtida por Programação Dinâmica. O processo de reconhecimento começa do nível básico, usando características mínimas. Então, o *decision maker* testa os resultados de reconhecimento iniciais, considerando a inter-relação entre as entradas do léxico dadas, e aceita as

respostas mais qualificadas, próximo ao que acontece aos métodos convencionais. Se os resultados não atingem o critério de aceitação, o *decision maker* então rejeita somente as opções ruins e estreita as opções possíveis para a próxima iteração. O léxico reduzido (sem as opções falsas) reforça o reconhecedor para otimizar o curso do reconhecimento. Este *feedback* ajuda o reconhecedor de maneira adaptativa a selecionar as operações mais eficientes para o léxico alcançar as condições de término. Características de gradiente são diretamente geradas de uma representação de contorno baseada nos *pixels* de um caractere. Uma dada imagem de caractere é dividida em *Nf* por *Nf* células de igual área, onde *Nf* é o tamanho das divisões. Um histograma de *Ng* aberturas de gradientes de componentes de contornos é obtido para cada célula. A medida de normalização do histograma de todas as células é utilizada como um conjunto de características. Duas características globais, em relação ao aspecto da *bounding box* e uma relação de componente vertical/horizontal de contornos globais, são incluídas no conjunto de características. Subimagens são obtidas por divisões utilizando as linhas verticais e horizontais que cruzam no centróide dos contornos. Os vetores de características extraídos são agrupados independentemente em cada nó ao longo de cada classe, utilizando o algoritmo de agrupamento *K-means*. O procedimento de agrupamento é terminado se o mínimo erro quadrático para os centros de agrupamentos é menor do que um limiar pré-determinado, ou, se o número de centros de agrupamento alcança o número máximo permitido. A Técnica de Programação Dinâmica é utilizada para encontrar o melhor ajuste entre as possíveis cadeias de caractere de uma entrada do léxico e o vetor de segmento gerado. Em seu experimento o autor utilizou um léxico de 10 palavras de pouca semelhança, formado por nomes de cidades e estados. Os resultados tem acima de 97% de acerto.

A metodologia adotada por Arica e Yarman-Vural em [AYV02] emprega uma abordagem analítica em imagens em níveis de cinza, que é suportado pela imagem binária e um conjunto de características globais. A imagem do documento não é pré-processada para ruídos e normalização. Entretanto, parâmetros globais, tais como linhas superiores e inferiores ao corpo das palavras e o ângulo de inclinação dos caracteres são estimados e incorporados para melhorar a exatidão dos estágios de segmentação dos caracteres e reconhecimento. O esquema faz uso juntamente das imagens em nível de cinza e binária para extrair a máxima quantidade de informação

para ambos segmentação e reconhecimento. O algoritmo de segmentação proposto segmenta a palavra inteira em pedaços, cada um representando um caractere ou uma parte dele. O reconhecimento de cada segmento é realizado em três estágios: no primeiro estágio, caracteres são rotulados em três classes como ascendentes, descendentes, e caracteres normais. No segundo estágio, um Modelo Escondido de Markov (MEM) é aplicado para reconhecimento de formas. As características extraídas dos pedaços de cada segmento são submetidas a um MEM *left-right*. Os parâmetros do espaço de características são também estimados no estágio de treinamento do MEM. Finalmente, um algoritmo de reconhecimento em nível de palavra soluciona a cadeia de manuscritos pela combinação da informação do léxico e as probabilidades do MEM. O resultado obtido tem 92,3% de acerto para um léxico de 50 palavras, onde um conjunto de 1000 palavras de letras minúsculas é segmentado e usado para o treinamento do MEM, e um outro conjunto de 2000 palavras é usado para testar a performance.

Em [MYS+01], [MSBS02] e [MOS+02] os autores apresentam um sistema híbrido MEM-MLP que utiliza uma estratégia de segmentação implícita para o reconhecimento de datas em cheques bancários brasileiros. Uma rede neural MLP é usada para reconhecer os números, como o dia e o ano. E um MEM é utilizado para o reconhecimento das palavras. A imagem da palavra referente ao mês, é segmentada em grafemas e então as seguintes características são extraídas: global, uma mistura de concavidades e informações de contorno sobre os pontos de segmentação. Os autores dividem os grafemas em duas zonas, resultando em dois vetores de concavidade de 9 componentes cada. Para cada vetor, introduzem mais 8 componentes para aumentar a discriminação entre alguns pares de letras (e.g., "L" e "N"). Assim, o vetor de características final tem $(2 \times (9 + 8))$ 34 componentes. Portanto, a saída da extração de características é um par de descrições simbólicas, cada uma consistindo de uma seqüência alternante de grafemas de forma e símbolos de pontos de segmentações associados. Uma base de 9.500 palavras manuscritas é utilizada para treinamento e testes. Os resultados obtidos pelos autores variam entre 89,5% e 91,5%.

No estudo de Oliveira *et al.* [JdCCFS02], os autores avaliam o desempenho das Redes Neurais Artificiais (RNA) e dos Modelos Escondidos de Markov (MEM) como classificadores. Para isto, utilizam uma segmentação implícita da palavra em 8

regiões para as RNA e outra baseada no número de pontos obtidos por um processo de pseudo-segmentação para os MEM, e as seguintes características: perceptivas, que correspondem a posição e tamanho de ascendentes e descendentes, tamanho e localização de laços, ângulos de concavidade e estimativa do comprimento da palavra; e direcionais, em que para cada pixel do fundo da imagem é verificado se um *pixel* preto pode ser encontrado em cada uma das quatro direções principais (Norte, Sul, Leste e Oeste), para alimentarem o classificador neural. A metodologia utilizada para os Modelos Escondidos de Markov é a mesma descrita em [Fre01]. Para a obtenção dos resultados experimentais, utilizou-se uma base de 6.000 imagens de palavras manuscritas do idioma Português referentes aos meses do ano. As taxas de reconhecimento obtidas são distribuídas da seguinte forma:

- Características Perceptivas + Rede Neural Artificial: 81,8%
- Características Direcionais + Rede Neural Artificial: 76,6%
- Metodologia HMM [Fre01]: 75,9%

O resultado final apresentado é a combinação por multiplicação dos três métodos, resultando numa taxa de reconhecimento de 90,4%.

Concluindo, os trabalhos citados neste capítulo contribuem na elaboração do presente trabalho, como na escolha de características e classificadores, e principalmente ajudam a entender a complexidade do problema do reconhecimento de palavras manuscritas. No capítulo seguinte, são descritos: o conjunto de características, os classificadores, o mecanismo de rejeição e o método para análise das características utilizados na metodologia do presente trabalho para o reconhecimento de palavras manuscritas no contexto de cheques bancários brasileiros.

Capítulo 3

Metodologia Proposta

Neste Capítulo é descrita a metodologia aplicada no presente trabalho. De acordo com Gonzalez e Woods [GW95], é conceitualmente útil dividir o espectro de técnicas empregadas em análise de imagens em três áreas básicas. As três áreas são:

1. Processamento de baixo nível: trata de funções que podem ser vistas como ações automáticas, onde pode não se requerer nenhuma inteligência por parte do sistema de análise de imagem. As tarefas que se enquadram neste nível são, em geral, aquisição de imagens e pré-processamento. Como no presente trabalho, a metodologia proposta é empregada em bases de imagens já coletadas, torna-se desnecessário um processo de aquisição.

2. Processamento de nível intermediário: é responsável por tarefas de extração e caracterização de componentes, ou regiões, em uma imagem resultante de um processo de baixo nível. Os processos de nível intermediário abrangem, em geral, tarefas de segmentação e extração de características de componentes ou regiões da imagem.

3. Processamento de alto nível: envolve reconhecimento e interpretação de padrões. Na metodologia proposta neste Capítulo, o reconhecimento abrange tarefas de classificação e rejeição de imagens das palavras manuscritas.

Embora as subdivisões entre os processamentos não possuam fronteiras definitivas, elas provêm uma arquitetura útil para a categorização de vários processos que são componentes inerentes de um sistema de análise de imagens autônomo. A

Figura 3.1 ilustra esses conceitos e as etapas aplicadas na metodologia proposta para o reconhecimento de palavras manuscritas.

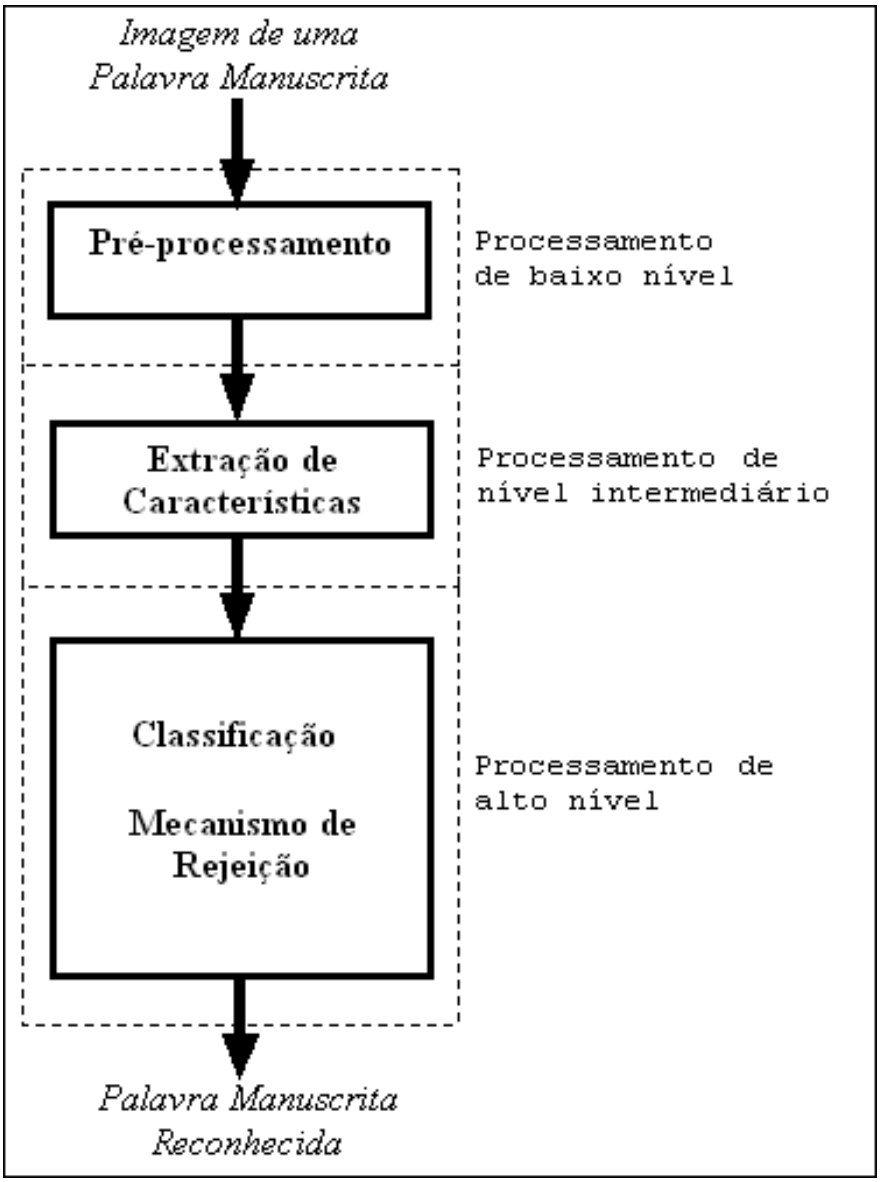

Figura 3.1: Diagrama aplicado para o reconhecimento de palavras manuscritas

A seguir são detalhadas cada uma das etapas para o reconhecimento de palavras manuscritas apresentadas na Figura 3.1.

3.1 Pré-processamento

O pré-processamento é necessário para diminuir os fatores de complexidade das palavras. Algumas das operações executadas antes do reconhecimento são: limiarização, que é a conversão de uma imagem em níveis de cinza numa imagem binária, a segmentação e remoção de ruídos, etapa onde a extração do texto de interesse ocorre pela remoção do fundo do documento, ruídos do tipo sal e pimenta e outros. Diversas segmentações podem ser feitas, assim como, segmentação do texto em linhas, segmentação das linhas em palavras e destas em caracteres.

Como no presente estudo, as imagens das bases de dados já passaram pelas etapas anteriores [Fre01] e [dOJ02], o pré-processamento envolve apenas as tarefas descritas a seguir:

a) Correção da inclinação da linha de base (*skew*): busca detectar o ângulo de inclinação com o eixo horizontal e corrigir este ângulo de tal forma que a escrita se torne horizontal. Em geral, os métodos de correção de linha de base podem ser locais ou globais. Os métodos globais realizam uma estimativa da inclinação da palavra considerando que a inclinação é valida para a palavra como um todo [Yac96]. Por outro lado, os métodos locais consideram que a inclinação horizontal das palavras não é constante e igual para a palavra inteira, realizando então pequenas correções localizadas. A Figura 3.2 exemplifica a correção de *skew* para a palavra manuscrita "fevereiro".

a) b)

Figura 3.2: Exemplo de correção da inclinação horizontal de uma palavra: a) Imagem original com inclinação horizontal e b) Imagem após correção de *skew*

b) Correção da inclinação vertical dos caracteres (*slant*): Inclinação vertical do caractere corresponde ao ângulo formado entre o eixo da direção de escrita dos caracteres e o eixo da vertical [Fre01]. O objetivo é reduzir a variabilidade da

escrita, tornando a palavra o mais vertical possível. A Figura 3.3 mostra a correção de *slant* aplicada para a imagem da palavra "sete".

a) b)

Figura 3.3: Exemplo de correção da inclinação vertical de uma palavra: a) Imagem original com inclinação vertical e b) Imagem após correção de *slant*

c) Detecção do corpo das palavras: A parte correspondente às letras minúsculas da palavra é denominada corpo da palavra. É onde se encontra a maioria das letras e conseqüentemente é de grande importância para uma abordagem baseada em primitivas perceptivas, conforme apresentado na Figura 3.4. Utiliza-se para a detecção informações obtidas através do pico do histograma horizontal de transição branco-preto da palavra.

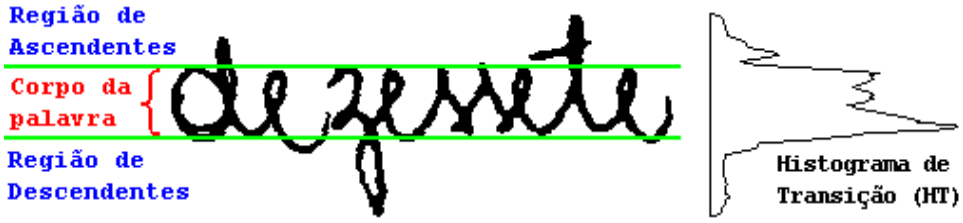

Figura 3.4: Exemplo de detecção do corpo da palavra e separação das regiões de ascendentes e descendentes

3.2 Extração de Primitivas

A maioria dos estudos de reconhecimento de padrões e mais especificamente de palavras manuscritas tem seu ponto forte na seleção de conjuntos de primitivas capazes de representar e discriminar as diferentes formas encontradas. Escolher primitivas "adequadas" não é uma tarefa fácil ou trivial. Muitas técnicas de extração

de primitivas têm sido desenvolvidas e aplicadas ao reconhecimento de manuscritos, podendo-se resumir em 3 classes [Heu94]:

a) Primitivas baseadas em transformadas globais e séries de expansão: Transformadas e séries de expansão, tais como, Fourier, Walsh, Harr e outras, fornecem primitivas invariantes a algumas deformações globais, por exemplo, translação e rotação. Entretanto, tais técnicas apresentam um custo computacional alto no que se refere a tempo;

b) Primitivas baseadas na distribuição estatísticas dos pontos: Estas primitivas incluem momentos, n-tuplas, *crossing* e distâncias. São tolerantes a distorções e levam em conta, para alguns casos, as variações de estilo. Implicam em baixa complexidade de implementação;

c) Primitivas geométricas e perceptivas: Estas são as primitivas mais empregadas para representar global e localmente as propriedades dos caracteres. Estão incluídos nesta classe os ascendentes, descendentes, laços, traços, barras em diferentes direções, pontos finalizadores, interseções de segmentos de linha, laços, relação entre traços e propriedades angulares. Estas primitivas tem alta tolerância a distorções, variações de estilo, translação e rotação.

No presente estudo, o conjunto de características adotado é formado por primitivas geométricas e perceptivas. Trata-se na maior parte de contagens de ocorrências de número de laços, concavidades, convexidades, traços horizontais, verticais, etc. Um maior detalhamento é dado mais adiante. Entretanto, de acordo com [TJT96], somente estas primitivas discretas não conduzem a sistemas de reconhecimento robustos, então com o intuito de aumentar a discriminação entre as formas a classificar, adicionou-se ao conjunto de característica a proporção de pixels que fazem parte do traçado e um mecanismo de zoneamento (*zoning*) no momento da captura de cada característica.

O zoneamento faz-se em somente duas regiões separadas pelo centro de gravidade da palavra, a região da esquerda e da direita da palavra, conforme mostrado na Figura 3.5, isto se justifica visto que separando as ocorrências das primitivas obtêm-se a informação sobre o posicionamento das mesmas dentro da palavra, o que dá mais precisão na classificação das formas. Tendo-se em mente o problema

da maldição da dimensionalidade abordado em [JDM00], em que a performance de um classificador depende do relacionamento entre número de amostras e número de primitivas. Assim, assume-se duas regiões para que o tamanho do vetor que aumenta proporcionalmente à quantidade de regiões não se estenda e para que primitivas irrelevantes não sejam processadas, ou ainda para evitar que o número de amostra se torne pequeno em relação ao número total de primitivas.

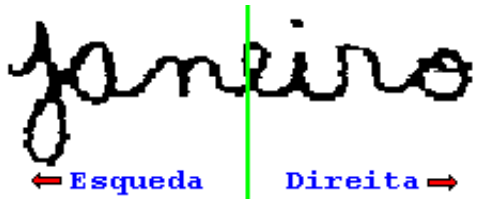

Figura 3.5: Exemplo do *zoning* aplicado

Demonstrando em detalhes, o conjunto proposto e implementado é composto por:

1. Número de laços contidos nos lados esquerdo e direito da linha vertical que passa pelo centro de gravidade (CG) da palavra (NLE e NLD), conforme mostrado na Figura 3.6, onde para este exemplo, os valores de NLE e NLD são 2 e 3 respectivamente.

Figura 3.6: Exemplo de laços identificados

2. Número de semicírculos côncavos que fazem parte do corpo da palavra nas regiões esquerda e direita do CG (NSCVE=3 e NSCVD=5). A Figura 3.7 exemplifica a obtenção desta primitiva.

3. Número de semicírculos convexos que fazem parte do corpo da palavra nas regiões esquerda e direita do CG (NSCXE=3 e NSCXD=3), como demonstrado na Figura 3.8. As concavidades e convexidades são extraídas somente

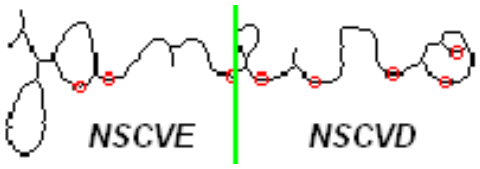

Figura 3.7: Exemplo de obtenção das concavidades

no corpo das palavras esqueletizadas, obtidas através do algoritmo de Holt [HSCP97]. Os pontos convexos são determinados com o auxílio de 5 elementos estruturantes diferentes, e os pontos côncavos com uma família de 10 elementos estruturantes. São primitivas complementares, ou seja, auxiliam na representação das curvaturas das letras e ligações entre letras, ou ainda, de laços abertos existentes no corpo das palavras.

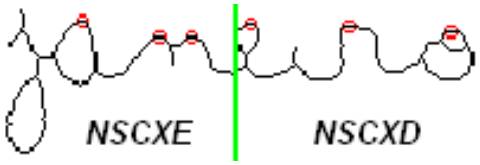

Figura 3.8: Exemplo de obtenção das convexidades

4. Número de pontos de cruzamento (*crossing points*) à esquerda e direita do CG (NCPE=1 e NCPD=1), como ilustrado na Figura 3.9.

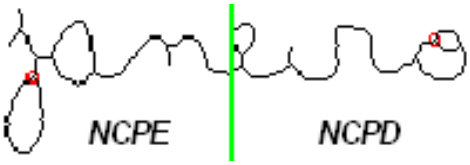

Figura 3.9: Exemplo da obtenção de *crossing points*

5. Número de pontos de ramificação (*branch points*) à esquerda e direita do CG (NBPE=3 e NBPD=6), conforme exemplificado na Figura 3.10.

6. Número de pontos finalizadores (*endpoints*) à esquerda e à direita do CG (NEPE=3 e NEPD=1). A Figura 3.11 apresenta a obtenção de *endpoints*.

   As primitivas referentes aos itens 4, 5 e 6 também são obtidas basicamente através de elementos estruturantes, sendo que a busca destas envolve toda a

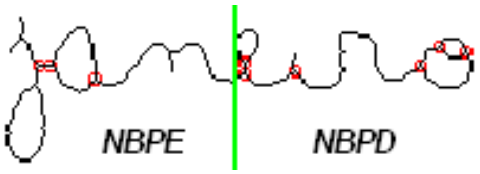

Figura 3.10: Exemplo da obtenção de *branch points*

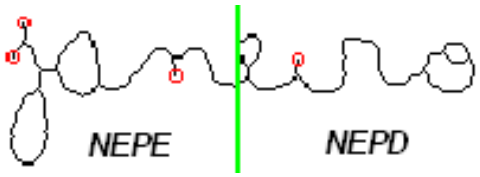

Figura 3.11: Exemplo da obtenção de *endpoints*

palavra, ou seja, o corpo da palavras juntamente com as regiões de ascendentes e descendentes.

7. Número de cruzamentos com o eixo horizontal que passa pela palavra (NCH=14). O eixo corresponde a linha média obtida considerando-se a altura do corpo da palavra. A Figura 3.12 ilustra a obtenção dessa primitiva.

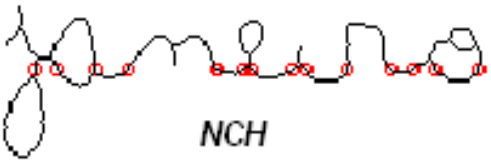

Figura 3.12: Exemplo da obtenção de cruzamentos com o eixo horizontal

8. Número de ascendentes no lado esquerdo e direito da linha vertical (NAE=0 e NAD=0) que passa pelo CG. Representam o que está acima do limite superior do corpo da palavra.

9. Número de descendentes no lado esquerdo e direito da linha vertical que passa pelo centro de gravidade da palavra (NDE=1 e NDD=0). Representam o que está abaixo do limite inferior do corpo da palavra.

10. Proporção de pixels que fazem parte do traçado em relação ao contexto da palavra (NPP=0,955324). Utiliza-se a *minimal bounding box* ao redor da palavra, para que a proporção obtida pela Equação 3.1 seja calculada sobre os

limites reais da palavra, como mostrado na Figura 3.13.

$$prop = \frac{(tp - tpp)}{tp} \tag{3.1}$$

Na equação 3.1, *prop, tp, tpp* correspondem à proporção de branco não preenchida, total de pixels no interior da *minimal bouding box* e total de pixels do traçado da palavra esqueletizada, respectivamente.

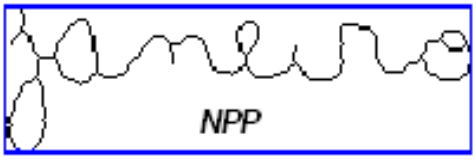

Figura 3.13: Exemplo da obtenção da proporção de pixels

11. Número de traços verticais (NTV=7), conforme ilustrado na Figura 3.14, determinados através da utilização de elementos estruturantes que representam linhas verticais.

Figura 3.14: Exemplo da obtenção de traços verticais

12. Número de traços horizontais (NTH=0), obtidos através de elementos estruturantes que representam linhas horizontais.

13. Número de laços ascendentes contidos no lado esquerdo e direito (NLAE=0 e NLAD=0).

14. Número de laços descendentes contidos no lado esquerdo e direito (NLDE=1 e NLDD=0). A Figura 3.15 exemplifica a obtenção desta primitiva.

Figura 3.15: Exemplo da obtenção de laços descendentes

3.3 Representação das Primitivas

Nas primitivas estruturais a representação é função direta das próprias primitivas extraídas e da localização destas na imagem analisada. Nas primitivas estatísticas a representação é em termos de d primitivas ou medidas e é vista como um ponto no espaço de d-dimensões. Assim, as principais formas de representação são [Heu94, MG01]:

- Vetores de Características e Matrizes: Normalmente a imagem é dividida em zonas (*zoning*) utilizando-se uma grade fixa ou variável (segmentação implícita) Para cada zona se extraem vetores ou matrizes de dados. Contagem de pixels, número de ascendentes, descendentes, laços, entre outros. A verificação da presença ou ausência de barras (letra T), entre outros. É muito utilizada em abordagens globais. Freqüentemente utilizada para descartar objetos não similares.

- Seqüências: A imagem é representada por uma seqüência de símbolos (*codebooks*). A obtenção da seqüência respeita a ordem de ocorrência dos símbolos na imagem. Quando se trata de palavras define-se um conjunto de símbolos com base nos grafemas extraídos. Grafema é o conjunto de todas as combinações de primitivas extraídas para as palavras [Fre01].

- Estruturas de Grafos: A imagem é representada por um grafo tendo as primitivas como nós e a relação espacial entre estas como as arestas ou ligações.

Como no presente estudo as primitivas são formadas pela combinação entre *zoning* e contagens opta-se por uma representação das primitivas por vetores de 24 dimensões, assim como mostrado na Figura 3.16.

1	2	3	4	5	6	7	8	9	10	11	12
NLE	NLD	NSCVE	NSCVD	NSCXE	NSCXD	NCPE	NCPD	NBPE	NBPD	NEPE	NEPD

13	14	15	16	17	18	19	20	21	22	23	24
NAE	NAD	NDE	NDD	NCH	NPP	NTV	NTH	NLAE	NLDE	NLAD	NLDD

Figura 3.16: Representação das primitivas

3.4 Classificação

Classificação é uma das mais freqüentes tarefas de tomada de decisão da atividade humana. Um problema de classificação acontece quando um objeto precisa ser associado a um determinado grupo ou classe baseando-se em um número de atributos observados e relacionados àquele objeto [Zha00]. Muitos problemas em negócios, ciências, indústrias, e medicamentos podem ser tratados como problemas de classificação, por exemplo: crédito *scoring*, diagnose médica, controle de qualidade, reconhecimento de voz e caracteres manuscritos.

Os procedimentos tradicionais de classificação estatística são realizados, na maioria dos casos, baseando-se na teoria de decisão de Bayes, assim como ocorre em análises discriminantes. Nestes procedimentos, um modelo de probabilidade *a priori* deve ser assumido para que a probabilidade *a posteriori* possa ser calculada e a tomada de decisão de classificação realizada. Porém, uma das limitações principais dos modelos estatísticos é que eles trabalham bem apenas quando as suposições criadas inicialmente são satisfatórias. A efetividade destes métodos depende de uma grande quantidade de suposições ou condições sobre as quais os modelos são desenvolvidos [Zha00]. Os usuários devem ter um bom conhecimento das propriedades dos dados e das capacidades do modelo, antes que os mesmos possam ser aplicados definitivamente.

As redes neurais artificiais (RNAs) emergiram como uma ferramenta importante para classificação. As recentes atividades de pesquisa são vastas em classificação neural, estabelecendo-as como uma alternativa promissora para vários métodos de classificação convencionais [Zha00].

No presente trabalho, a classificação é realizada com a utilização de redes neurais artificiais MLP (*Multilayer Perceptron*). A vantagem dessas redes neurais artificiais situa-se nos seguintes aspectos teóricos. Primeiramente, as RNAs são métodos

auto-adaptativos e dirigidos pelos dados, ou seja, ajustam-se aos dados por conta própria, sem qualquer especificação explícita da forma de distribuição ou função para um dado modelo. Segundo, elas são aproximadores funcionais universais, pois aproximam qualquer função com precisão arbitrária [Cyb89], [Hor91], [HSW89]. Considerando que qualquer procedimento de classificação busca uma relação funcional entre um grupo relacionado e os atributos do objeto, a identificação precisa desta função é sem dúvida importante. Terceiro, RNAs são modelos não-lineares, o que as fazem flexíveis na modelagem de relacionamentos complexos do mundo real. Finalmente, RNAs podem calcular as probabilidades *a posteriori* que provêem a base para estabelecer regras de classificação e desempenhar análises estatísticas [RL91], [Jor95].

As redes neurais foram desenvolvidas, originalmente, na década de 40, pelo neurofisiologista Warren McCulloch, do *Massachusetts Institute of Technology* (MIT), e pelo matemático Walter Pitts, da Universidade de Illinois, os quais, dentro do espírito cibernético, fizeram uma analogia entre células nervosas vivas e o processo eletrônico num trabalho publicado sobre "neurônios formais"[MP43]. O trabalho consistia num modelo de resistores variáveis e amplificadores representando conexões sinápticas de um neurônio biológico.

Desde então, mais enfaticamente a partir da década de 80, diversos modelos de redes neurais artificiais têm surgido com o propósito de aperfeiçoar e aplicar este método. O neurônio artificial é uma estrutura lógico-matemática que procura simular a forma, o comportamento e as funções de um neurônio biológico. Assim sendo, os dendritos foram substituídos por **entradas**, cujas ligações com o corpo celular artificial são realizadas através de elementos chamados de **pesos** (simulando as sinapses). Os estímulos captados pelas entradas são processados pela **função de soma**, e o limiar de disparo do neurônio biológico foi substituído pela **função de transferência**, conforme mostrado na Figura 3.17.

Combinando diversos neurônios artificiais, ou elementos de processamento (PEs) como também são chamados [PEL99], podemos formar o que é chamado de rede neural artificial, Figura 3.18. As entradas, simulando uma área de captação de estímulos, podem ser conectadas em muitos neurônios, resultando, assim, em uma série de saídas, em que cada neurônio representa uma saída. Essas conexões, em comparação com o sistema biológico, representam o contato dos dendritos com ou-

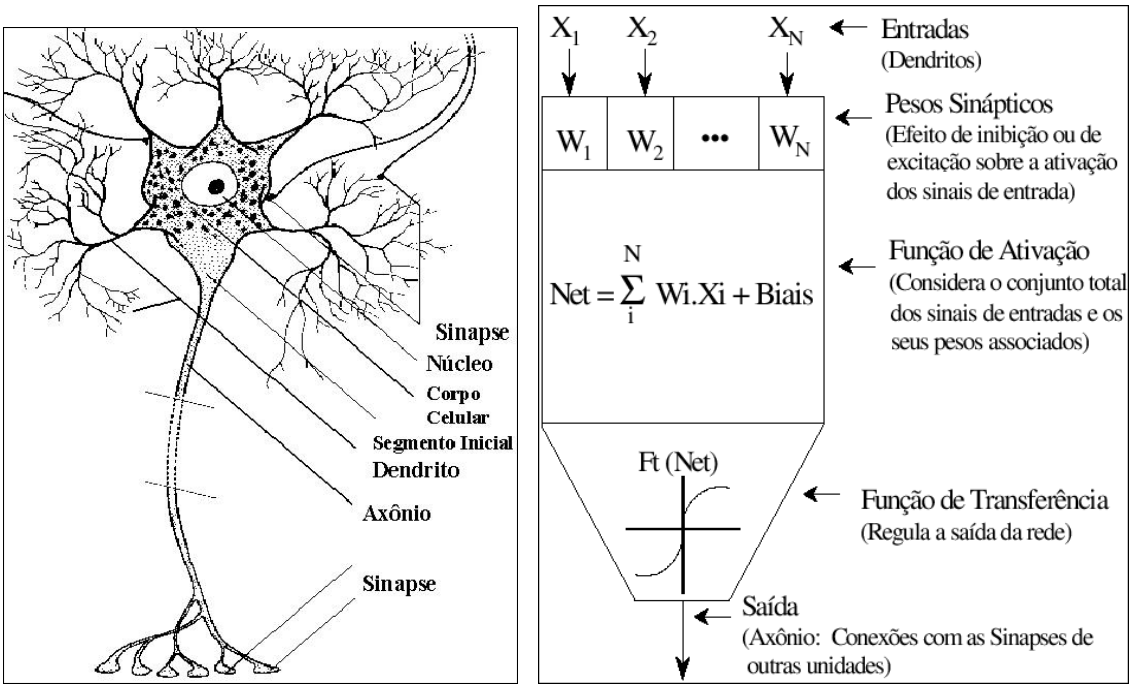

Figura 3.17: Ilustração de: a) Um neurônio biológico e b) Um neurônio artificial

tros neurônios, formando assim as sinapses. A função da conexão é tornar o sinal de saída de um neurônio em um sinal de entrada de outro, ou ainda, orientar o sinal de saída para o mundo real. As diferentes possibilidades de conexões entre as camadas de neurônios podem gerar várias estruturas diferentes.

As variantes de uma rede neural são muitas, e combinando-as, podemos mudar a arquitetura conforme a necessidade da aplicação. Basicamente, os itens que compõem uma rede neural, portanto, sujeitos a modificações, são os seguintes: conexões entre camadas, número de camadas intermediárias, quantidade de neurônios, função de transferência e algoritmo de aprendizado.

Na metodologia proposta neste trabalho, utilizam-se duas arquiteturas de RNAs-MLP denominadas: Arquitetura Convencional e Arquitetura Classe-Modular, assim como em [OS02] e [KFNS03], com o intuito de testá-las juntamente com o conjunto de características proposto. Um detalhamento sobre as arquiteturas é descrito a seguir.

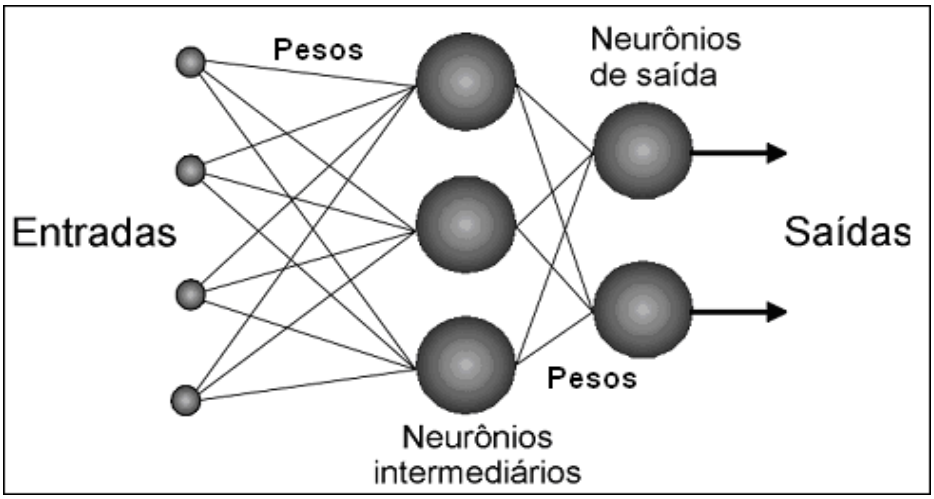

Figura 3.18: Exemplo de uma Rede Neural Artificial MLP de 2 camadas com 4 entradas e 2 saídas

3.4.1 Arquitetura Convencional

A arquitetura convencional é similar a uma rede neural artificial MLP utilizada para a classificação de todo um léxico. No presente trabalho, a RNA-MLP é formada como segue:

- A quantidade de entradas da rede é correspondente à quantidade de elementos do vetor de características mostrado na Figura 3.16 da Seção 3.3.
- O número de neurônios na camada escondida é variável.
- A quantidade de neuronios na saída está relacionada ao tamanho do léxico tratado.

As Figuras 3.18 e 3.19 exemplificam RNA-MLP convencionais. Detalhes sobre as quantidades de neurônios nas camadas, algoritmo de aprendizagem utilizado, inicialização de pesos entre outros, estão descritos no Capítulo 4 para cada léxico relacionado no presente trabalho.

3.4.2 Arquitetura Classe-Modular

A sugestão nesta arquitetura é a modularidade das classes, da forma que cada módulo é responsável por um problema 2-classificação, sendo que a discriminação

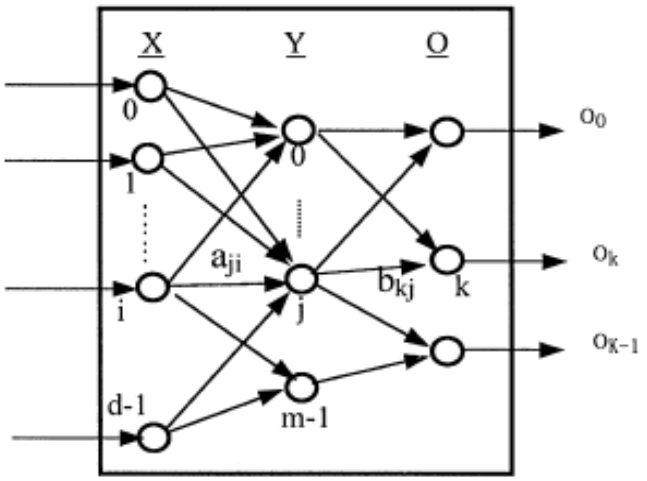

Figura 3.19: Arquitetura convencional onde K classes estão misturadas [OS02]

das amostras é feita em $K-1$ classes.

Em [OS02], os autores iniciam com uma nota sobre o fato das redes neurais convencionais possuirem uma natureza modular somente nos níveis dos neurônios (granularidade fina) e nas camadas (granularidade grossa). Consideram as RNAs convencionais como boas, porém argumentam que a determinação de ótimas bordas de decisões para K-classificações em reconhecimento de caracteres em um espaço de características com grande dimensionalidade torna-se uma tarefa muito complexa, e pode seriamente limitar a performance dos sistemas de reconhecimento de caracteres usando redes neurais artificiais MLP.

Principe *et al.* [PEL99], Oh e Suen [OS02] citam problemas de convergência quando utiliza-se uma rede grande em uma aplicação específica. Um dos problemas que pode ocorrer na convergência é, principalmente, quando um conjunto de treinamento não é grande o suficiente comparado com o tamanho do classificador, isto é, com o número de parâmetros livres no classificador (pesos). Então uma solução seria possuir um conjunto de treinamento tão grande quanto à rede, o que nem sempre é possível ou trivial de se obter. De acordo com Oh e Suen, a arquitetura convencional tem uma estrutura rígida composta de uma caixa preta em que todas as K classes estão juntas e misturadas. Os módulos não podem ser modificados ou otimizados localmente para cada classe.

Na arquitetura classe-modular, o módulo de classificação da linha tradicional de

reconhecimento apresentada na Figura 3.19 é substituído por K subredes, M_{wi} para $0 \leqslant i < K$, cada uma referente a uma classe. A tarefa específica de cada M_{wi} é selecionar entre dois grupos de classes, conforme mostrado na Figura 3.20. Ω_0 e Ω_1, com $\Omega_0 = \{w_i\}$ e $\Omega_1 = \{w_k \mid 0 \leqslant k < K\}$ e $k \neq i$, ou seja, com apenas duas saídas, classificando se determinado exemplo pertence a classe ou não. As redes M_{wi} foram projetadas da mesma maneira como uma não-modular RNA-MLP mostrada na Figura 3.19.

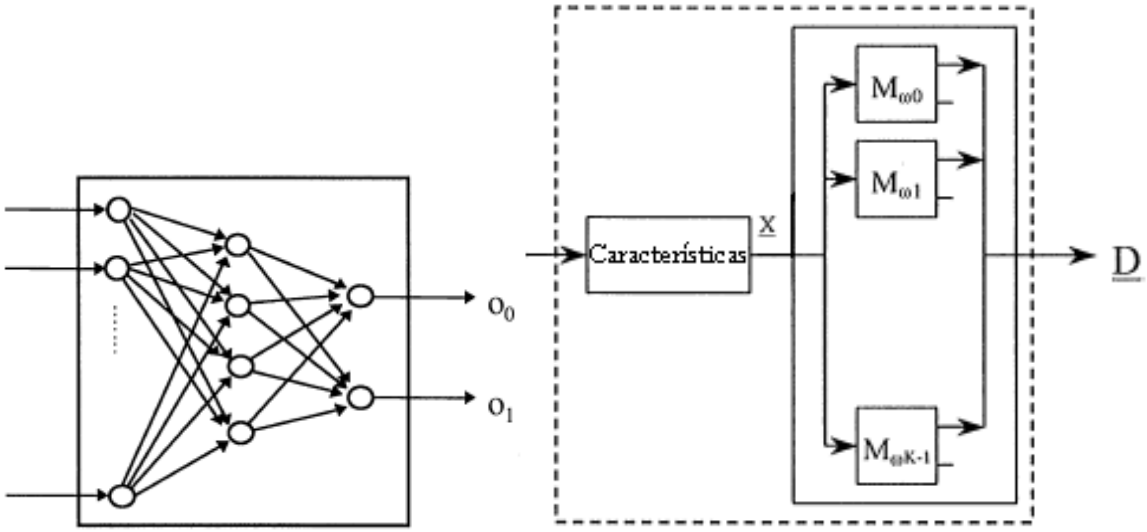

Figura 3.20: Arquitetura para uma RNA-MLP classe-modular [OS02]: a) Uma subrede M_{wi} e b) A RNA-MLP classe-modular inteira.

As três camadas são totalmente conectadas. A camada de entrada tem d nós de entrada para aceitar o vetor de característica d-dimensional, a camada de saída tem dois nós de saída, denotados por O_0 e O_1 para Ω_0 e Ω_1 respectivamente.

A arquitetura para a rede inteira formada por K sub-redes é mostrada na Figura 3.20. O módulo de extração de características extrai um vetor X que será usado comumente para todas as K classes. X é aplicado para a camada de entrada de todas as sub-redes e cada M_{wi} efetua os calculos do processo *forward* usando seu próprio conjunto de pesos para produzir um vetor de saída $D = (O_0, O_1)$. Depois os valores de O_0 constituem o vetor de decisão final assim como também ocorre em [OS02] e [KFNS03].

O treinamento e o reconhecimento na arquitetura classe-modular é o mesmo

realizado em [KFNS03]: cada um dos K 2-classificadores é treinado independentemente de outras classes. O algoritmo de aprendizagem *backpropagation* do erro é aplicado para cada um dos 2-classificadores da mesma maneira como ocorre na arquitetura convencional MLP. Os conjuntos de treinamento são preparados para os K 2-classificadores, separando-o em dois grupos, Z_{Ω_0} e Z_{Ω_1}, tais que Z_{Ω_0} contém as amostras das classes em Ω_0 e Z_{Ω_1} para as restantes Ω_1. A mesma separação é feita no conjunto de validação. No estágio de reconhecimento os valores obtidos das saídas das sub-redes, os O_0 são utilizados juntamente com um simples "vencedor-leva-tudo"(*winner-take-all*) para determinar qual é a classe final. Os conjuntos de treinamento e validação para um 2-classificador não são balanceado entre as 2 classes Ω_0 e Ω_1, o que acarreta em mais exemplos para Ω_1 do que para Ω_0.

Assim como para arquitetura convencional, no Capítulo 4 são detalhadas as quantidades de neurônios nas camadas intermediárias, o algoritmo de aprendizagem utilizado, a inicialização de pesos entre outros, para cada léxico estudado neste trabalho.

3.5 Rejeição

Geralmente, sistemas de reconhecimento aplicam uma decisão global que decide entre aceitar o resultado do reconhecimento ou rejeitá-lo. Em classificação, um padrão é considerado ambíguo se ele não pode ser associado a uma classe com determinada certeza, enquanto que um padrão associado com baixa confiança para todas as classes em hipótese pode ser tratado como um "dado falso" (*outlier*).

O objetivo do mecanismo de rejeição é minimizar o número de erros de reconhecimento para um dado número de rejeições. Um esquema simples de rejeição é rejeitar a imagem que tem uma probabilidade global menor do que um determinado limiar, como denotado pela regra de Chow [Cho70].

Agora, considere uma tarefa simples de classificação unidimensional com duas classes w_1 e w_2 caracterizadas por distribuições Gaussianas, como mostrado na Figura 3.21. Os termos $P(w_i \mid x)$ e $\hat{P}(w_i \mid x)$, $i = 1, 2$, indicam as probabilidades *a posteriori* "verdadeiras" e "estimadas", respectivamente. Fumera *et al.* em [FRG00], apontam a hipótese de que erros significantes afetam as probabilidades estimadas nas variações dos valores das características nas quais duas classes estão

"sobrepostas". As regiões ótimas de decisão e rejeição providas pela regra de Chow aplicada para as probabilidades "verdadeiras" são indicados pelos termos D_1, D_2 e D_0. O termo T indica um limiar de rejeição de Chow. Analogamente, os termos $\hat{D}_1, \hat{D}_2$ e $\hat{D}_0$ indicam regiões de decisão e rejeição providas pela regra de Chow aplicada para probabilidades estimadas.

Uma análise cuidadosa da Figura 3.21 sugere uma abordagem diferente da regra de Chow para a obtenção de ótimas bordas de erro-rejeição, principalmente quando as probabilidades *a posteriori* são afetadas pelos erros. A Figura 3.21 mostra que as regiões estimadas diferem-se das ótimas nos intervalos $(\hat{D}_1 - D_1)$ e $(D_2 - \hat{D}_2)$. Em particular, a regra de Chow erroneamente aceita os padrões pertencentes ao intervalo $(\hat{D}_1 - D_1)$, visto que a probabilidade *a posteriori* $\hat{P}(w_1 \mid x)$ contém valores superiores aos "verdadeiros" neste intervalo. Sendo que o correto seria que tais valores fossem rejeitados utilizando um valor de limiar $T_1 > T$. Da mesma forma, os padrões pertencentes a $(D_2 - \hat{D}_2)$ são erroneamente rejeitados, pois a probabilidade *a posteriori* $\hat{P}(w_2 \mid x)$ contém valores inferiores aos "verdadeiros" dentro deste intervalo. Tais padrões deveriam ser corretamente aceitos utilizando um valor de limiar $T_2 < T$.

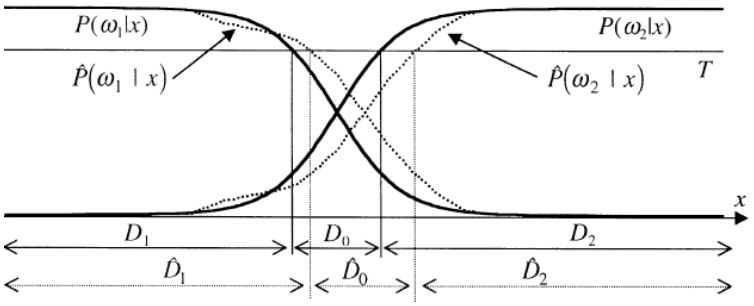

Figura 3.21: Aplicação da regra de Chow para as probabilidades *a posteriori* "verdadeiras" e "estimadas" [FRG00]

Portanto, é fácil ver, como ilustrado na Figura 3.21, que este limiar T aplicado para as probabilidades estimadas não permite obter para ambas as classes, as suas ótimas regiões de decisão e rejeição. De acordo com este exemplo, Fumera *et al.* sugerem o uso de N limiares de rejeição relacionados a cada classe (*Class-Related Thresholds-CRTs*). A Figura 3.22 mostra o uso de dois limiares de rejeição diferentes T_1 e T_2 para a tarefa de classificação da Figura 3.21.

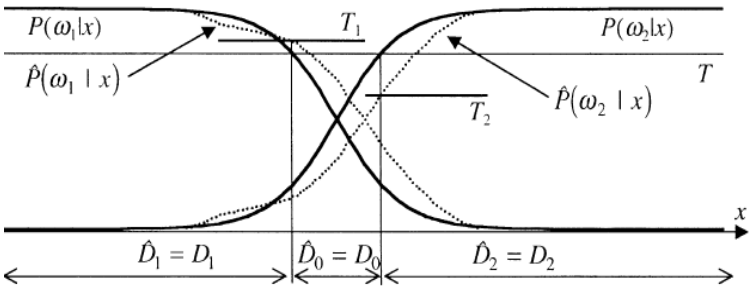

Figura 3.22: Dois limiares diferentes T_1 e T_2, aplicados para as probabilidades *a posteriori* estimadas da tarefa de classificação na Figura 3.21 [FRG00]

No presente trabalho, são investigados os efeitos das estimativas de erro da regra de Chow e das CRTs baseadas em múltiplos limiares de rejeição relacionados aos dados das classes.

3.5.1 Opção de Rejeição com Múltiplos Limiares

Um classificador de N-classes é utilizado para subdividir o espaço de características em N regiões de decisão D_i, $i = 0, \ldots, N-1$, tais que os padrões das classes w_i pertençam a região D_i. De acordo com a teoria de reconhecimento de padrões estatística, tais regiões de decisão são definidas para maximizar a probabilidade de reconhecimento correto, comumente chamada de precisão do classificador, conforme a Equação (3.2):

$$Precisao = P(correto) \sum_{i=0}^{N-1} \int_{D_i} p(x \mid w_i) P(w_i) dx \tag{3.2}$$

E, consequentemente, para minimizar a probabilidade de erro do classificador, conforme a Equação (3.3):

$$P(erro) = \sum_{i=0}^{N-1} \int_{D_i} \sum_{\substack{j \neq i, \\ j=0}}^{N-1} p(x \mid w_j) P(w_i) dx \tag{3.3}$$

Para este fim, a então denominada regra de decisão de Bayes associa cada padrão x à classe para qual a probabilidade *a posteriori* $P(w_i \mid x)$ é máxima. Uma probabilidade mais baixa do que uma provida pela regra de Bayes pode ser obtida utilizando a chamada opção de rejeição [FRG00]. Nominalmente, os padrões que são os mais

propensos a serem classificados erroneamente são rejeitados, ou seja, não classificados.

A formulação de uma melhor borda entre rejeição e erro foi relacionada por Chow em [Cho70]. De acordo com a regra de Chow, um padrão x é rejeitado se:

$$max_{k=0,\ldots,N-1} P(w_k \mid x) = P(w_i \mid x) < T \tag{3.4}$$

Onde $T \in [0, 1]$. Por outro lado, o padrão x é aceito e associado a classe w_i, se:

$$max_{k=0,\ldots,N-1} P(w_k \mid x) = P(w_i \mid x) \geq T \tag{3.5}$$

O espaço de características é subdividido em $N + 1$ regiões. A região de rejeição D_n é definida de acordo com a Equação (3.4), enquanto a região de decisão $D_0, \ldots, D_{n-1}$ são definidas de acordo com a Equação (3.5). É fácil ver que a probabilidade de um padrão ser rejeitado pode ser computada como segue:

$$P(rejeicao) = \int_{D_n} p(x)dx \tag{3.6}$$

Em que $p(x)$ representa uma função densidade de probabilidade. Em contraste, a precisão do classificador é definida como a probabilidade condicional que um padrão classificado corretamente, dado que ele tenha sido aceito, Equação (3.7):

$$Precisao = P(correto \mid aceito) = \frac{P(correto)}{P(correto) + P(erro)} \tag{3.7}$$

De acordo com Fumera *et al.* em [FRG00], uma análise do trabalho de Chow permite apontar que sua regra provê uma borda ótima de erro-rejeição, somente se as probabilidades *a posteriori* são exatamente conhecidas. Infelizmente, em aplicações do mundo real, tais probabilidades são afetadas estimativas de erro significantes, como observado em reconhecimento de palavras manuscritas.

Em [FRG00], os autores sugerem o uso de múltiplos limiares de rejeição, para várias classes de dados, para obter as ótimas regiões de decisão e rejeição, mesmo se as probabilidades *a posteriori* são afetadas por erros.

Tais limiares aplicados para as probabilidades estimadas permitem obter ambas regiões de decisão e rejeição. Portanto, baseados no exemplo da Figura 3.22, Fumera

et al. sugerem que o uso de N limiares de rejeição classe-relacionados (CRTs), podem prover uma melhor borda de decisão erro-rejeição do que a regra de Chow [Cho70].

Em particular, sobre a suposição de que as probabilidades *a posteriori* são afetadas por erros significantes, em [FRG00], os autores tem provado em seus experimentos que, para qualquer taxa de rejeição R, existem tais valores dos CRTs $T_0, \ldots, T_{N-1}$ que correspondem a precisão de um classificador $A(T_0, \ldots, T_{N-1})$ ser igual ou superior à precisão $A(T)$ provida pela regra de Chow, dada pela Equação (3.8):

$$\forall R \; \exists \; T_0, T_1, \ldots, T_{N-1} : A(T_0, T_1, \ldots, T_{N-1}) \geq A(T) \tag{3.8}$$

Portanto, em [FRG00], os autores propõem a seguinte regra de rejeição, denominada de regra CRT, para uma tarefa de classificação com N classes de dados que são caracterizadas por probabilidades *a posteriori* estimadas $\hat{P}(w_i \mid x), i = 0, \ldots, N-1$.

Um padrão x é rejeitado se:

$$max_{k=0,\ldots,N-1}\hat{P}(w_k \mid x) = \hat{P}(w_i \mid x) < T_i \tag{3.9}$$

Enquanto um padrão x é aceito e associado a classe w_i, se:

$$max_{k=0,\ldots,N-1}\hat{P}(w_k \mid x) = \hat{P}(w_i \mid x) \geq T_i \tag{3.10}$$

Os CRTs são valores entre $[0, 1]$. Por analogia com a regra de Chow, os valores dos CRTs deve ser calculados de acordo com a tarefa de classificação à mão em aplicações reais [FRG00].

Nos experimentos realizados no presente trabalho, tais como [FRG00], considera-se a habitual exigência erro-rejeição de aplicações de reconhecimento de padrões reais, isto é, obtendo a mais alta precisão e uma taxa de rejeição abaixo de um dado valor R_{Max}. Adaptações podem ser feitas a [FRG00], como a busca de uma taxa de erro abaixo de um determinado valor E_{Min}, validando também a Equação (3.11):

$$\forall E \; \exists \; T_0, T_1, \ldots, T_{N-1} : A(T_0, T_1, \ldots, T_{N-1}) \geq A(T) \tag{3.11}$$

Os valores CRT são estimados através da resolução do seguinte problema de

maximização, conforme mostram as Equações (3.12) ou (3.13).

$$\begin{cases} max_{T_0,\ldots,T_{N-1}} A(T_0,\ldots,T_{N-1}) \\ R(T_0,\ldots,T_{N-1}) \leq R_{Max} \end{cases} \tag{3.12}$$

Observa-se que, de acordo com as Equações (3.8) e (3.11), para qualquer dado R_{Max} ou E_{Min}, os valores CRT obtidos como soluções do problema de maximização acima provê uma precisão igual ou superior a obtida na regra Chow.

$$\begin{cases} max_{T_0,\ldots,T_{N-1}} A(T_0,\ldots,T_{N-1}) \\ E(T_0,\ldots,T_{N-1}) \leq E_{Min} \end{cases} \tag{3.13}$$

Em aplicações reais, as funções $R(T_0,\ldots,T_{N-1})$, $E(T_0,\ldots,T_{N-1})$ e $A(T_0,\ldots,T_{N-1})$ pode ser estimada conforme as Equações (3.6) e (3.7) utilizando um conjunto de validação finito. Para isto, utiliza-se um número de valores finitos no intervalo $[0,1]$ e a Equação (3.10), representando um problema de maximização restringida, cujas funções "alvo" e "restrição" são funções estimadas discretas de variáveis contínuas.

O algoritmo proposto no presente trabalho, leva em conta que $R(T_0,\ldots,T_{N-1})$ e $E(T_0,\ldots,T_{N-1})$ são funções crescentes das variáveis $T_0,\ldots,T_{N-1}$, isto é, o número de padrões rejeitados não pode diminuir, assim como os erros aumentar, pelo crescimento dos valores dos CRTs. Consequemente, assume-se também que $A(T_0,\ldots,T_{N-1})$ é uma função crescente de $T_0,\ldots,T_{N-1}$.

Resumindo, a idéia básica é resolver a Equação (3.12) ou (3.13) iterativamente. Inicia-se com os valores dos CRTs que provêm uma taxa de rejeição igual a zero, e em cada passo aumenta-se o valor de um dos CRTs. Este fato gera o aumento da precisão até que a taxa de rejeição exceda o valor de R_{Max} ou E_{Min} no caso da utilização da Equação (3.13). Porém, tal algoritmo não garante uma solução ótima para as Equações (3.12) e (3.13).

3.5.2 Obtendo e Testando Múltiplos Limiares Utilizando a Arquitetura Convencional

Para obter os limiares é necessário o uso de vários subconjuntos de validação para cada classe do problema. Cada subconjunto é submetido na arquitetura convencional, como mostrado na Figura 3.23, que ilustra a maneira na qual limiares são obtidos para uma classe w_0, onde o conjunto de validação possui somente amostras

da classe w_0 e também a saída da RNA-MLP é correspondente a y_0. Note que no exemplo da Figura 3.23 a estimação de todos os limiares termina quando a taxa de rejeição alcançada é superior a 20%.

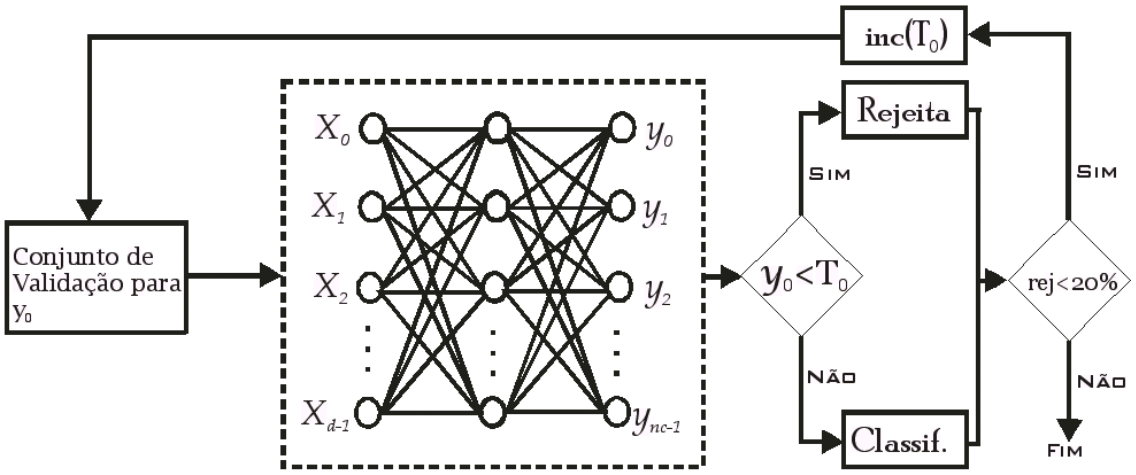

Figura 3.23: Obtendo um limiar T_0 para uma classe w_0 utilizando uma arquitetura convencional

A Figura 3.24 ilustra a aplicação dos limiares obtidos com um conjunto de teste, quando denota-se a máxima saída como y_i com o correspondente limiar T_i onde o padrão x é aceito ou rejeitado, conforme a Equação (3.9).

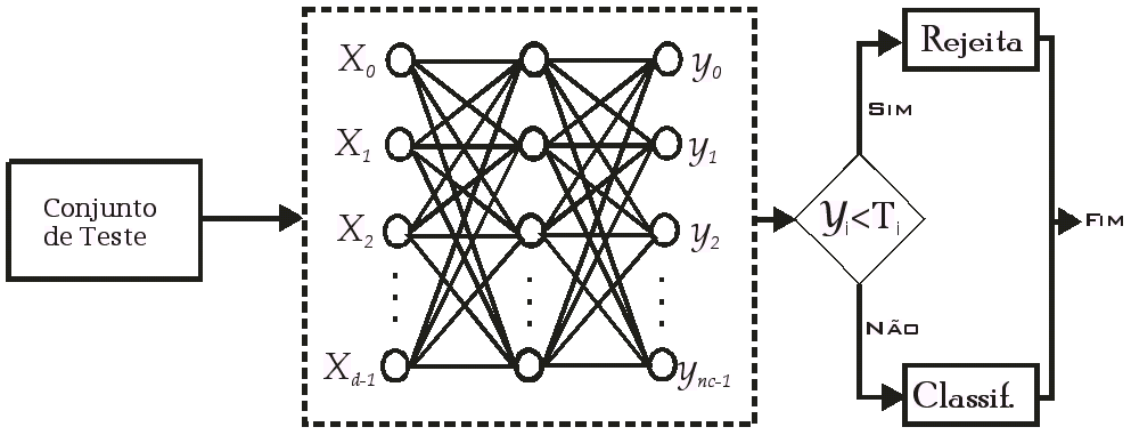

Figura 3.24: Testando um limiar T_0 para uma classe w_0 utilizando uma arquitetura convencional

3.5.3 Obtendo e Testando Múltiplos Limiares Utilizando a Arquitetura Classe-Modular

A Figura 3.25 mostra como obter um limiar T_0 para uma classe w_0 utilizando a arquitetura classe-modular. A maneira que os limiares foram computados aqui está baseado no procedimento anterior utilizado para uma arquitetura convencional.

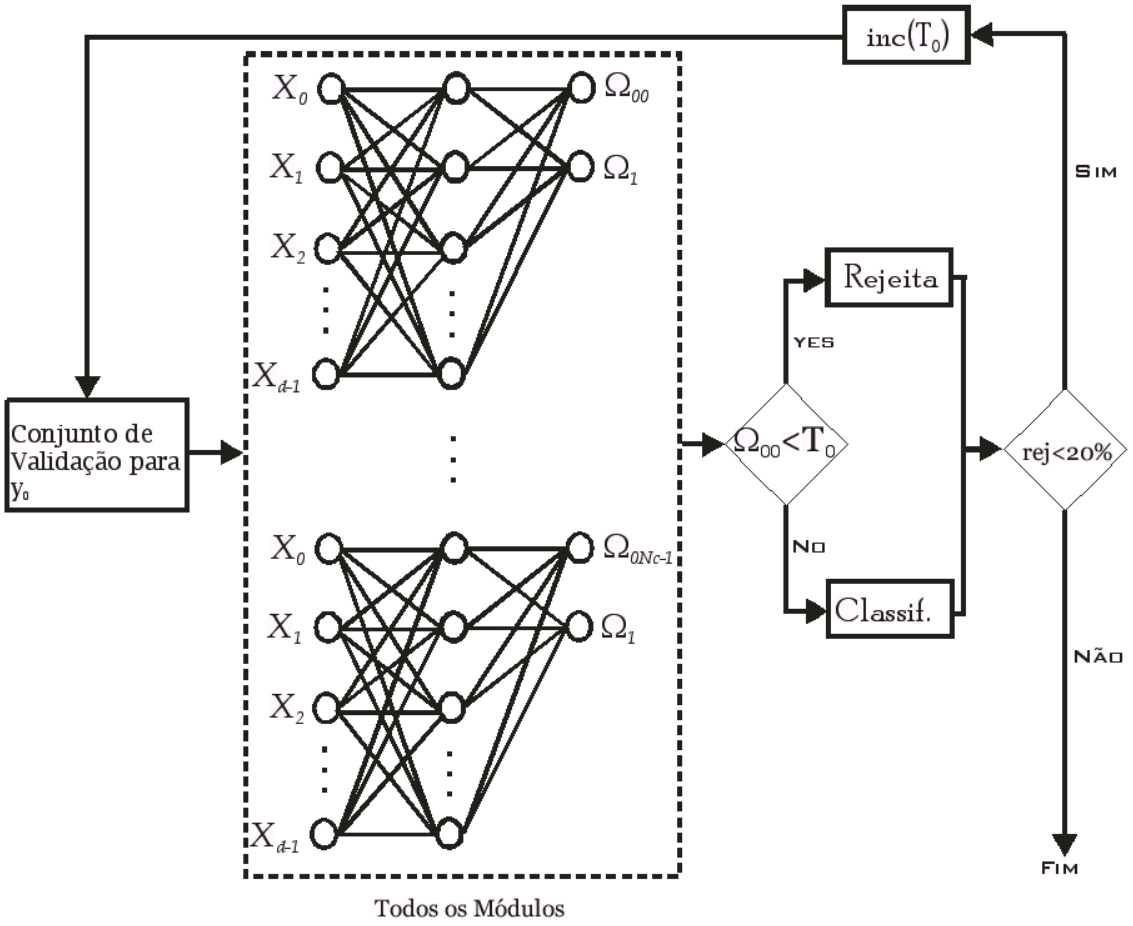

Figura 3.25: Obtendo um limiar T_0 para uma classe w_0 utilizando uma arquitetura classe-modular

A Figura 3.26 mostra a aplicação dos limiares com um conjunto de teste, onde denota-se a saída máxima como y_i com o limiar T_i correspondente e um padrão x é aceito ou rejeitado também de acordo com a Equação (3.9).

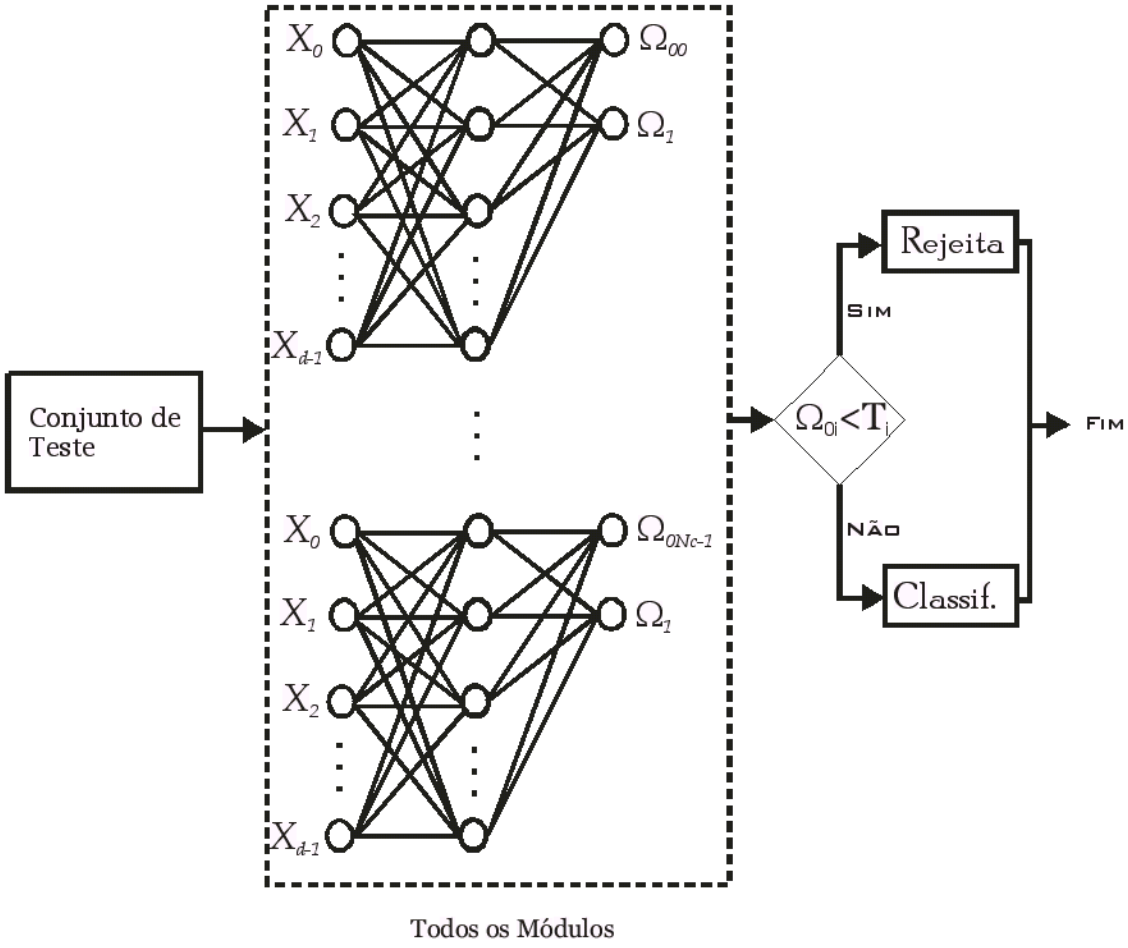

Figura 3.26: Testando um limiar T_0 para uma classe w_0 utilizando uma arquitetura classe-modular

3.6 Seleção de Características - Abordagem *Wrapper/Hill Climbing*

Nesta metodologia, a análise do conjunto de características proposto é realizada a partir de um processo de seleção de características. Métodos automáticos de seleção de características são importantes em muitas situações em que se tem disponível um conjunto grande de características e deseja-se selecionar um subconjunto adequado. Além de ser uma forma de redução de dimensionalidade, uma aplicação importante é a fusão de dados procedentes de múltiplas modalidades de sensores ou de múltiplos modelos de dados. A seleção automática de características é uma técnica de otimização que, dado um conjunto de m características, tenta selecionar

um subconjunto de tamanho w que maximiza uma função critério [JDM00].

É desejável que a função critério seja maior quanto menor for a redundância entre as características e quanto maior a facilidade de discriminar padrões de classes diferentes. Apesar da importância de seleção de atributos, não há regras ou procedimentos definitivos para essa tarefa em cada aplicação particular, principalmente quando o número de características disponíveis for grande. O problema da dimensionalidade, também conhecido como *curse of dimensionality* e como comportamento de curva em U, é um fator muito relevante para decidir-se a dimensionalidade ideal a ser adotada em um problema de reconhecimento de padrões. Trata-se do seguinte fenômeno: o número de elementos de treinamento requeridos para que um classificador tenha um bom desempenho é uma função monotonicamente crescente da dimensão do espaço de características.

Apesar de ser teoricamente clara a relação entre a dimensionalidade e o tamanho do conjunto de treinamento, há outros fatores que, quando considerados, ofuscam a exatidão dessa relação, tais como a complexidade do classificador e o número de classes. Segundo Jain *et al.* em [JDM00], resultados empíricos sugerem que no mínimo deve-se utilizar um número de exemplos de treinamento por classe dez vezes maior que a dimensionalidade.

Em aprendizado supervisionado de máquina, um algoritmo de aprendizagem é colocado de frente com o problema de seleção de alguns subconjuntos de características no qual foca sua atenção, enquanto ignora o resto [KJ97]. Há duas abordagens que podem ser seguidas: a abordagem com filtros e a *wrapper*.

A abordagem com filtros seleciona características utilizando um passo de pré-processamento, ignorando o algoritmo de indução. A sua principal desvantagem é que ele ignora totalmente os efeitos do conjunto de características selecionado no desempenho do algoritmo de indução. Em [KJ97], os autores apresentam alguns algoritmos para aplicação da abordagem com filtros, como os algoritmos *Focus*, *Relief* e filtragem através de árvores de decisão.

Na abordagem *wrapper* [Koh94], o algoritmo de seleção de características funciona como um "empacotador"ao redor do algoritmo de indução. O subconjunto de características conduz uma busca por um subconjunto bom utilizando o próprio algoritmo de indução como parte da função de avaliação de subconjuntos de carac-

terísticas. A idéia da abordagem *wrapper* é relativamente simples: o algoritmo de indução é considerado como uma caixa preta, isto é, não é necessário conhecimento do algoritmo, apenas da sua interface. O algoritmo de indução é rodado sobre um conjunto de dados, usualmente um conjunto de validação, com diferentes conjuntos de características removidos dos dados. O conjunto de características com a mais alta avaliação é escolhido como o conjunto final para rodar o algoritmo de indução. O classificador resultante é então avaliado sobre um conjunto de teste independente, ou seja, que não foi usado durante a busca [KJ97].

Nesta metodologia a abordagem de seleção de características adotada é a *wrapper*. A escolha pela abordagem *wrapper* é devido principalmente ao fato de que, nesta fase, já possui-se as arquiteturas de RNAs já treinadas, ou seja, nossos algoritmos de indução já estão preparados, e ainda há o interesse em investigar os efeitos do conjunto de características proposto sobre os classificadores, e não como um pré-processamento dos dados juntamente com a retirada de algumas características, necessitando de um processo de retreinamento dos classificadores como ocorre na abordagem com filtros. Esquemas da utilização da abordagem *wrapper* no presente trabalho estão ilustrados nas Figuras 3.27 para a arquitetura convencional e 3.28, 3.29 para a arquitetura classe-modular e também pelo Algoritmo 1 para ambas.

A abordagem *wrapper* conduz uma busca no espaço de possíveis parâmetros. Uma busca requer um espaço de estados, como um estado inicial, uma condição de término, e um processo de busca. A organização do espaço de busca utilizado nesta metodologia está representado da seguinte maneira, para cada característica do conjunto há um *bit* que indica a presença (1) ou a remoção (0) de uma referida característica. Operadores determinam a conectividade entre os estados, como operadores de remoção e adição de características de um estado (0 ou 1). O tamanho do espaço de busca para n características é $O(2^n)$, então é impraticável buscar o espaço inteiro exaustivamente, a menos que n seja pequeno. O objetivo da busca é encontrar o conjunto de estados com a mais alta avaliação, usando uma função de heurística para guiá-lo.

No presente trabalho, o processo de busca adotado para a abordagem *wrapper* é o *hill climbing*. Assim como Skalak em [Ska94], utiliza-se um *hill climbing* randômico que continua por um número de ciclos específico. É denominado de randômico devido a mutação ser executada de forma randômica. A idéia básica deste algoritmo é como

descrita em [Ska94] como:

1. Escolhe-se uma cadeia de *bits*. Chama-se esta cadeia de *melhor-avaliada*;
2. Muta-se um *bit* escolhido aleatoriamente na cadeia *melhor-avaliada*;
3. Computa-se o *fitness* da cadeia mutada. Se o *fitness* é maior do que o *fitness* da *melhor-avaliada*, então seta-se a *melhor-avaliada* para a cadeia mutada;
4. Se o número máximo de iterações é atingido, retorna-se a *melhor-avaliada*, caso contrário, retorna-se ao passo 2.

Além da aplicação da abordagem *Wrapper/Hill Climbing*, deve-se destacar um detalhe importante em relação aos estados das características. Os estados de ausência e presença de determinadas características são representados por $0's$ e $1's$. Porém, quando há ausência de determinada característica, isto é, seu estado é 0, ocorre uma substituição desta característica pela média referente a sua posição no conjunto de treinamento do classificador, isto é, do algoritmo de indução, dando origem a um *wrapper* modificado.

Especificando, os passos para realizar a abordagem *Wrapper/Hill Climbing* neste estudo são:

1. Obter o valor médio das características: tais valores são obtidos através da média do conjunto de características de treinamento. Por exemplo, o cálculo para a primeira primitiva:

$$vetor\_caracteristicas\_medio[0] = \frac{\sum_{i=0}^{N-1} vetores\_caracteristicas[i][0]}{N} \quad (3.14)$$

   Sendo N o número total de amostras do conjunto de treinamento. Aplica-se então a Equação (3.14) para todas as características.

2. O próximo passo é utilizar o conjunto de validação para a aplicação do *Wrapper/Hill Climbing*, com 1000 iterações cada vez, e 10 vezes no total. O Algoritmo 1 ilustra esses passos.

```
Dados     : Conjunto de validação e treinamento
Resultado: Conjuntos de quantidades de características com suas respecti-
           vas taxas de reconhecimento
inicialização;
vetor_médio=obter_vetor_médio(Conjunto_Treinamento);
para (i = 0; i < 10; i++) faça
    vetor_estados=[1 1 1 1 1...1];
    para (j = 0; j < 1000; j++) faça
        se (j > 0) então
            mutação(vetor_estados) //muda aleatoriamente;
        fim
        recog=executa_rna(Conjunto_Validação,vetor_estados, vetor_médio)
        //executa a rna com o número de entradas de acordo com a
        //quantidade de 1's do vetor_estados;
        imprime("Iteração i=%d Iteração j=%d Qtde features:%d
        Recog:%f", i,j, qtde_features(vetor_medio),recog);
    fim para
fim para
```

Algoritmo 1: Algoritmo *Wrapper/Hill Climbing* aplicado

3. Montar um gráfico *Reconhecimento x Quantidade de características*. Detalhe: mesmas quantidades de características podem gerar taxas de reconhecimento diferentes e tais quantidades de características ocorrem de maneira aleatória e em quantidades diferentes.

4. Após análise dos gráficos, selecionar os vetores de estado e médio obtidos ao final do processo, correspondente à determinada quantidade de características/reconhecimento e aplicá-lo sobre o conjunto de teste.

Os passos 2, 3 e 4 são feitos para ambas arquiteturas convencional e classe-modular, sendo que para class-modular, são duas fases: na primeira, tudo ocorre como na convencional e na segunda (mais complexa) todos os passos são realizados para cada subrede. As Figuras 3.27, 3.28 e 3.29 ilustram os respectivos esquemas.

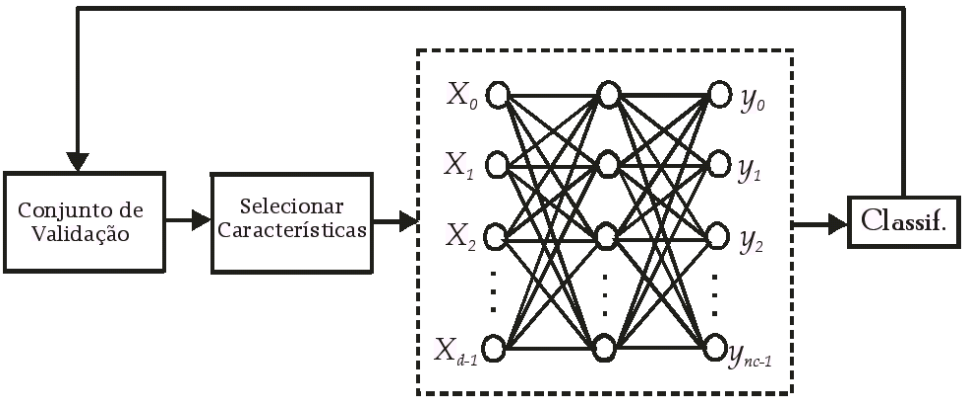

Figura 3.27: Esquema para a aplicação do método *Wrapper/Hill Climbing* na arquitetura convencional para o conjunto de características proposto

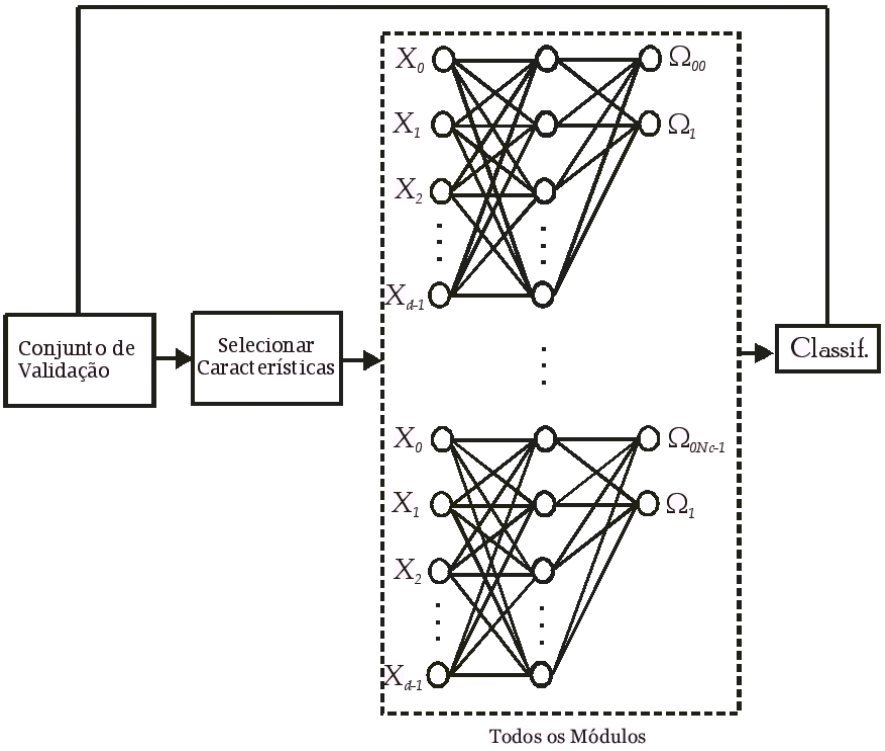

Figura 3.28: Esquema para a aplicação do método *Wrapper/Hill Climbing* na arquitetura classe-modular para o conjunto de características proposto

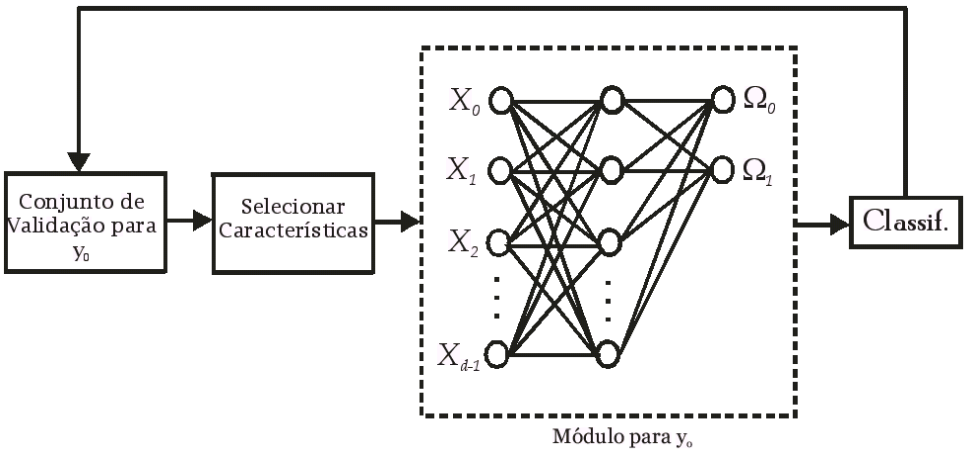

Figura 3.29: Esquema para a aplicação sepadaradamente do método *Wrapper/Hill Climbing* nos módulos da arquitetura classe-modular para o conjunto de características proposto

3.7 Comentários Finais

Neste Capítulo foi apresentada a metodologia proposta ao problema de reconhecimento de palavras manuscritas no contexto de cheques bancários brasileiros, detalhando os pontos essenciais investigados da abordagem, como: o conjunto de características proposto; as arquiteturas de RNAs MLP utilizadas; o mecanismo de rejeição utilizado; e uma análise do conjunto de características proposto através de uma abordagem *Wrapper/Hill climbing*. No próximo Capítulo são mostrados os experimentos realizados para validar a metodologia proposta, os resultados obtidos e a análise dos mesmos.

Capítulo 4

Experimentos

Neste capítulo, são apresentados os experimentos realizados e os resultados obtidos pela metodologia proposta com o objetivo de investigar a sua eficiência no contexto proposto. Como descrito na Seção 1.1, o problema abordado no presente trabalho é o reconhecimento de dois conjuntos de palavras manuscritas no contexto de cheques bancários. O primeiro conjunto é formado pelos nomes dos meses do ano, onde a base de imagens de palavras manuscritas utilizada é a da Universidade Federal de Campina Grande (UFCG). O segundo é formado pelas palavras referentes aos valores por extenso. Neste, a base de imagens utilizada é a da Pontifícia Universidade Católica do Paraná (PUC-PR). Para ambos os conjuntos, são descritos os experimentos e os resultados em partes, na mesma ordem da metodologia proposta no Capítulo 3, juntamente com a separação das bases de imagens em conjuntos de treinamento, validação e teste, encerrando com uma análise de resultados para cada base.

4.1 Experimento 1 - Base de Dados UFCG

Neste experimento, toda a metodologia proposta no Capítulo 3 é aplicada para a base de imagens da UFCG, que é formada por palavras manuscritas referentes aos meses do ano. A Figura 4.1 exemplifica algumas amostras coletadas da base. A quantidade total de amostras é de 6000, sendo 500 para cada mês. Segundo [dOJ02], a base foi coletada a partir de 500 escritores diferentes, na maioria estudantes do ensino médio e superior de instituições públicas e privadas.

A divisão do total de amostras em conjuntos de treinamento, validação e teste

é realizado da seguinte forma: 60% do total de amostras para o conjunto de treinamento, 20% para o conjunto de validação e os 20% restantes para o conjunto de teste, conforme apresentado na Tabela 4.1. Na Tabela 4.2 são descritas informações sobre a distribuição das amostras nesses conjuntos de acordo com os estilos de escritas encontrados.

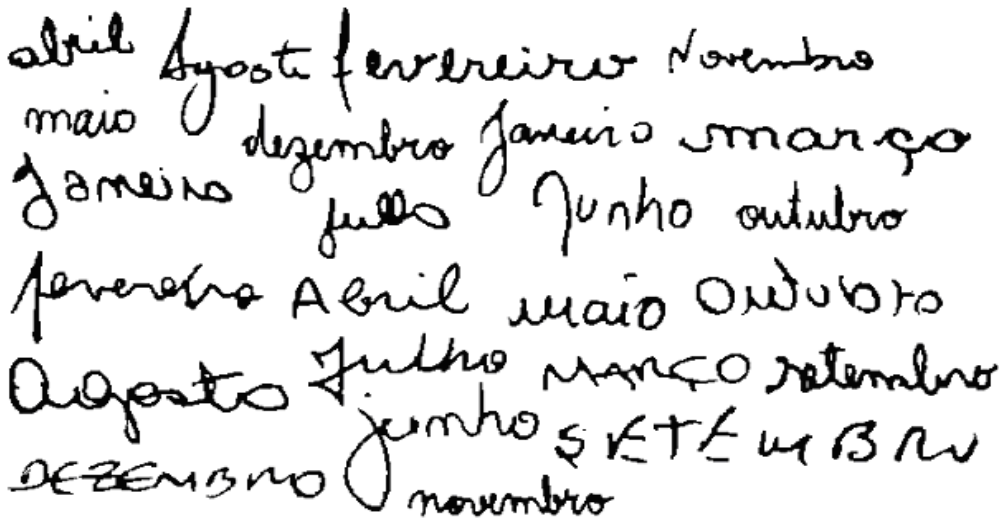

Figura 4.1: Amostras da base de imagens UFCG

Tabela 4.1: Distribuição das amostras da base UFCG nos conjuntos de treinamento, validação e teste

Palavras	**Treinamento**	**Validação**	**Teste**	**Total**
Janeiro	300	100	100	500
Fevereiro	300	100	100	500
Março	300	100	100	500
Abril	300	100	100	500
Maio	300	100	100	500
Junho	300	100	100	500
Julho	300	100	100	500
Agosto	300	100	100	500
Setembro	300	100	100	500
Outubro	300	100	100	500
Novembro	300	100	100	500
Dezembro	300	100	100	500
Total	3600	1200	1200	6000

Tabela 4.2: Informações sobre a distribuição dos estilos de escritas nos conjuntos

Estilo de Escrita	**Treinamento**	**Validação**	**Teste**
Cursiva pura	70,97%	71,25%	72,92%
Caixa alta	5,83%	5,75%	4,83%
Caracteres disjuntos	12,06%	11,00%	9,08%
Mista	11,14%	12,00%	13,17%

4.1.1 Aplicação de um classificador K-NN (*K-Nearest Neighbour*)

Com o intuito de apenas analisar a distribuição das amostras perante um classificador linear simples, submeteu-se cada um dos conjuntos separadamente (treinamento, validação e teste) a um algoritmo K-NN (*K-Nearest Neighbour*), neste caso, com $K = 1$. Os resultados são mostrados na Tabela 4.3. Os passos realizados são:

1. Para cada amostra e calcula-se a distância a cada uma das amostras restantes, menos de si própria, o que equivaleria a 0.

2. Seleciona-se o vizinho mais próximo de cada elemento e atribui-se a este mesma classificação que esse vizinho.

3. Determina-se a porcentagem de erro associado a este algoritmo no problema, neste caso, representada pela Equação 4.1.

$$Erro(\%) = (1 - (Qtde_{Acertos}/Qtde_{Exemplos})) * 100 \tag{4.1}$$

Tabela 4.3: Taxas de reconhecimento obtidas para cada conjunto utilizando um classificador K-NN

Aplicação de um K-NN	
Conjunto	**Reconhecimento(%)**
Treinamento	72,97
Validação	58,92
Teste	61,83

4.1.2 Resultados para a Arquitetura Convencional

Assim como descrito na Seção 3.4.1, a arquitetura RNA-MLP convencional é uma rede neural artificial MLP, onde sua entrada é composta por 24 neurônios, um para cada posição dos vetores de características, uma camada escondida de tamanho variável, e 12 neurônios de saída, em que cada saída equivale a uma palavra do léxico. O critério de parada do treinamento para ambas arquiteturas convencional e classe-modular é realizado através de um procedimento de validação cruzada.

Os melhores resultados foram obtidos com a utilização do algoritmo de aprendizagem *backpropagation* padrão com o valor do parâmetro de aprendizagem estimado em 0,2, uma camada escondida de 45 neurônios e 900 épocas. Os pesos são iniciados aleatoriamente com valores entre -0,1 e 0,1 e suas atualizações realizadas de forma topológica da entrada para a saída. Quando uma primitiva não é encontrada assume-se o valor 0,001. Tal valor é assumido após uma análise do conjunto de dados resultante de um processo de normalização, em que o menor valor encontrado é superior a este em pelo menos uma casa decimal. A Tabela 4.4 mostra a matriz de confusão obtida após a submissão do conjunto de teste à arquitetura, mostrando as taxas de reconhecimento e erros obtidas.

Tabela 4.4: Matriz de confusão para o conjunto de teste - Arquitetura Convencional

Des/Obt	**J**	**F**	**M**	**A**	**M**	**J**	**J**	**A**	**S**	**O**	**N**	**D**	**Rec.(%)**
Janeiro	**77**	5	4	0	1	1	1	2	1	2	5	1	77
Fevereiro	6	**87**	0	0	0	0	0	0	2	4	0	1	87
Março	5	1	**74**	2	12	2	2	1	0	1	0	0	74
Abril	0	1	5	**84**	4	3	2	1	0	0	0	0	84
Maio	1	2	9	5	**77**	0	2	1	0	2	1	0	77
Junho	2	2	1	0	4	**77**	10	0	1	3	0	0	77
Julho	1	1	2	3	4	10	**73**	3	0	3	0	0	73
Agosto	6	2	5	4	0	1	2	**73**	0	0	2	5	73
Setembro	2	8	0	1	0	1	0	0	**71**	7	6	4	71
Outubro	3	3	3	0	0	2	1	0	9	**76**	3	0	76
Novembro	0	2	6	0	0	1	0	0	3	1	**82**	5	82
Dezembro	4	6	0	0	1	1	1	2	4	1	6	**74**	74
											Taxa de Reconhecimento		**77,08%**

4.1.3 Resultados para a Arquitetura Classe-Modular

O processo de obtenção de resultados para a arquitetura classe-modular segue como descrito na Seção 3.4.2, onde utiliza-se K RNA-MLP, uma para cada palavra do léxico do problema. Cada um dos K 2-classificadores é treinado independentemente das outras classes, utilizando um conjunto de treinamento e validação. O algoritmo *backpropagation* padrão é usado em cada um dos 2-classificadores da mesma maneira como ocorre na arquitetura convencional.

Cada um dos K 2-classificadores é composto por 24 neurônios de entrada, 45 na camada escondida e 2 neurônios de saída. Os melhores resultados foram obtidos com a utilização do algoritmo de aprendizagem *backpropagation* padrão com o valor do parâmetro de aprendizagem estimado entre 0,5 e 0,8, e número de épocas variando entre 400 a 500 épocas. Os pesos são iniciados aleatoriamente com valores entre -0,1 e 0,1 e suas atualizações realizadas de forma topológica da entrada para a saída. A Tabela 4.5 mostra a matriz de confusão obtida após a submissão do conjunto de teste à arquitetura apresentando as taxas de reconhecimento e erros obtidas.

Tabela 4.5: Matriz de confusão para o conjunto de teste - Arquitetura Classe-Modular

Des/Obt	**J**	**F**	**M**	**A**	**M**	**J**	**J**	**A**	**S**	**O**	**N**	**D**	**Rec. (%)**
Janeiro	**83**	8	2	0	0	2	0	0	0	2	1	2	83
Fevereiro	5	**83**	1	1	1	0	0	1	3	1	2	2	83
Março	3	3	**75**	5	10	0	0	0	0	2	2	0	75
Abril	1	1	1	**93**	2	0	2	0	0	0	0	0	93
Maio	1	0	10	5	**80**	2	1	0	0	0	1	0	80
Junho	1	3	0	0	5	**84**	4	0	1	2	0	0	84
Julho	1	0	0	6	4	9	**76**	0	0	3	1	0	76
Agosto	2	4	2	3	0	3	0	**78**	0	3	3	2	78
Setembro	1	10	0	0	0	0	0	0	**73**	6	8	2	73
Outubro	4	4	2	0	0	0	0	0	4	**85**	1	0	85
Novembro	3	2	0	0	0	0	0	0	4	1	**89**	1	89
Dezembro	3	5	0	0	0	2	1	1	0	0	6	**82**	82
											Taxa de Reconhecimento		**81,75%**

4.1.4 Mecanismo de Rejeição Utilizando a Regra de Chow

Nesta Subseção realiza-se experimentos para um dos mecanismos de rejeição tratados no presente trabalho, seguindo parte da metodologia proposta na Seção 3.5. Tal mecanismo é proporcionado pela regra de Chow [Cho70], em que apenas um limiar é utilizado para rejeição em todas as classes do problema. A obtenção do limiar utilizando esta regra e o conjunto de validação é apresentada na Tabela 4.6 para a arquitetura convencional e na Tabela 4.7 para a arquitetura classe-modular. Para ambas as arquiteturas as taxas de erro são fixadas em 1%, 2% e 5%.

Tabela 4.6: Limiares obtidos pela regra de Chow na arquitetura convencional

Limiares	**Rec.%**	**Rej.%**	**Erro%**	**Conf.%**	**Erro Conf.%**
0,99996	21,34	77,66	1,00	95,52	4,48
0,99976	26,75	71,25	2,00	93,04	6,96
0,993859	40,25	54,75	5,00	88,95	11,05

Tabela 4.7: Limiares obtidos pela regra de Chow na arquitetura classe-modular

Limiares	**Rec.%**	**Rej.%**	**Erro%**	**Conf.%**	**Erro Conf.%**
0,972505	23,00	76,00	1,00	95,83	4,17
0,948801	29,75	68,25	2,00	93,70	6,30
0,841883	46,08	48,92	5,00	90,21	9,79

4.1.5 Mecanismo de Rejeição com Múltiplos Limiares

Nesta Subseção relata-se experimentos referentes ao mecanismo de rejeição proporcionado pela regra dos CRTs de Fumera *et al.* [FRG00] descrita na Subseção 3.5.1, em que utiliza-se múltiplos limiares, ou seja, um para cada classe do problema. A obtenção dos limiares utilizando esta regra e o conjunto de validação são apresentadas nas Tabelas 4.8, 4.9 e 4.10 para a arquitetura convencional e nas Tabelas 4.11, 4.12 e 4.13 para a classe-modular. As taxas de erro também são fixadas em 1%, 2% e 5% para ambas as arquiteturas. A Figura 4.2 apresenta um gráfico comparativo entre as arquiteturas na obtenção dos múltiplos limiares até uma taxa de rejeição de

20%. Após a estimação dos múltiplos limiares utilizando o conjunto de validação, os mesmos são aplicados sobre o conjunto de teste.

Tabela 4.8: Múltiplos limiares obtidos e suas performance para a arquitetura convencional no conjunto de validação com taxa de erro fixada em 1%

Mês	**Limiares**	**Rec.%**	**Rej.%**	**Erro 1%**	**Conf.%**	**Erro Conf.%**
Janeiro	0,298393	73	26	1	98,65	1,35
Fevereiro	0,997559	23	76	1	95,83	4,17
Março	0,725215	69	30	1	98,57	1,43
Abril	0,991709	56	43	1	98,25	1,75
Maio	0,804127	59	40	1	98,33	1,67
Junho	0,358403	66	33	1	98,51	1,49
Julho	0,69891	73	26	1	98,65	1,35
Agosto	0,947301	61	38	1	98,39	1,61
Setembro	0,957002	48	51	1	97,96	2,04
Outubro	0,993358	52	47	1	98,11	1,89
Novembro	0,678156	75	24	1	98,68	1,32
Dezembro	0,861186	50	49	1	98,04	1,96
Média		58,75	40,25	1	98,16	1,84

Tabela 4.9: Múltiplos limiares obtidos e suas performance para a arquitetura convencional no conjunto de validação com taxa de erro fixada em 2%

Mês	**Limiares**	**Rec.%**	**Rej.%**	**Erro 2%**	**Conf.%**	**Erro Conf.%**
Janeiro	0,060451	73	25	2	97,35	2,65
Fevereiro	0,994809	25	73	2	92,59	7,41
Março	0,571389	72	26	2	97,30	2,70
Abril	0,45822	81	17	2	97,59	2,41
Maio	0,484624	65	33	2	97,04	2,96
Junho	0,074551	67	31	2	97,10	2,90
Julho	0,200942	76	22	2	97,44	2,56
Agosto	0,934349	63	35	2	96,92	3,08
Setembro	0,564437	71	27	2	97,26	2,74
Outubro	0,980306	60	38	2	96,77	3,23
Novembro	0,085401	79	19	2	97,53	2,47
Dezembro	0,58164	61	37	2	96,83	3,17
Média		66,08	31,92	2	96,81	3,19

Tabela 4.10: Múltiplos limiares obtidos e suas performance para a arquitetura convencional no conjunto de validação com taxa de erro fixada em 5%

Mês	**Limiares**	**Rec.%**	**Rej.%**	**Erro 5%**	**Conf.%**	**Erro Conf.%**
Janeiro	0,0054	74	21	5	93,67	6,33
Fevereiro	0,076201	65	30	5	92,86	7,14
Março	0,080201	75	20	5	93,75	6,25
Abril	0,385358	81	14	5	94,19	5,81
Maio	0,078901	66	29	5	92,98	7,02
Junho	0,0275	68	27	5	93,15	6,85
Julho	0,058001	79	16	5	94,05	5,95
Agosto	0,004	74	21	5	93,67	6,33
Setembro	0,240237	72	23	5	93,51	6,49
Outubro	0,799977	69	26	5	93,24	6,76
Novembro	0,00805	79	16	5	94,05	5,95
Dezembro	0,052001	67	28	5	93,06	6,94
Média		72,42	22,58	5	93,51	6,49

Tabela 4.11: Múltiplos limiares obtidos e suas performance para a arquitetura classe-modular no conjunto de validação com taxa de erro fixada em 1%

Mês	**Limiares**	**Rec.%**	**Rej.%**	**Erro 1%**	**Conf.%**	**Erro Conf.%**
Janeiro	0,671855	66	33	1	98,51	1,49
Fevereiro	0,712062	32	67	1	97,06	2,94
Março	0,576289	56	43	1	98,25	1,75
Abril	0,307345	84	15	1	98,82	1,18
Maio	0,779573	44	55	1	97,78	2,22
Junho	0,732465	47	52	1	97,92	2,08
Julho	0,21579	81	18	1	98,78	1,22
Agosto	0,644851	63	36	1	98,44	1,56
Setembro	0,732115	56	43	1	98,25	1,75
Outubro	0,810228	59	40	1	98,33	1,67
Novembro	0,343251	81	18	1	98,78	1,22
Dezembro	0,511579	67	32	1	98,53	1,47
Média		61,33	37,67	1	98,29	1,71

Tabela 4.12: Múltiplos limiares obtidos e suas performance para a arquitetura classe-modular no conjunto de validação com taxa de erro fixada em 2%

Mês	**Limiares**	**Rec.%**	**Rej.%**	**Erro 2%**	**Conf.%**	**Erro Conf.%**
Janeiro	0,558436	66	32	2	97,06	2,94
Fevereiro	0,712012	32	66	2	94,12	5,88
Março	0,385858	69	29	2	97,18	2,82
Abril	0,170646	87	11	2	97,75	2,25
Maio	0,599343	55	43	2	96,49	3,51
Junho	0,538633	55	43	2	96,49	3,51
Julho	0,143499	81	17	2	97,59	2,41
Agosto	0,122852	74	24	2	97,37	2,63
Setembro	0,484824	68	30	2	97,14	2,86
Outubro	0,286591	79	19	2	97,53	2,47
Novembro	0,252236	87	11	2	97,75	2,25
Dezembro	0,464771	67	31	2	97,10	2,90
Média		68,33	29,67	2	96,97	3,03

Tabela 4.13: Múltiplos limiares obtidos e suas performance para a arquitetura classe-modular no conjunto de validação com taxa de erro fixada em 5%

Mês	**Limiares**	**Rec.%**	**Rej.%**	**Erro 5%**	**Conf.%**	**Erro Conf.%**
Janeiro	0,28024	75	20	5	93,75	6,25
Fevereiro	0,265838	67	28	5	93,06	6,94
Março	0,295893	71	24	5	93,42	6,58
Abril	0,0096	88	7	5	94,62	5,38
Maio	0,3369	67	28	5	93,06	6,94
Junho	0,208941	64	31	5	92,75	7,25
Julho	0,051701	83	12	5	94,32	5,68
Agosto	0,00815	76	19	5	93,83	6,17
Setembro	0,192193	75	20	5	93,75	6,25
Outubro	0,093151	80	15	5	94,12	5,88
Novembro	0,001	88	7	5	94,62	5,38
Dezembro	0,192693	72	23	5	93,51	6,49
Média		75,50	19,50	5	93,73	6,27

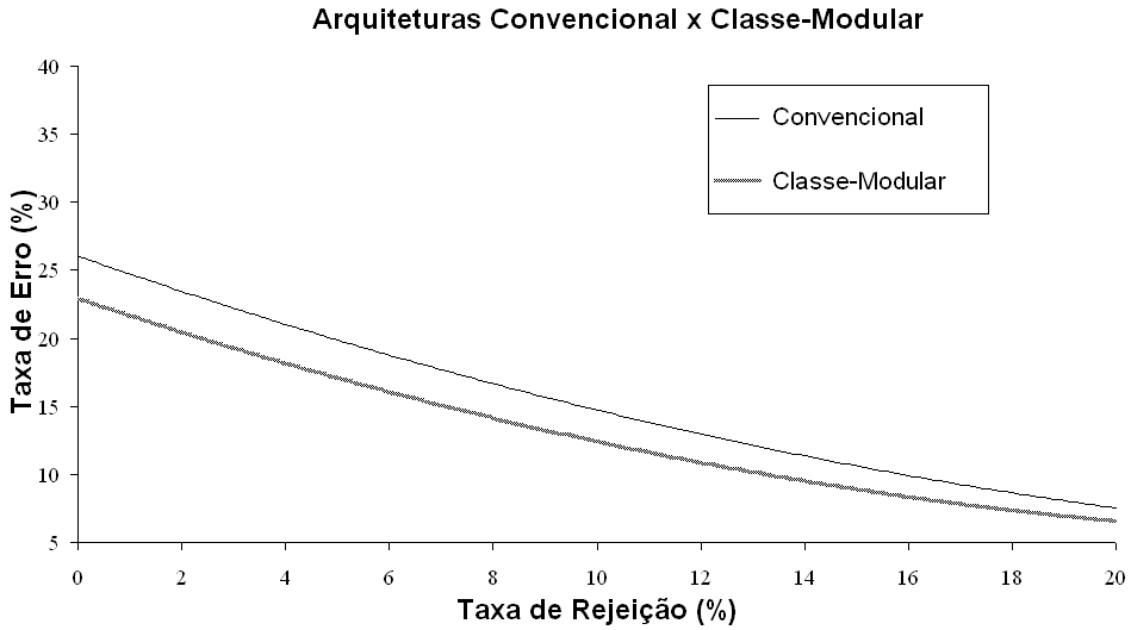

Figura 4.2: Obtenção dos limiares na base UFCG(meses do ano)

Testando os Múltiplos Limiares em Ambas Arquiteturas

Os resultados da obtenção dos limiares no conjunto de validação em ambos mecanismos de rejeição e arquiteturas estão resumidos na Tabela 4.14 e na Tabela 4.15. Devido ao melhor ajuste de limiares proporcionado pela regra CRT de Fumera *et al.* [FRG00]. Analisando as Tabelas 4.14 e 4.15, observam-se menores taxas de rejeição e maiores taxas de reconhecimento utilizando os múltiplos limiares. Assim, os resultados finais obtidos no conjunto de teste utilizando os múltiplos limiares estimados através do conjunto de validação são apresentados na Tabela 4.16.

Tabela 4.14: Resumo dos resultados obtidos com a regra de Chow [Cho70] para as arquiteturas convencional e classe-modular

Taxas de	**Convencional**			**Classe-Modular**		
Erro	**Rec.(%)**	**Rej.(%)**	**Conf.(%)**	**Rec.(%)**	**Rej.(%)**	**Conf.(%)**
1%	21,34	77,66	95,52	23,00	76,00	95,83
2%	26,75	71,25	93,04	29,75	68,25	93,70
5%	40,25	54,75	88,95	46,08	48,92	90,21

Tabela 4.15: Resumo dos resultados obtidos com a regra CRT de Fumera et al.[FRG00] para as arquiteturas convencional e classe-modular

Taxas de	**Convencional**			**Classe-Modular**		
Erro	**Rec.(%)**	**Rej.(%)**	**Conf.(%)**	**Rec.(%)**	**Rej.(%)**	**Conf.(%)**
1%	58,75	40,25	98,16	61,03	37,67	98,29
2%	66,08	31,92	96,81	68,33	29,67	96,97
5%	72,42	22,58	93,51	75,50	19,50	93,73

Tabela 4.16: Resultados obtidos após a aplicação dos múltiplos limiares no conjunto de teste em ambas arquiteturas convencional e classe-modular

Taxas	**Convencional**			**Classe-Modular**		
	1%	**2%**	**5%**	**1%**	**2%**	**5%**
Reconhecimento	57,17	67,00	75,42	68,33	74,17	79,75
Rejeição	34,00	20,42	7,33	25,33	17,08	6,92
Erro	8,83	12,58	17,25	6,33	8,75	13,33
Confiabilidade	86,62	84,19	81,38	91,52	89,45	85,68

4.1.6 Seleção de Características - Abordagem *Wrapper/Hill Climbing*

Nesta Subseção são apresentados os resultados experimentais obtidos pela seleção de características descrita na metodologia proposta, Seção 3.6. Primeiramente, são relacionados os resultados para a arquitetura convencional e classe-modular. A seleção de características para cada módulo separadamente também é realizada.

Resultados Obtidos com as Arquiteturas Convencional e Classe-Modular

Para a arquitetura convencional no conjunto de validação, os resultados foram:

- Qtde Características: 24 = Taxa Rec.:73,92;
- Qtde Características: 22 = Taxa Rec.: 74,17;
- SubConjunto:NLE, NSCVE, NSCVD, NSCXE, NSCXD, NCPE, NCPD, NBPE, NBPD, NEPE, NEPD, NAE, NAD, NDE, NDD, NCH, NPP, NTV, NTH,

NLAE, NLAD, NLDE.

Para a arquitetura classe-modular no conjunto de validação, os resultados foram:

- Qtde Características: 24 = Taxa Rec.:77,08;
- SubConjunto: Não houve modificações na performance, conjunto permanece o inicial.

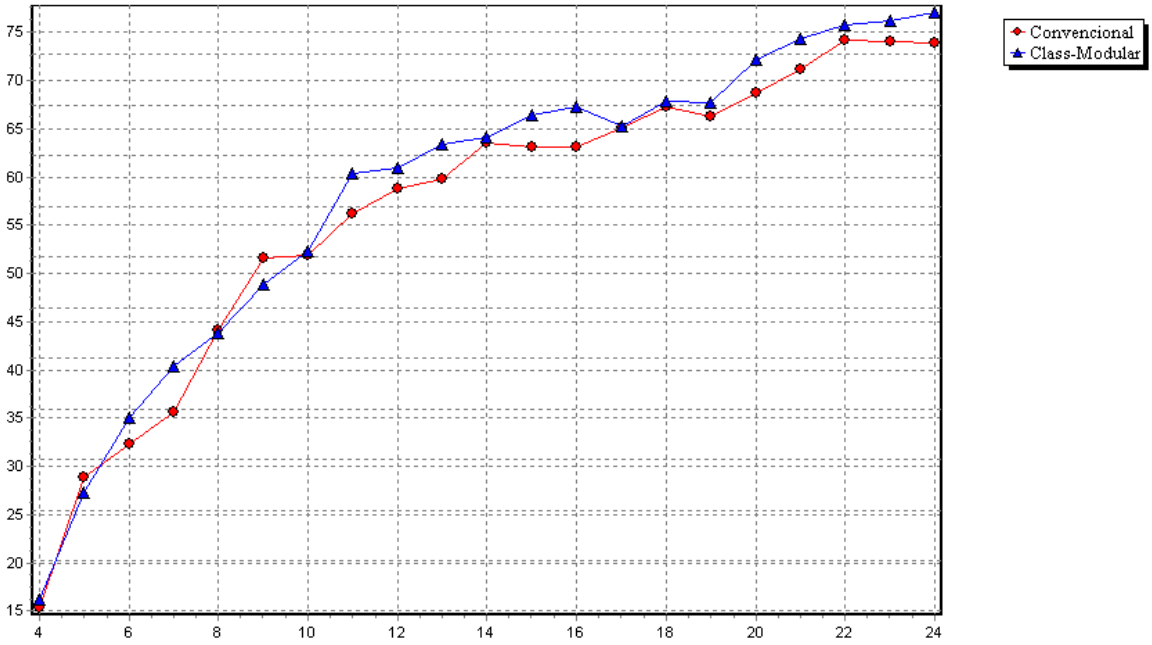

Figura 4.3: Resultados (Reconhecimento x Quantidade de características) da abordagem *Wrapper/Hill Climbing* para as arquiteturas convencional e classe-modular na base UFCG

Resultados Obtidos para cada Módulo Separadamente da Arquitetura Classe-Modular

Nesta etapa, a abordagem *wrapper/hill climbing* é aplicada para cada módulo da arquitetura classe-modular. Os gráficos para cada módulo são apresentados nas Figuras 4.4 a 4.16 juntamente com a descrição dos novos conjuntos de características e taxas de reconhecimento obtidas.

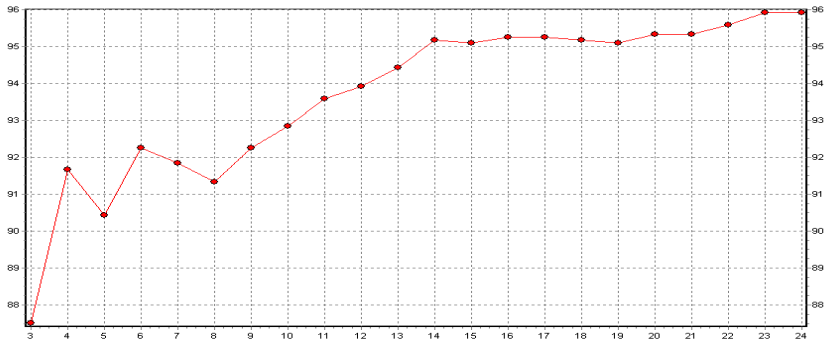

Figura 4.4: Gráfico para o Módulo Janeiro

Módulo Janeiro

- Qtde Características: 24 = Taxa Rec.: 95,92;
- Qtde Características: 23 = Taxa Rec.: 95,92;
- SubConjunto: NLE, NLD, NSCVE, NSCVD, NSCXE, NSCXD, NCPE, NCPD, NBPE, NBPD, NEPE, NEPD, NAE, NAD, NDE, NDD, NCH, NTV, NTH, NLAE, NLAD, NLDE, NLDD.

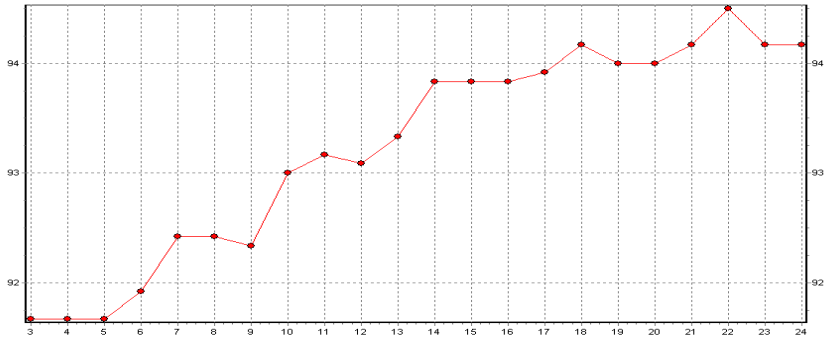

Figura 4.5: Gráfico para o Módulo Fevereiro

Módulo Fevereiro

- Qtde Características: 24 = Taxa Rec.: 94,17;
- Qtde Características: 22 = Taxa Rec.: 94,5;
- SubConjunto: NLD, NSCVE, NSCVD, NSCXE, NSCXD, NCPE, NCPD, NBPE, NBPD, NEPE, NEPD, NAE, NAD, NDE, NDD, NCH, NPP, NTV, NTH, NLAD, NLDE, NLDD.

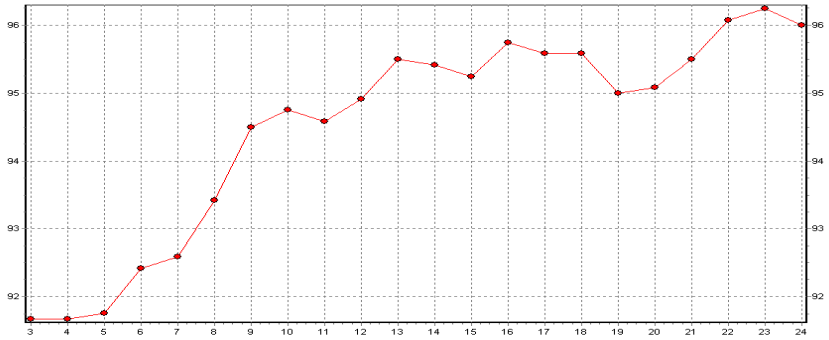

Figura 4.6: Gráfico para o Módulo Março

Módulo Março

- Qtde Características: 24 = Taxa Rec.: 96;
- Qtde Características: 23 = Taxa Rec.: 96,25;
- SubConjunto: NLE, NSCVE, NSCVD, NSCXE, NSCXD, NCPE, NCPD, NBPE, NBPD, NEPE, NEPDNAE, NAD, NDE, NDD, NCH, NPP, NTV, NTH, NLAE, NLAD, NLDE, NLDD.

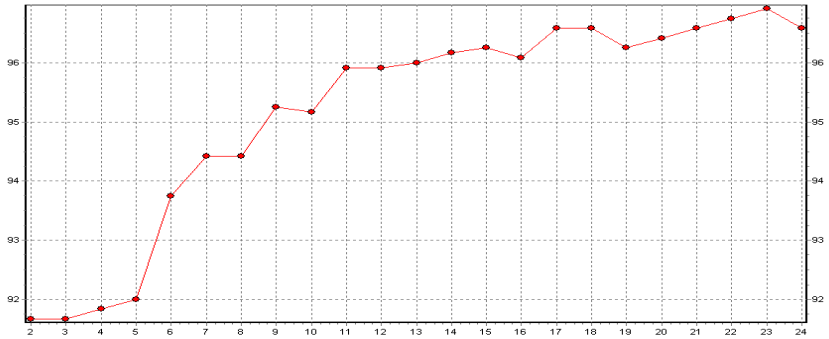

Figura 4.7: Gráfico para o Módulo Abril

Módulo Abril

- Qtde Características: 24 = Taxa Rec.: 96,58;
- Qtde Características: 23 = Taxa Rec.: 96,92;
- SubConjunto: NLE, NLD, NSCVD, NSCXE, NSCXD, NCPE, NCPD, NBPE, NBPD, NEPE, NEPD, NAE, NAD, NDE, NDD, NCH, NPP, NTV, NTH, NLAE, NLAD, NLDE, NLDD.

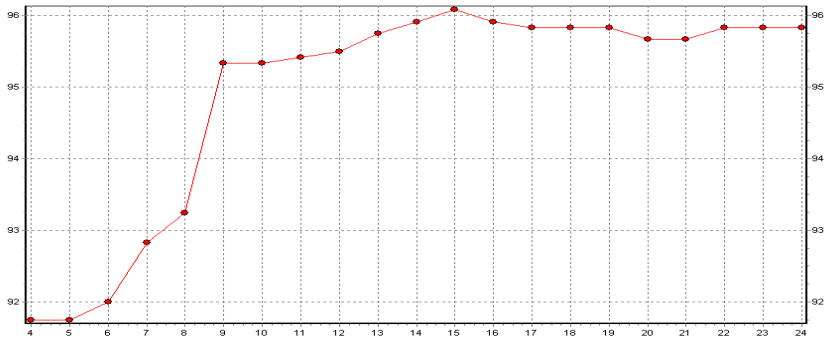

Figura 4.8: Gráfico para o Módulo Maio

Módulo Maio

- Qtde Características: 24 = Taxa Rec.: 95,83;
- Qtde Características: 15 = Taxa Rec.: 96,08;
- SubConjunto: NLE, NLD, NSCVE, NSCXE, NSCXD, NCPD, NEPE, NEPD, NDE, NCH, NPP, NTVNTH, NLAD, NLDE.

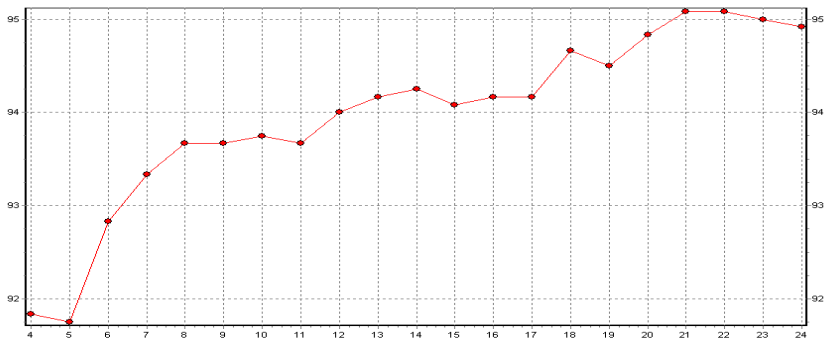

Figura 4.9: Gráfico para o Módulo Junho

Módulo Junho

- Qtde Características: 24 = Taxa Rec.: 94,92;
- Qtde Características: 21 = Taxa Rec.: 95,08;
- SubConjunto: NLE, NLD, NSCVE, NSCVD, NSCXE, NSCXD, NCPD, NBPE, NBPD, NEPE, NEPD, NAE, NAD, NDD, NCH, NPP, NTV, NTH, NLAD, NLDE, NLDD.

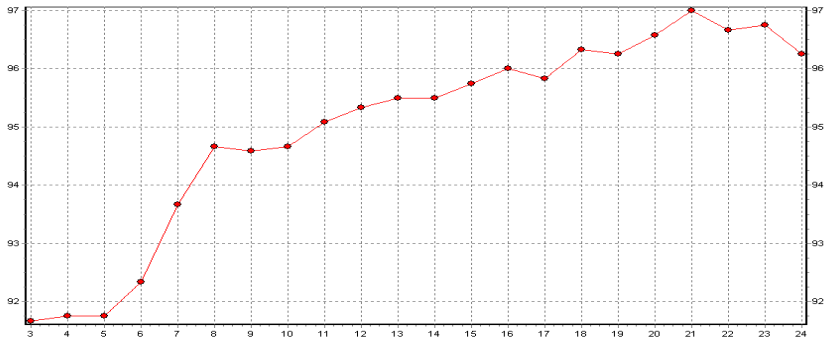

Figura 4.10: Gráfico para o Módulo Julho

Módulo Julho

- Qtde Características: 24 = Taxa Rec.: 96,25;
- Qtde Características: 21 = Taxa Rec.: 97;
- SubConjunto: NLD, NSCVE, NSCVD, NSCXE, NSCXD, NCPD, NBPE, NBPD, NEPE, NEPD, NAENAD, NDE, NDD, NCH, NPP, NTH, NLAE, NLAD, NLDE, NLDD.

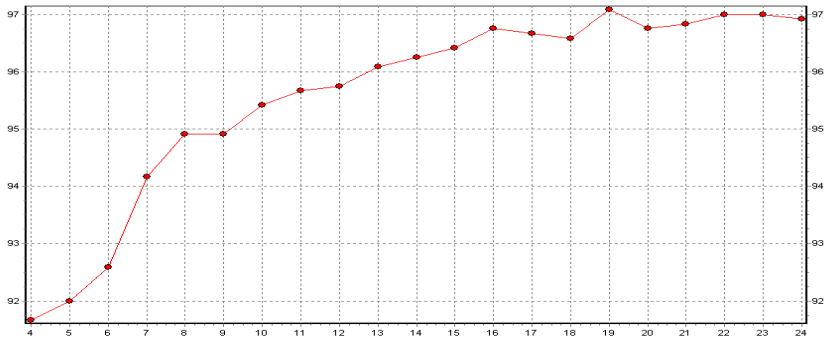

Figura 4.11: Gráfico para o Módulo Agosto

Módulo Agosto

- Qtde Características: 24 = Taxa Rec.: 96,92;
- Qtde Características: 19 = Taxa Rec.: 97,08;
- SubConjunto: NLE, NLD, NSCVE, NSCVD, NSCXE, NSCXD, NCPE, NBPE, NBPD, NEPE, NEPD, NAE, NAD, NDD, NPP, NTV, NLAE, NLAD, NLDD.

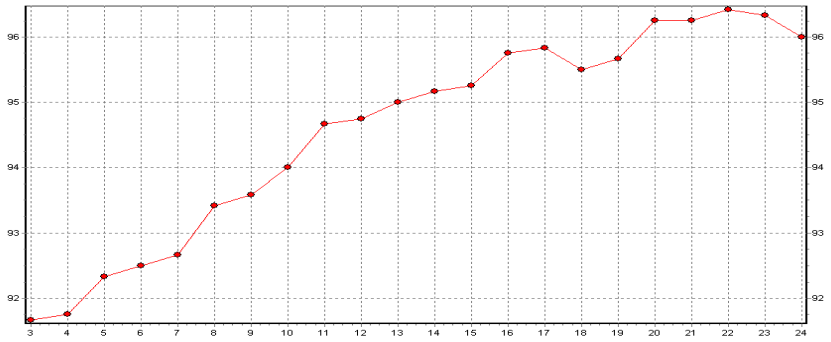

Figura 4.12: Gráfico para o Módulo Setembro

Módulo Setembro

- Qtde Características: 24 = Taxa Rec.: 96;
- Qtde Características: 22 = Taxa Rec.: 96,42;
- SubConjunto: NLE, NSCVE, NSCVD, NSCXE, NSCXD, NCPE, NCPD, NBPE, NBPD, NEPE, NEPD, NAE, NAD, NDE, NCH, NPP, NTV, NTH, NLAE, NLAD, NLDE, NLDD.

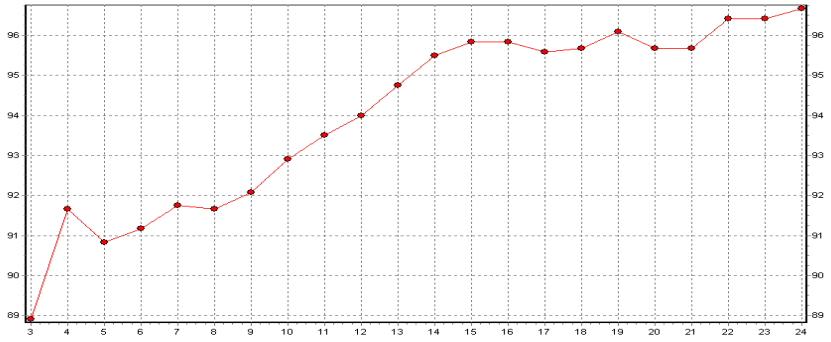

Figura 4.13: Gráfico para o Módulo Outubro

Módulo Outubro

- Qtde Características: 24 = Taxa Rec.: 96,66;
- SubConjunto: Não houve modificações na performance, conjunto permanece o inicial.

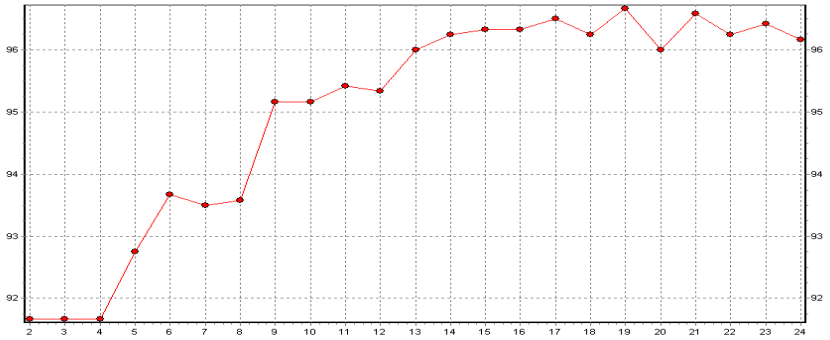

Figura 4.14: Gráfico para o Módulo Novembro

Módulo Novembro

- Qtde Características: 24 = Taxa Rec.: 96,12;
- Qtde Características: 19 = Taxa Rec.: 96,67;
- SubConjunto: NSCVE, NSCVD, NSCXD, NCPE, NBPE, NBPD, NEPE, NAE, NAD, NDE, NDD, NCH, NPP, NTV, NTH, NLAE, NLAD, NLDE, NLDD.

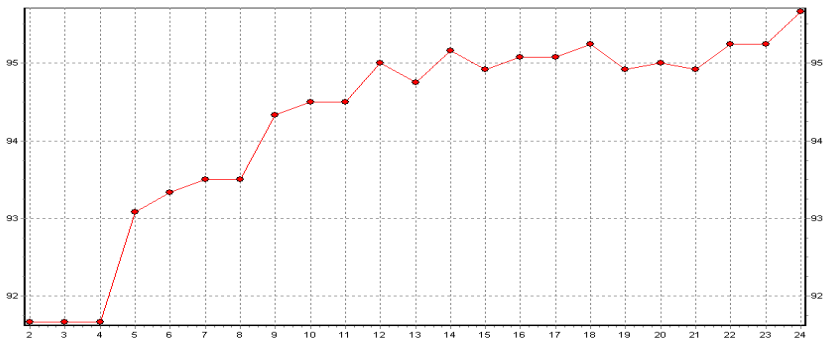

Figura 4.15: Gráfico para o Módulo Dezembro

Módulo Dezembro

- Qtde Características: 24 = Taxa Rec.: 95,66;
- SubConjunto: Não houve modificações na performance, conjunto permanece o inicial.

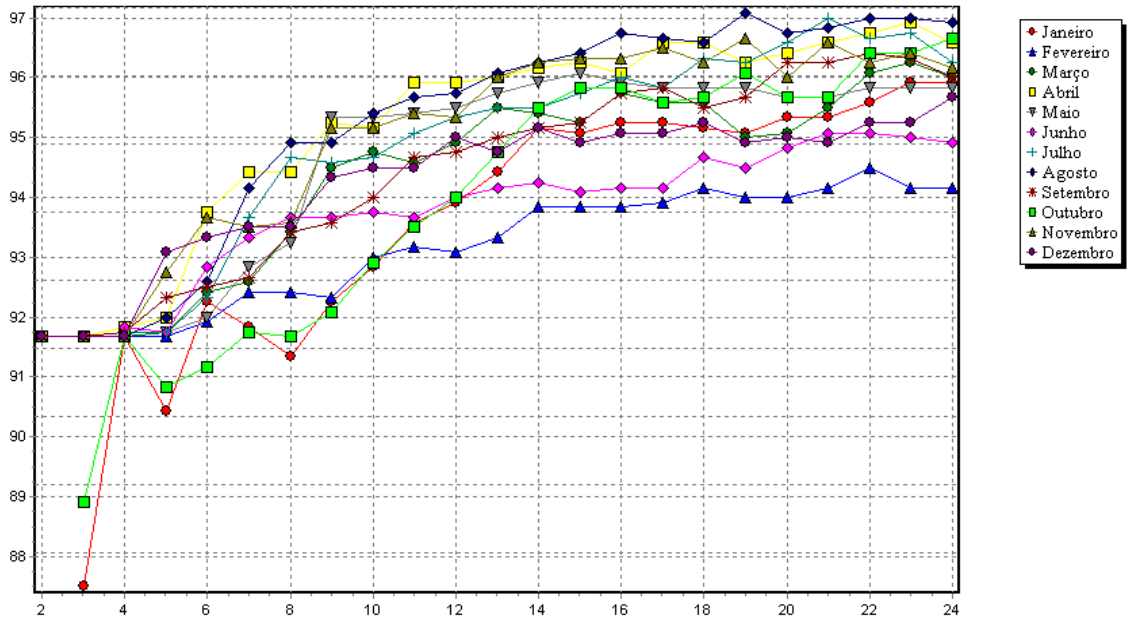

Figura 4.16: Gráfico para os Módulos Juntos (Reconhecimento x Qtde características)

A Tabela 4.17 é referente a experimentos realizados com o conjunto inicial de características e os novos subconjuntos.

Tabela 4.17: Taxas de reconhecimento sem mecanismo de rejeição no conjunto de teste

Arquitetura	**Qtde de Características**	**Rec.(%)**
Convencional	24	77,08
Convencional	22	74,58
Classe-Modular	24	81,75
Classe-Modular	Dependente	80,16

4.1.7 Análise dos Resultados

Nesta Subseção são analisados os resultados obtidos pela metodologia proposta no léxico dos meses do ano. A comparação dos resultados obtidos com os já encontrados na literatura é realizada para posicionar o presente trabalho diante da comunidade

científica. Os trabalhos adotados para comparação de resultados são dois também descritos no Capítulo 2:

- O estudo [JdCCFS02] foi selecionado pois a base de imagens trabalhada neste estudo é a mesma do presente trabalho, porém sem mecanismos de rejeição. Detalhes dos resultados são apresentados na Tabela 4.18.
- Nos estudos [MYS+01], [MSBS02] e [MOS+02] os autores trabalham, em parte, com os nomes dos meses do ano em português, assim como no presente trabalho, porém com uma base de imagens diferente. De acordo com os autores, os resultados variaram em 89.5% e 91,5% sem rejeição.

A organização das análises segue a mesma ordem da metodologia, com comentários sobre os resultados das arquiteturas de RNAs que também foram publicados em Kapp *et al.* [KFNS03], mecanismos de rejeição e seleção de características.

Arquiteturas Convencional e Classe-Modular

Para comparação entre arquitetura clássica não-modular e a modular, dois critérios são adotados, convergência e capacidade de reconhecimento. Sobre a convergência, observam-se grandes diferenças entre os MSEs (*Mean Square Error*) da convencional e da modular.

O MSE da modular é obtido através da média dos MSEs das K subredes, e ainda assim é menor do que a da rede não-modular. Duas conclusões podem ser feitas baseadas nos resultados experimentais referentes aos desempenhos das arquiteturas:

- A arquitetura classe-modular teve superioridade em termos de término de convergência em relação a convencional;
- A arquitetura classe-modular também foi superior em termos de capacidade de reconhecimento do que a arquitetura convencional neste trabalho.

Outra observação importante é em relação ao tamanho do classificador *versus* tamanho do conjunto de treinamento. O treinamento da rede é um processo para estimar um conjunto de valores ótimos para os pesos das redes pelo uso de um conjunto de amostras de treinamento.

Na arquitetura classe-modular, boa parte dos pesos é reduzida no treinamento individual, pois temos um conjunto de subredes, treinadas separadamente, ou seja, menos parâmetros a serem estimados. Outra vantagem é que cada amostra passa em determinada época K vezes e não uma vez como na convencional. Em geral, as redes modulares ocupam mais tempo computacional do que as não-modulares, porém em conjuntos de classificação grandes, ocorre o contrário [OS02].

Em relação a tempo computacional, ou seja, para alguns casos o número de pesos (ligações entre os nós) é maior para as redes modulares, o que acarreta num maior tempo de processamento. Se para uma RNA-MLP convencional, agindo como um 2-classificador a quantidade de pesos é igual a $dm+2m$, sendo m o número de neurônios na camada escondida, no caso modular onde há K subredes, esta quantidade passa a ser a soma total de pesos de todas as subredes, portanto $(dm + 2m)K$, sendo que para uma arquitetura convencional tem-se $dm + mK$.

A Tabela 4.18 foi publicada em [KFNS03] e resume os resultados obtidos neste trabalho e alguns outros estudos. A arquitetura classe-modular obteve uma taxa de reconhecimento similar à de uma RNA-MLP utilizando características perceptivas (PF).

As diferentes taxas de reconhecimentos entre uma mesma arquitetura podem ocorrer principalmente devido a fatores como o conjunto de características e o tamanho do vetor de características trabalhados, e consequentemente o tamanho das RNAs, ou ainda devido a diferentes conjuntos de treinamento, validação e teste utilizados nos experimentos, como neste caso.

Tabela 4.18: Resultados comparativos

Método	**Reconhecimento (%)**
HMM [JdCCFS02]	75,90
Carac. Direcionais / RNA [JdCCFS02]	76,60
Carac. Perceptivas / RNA [JdCCFS02]	81,80
Arquitetura Convencional [KFNS03]	77,08
Arquitetura Classe-Modular [KFNS03]	81,75

Rejeição

Pode-se observar através de uma análise das Tabelas 4.14 e 4.15, que assim como em [FRG00], a obtenção dos limiares através dos CRTs foi melhor do que utilizando a regra de Chow para uma arquitetura RNA-MLP convencional. Complementando, pode-se observar principalmente que entre o desempenho de obtenção e teste das duas arquiteturas, a arquitetura classe-modular é superior a convencional, em termos de maiores taxas de reconhecimento em menores taxas de rejeição com taxas de erro fixas, veja Tabela 4.15.

As explicações para este fato são, primeiramente, devido melhor mapeamento entre espaço de características e classes proporcionado pela arquitetura classe-modular antes mesmo do processo de rejeição, e principalmente a idéia de múltiplos limiares, obtidos de forma local em cada módulo para cada classe, e não de forma global como na Tabela 4.14.

A superioridade também estende-se a um conjunto de teste totalmente desconhecido pelas arquiteturas, veja Tabela 4.16. Entretanto, as taxas de erro se diferenciam das fixadas no conjunto de validação principalmente devido à variabilidade de estilos de escrita (cursivas pura, de forma, caracteres disjuntos e mistas) e suas respectivas quantidades que formam o conjunto dos dados, o que influencia nos resultados dos conjuntos formados de maneira aleatória.

Seleção de Características - Abordagem *Wrapper/Hill Climbing*

Considerando uma análise classe-dependente da arquitetura classe-modular, temos 288 características no total ($24 \times 12 = 288$). A diferença entre a quantidade total de características (288) e as novas (256) são de 32 características, sendo que de modo geral há um equilíbrio entre características do lado esquerdo e direito, 12 são responsáveis pelo lado direito, 16 pelo lado esquerdo e 4 "neutras de lado"(NPP em janeiro, NCH, NTH em agosto e NTV em julho).

A Tabela 4.17 é referente a experimentos realizados com ambos, o conjunto inicial de características e os novos subconjuntos. Analisando a Tabela 4.17, observa-se que na arquitetura convencional utilizando o conjunto de validação, há um pequeno aumento de reconhecimento com menos características, porém com o conjunto de testes sua performance é comprometida.

Já a investigação das características módulo a módulo, mostra através dos gráficos como os subconjuntos de características são importantes para a separação de uma determinada classe. Porém quando os subconjuntos de cada módulo são organizados de forma a trabalharem juntos, há maior dificuldade de separação e a performance do sistema é comprometida novamente.

Entretanto, de acordo com Raman e Ioerger em [RI03], há dois problemas principais no desenvolvimento de arquiteturas de seleção de características. Primeiramente, encontrar um conjunto de características "mínimo"é um problema NP-Completo e segundo, o conjunto de características que provê o máximo aumento de precisão na classificação não necessariamente define uma hipótese consistente, muito provavelmente por um problema de relevância e irrelevância de exemplos descrito em [RI03].

Concluindo, os estudos utilizando a base de validação e os efeitos no conjunto de teste mostram que para aumentar a performance do sistema, não é possível utilizar um conjunto menor de primitivas. Sendo assim, para melhorar os resultados, deve-se estudar novas primitivas.

O resultado final para a metodologia proposta na base de imagens de palavras manuscritas da UFCG, referentes aos nomes dos meses do ano, é de 81,75% de taxa de reconhecimento adotando a arquitetura classe-modular sem rejeição e torna-se 91,52% com a utilização do mecanismo de múltiplos limiares de rejeição no mesmo conjunto de teste. Os valores dos limiares são estimados no conjunto de validação com taxas de erro fixadas e as taxas de rejeição variam entre um valor "ideal"de 20% para aplicações reais, de acordo com Liu *et al.* [LSF02].

4.2 Experimento 2 - Base de Dados PUC-PR

Nesta etapa, a base de imagens foi pré-processada, ambas correções de *skew* e *slant* foram realizadas. A divisão da base de dados nos conjuntos de treinamento, validação e teste seguiu de um balanceamento na quantidade de amostras das palavras mil, reais e centavos que se encontravam em número muito acima da média das outras classes e assim novas quantias foram selecionadas aleatoriamente, condizentes com o número de amostras de outras classes. A Figura 4.17 exemplifica imagens da base de dados.

Uma estratificação das amostras das classes também foi realizada para garantir que as quantidades de amostras existentes nos conjuntos de treinamento, validação e teste fossem similares em cada classe. A Tabela 4.19 mostra em detalhes a montagem dos conjuntos.

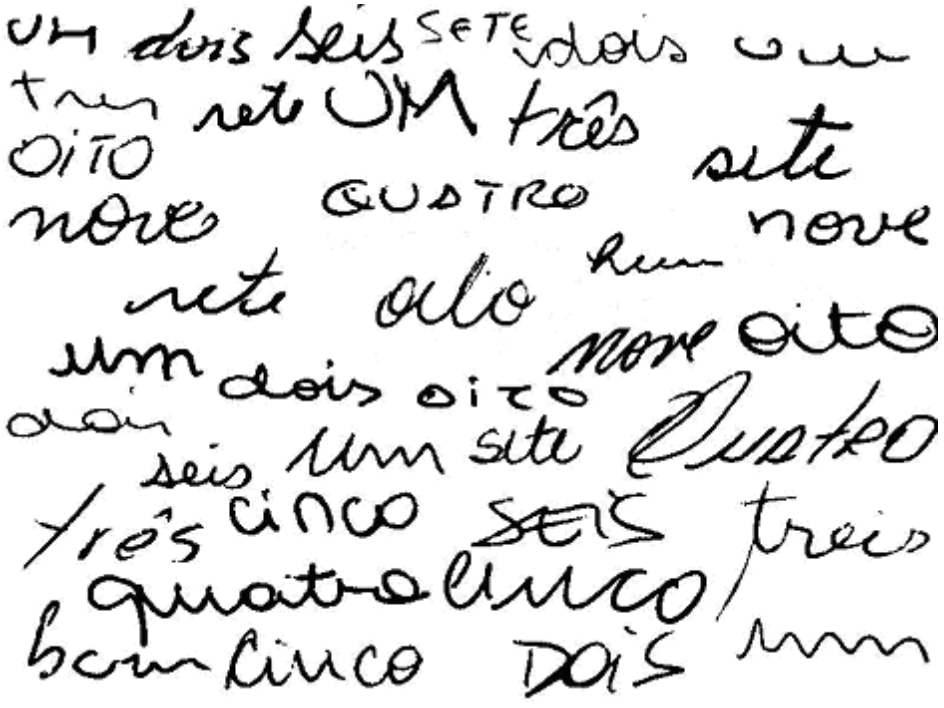

Figura 4.17: Amostras da base de imagens PUC-PR referentes aos valores por extenso

Tabela 4.19: Distribuição das amostras da base PUC-PR nos conjuntos

Índice	**Palavra**	**Treinamento**	**Validação**	**Teste**	**Total**
0	*Um*	191	64	64	319
1	*Dois*	209	69	69	347
2	*Três*	191	64	64	319
3	*Quatro*	182	60	61	303
4	*Cinco*	191	64	64	319
5	*Seis*	167	55	55	277
6	*Sete*	170	56	57	283
7	*Oito*	176	59	59	294
8	*Nove*	183	61	61	305
9	*Dez*	29	9	9	47
10	*Onze*	30	10	10	50
11	*Doze*	17	6	6	29
12	*Treze*	23	8	8	39
13	*Quatorze*	23	7	7	37
14	*Quinze*	27	9	9	45
15	*Dezesseis*	17	5	6	28
16	*Dezessete*	23	7	8	38
17	*Dezoito*	21	7	7	35
18	*Dezenove*	28	9	9	46
19	*Vinte*	221	74	74	369
20	*Trinta*	221	74	74	369
21	*Quarenta*	206	68	68	342
22	*Cinquenta*	224	74	75	373
23	*Sessenta*	215	71	72	358
24	*Setenta*	213	71	71	355
25	*Oitenta*	221	74	74	369
26	*Noventa*	196	65	65	326
27	*Cem/Cento*	163	54	54	271
28	*Duzentos*	152	50	51	253
29	*Trezentos*	141	47	47	235
30	*Quatrocentos*	132	44	44	220
31	*Quinhentos*	149	50	50	249
32	*Seiscentos*	137	46	46	229
33	*Setecentos*	152	50	50	252
34	*Oitocentos*	151	50	50	251
35	*Novecentos*	144	48	48	240
36	*Mil*	228	76	76	380
37	*Reais*	228	76	76	380
38	*Centavos*	228	76	76	380
	Total	5620	1867	1874	9361

4.2.1 Aplicação de um classificador K-NN (*K-Nearest Neighbour*)

Nesta fase, aplica-se um algoritmo K-NN para os conjuntos de treinamento, validação e teste da base de imagens da PUCPR, assim como realizada anteriormente na base de imagens UFCG. A Tabela 4.20 apresenta os resultados obtidos.

Tabela 4.20: Taxas de reconhecimento obtidas para cada conjunto com um K-NN

Conjunto	**Reconhecimento(%)**
Treinamento	33,83
Validação	31,12
Teste	29,35

4.2.2 Resultados para a Arquitetura Convencional

Após a experimentação de várias quantidades de neurônios na camada escondida e parâmetros, o melhor ajuste foi alcançado com uma RNA MLP com 24 neurônios de entrada, 48 na camada escondida e 39 de saída. Os valores dos parâmetros de constante de aprendizagem e momento são de 0,3 e 0,1 respectivamente, com a iniciação dos pesos aleatória com valores entre 0 e 1. O término do treinamento é alcançado em aproximadamente 250 épocas, com um MSE de 0,1190012 para o conjunto de treinamento e 0,1346084 para o conjunto de validação. A Tabela 4.21 apresenta os resultados obtidos para a aquitetura convencional.

4.2.3 Resultados para a Arquitetura Classe-Modular

Após a experimentação de várias RNAs, camadas escondidas e parâmetros, o melhor ajuste ocorre com camadas escondidas fixadas nas subredes em 48 neurônios. Os valores dos parâmetros de constante de aprendizagem e momento variam entre 0,2 e 0,4, o valor dos momentos em geral é 0,4 em conjuntos de treinamento pequenos (com menos do que 40 amostras). O término do treinamento ocorre, em geral, em 800 épocas. Os pesos iniciam aleatóriamente, com valores entre 0 e 1. A Tabela 4.22 apresenta os resultados obtidos para a aquitetura classe-modular.

Tabela 4.21: Resultados com a arquitetura convencional e o conjunto de teste

Classes	**Rec. (%)**	**Padrões/Classe**
Um	76,56	64
Dois	60,87	69
Três	68,75	64
Quatro	45,90	61
Cinco	56,25	64
Seis	54,55	55
Sete	42,11	57
Oito	40,68	59
Nove	63,93	61
Dez	33,33	9
Onze	20,00	10
Doze	0,00	6
Treze	50,00	8
Quatorze	14,29	7
Quinze	33,33	9
Dezesseis	33,33	6
Dezessete	0,00	8
Dezoito	14,29	7
Dezenove	55,56	9
Vinte	72,97	74
Trinta	55,41	74
Quarenta	36,76	68
Cinquenta	68,00	75
Sessenta	55,56	72
Setenta	23,94	71
Oitenta	41,89	74
Noventa	33,85	65
Cem/Cento	29,63	54
Duzentos	41,18	51
Trezentos	21,28	47
Quatrocentos	84,09	44
Quinhentos	50,00	50
Seiscentos	4,35	46
Setecentos	38,00	50
Oitocentos	30,00	50
Novecentos	35,42	48
Mil	65,79	76
Reais	36,84	76
Centavos	77,63	76
Total	48,93	1874

Tabela 4.22: Resultados com a arquitetura classe-modular e o conjunto de teste

Classes	**Rec. (%)**	**Padrões/Classe**
Um	89,06	64
Dois	66,67	69
Três	68,75	64
Quatro	47,54	61
Cinco	57,81	64
Seis	56,36	55
Sete	45,61	57
Oito	57,63	59
Nove	52,46	61
Dez	33,33	9
Onze	10,00	10
Doze	50,00	6
Treze	12,50	8
Quatorze	14,29	7
Quinze	22,22	9
Dezesseis	0,00	6
Dezessete	62,50	8
Dezoito	0,00	7
Dezenove	44,44	9
Vinte	68,92	74
Trinta	66,22	74
Quarenta	38,24	68
Cinquenta	54,67	75
Sessenta	51,39	72
Setenta	26,76	71
Oitenta	48,65	74
Noventa	56,92	65
Cem/Cento	33,33	54
Duzentos	41,18	51
Trezentos	31,91	47
Quatrocentos	75,00	44
Quinhentos	50,00	50
Seiscentos	19,57	46
Setecentos	40,00	50
Oitocentos	36,00	50
Novecentos	39,58	48
Mil	75,00	76
Reais	44,74	76
Centavos	78,95	76
Total	52,35	1874

4.2.4 Tentativa do uso de informação *a priori* das classes

A tentativa da utilização da informação *a priori* das classes é realizada com o propósito de aumentar o poder discriminatório e consequentemente a taxa de reconhecimento dos classificadores. Tal tentativa aqui, é baseada no trabalho de Lawrence *et al.* em [LBB+98], em que um procedimento denominado *post scaling* é feito após a obtenção das saídas das redes neurais artificiais.

O procedimento parte do princípio que é possível efetuar um treinamento comum numa RNA e então depois utilizar conjuntamente suas respostas de saídas y_i juntamente com as informações s_i de probabilidade *a priori* das classes através de uma multiplicação entre os valores de y_i e s_i. A Equação 4.2 apresenta como são calculadas as informações de probabilidade *a priori* das classes, em que s_i representa estas informações, c_s é uma constante que representa o quanto da informação das probabilidades *a priori* deve ser utilizada, p_i é a probabilidade *a priori* de uma classe e N_c o número total de classes.

$$s_i = 1 - c_s + \frac{c_s}{p_i.N_c} \qquad i = 0, 1, \ldots, N_c - 1 \tag{4.2}$$

A estimativa do valor de c_s, $0 \leq c_s \leq 1$, é realizada através do uso do conjunto de validação. Se $c_s = 0$ significa que não há escalonamento com a utilização da informação *a priori*. Caso contrário, $c_s = 1$, corresponde ao uso da informação *a priori* integralmente. Os resultados obtidos com as arquiteturas de RNA-MLP utilizando a informação *a priori* das classes são descritos a seguir.

Para a arquitetura convencional os resultados são os seguintes:

- Taxa de reconhecimento no conjunto de validação: 47,24%.
- Taxa de reconhecimento no conjunto de validação com fator c_s estimado em 0,2: 47,35% (melhor resultado) .
- Taxa de reconhecimento no conjunto de teste: 48,93%.
- Taxa de reconhecimento no conjunto de teste com fator c_s=0,2 (estimado através do conjunto de validação): 48,72%.

As Tabelas 4.23 e 4.24 resumem a tentativa da utilização da informação *a priori* na arquitetura convencional considerando as 3 melhores hipóteses de reconhecimento (Top 1, Top 2 e Top 3).

Tabela 4.23: Resultados dos experimentos com o conjunto de teste

Top 1	**Top 2**	**Top 3**
48,93	64,41	73,59

Tabela 4.24: Resultados dos experimentos com o conjunto de teste e fator CS(0,2)

Top 1	**Top 2**	**Top 3**
48,72	63,39	72,84

Para a arquitetura classe-modular os resultados são os seguintes:

- Taxa de reconhecimento no conjunto de validação: 47,78%.
- Taxa de reconhecimento no conjunto de validação com fator CS estimado em 0,1, com a taxa de reconhecimento no conjunto de validação torna-se 47,72% (melhor resultado).
- Taxa de reconhecimento no conjunto de teste: 52,35%.
- Taxa de reconhecimento no conjunto de teste com fator CS=0,1 (estimado através do conjunto de validação) torna-se: 52,24%.

As Tabelas 4.25 e 4.26 resumem a tentativa da utilização da informação *a priori* na arquitetura classe-modular:

Tabela 4.25: Resultados dos experimentos com o conjunto de teste

Top 1	**Top 2**	**Top 3**
52,35	66,12	73,96

Tabela 4.26: Resultados dos experimentos com o conjunto de teste e fator CS(0,1)

Top 1	**Top 2**	**Top 2**
52,24	65,85	73,37

4.2.5 Mecanismo de Rejeição com Múltiplos Limiares

A obtenção dos limiares nesse léxico de palavras difere-se um pouco da obtenção para os meses devido às taxas de erro fixadas. Para o léxico de meses fixou-se a taxa de erro em: 1%, 2% e 5%, porém neste léxico dos valores por extenso não consegue-se fixar o erro em por exemplo 1% para a classe "dez", visto que a menor taxa de erro que pode-se atingir diferente de 0% seria a de 1,11% e assim por diante. A quantidade disponível de amostras para as obtenções dificulta o trabalho. Assim fixou-se a obtenção dos limiares nos menores erros possíveis para cada classe. A seguir apresenta-se nas Tabelas 4.27 e 4.28 os dados referentes à arquitetura convencional, e nas Tabelas 4.29 e 4.30 os dados referentes à arquitetura classe-modular.

4.2.6 Seleção de Características - Abordagem *Wrapper/Hill Climbing*

Resultados Obtidos com as Arquiteturas Convencional e Classe-Modular

Os resultados obtidos com 23 primitivas no conjunto de validação foram:

- Arquitetura Convencional: 47,40%.
- Arquitetura Classe-Modular: 47,93%

Os resultados obtidos com 23 primitivas no conjunto de teste foram:

- Arquitetura Convencional: 48,02%.
- Arquitetura Classe-Modular: 52,18%

Tabela 4.27: Limiares e resultados no conj. de validação com a arq. convencional

Classe	**Limiar**	**Rec.(%)**	**Rej.(%)**	**Erro(%)**	**Conf.(%)**	**E.C.(%)**
Um	0,047	89,06	9,38	1,56	98,28	1,72
Dois	0,352	43,48	53,62	2,90	93,75	6,25
Três	0,579	51,56	46,88	1,56	97,06	2,94
Quatro	0,328	38,33	60,00	1,67	95,83	4,17
Cinco	0,736	23,44	75,00	1,56	93,75	6,25
Seis	0,703	27,27	70,91	1,82	93,75	6,25
Sete	0,381	42,86	55,36	1,79	96,00	4,00
Oito	0,715	16,95	81,36	1,69	90,91	9,09
Nove	0,41	40,98	55,74	3,28	92,59	7,41
Dez	0,197	44,44	44,44	11,11	80,00	20,00
Onze	0,797	40,00	50,00	10,00	80,00	20,00
Doze	0,0024	0,00	83,33	16,67	0,00	100,00
Treze	0,148	0,00	87,50	12,50	0,00	100,00
Quatorze	0,501	0,00	85,71	14,29	0,00	100,00
Quinze	0,203	11,11	77,78	11,11	50,00	50,00
Dezesseis	0,352	0,00	80,00	20,00	0,00	100,00
Dezessete	0,044	0,00	85,71	14,29	0,00	100,00
Dezoito	0,081	0,00	85,71	14,29	0,00	100,00
Dezenove	0,314	22,22	66,67	11,11	66,67	33,33
Vinte	0,491	64,86	32,43	2,70	96,00	4,00
Trinta	0,404	40,54	56,76	2,70	93,75	6,25
Quarenta	0,572	20,59	76,47	2,94	87,50	12,50
Cinquenta	0,524	32,43	64,86	2,70	92,31	7,69
Sessenta	0,421	46,48	52,11	1,41	97,06	2,94
Setenta	0,657	9,86	88,73	1,41	87,50	12,50
Oitenta	0,3605	18,92	79,73	1,35	93,33	6,67
Noventa	0,592	21,54	75,38	3,08	87,50	12,50
Cem/Cento	0,381	18,52	77,78	3,70	83,33	16,67
Duzentos	0,483	20,00	78,00	2,00	90,91	9,09
Trezentos	0,386	17,02	78,72	4,26	80,00	20,00
Quatrocentos	0,008	88,64	9,09	2,27	97,50	2,50
Quinhentos	0,585	34,00	64,00	2,00	94,44	5,56
Seiscentos	0,288	0,00	95,65	4,35	0,00	100,00
Setecentos	0,326	14,00	84,00	2,00	87,50	12,50
Oitocentos	0,163	32,00	66,00	2,00	94,12	5,88
Novecentos	0,394	20,83	75,00	4,17	83,33	16,67
Mil	0,714	50,00	48,68	1,32	97,44	2,56
Reais	0,641	15,79	82,89	1,32	92,31	7,69
Centavos	0,173	75,00	23,68	1,32	98,28	1,72
Média		34,83	62,47	2,69	87,84	12,16

Tabela 4.28: Resultados com a arq. convencional, conj. de teste e os limiares estimados pelo conj. de validação

Classe	**Rec.(%)**	**Rej.(%)**	**Erro(%)**	**Conf.(%)**	**E.C.(%)**
Um	76,56	23,44	0,00	100,00	0,00
Dois	49,28	50,72	0,00	100,00	0,00
Três	53,13	42,19	4,69	91,89	8,11
Quatro	44,26	54,10	1,64	96,43	3,57
Cinco	32,81	65,63	1,56	95,45	4,55
Seis	23,64	72,73	3,64	86,67	13,33
Sete	31,58	64,91	3,51	90,00	10,00
Oito	11,86	79,66	8,47	58,33	41,67
Nove	50,82	49,18	0,00	100,00	0,00
Dez	33,33	66,67	0,00	100,00	0,00
Onze	10,00	80,00	10,00	50,00	50,00
Doze	0,00	100,00	0,00	0,00	100,00
Treze	50,00	50,00	0,00	100,00	0,00
Quatorze	14,29	85,71	0,00	100,00	0,00
Quinze	33,33	66,67	0,00	100,00	0,00
Dezesseis	33,33	50,00	16,67	66,67	33,33
Dezessete	0,00	100,00	0,00	0,00	100,00
Dezoito	14,29	85,71	0,00	100,00	0,00
Dezenove	44,44	44,44	11,11	80,00	20,00
Vinte	67,57	29,73	2,70	96,15	3,85
Trinta	41,89	55,41	2,70	93,94	6,06
Quarenta	20,59	77,94	1,47	93,33	6,67
Cinquenta	54,67	40,00	5,33	91,11	8,89
Sessenta	44,44	54,17	1,39	96,97	3,03
Setenta	1,41	92,96	5,63	20,00	80,00
Oitenta	22,97	77,03	0,00	100,00	0,00
Noventa	15,38	80,00	4,62	76,92	23,08
Cem/Cento	24,07	66,67	9,26	72,22	27,78
Duzentos	25,49	74,51	0,00	100,00	0,00
Trezentos	12,77	87,23	0,00	100,00	0,00
Quatrocentos	84,09	13,64	2,27	97,37	2,63
Quinhentos	32,00	64,00	4,00	88,89	11,11
Seiscentos	0,00	97,83	2,17	0,00	100,00
Setecentos	30,00	54,00	16,00	65,22	34,78
Oitocentos	28,00	60,00	12,00	70,00	30,00
Novecentos	25,00	75,00	0,00	100,00	0,00
Mil	51,32	46,05	2,63	95,12	4,88
Reais	13,16	77,63	9,21	58,82	41,18
Centavos	76,32	23,68	0,00	100,00	0,00
Média	36,39	60,08	3,52	84,08	15,92

Tabela 4.29: Limiares e resultados no conj. de validação com a arq. classe-mod.

Classe	**Limiar**	**Rec.(%)**	**Rej.(%)**	**Erro(%)**	**Conf.(%)**	**E.C.(%)**
Um	0,335	92,19	6,25	1,56	98,33	1,67
Dois	0,175	53,62	43,48	2,90	94,87	5,13
Três	0,635	45,31	53,13	1,56	96,67	3,33
Quatro	0,8	16,67	81,67	1,67	90,91	9,09
Cinco	0,69	25,00	73,44	1,56	94,12	5,88
Seis	0,97	9,09	89,09	1,82	83,33	16,67
Sete	0,785	16,07	82,14	1,79	90,00	10,00
Oito	0,775	28,81	69,49	1,69	94,44	5,56
Nove	0,115	40,98	55,74	3,28	92,59	7,41
Dez	0,01	55,56	33,33	11,11	83,33	16,67
Onze	0,0435	30,00	60,00	10,00	75,00	25,00
Doze	0,23	16,67	66,67	16,67	50,00	50,00
Treze	0,03	0,00	87,50	12,50	0,00	100,00
Quatorze	0,0065	28,57	57,14	14,29	66,67	33,33
Quinze	0,18	0,00	88,89	11,11	0,00	100,00
Dezesseis	0,005	0,00	80,00	20,00	0,00	100,00
Dezessete	0,00003	0,00	85,71	14,29	0,00	100,00
Dezoito	0,0065	0,00	85,71	14,29	0,00	100,00
Dezenove	0,195	44,44	44,44	11,11	80,00	20,00
Vinte	0,49	45,95	51,35	2,70	94,44	5,56
Trinta	0,325	63,51	33,78	2,70	95,92	4,08
Quarenta	0,68	23,53	73,53	2,94	88,89	11,11
Cinquenta	0,41	25,68	71,62	2,70	90,48	9,52
Sessenta	0,655	15,49	83,10	1,41	91,67	8,33
Setenta	0,5	18,31	80,28	1,41	92,86	7,14
Oitenta	0,79	4,05	94,59	1,35	75,00	25,00
Noventa	0,67	26,15	70,77	3,08	89,47	10,53
Cem/Cento	0,19	24,07	72,22	3,70	86,67	13,33
Duzentos	0,725	24,00	74,00	2,00	92,31	7,69
Trezentos	0,56	17,02	78,72	4,26	80,00	20,00
Quatrocentos	0,985	50,00	47,73	2,27	95,65	4,35
Quinhentos	0,895	28,00	70,00	2,00	93,33	6,67
Seiscentos	0,625	0,00	95,65	4,35	0,00	100,00
Setecentos	0,165	32,00	66,00	2,00	94,12	5,88
Oitocentos	0,615	18,00	80,00	2,00	90,00	10,00
Novecentos	0,7	18,75	77,08	4,17	81,82	18,18
Mil	0,93	44,74	53,95	1,32	97,14	2,86
Reais	0,455	14,47	84,21	1,32	91,67	8,33
Centavos	0,12	65,79	32,89	1,32	98,04	1,96
Média		31,00	66,30	2,69	86,95	13,05

Tabela 4.30: Resultados com a arquitetura classe-modular, conj. de teste e os limiares estimados através do conj. de validação

Classe	**Rec.(%)**	**Rej.(%)**	**Erro(%)**	**Conf.(%)**	**E.C.(%)**
Um	87,50	12,50	0,00	100,00	0,00
Dois	60,87	39,13	0,00	100,00	0,00
Três	53,13	45,31	1,56	97,14	2,86
Quatro	27,87	70,49	1,64	94,44	5,56
Cinco	37,50	56,25	6,25	85,71	14,29
Seis	5,45	89,09	5,45	50,00	50,00
Sete	7,02	91,23	1,75	80,00	20,00
Oito	44,07	49,15	6,78	86,67	13,33
Nove	49,18	50,82	0,00	100,00	0,00
Dez	33,33	66,67	0,00	100,00	0,00
Onze	10,00	90,00	0,00	100,00	0,00
Doze	16,67	66,67	16,67	50,00	50,00
Treze	12,50	87,50	0,00	100,00	0,00
Quatorze	14,29	71,43	14,29	50,00	50,00
Quinze	22,22	77,78	0,00	100,00	0,00
Dezesseis	0,00	100,00	0,00	0,00	100,00
Dezessete	62,50	37,50	0,00	100,00	0,00
Dezoito	0,00	100,00	0,00	0,00	100,00
Dezenove	44,44	44,44	11,11	80,00	20,00
Vinte	52,70	45,95	1,35	97,50	2,50
Trinta	60,81	37,84	1,35	97,83	2,17
Quarenta	22,06	76,47	1,47	93,75	6,25
Cinquenta	44,00	56,00	0,00	100,00	0,00
Sessenta	30,56	68,06	1,39	95,65	4,35
Setenta	15,49	81,69	2,82	84,62	15,38
Oitenta	12,16	86,49	1,35	90,00	10,00
Noventa	32,31	67,69	0,00	100,00	0,00
Cem/Cento	29,63	70,37	0,00	100,00	0,00
Duzentos	29,41	68,63	1,96	93,75	6,25
Trezentos	14,89	80,85	4,26	77,78	22,22
Quatrocentos	38,64	59,09	2,27	94,44	5,56
Quinhentos	34,00	66,00	0,00	100,00	0,00
Seiscentos	6,52	93,48	0,00	100,00	0,00
Setecentos	34,00	64,00	2,00	94,44	5,56
Oitocentos	22,00	78,00	0,00	100,00	0,00
Novecentos	25,00	72,92	2,08	92,31	7,69
Mil	42,11	57,89	0,00	100,00	0,00
Reais	18,42	77,63	3,95	82,35	17,65
Centavos	77,63	21,05	1,32	98,33	1,67
Média	35,70	62,49	1,81	92,06	7,94

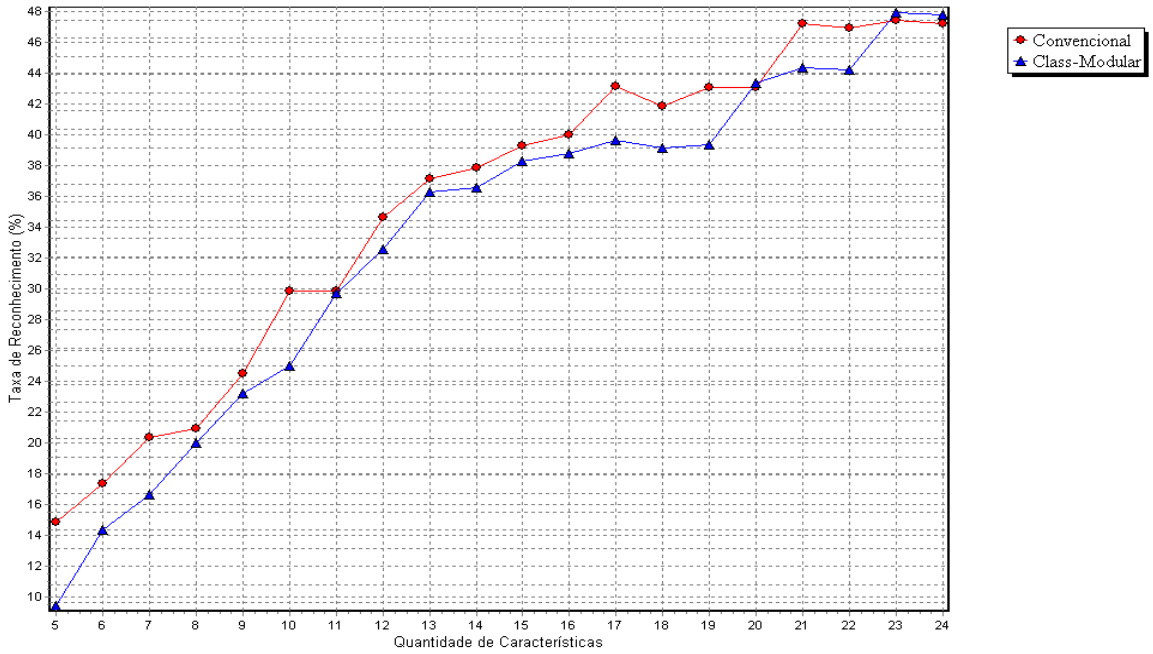

Figura 4.18: Resultados da abordagem *Wrapper/Hill Climbing* para as arquiteturas convencional e classe-modular na base PUC-PR

4.2.7 Análise dos Resultados

Nesta Subseção são descritas as análises dos resultados obtidos pela metodologia proposta no léxico dos valores por extenso. A comparação dos resultados obtidos com trabalhos já encontrados na literatura também é realizada assim como realizado para o léxico dos meses. Os trabalhos adotados para comparação de resultados são dois também descritos no Capítulo 2:

- O estudo [Fre01] foi selecionado pois a base de imagens trabalhada neste estudo é a mesma do presente trabalho, porém sem mecanismos de rejeição. Detalhes dos resultados são apresentados na Tabela 4.31;
- E [GL01], onde os autores também trabalham com palavras do extenso de cheques bancários brasileiros, assim como no presente trabalho, porém com uma outra base de imagens isoladas. De acordo com os autores, o resultado é de 50% de taxa de reconhecimento em média.

Tabela 4.31: Resultados comparativos

Método	**Reconhecimento (%)**
Freitas em [Fre01]	70,60
Arq. Classe-Modular	52,35
Gomes em [GL01]	50,00

Grande parte das baixas taxas de reconhecimento produzidas pelas arquiteturas são devidas principalmente à excassez de amostras. Pode-se observar que a maioria das classes com baixo reconhecimento, são aquelas que possuem menos amostras. A falta de amostras compromete também a obtenção de limiares no mecanismo de rejeição e na análise e seleção de características classe-dependente na arquitetura classe-modular, tornando inviável a análise devido às baixas taxas de reconhecimento. As informações *a priori* das classes não foram utilizadas na obtenção dos limiares e também na seleção de características porque quando elas são aplicadas, não há melhora significativa nos resultados. Também, de acordo com Lawrence *et al.* em [LBB+98] o uso da informação *a priori* pode modificar a distribuição de probabilidade, podendo causar distúrbios na obtenção de limiares e também na classificação.

Capítulo 5

Conclusão

O foco principal desta dissertação foi o desenvolvimento de um método automático para o reconhecimento de palavras manuscritas referentes aos meses do ano e valores por extenso de cheques bancários. Observou-se a dificuldade da realização desta tarefa principalmente devido à grande variabilidade de estilos de escrita encontrados. O método desenvolvido basicamente utiliza redes neurais artificiais como classificador, porém diferentes arquiteturas e mecanismos de rejeição foram testados juntamente com uma seleção de características.

Em relação aos estudos das arquiteturas de RNA-MLP convencional e classe-modular, os resultados obtidos nos estudos, indicam que esta pesquisa é totalmente promissora e prova ser merecedora de investigações adicionais do paradigma de modularidade de classes. Porém, uma consideração deve ser feita sobre classificações em grandes conjuntos de dados, com o propósito de investigar o efeito do número de classes na capacidade de reconhecimento.

No presente trabalho, propõe-se e implementa-se um novo conjunto de características baseado em características também trabalhadas por autores citados no Capítulo 2, entretanto obtidas sobre um zoneamento da palavra que enfatiza uma não supersegmentação das palavras. Pelo contrário, as mesmas são analisadas em apenas duas regiões, numa espécie de prefixo e sufixo. Para o léxico de palavras pertencentes aos meses do ano, o conjunto de características proposto e aplicado também em Kapp *et al.* [KFNS03], apresenta menor dimensão e resultados muito próximos de Oliveira [dOJ02]. Então gera-se menos parâmetros a serem calculados nas RNAs, diminuindo a complexidade computacional sem comprometer o desem-

penho. Observa-se que a arquitetura classe-modular foi superior em termos de convergência e capacidade de reconhecimento do que a arquitetura convencional, como em [OS02].

Observou-se também que a arquitetura convencional tem uma estrutura rígida composta de uma "caixa preta"na qual todas as classes K são misturadas completamente. Os módulos não podem ser modificados ou aperfeiçoados localmente para cada classe, como pode ocorrer na classe-modular. Porém, as desvantagens da arquitetura classe-modular são primeiramente as reorganizações de treinamento, validação e teste fixados para ajustar cada classe como descrito no Capítulo 3, e o treinamento de K redes para as classes do problema.

Os resultados obtidos motivaram a continuidade do desenvolvimento do método considerando também um mecanismo de rejeição e uma seleção de características. Dois mecanismos de rejeição foram testados e avaliados. O primeiro, baseado na regra de Chow [Cho70] e o segundo na regra dos CRT de Fumera *et al.* em [FRG00]. Ambos mecanismos são aplicados no conjunto de validação e têm suas performances analisadas. Devido ao melhores desempenhos da regra dos CRT na obtenção dos limiares de rejeição em relação a regra de Chow, observados no Capítulo 4, apenas a regra dos CRT é aplicada no conjunto de teste, obtendo-se resultados incentivadores, principalmente para o léxico dos meses. Enfatiza-se, porém que as porcentagens de erro obtidas no conjunto de teste podem ainda ser ajustadas para as mesmas fixadas no conjunto de validação. Entretanto, como o objetivo principal é a análise do comportamento da obtenção dos limiares e sua aplicação em um conjunto de teste desconhecido para o método, verificando as possíveis diferenças e dificuldades que os métodos de reconhecimento encontram em conjuntos diferentes, muito provavelmente pela quantidade de amostras e variabilidade dos estilos de escrita.

Os resultados e gráficos da seleção de característica apresentados no Capítulo 4, mostram a importância da utilização das primitivas estruturais e perceptivas selecionadas no conjunto de características proposto, e apontam para a necessidade da adição de novas características, visto que em geral, as já utilizadas juntamente com um zoneamento de baixa segmentação implícita não demonstraram ser desnecessárias.

Durante o desenvolvimento desta dissertação, não se teve a oportunidade de pesquisar ou testar alguns assuntos devido a constrangimento de tempo ou adquiri-

dos durante a conclusão desta. Aqui esboça-se algumas diretivas futuras que acredita-se serem merecedoras de investigação:

- Aumento das bases de imagens de palavras, principalmente referentes aos valores por extenso, buscando aumentar o nível de reconhecimento de classes que atualmente possuem poucas amostras, onde se obtêm taxas muito baixas, como por exemplo, a classe “dezessete”.

- Adição de novas características, principalmente ligadas a transformadas e a letras de forma numa abordagem local.

- Adicionar no método um mecanismo de segmentação e reconhecimento de uma sentença inteira. Assim, baseando-se em informação de contexto, espera-se aumentar as porcentagens de reconhecimento para o léxico de valores por extenso.

- O zoneamento aplicado nas palavras, de somente esquerda e direita, foi baseado principalmente nas regiões de ascendentes e descendentes das palavras referentes aos meses, e devido à ausência dessas características na região central das palavras. De acordo com o método do presente trabalho, não há necessidade de maiores segmentações para tais características. Porém, um novo trabalho pode ser realizado para estimar novas regiões de zoneamento em palavras que possuem ascendentes e descendentes em mais de duas regiões, como ocorre com o léxico dos valores por extenso, como, por exemplo, as palavras “quatrocentos”e “duzentos”.

Referências Bibliográficas

[AGHG95] A. Agarwal, L. Granowetter, K. Hussein, and A. Gupta. Detection of courtesy amount block on bank checks. In *Internacional Conference in Document Analisys and Recognition*, pages 748–751. IEEE, 1995.

[Avi96] M. Avila. *Optimisation de Modeles Markoviens pour la Reconnaissance de L'escrit*. PhD thesis, Université de Rouen, France, 1996.

[AYV02] Nafiz Arica and Fatos T. Yarman-Vural. Optical character recognition for cursive handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):801–813, June 2002.

[CDIP95] G. Congedo, G. Dimauro, S. Impedovo, and G. A. Pirlo. Structural method with local refining for handwritten character recognition. In *Internacional Conference in Document Analisys and Recognition*, pages 853–856. IEEE, 1995.

[Cho70] C. K. Chow. On optimum error and reject tradeoff. *IEEE Trans. Inform. Theory*, 16(1):41–46, 1970.

[Com03] Compe. Reestruturação do sistema de pagamentos - versão 25.3.2002. www.bcb.gov.br, November 2003.

[Côt97] M. Côté. *Utilisation d'un modèle d'accès lexical et de concepts perceptifs pour la reconnaissance d'images de mots cursifs*. PhD thesis, Ècole Nationale Supérieure des Télécommunications, France, 1997.

[Cyb89] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Contr. Signals Syst.*, 2:303–314, 1989.

[dOJ02] José Josemar de Oliveira Júnior. Avaliação de conjuntos de características no reconhecimento de palavras manuscritas. Master's thesis, Universidade Federal de Campina Grande, Curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campina Grande, 2002.

[FGK95] A. Filatov, A. Gitis, and I. Kil. Graph-based handwritten digit string recognition. In *Internacional Conference in Document Analisys and Recognition*, pages 845–848. IEEE, 1995.

[Fre01] C. O. A. Freitas. *Uso de modelos escondidos de Markov para reconhecimento de palavras manuscritas*. PhD thesis, Pontifícia Universidade Católica do Paraná-Programa de Pós-Graduação em Informática Aplicada, Curitiba, Brasil, 2001.

[FRG00] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Reject option with multiple thresholds. *Pattern Recognition*, 33:2099–2101, March 2000.

[GL01] N. R. Gomes and L. L. Lee. Feature extraction based on fuzzy set theory for handwriting recogntion. In *Sixth International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2001.

[Gui95] D. Guillevic. *Unconstrained handwriting recognition applied to the processing of banks cheques*. PhD thesis, Concordia University, Canada, December 1995.

[GW95] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, New York, 1 edition, 1995.

[Heu94] L. Heutte. *Reconnaissance de caractères manuscrits: application à la lecture automatique des chèques et des enveloppes postales*. PhD thesis, L'Université de Rouen, Rouen, France, December 1994.

[Hor91] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

[HSCP97] C. M. Holt, A. Stewart, M. Clint, and R. H. Perrott. *Algorithms for Image Processing and Computer Vision*. JR Parker- John Wiley & Sons, 1997.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[JdCCFS02] J. J. O. Júnior, J. M. de C. Carvalho, C. O. A. Freitas, and R. Sabourin. Evaluating nn and hmm classifiers for handwritten word recognition. In *Proceedings of XV Brazilian Symposium on Computer Graphics and Image Processing*, pages 210–217. IEEE, 2002.

[JDM00] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, August 2000.

[Jor95] Michael I. Jordan. Why the logistic function? a tutorial discussion on probabilities and neural networks. Technical Report 9503, Massachusetts Institute of Technology, 1995.

[KAB+96] S. Knerr, V. Anisimov, O. Baret, D. E. Price, and J. C. Simon. The a2ia recognition system for handwritten checks. Technical Report 11, A2iA Technical Report, 1996.

[KFNS03] Marcelo N. Kapp, Cinthia O. A. Freitas, Júlio C. Nievola, and Robert Sabourin. Evaluating the conventional and class-modular architectures feedforward neural network for handwritten word recognition. In *Proceedings of XVI Brazilian Symposium on Computer Graphics and Image Processing*, pages 315–319. IEEE Computer Society, 2003.

[Kim96] G. Kim. *Recognition of offline handwritten words and its extension to phrase recognition.* PhD thesis, University of New York at Buffalo, USA, March 1996.

[KJ97] Ron Kohavi and George H. John. Wrappers for feature subset selection. *AIJ special issue on relevance*, pages 1–43, 1997.

[Koh94] Ron Kohavi. Feature subset selection as search with probabilistic estimates. *AAAI Fall Symposium on Relevance*, pages 122–126, 1994.

[LBB+98] Steve Lawrence, Ian Burns, Andrew Back, Ah Chung Tsoi, and C. Lee Giles. Neural network classification and prior class probabilities. *Tricks of the Trade, Lecture Notes in Computer Science State-of-the-Art Surveys*, pages 299–314, 1998.

[LSF02] Cheng-Lin Liu, Hiroshi Sako, and Hiromichi Fukisawa. Performance evaluation of pattern classifiers for handwritten character recognition. *International Journal on Document Analysis and Recognition*, 4:191–204, 2002.

[MG01] Sriganesh Madhvanath and Venu Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):149–164, 2001.

[Mon95] L. Montoliu. *Architecture multi-agents et réseaux connexionnistes. Application à la lecture de chèques manuscrits.* PhD thesis, École Polytechnique, France, 1995.

[MOS+02] M. Morita, L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. An hmm-mlp hybrid system to recognize handwritten dates. In *Proceedings of the International Joint Conference on Neural Networks*, pages 867–872. IEEE Press, 2002.

[MP43] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[MSBS02] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Segmentation and recognition of handwritten dates. In *Proceedings of Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 105–110. IEEE Computer Society Press, 2002.

[MYS+01] M. Morita, A. El Yacoubi, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Handwritten month word recognition on brasilian bank cheques. In *Sixth International Conference on Document Analysis and Recognition*, pages 972–976. IEEE, 2001.

[Oll99] D. Ollivier. *Une approche économisant les traitements pour reconnaître l'écriture manuscrite:application à la reconnaissance des montants littéraux de chèques bancaires*. PhD thesis, Université de Paris XI Orsay, France, 1999.

[OS02] I-S. Oh and C. Y. Suen. A class-modular feedforward neural network for handwriting recognition. *Pattern Recognition*, 35:229–244, 2002.

[Par02] Jaehwa Park. An adaptive approach to offline handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):920–931, July 2002.

[PEL99] José C. Principe, Neil R. Euliano, and W. Curt Lefebvre. *Neural and Adaptive Systems: Fundamentals through Simulations*. John Wiley & Sons, Inc., New York, 1999.

[PS00] Réjean Plamondon and Sargur N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, January 2000.

[RI03] Baranidharan Raman and Thomas R. Ioerger. Enhancing learning using feature and example selection. Master's thesis, Department of Computer Science, Texas A&M University, College Station, Tx, USA, 2003.

[RL91] M. D. Richard and R. Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Comput.*, 3:461–483, 1991.

[Ska94] David B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. *In Eleventh International Machine Learning Conference (ICML-94)*, pages 293–301, 1994.

[SR98] A. W. Senior and A. J. Robinson. An off-line cursive handwritting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, March 1998.

[TJT96] O. D. Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29(4):641–662, 1996.

[TSW90] C. C. Tappert, C. Y. Suen, and T. Wakahara. The state of art in online handwritting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 787–808, August 1990.

[Yac96] A. El Yacoubi. *Modelisation Markovienne de l'écriture manuscrite. Application à la reconnaissance des adresses postales*. PhD thesis, Universite de Rennes, France, 1996.

[Zha00] Guoqiang Peter Zhang. Neural networks for classification: A survey. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 30(4):451–462, November 2000.