

UFRJ

VIVIANE SOARES RODRIGUES SILVA

**UM COMITÊ DE REDES NEURAIS
PARA RECONHECIMENTO DE
CARACTERES MANUSCRITOS**

Dissertação de Mestrado apresentada ao programa de pós-graduação do Núcleo de Computação Eletrônica - Instituto de Matemática da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários a obtenção do título de Mestre em Informática.

RIO DE JANEIRO
2006

S586 SILVA, VIVIANE SOARES RODRIGUES

UM COMITÊ DE REDES NEURAIS PARA RECONHECIMENTO DE CARACTERES MANUSCRITOS / VIVIANE SOARES RODRIGUES SILVA – RIO DE JANEIRO, 2006.

121 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, 2006.

Orientador: Antonio Carlos Gay Thomé.

1. Reconhecimento de Caracteres Manuscritos - Teses. 2. Processamento de Imagem - Teses. 3. Redes Neurais - Teses. I. Antonio Carlos Gay Thomé (Orientador). II Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica. III - Título.

CDD

Viviane Soares Rodrigues Silva

**UM COMITÊ DE REDES NEURAIS PARA RECONHECIMENTO
DE CARACTERES MANUSCRITOS**

Dissertação de Mestrado apresentada ao programa de pós-graduação do Núcleo de Computação Eletrônica - Instituto de Matemática da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários a obtenção do título de Mestre em Informática.

Orientador: Antonio Carlos Gay Thomé

Rio de Janeiro
2006

Viviane Soares Rodrigues Silva

**UM COMITÊ DE REDES NEURAIIS PARA RECONHECIMENTO
DE CARACTERES MANUSCRITOS**

Dissertação de Mestrado apresentada ao programa de pós-graduação do Núcleo de Computação Eletrônica - Instituto de Matemática da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários a obtenção do título de Mestre em Informática.

Aprovado em

Antonio Carlos Gay Thomé - Ph.D. - DCC/IM/UFRJ

José Manoel Seixas - Dsc. - COPPE/UFRJ

Josefino Cabral Melo Lima - Ph.D - DCC/IM/UFRJ

Aos meus amores Anderson e Leticia.

AGRADECIMENTOS

Agradeço de todo coração a Deus que certamente sonhava com os meus dias antes mesmo que eu tivesse nascido. Que não resumiu sua presença em nenhum instante, deixando claro sua torcida por mim.

Minha profunda gratidão ao meu marido. Você sim, sabe muito bem o que foi preciso para chegar até aqui. Obrigada por me encorajar, apoiar, sofrer e comemorar comigo! Eu também agradeço a minha filhinha, que, enquanto eu treinava as redes neurais para distinguirem as letras e os números, ela aprendia a desenhar seu nome e a escrever te amo. Também te amo.

Sou imensamente grata aos meus pais. Batalharam muito para que eu pudesse me dedicar exclusivamente aos estudos e me ensinaram a valorizar e aproveitar as oportunidades. Também agradeço ao meu irmão pela grande força para que eu pudesse retornar a área de informática.

Ao meu sempre orientador Thomé. Desde a participação decisiva para minha conclusão do curso de Bacharelado, a orientação como aluna de mestrado. Já te disse e repito: “Que bom ter suas críticas ao longo do processo. Sei que estará ao meu lado e ficarei segura no momento da defesa!”.

Um obrigada especial para a professora Cíntia Freitas da PUCPR/PPGIA, pela atenção e ajuda a este trabalho.

Agradeço a oportunidade de estudar oferecida pela Capes, UFRJ e NCE.

Aos colegas de trabalho que sempre se mostravam dispostos a me ajudar: Fernando, Eugênio, Rogério, Marcelo, Adriano e Simone. Valeu a força!

Sem dúvida, minha gratidão vai para aqueles que acompanharam todos os bastidores desta empreitada. Aos meus amigos, Ana, Marco, Cremilda, Emília e Raquel. Colaboraram comigo, se interessando e me oferecendo ânimo para eu continuar.

RESUMO

SILVA, Viviane Soares Rodrigues. **Um Comitê de Redes Neurais para Reconhecimento de Caracteres Manuscritos**. Rio de Janeiro, 2006. Dissertação (Mestrado em Informática) - Instituto de Matemática, Universidade Federal do Rio de Janeiro, 2006.

A escrita a mão tem persistido aos grandes avanços tecnológicos para a produção de documentos. A necessidade de leitura automática de cheques bancários, endereçamento de cartas, formulários diversos, por exemplo, tem incentivado pesquisas no campo do reconhecimento inteligente de caracteres manuscritos. Este trabalho traz a proposta de um comitê formado por redes neurais do tipo Multi Layer Perceptron (MLP), para o reconhecimento de caracteres manuscritos. O modelo proposto deve fazer parte de um sistema mais abrangente que visa o processamento automático de formulários de concurso. Foram feitos estudos sobre a diversidade e exatidão entre os agentes componentes do comitê a fim de que fosse possível prever os resultados de cada combinação de agentes experimentada. Os resultados de cada tentativa são avaliados individualmente e em seguida são comparados com uma outra técnica de combinação de agentes (SILVA, 2002), utilizando a mesma massa de dados. Ao final, os resultados obtidos são comparados com a percentagem de acerto quando a classificação é feita por seres humanos.

A Committee of Neural Nets for Recognition of Characters Manuscripts

ABSTRACT

SILVA, Viviane Soares Rodrigues. **A Committee of Neural Nets for Recognition of Characters Manuscripts**. Rio de Janeiro, 2005. Dissertação (Mestrado em Informática) - Instituto de Matemática, Universidade Federal do Rio de Janeiro, 2006.

The handwriting has persisted during the time evolution despite the great technological advances in the document production technology. The research in the field of the intelligent recognition of handwrite characters has been strongly stimulated by the necessity of several comercial applications such as bank and post office automation. In particular, for the identification of the information contained in printed forms, a system named NEUROTERTO, has being developed by the LABIC/NCE since the year of 2000. The NEUROTERTO uses a Teams of Neural Nets as recognition technique. In order to enlarge the research on alternative ways to use of Neural Nets in such applications, and to establish a comparison with the technique of the Teams of Neural Nets, this work proposes the use of a committee formed for neural nets of the type of Multi Layer Perceptron (MLP). The obtained results are analyzed and compared with those provided by the team.

LISTA DE FIGURAS

Figura 1.1: Abordagens dos Sistemas de Reconhecimento de Caracteres.	20
Figura 1.2: Exemplos de Caracteres Impressos e Manuscritos	21
Figura 1.3: Caracteres Vistos de Modo Isolado	22
Figura 1.4: Caracteres Vistos Dentro de um Contexto	23
Figura 2.1: Sistema NeuroTexto	27
Figura 2.2: (a)Campo Extraído do Formulário (b)Caracteres Segmentados	30
Figura 2.3: (a)círculo circunscrito à imagem (b)polígono circunscrito ao círculo. Adaptado de (SILVA, 2002)	31
Figura 2.4: Seqüência de Projeções para o Quadrado Rotacionado. Adaptado de (SILVA, 2002)	32
Figura 2.5: Modelos Deformáveis. (a)Deformações do Modelo de Dígitos (b)Modelo Deformado Superposto a Imagem Alvo. Adaptado de (ARICA & VU- RAL, 2001)	33
Figura 2.6: HMM Modelo-Discriminante de cinco estágios do caractere “d”. Adap- tado de (ARICA & VURAL, 2001)	35
Figura 3.1: Exemplo de Arquitetura em Série do Tipo Sequencial	39
Figura 3.2: Exemplo de Arquitetura em Paralelo	40
Figura 3.3: Exemplo de Arquitetura em Série do Tipo Sequencial: Time de Redes. Adaptado de (SILVA, 2002)	42
Figura 3.4: Arquitetura de uma Combinação de Técnicas de Reconhecimento Classe- Modular	44
Figura 3.5: Arquitetura de uma Combinação de Técnicas de Reconhecimento do Tipo Seletiva e Heterogênea	45

Figura 3.6: Estruturas Estáticas: rede do tipo Adaline	50
Figura 3.7: Diagrama em blocos do modelo: Comitê com Estrutura Dinâmica . . .	51
Figura 3.8: (a)Grafo arquitetural da rede de passagem.(b)Grafo do fluxo de sinal do neurônio k	52
Figura 3.9: Modelo de Comitê Hierárquico com dois níveis de hierarquia.	53
Figura 4.1: Modelo de um Comitê Dinâmico composto por Redes Neurais do tipo MLP	56
Figura 4.2: Representação de um Nodo Não-Linear da Rede de Passagem	56
Figura 4.3: Diagrama Esquemático dos Nodos da Rede de Passagem com uma Camada Intermediária	59
Figura 5.1: Modelo do Comitê para Classificação de Dígitos - Primeira Tentativa .	69
Figura 5.2: Modelo do Comitê para Classificação de Dígitos - Segunda Tentativa .	76
Figura 5.3: Dígitos de Classes Diferentes Próximos Espacialmente	79
Figura 5.4: Dígitos Semelhantes.	80
Figura 5.5: Dígitos Agrupados como Mais Semelhantes.	84
Figura 5.6: Exemplo de Letras Agrupadas: (a)longelíneas (b)arredondadas	91
Figura 5.7: Modelo da Página de Coleta da Avaliação Humana	97
Figura 7.1: Matriz de Confusão da Rede Neural Selecionada - Letras	117
Figura 7.2: Matriz de Confusão da Rede Neural Selecionada Após PCA - Letras .	120

LISTA DE TABELAS

Tabela 3.1: MT - Matriz de Targets	48
Tabela 3.2: MA_a - Matriz de Confusão Gerada pelo Agente “a”	49
Tabela 5.1: Grupos de Dígitos	67
Tabela 5.2: Resultados dos Agentes Individuais	68
Tabela 5.3: Características do Treinamento da Rede de Passagem no Comitê Preliminar - Dígitos	69
Tabela 5.4: Resultado nos Testes da Primeira Tentativa para Classificação dos Dígitos	70
Tabela 5.5: Matriz de Probabilidade de Discordância	71
Tabela 5.6: Discordância Através das Matrizes de Confusão	71
Tabela 5.7: Matriz de Confusão do “Agente 1”	72
Tabela 5.8: Quantidade de Dígitos Representantes das Dez Classes em Cada Conjunto	73
Tabela 5.9: Matriz de Confusão do Agente Especialista nos Dígitos 0 e 8	74
Tabela 5.10: Matriz de Confusão do Agente Especialista nos Dígitos 2 e 7	74
Tabela 5.11: Desempenho dos Agentes - Segunda Tentativa	75
Tabela 5.12: Matriz de Probabilidade de Discordância	75
Tabela 5.13: Discordância Através das Matrizes de Confusão	75
Tabela 5.14: Características do Treinamento da Rede De Passagem no Comitê - Segunda Tentativa	77
Tabela 5.15: Resultado nos Testes da Segunda Tentativa para Classificação dos Dígitos	78
Tabela 5.16: Quantidade de Dígitos Representantes das Dez Classes em Cada Conjunto	80
Tabela 5.17: Desempenho dos Agentes - Terceira Tentativa com Dois Grupos	81
Tabela 5.18: Matriz de Probabilidade de Discordância	81

Tabela 5.19:Discordância Através das Matrizes de Confusão	82
Tabela 5.20:Resultado nos Testes da Terceira Tentativa com Dois Grupos	83
Tabela 5.21:Quantidade de Dígitos Representantes das Dez Classes em Cada Con- junto	84
Tabela 5.22:Matriz de Probabilidade de Discordância	85
Tabela 5.23:Discordância Através das Matrizes de Confusão	85
Tabela 5.24:Resultado nos Testes da Terceira Tentativa com Três Grupos	87
Tabela 5.25:Grupos de Letras	88
Tabela 5.26:Resultados dos Agentes Individuais	88
Tabela 5.27:Características do Treinamento da Rede De Passagem no Comitê Pre- liminar - Letras	89
Tabela 5.28:Matriz de Probabilidade de Discordância	90
Tabela 5.29:Discordância Através das Matrizes de Confusão	90
Tabela 5.30:Número de Letras em Cada Grupo Particionado	92
Tabela 5.31:Matriz de Probabilidade de Discordância	93
Tabela 5.32:Discordância Através das Matrizes de Confusão	93
Tabela 5.33:Resultado Final nos Testes dos Agentes para Classificação das Letras .	95
Tabela 5.34:Resultado Final nos Testes dos Comitês para Classificação das Letras .	95
Tabela 5.35:Tabela Comparativa entre Desempenho do Ser Humano e Modelos de Comitê	98
Tabela 6.1: Tabela Comparativa entre Desempenho do Time de Redes e Modelos de Comitê - Dígitos	102
Tabela 6.2: Tabela Comparativa entre Desempenho do Time de Redes e Modelos de Comitê - Letras	103
Tabela 7.1: Quantidade de dígitos para o treinamento, teste e validação das Redes Neurais	110
Tabela 7.2: Parâmetros de Treinamento das Redes Neurais	111
Tabela 7.3: Resultados da Melhor Rede Neural Obtida - Rede nº11	111
Tabela 7.4: Matriz de Confusão Resultante da Rede Neural Selecionada - Rede nº11111	
Tabela 7.5: Na tabela a) as variáveis mais relevantes e, em b) as variáveis menos relevantes no desempenho da rede neural - Dígitos	112

Tabela 7.6: Taxas de Acerto Relativas ao Fator de Corte Utilizado - Dígitos	112
Tabela 7.7: (a) as variáveis menos relevantes e, (b) as variáveis mais relevantes no resultado da análise de sensibilidade após aplicação de PCA - Dígitos.	113
Tabela 7.8: Resultados da Melhor Rede Neural Obtida Após PCA - Dígitos .	114
Tabela 7.9: Matriz de Confusão Resultante da Melhor Rede Neural Após PCA - Dígitos	114
Tabela 7.10: Quantidade de Letras para o Treinamento, Teste e Validação das Redes Neurais	115
Tabela 7.11: Resultados no Cálculo do Número de Neurônios na Camada Es- condida - Letras	115
Tabela 7.12: Resultados da Melhor Rede Neural Obtida - Letras	116
Tabela 7.13: (a) as variáveis menos relevantes e, (b) as variáveis mais relevantes no resultado da análise de sensibilidade - Letras.	118
Tabela 7.14: Taxas de Acerto Relativas ao Fator de Corte Utilizado - Letras	118
Tabela 7.15: (a) as variáveis menos relevantes e, (b) as variáveis mais relevantes no resultado da análise de sensibilidade após aplicação de PCA - Letras. .	119
Tabela 7.16: Resultados da Melhor Rede Neural Obtida Após PCA - Letras . .	120
Tabela 7.17: Resultados sobre Taxa de Aprendizagem e Momento	121

LISTA DE ABREVIATURAS E SIGLAS

MSC	Models System Classifiers
MLP	Multi Layer Perceptron
PCA	Principal Component Analysis

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Sistemas de Reconhecimento de Caracteres	18
1.1.1	Histórico	20
1.2	Motivações	22
1.3	Proposta de Pesquisa	24
1.4	Organização da Dissertação	25
2	ETAPAS DO PROCESSO DE RECONHECIMENTO DE CARACTERES MANUSCRITOS	26
2.1	Pré-processamento	27
2.2	Segmentação	29
2.3	Representação e Extração de Características	30
2.4	Reconhecimento e Classificação	32
2.4.1	Técnicas de Reconhecimento	32
2.5	Pós-Processamento	36
3	COMBINAÇÕES DE TÉCNICAS DE RECONHECIMENTO	38
3.1	Métodos de Combinação de Agentes	39
3.1.1	Exemplos de Técnicas Combinadas	42
3.2	Medidas de Eficácia Numa Combinação de Técnicas	44
3.2.1	Relação entre Diversidade e Exatidão	45
3.2.2	Medidas de Diversidade	47
3.2.3	Matriz de Probabilidade de Discordância	47
3.2.4	Discordâncias Através das Matrizes de Confusão	48

3.3	Combinação de Classificadores em Comitês	49
3.3.1	Comitê de Agentes Especialistas	50
3.3.2	Comitê Hierárquico de Agentes Especialistas	53
4	PROPOSTA DE UM COMITÊ DE AGENTES NEURAIIS	55
4.1	Formulação Teórica para o Modelo	55
4.1.1	Comitê Dinâmico com Uma Camada: Não Linear	55
4.1.2	Algoritmo de Treinamento	58
4.1.3	Comitê Dinâmico com Duas Camadas: Não Linear	59
4.1.4	Algoritmo de Treinamento	61
4.1.5	Comitê Dinâmico com Duas Camadas: Linear	62
4.1.6	Algoritmo de Treinamento	63
5	ESTUDO DE CASOS E AVALIAÇÃO DOS RESULTADOS	65
5.1	Experimentos com Dígitos Manuscritos	67
5.1.1	Treinamento dos Agentes - Primeira Tentativa	68
5.1.2	Treinamento dos Agentes - Segunda Tentativa	72
5.1.3	Treinamento dos Agentes - Terceira Tentativa	78
5.2	Experimentos com Letras Manuscritas	87
5.2.1	Treinamento dos Agentes - Primeira Tentativa	88
5.2.2	Treinamento dos Agentes - Segunda Tentativa	91
5.3	Máquina <i>versus</i> Ser Humano	96
6	CONCLUSÃO	99
6.1	Trabalhos Futuros	104
	REFERÊNCIAS	105
7	APÊNDICE - EXPERIMENTOS COM REDES MLP	109
7.1	Introdução	109
7.2	Experimentos para Reconhecimento de Dígitos Manuscritos	110
7.2.1	Descrição dos Experimentos e Resultados	110
7.2.2	Experimento com Dígitos Manuscritos - Conclusões	113

7.3	Experimento para Reconhecimento de Letras Manuscritas	115
7.3.1	Descrição dos Experimentos e Resultados	115
7.3.2	Experimento com Letras Manuscritas - Conclusões	117
7.4	Experimento Complementar	118

1 INTRODUÇÃO

A concepção de máquinas que simulem as funções humanas é tema em diversas áreas de pesquisas. Simular a função humana de ler um texto é uma destas áreas e tem sido um grande desafio para os pesquisadores nas últimas décadas.

Os autores em (PLAMONDON & SRIHARI, 2000) definem três características principais sobre a linguagem escrita: são marcas gráficas artificiais sobre uma superfície, têm a meta de comunicar algo e esta meta é atingida quando há uma relação convencional entre as marcas numa linguagem sendo utilizada. Sendo assim, para a realização de uma leitura automática de um texto, é necessário que a imagem dos traços que transmitem a informação desejada seja transcrita para um formato armazenável em um computador. A partir daí, deve-se submeter tal imagem capturada a uma seqüência de técnicas a fim de que, ao final, seja possível armazenar não uma seqüência de traços e sim o significado que possui dentro da linguagem.

Em geral, funções que fazem o processamento de imagens digitais, visando o reconhecimento de padrões, estão associadas à extração automática de informação a partir da imagem. Um exemplo disso são os Sistemas de Reconhecimento de Caracteres (GONZALES & WOODS, 2000).

1.1 Sistemas de Reconhecimento de Caracteres

Um sistema para reconhecimento de um caractere pode estar associado a diferentes domínios de aplicação. As emergentes áreas que fomentam as pesquisas e o desenvolvimento de produtos são bibliotecas eletrônicas, bancos de dados multimídia, e sistemas que são alimentados com imagens de textos manuscritos, como, por exemplo, os códigos postais, cheques bancários e formulários diversos (ARICA &

VURAL, 2001).

Os sistemas de reconhecimento de caracteres podem ser classificados sob dois critérios:

- Como a imagem do caractere é adquirida e processada,
- Que tipo de texto está sendo avaliado.

Para o primeiro critério existem dois tipos de processo de aquisição: *on-line* e *off-line*. Em aquisições do tipo *on-line*, a obtenção de dados é feita escrevendo-se com o uso de uma caneta especial numa superfície eletrônica, que trabalha como um digitalizador combinado a um display de cristal líquido, a exemplo disso tem-se os *palm tops*. A medida em que o caractere vai sendo escrito é imediatamente apresentado ao sistema para ser processado e reconhecido. Já no processo *off-line*, o dado de entrada é convertido para forma digital através da captura da imagem em um papel através de um leitor (*scanner*). Portanto, o processamento é feito após sua imagem ter sido completamente construída (PLAMONDON & SRIHARI, 2000).

O segundo modo de classificar os sistemas de reconhecimento está relacionado com a maneira com a qual o texto foi produzido, se foi impresso ou manuscrito. Um texto manuscrito pode ser classificado quanto à apresentação dos caracteres. Palavras ou sentenças numéricas podem ser compostas por caracteres isolados, contínuos (cursivos, de forma conectada), na forma cursiva discreta (grupo de caracteres escritos com um único movimento contínuo), ou a combinação destes (SUEN *et al*, 2000). Deve-se acrescentar que a identificação de um caractere na forma manuscrita traz consigo uma complexidade maior devido às variedades de formas de escrita relativa a cada pessoa.

A figura 1.1 mostra uma distribuição hierárquica da classificação dos sistemas de reconhecimento de caracteres.

Nota-se que os sistemas de reconhecimento *on-line* aplicam-se somente a caracteres manuscritos enquanto que os *off-line* subdividem-se tanto em caracteres impressos quanto em manuscritos. Estes últimos apresentam-se de forma isolada ou cursiva.

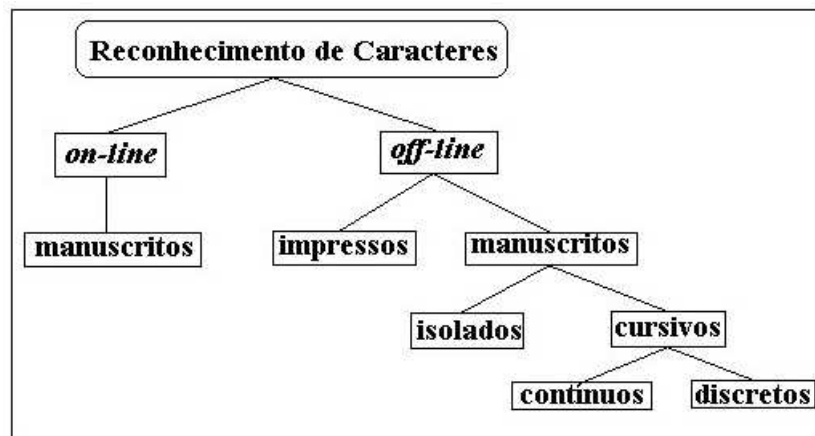


Figura 1.1: Abordagens dos Sistemas de Reconhecimento de Caracteres.

1.1.1 Histórico

A história do reconhecimento de caracteres pode ser dividida em três gerações: *A primeira, desde 1900 até o início dos anos 80*, foi iniciada pelas pesquisas do cientista russo Turing (ARICA & VURAL, 2001), com estudos voltados para o auxílio a deficientes visuais. Os primeiros sistemas para reconhecimentos de caracteres apareceram em meados dos anos 40, com o desenvolvimento do computador digital.

Em 1951 M. Sheppard, fundador da Intelligent Machine Research Co., criou o GISMO, um robô leitor-escritor (VERMA *et al*, 1998). Em 1954, um protótipo foi utilizado para ler letras maiúsculas datilografadas cuja velocidade era de um caractere por minuto (SRIHARI & LAM, 2006).

Textos manuscritos, num escopo bem limitado, recebiam alguma atenção dos pesquisadores. Conhecidos como sistema para o Reconhecimento inteligente de Caracteres (ICR), tiveram suas primeiras conquistas também em meados dos anos 50, com o reconhecimento de caracteres escritos numa tela eletrônica através da captura das coordenadas x-y do movimento de uma caneta especial.

Até o final dos anos 60 ocorreram muitos avanços, porém os sistemas de reconhecimento de caracteres ainda eram muito caros e só podiam ser adquiridos por grandes empresas privadas ou estatais.

Os sistemas comercialmente viáveis eram basicamente voltados para textos impressos, conhecidos como Sistemas de Reconhecimento de Caracteres Óticos (OCR), que, a princípio, eram capazes de reconhecer apenas tipos específicos de fontes.

A figura 1.2 traz exemplos de caracteres impressos e os mesmos escritos a mão.

Os sistemas OCR possuem a vantagem de avaliarem caracteres com uma forma pré-definida. Já os sistemas ICR, além de outros problemas relativos ao reconhecimento de padrões em geral, lidam com as diversas maneiras que um mesmo caractere pode ser escrito.



Figura 1.2: Exemplos de Caracteres Impressos e Manuscritos

Somente a partir de 1967 é que grandes companhias começaram a produzir e comercializar estes sistemas. Dentre elas estão: IBM, Recognition Equipment, Inc., Farrington, Control Data e Optical Scanning Corporation (SRIHARI & LAM, 2006).

*A segunda geração, a década de 80, beneficiada pelos avanços tecnológicos, foi marcada por experimentos que focalizavam na forma do caractere sem uma preocupação com a semântica envolvida. A precisão de tais sistemas, para caracteres manuscritos, ainda era baixa, não sendo suficiente para a comercialização. Sobre o estado da arte dos sistemas reconhecedores destes caracteres até o início desta década, pode-se encontrar uma boa referência em (SUEN *et al*, 1980).*

Somente de 1990 para os dias de hoje é que um real progresso foi visto. Uma combinação eficiente foi feita entre técnicas de processamento de imagem em reconhecimento de padrões e inteligência computacional. Algoritmos mais complexos foram implementados em máquinas de maior poder computacional.

Os produtos mais recentes, para o reconhecimento *off-line* de caracteres impressos e os sistemas *on-line* de caracteres manuscritos, para um vocabulário restrito, têm apresentado taxas de precisão satisfatórias em algumas aplicações. Ainda há um longo caminho a percorrer até alcance do desempenho de leitura humano, principalmente para os sistemas *on-line* sem restrições e para o reconhecimento *off-line* de manuscritos.

1.2 Motivações

A razão para a persistência de manuscritos numa era digital é a conveniência do papel e da caneta para as diversas situações do dia-a-dia. A automatização da leitura dos formulários para concursos, por exemplo, além de acelerar o processo de cadastramento dos candidatos, contorna de alguma forma a falta de acesso aos meios digitais, ainda uma realidade para um grande número de brasileiros. Porém, de modo geral, a digitalização e a extração de informações de um documento de forma automática possibilitam o acesso ao seu conteúdo de maneira rápida e objetiva, mantendo fidelidade aos documentos originais.

Atualmente existem variados sistemas que realizam o reconhecimento *on-line* de textos manuscritos. Mas, para o reconhecimento *off-line*, muitas pesquisas têm sido feitas com o propósito de gerar produtos comercialmente eficazes. Com o objetivo de desenvolver algoritmos inovadores a fim de alcançar resultados próximos aos dos humanos no reconhecimento de textos manuscritos e, por existirem inúmeras aplicações no processamento deste tipo de dado, vários pesquisadores têm se empenhado neste sentido (SUEN *et al*, 2000).

Embora mais simples que os caracteres manuscritos cursivos, existem problemas intrínsecos ao processo de reconhecimento de um caractere isolado manuscrito num sistema *off-line*, como por exemplo: ausência de contexto (pelo fato de avaliar-se o caractere de modo isolado, fora da palavra que o contém), qualidade da imagem (nível de ruído), escolha de métodos não tão adequados para a execução das etapas que antecedem a classificação, como por exemplo, o método mais adequado para a extração de descritores.

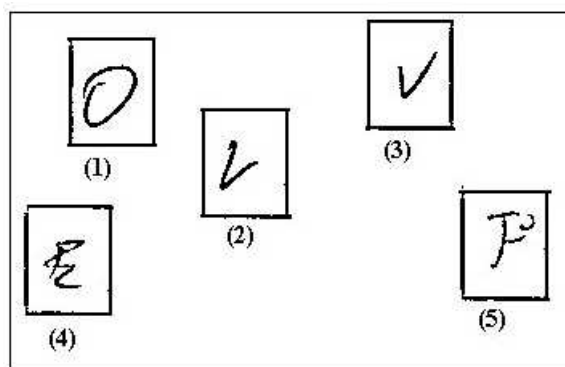


Figura 1.3: Caracteres Vistos de Modo Isolado

As figuras 1.3 e 1.4 representam respectivamente caracteres vistos fora da palavra da qual fazem parte e a palavra que os contém. A figura 1.4 contém índices numéricos que fazem referência aos caracteres mostrados na figura anterior. É perceptível o quanto a falta de contexto dificulta a distinção entre caracteres com grafias parecidas e também, dependendo da maneira como foi escrito, um caractere pode tornar-se irreconhecível quando analisado isoladamente.

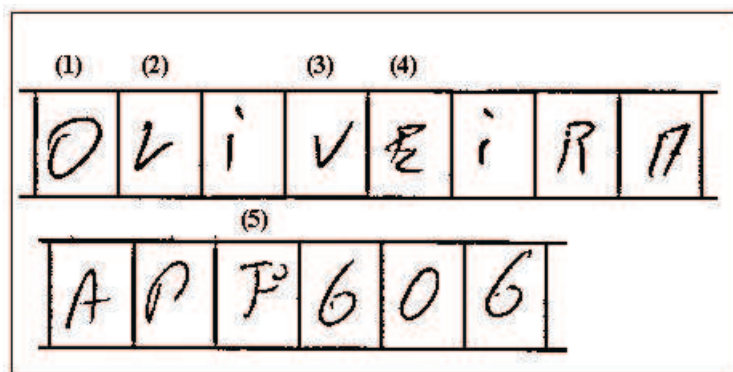


Figura 1.4: Caracteres Vistos Dentro de um Contexto

Numa revisão da literatura sobre as técnicas comumente utilizadas para o reconhecimento off-line de caracteres manuscritos, (HANMANDLU *et al*, 2003), (AIRES, 2003), (MORITA *et al*, 2002), (OH & SUEN, 2002) entre outros, tais autores e suas respectivas propostas apostam numa combinação destas técnicas. Apoiados no fato de que, de acordo com a complexidade do problema em questão, o emprego de uma técnica isolada não é capaz de resolvê-lo completamente.

O projeto NeuroTexto, que vem sendo desenvolvido desde o ano de 2000 pela equipe do Labic (Laboratório de Inteligência Computacional NCE/UFRJ), cujo objetivo é o desenvolvimento de um sistema capaz de reconhecer caracteres (dígitos e letras maiúsculas) provenientes de formulários para concurso, é um exemplo de uso de tecnologias de processamento de imagens para uma aplicação prática de reconhecimento automático de caracteres.

O sistema é composto basicamente por sete etapas:

1. **Aquisição da imagem do formulário:** armazenamento da imagem em modo digital.

2. **Extração das regiões de interesse:** definição dos campos do formulário que serão recortados e enviados ao próximo módulo.
3. **Tratamento e segmentação:** a qualidade da imagem do campo é melhorada e segmentada para a obtenção dos caracteres de modo isolado.
4. **Extração e Codificação das Características:** o caractere contido na imagem é representado numericamente.
5. **Reconhecimento:** utiliza a representação numérica do caractere para associá-lo a classe a qual pertence, para os dígitos 10 classes possíveis e para as letras 26 classes.
6. **Validação:** ação de um algoritmo de validação sobre o resultado da classificação.
7. **Apresentação:** realiza-se uma verificação visual dos formulários como um todo após feita a identificação caractere a caractere.

O módulo de classificação do sistema NeuroTexto conta hoje com um Time de Redes Neurais (SILVA, 2002), que consiste no uso de uma combinação de redes neurais artificiais formando um time de redes.

1.3 Proposta de Pesquisa

Este trabalho tem por objetivo estudar, conceber, implementar e avaliar o protótipo de um classificador baseado num comitê formado por redes neurais. Diferente de um time, que possui uma estrutura estática, o comitê proposto apresenta uma estrutura dinâmica, ou seja, sua construção é modelada com base nos dados de entrada.

Objetiva-se ainda, além do alcance de um bom desempenho para as taxas de acerto no reconhecimento de caracteres manuscritos, a realização de uma comparação entre o modelo em (SILVA, 2002) e o modelo proposto neste trabalho, utilizando-se, para isto, um mesmo banco de imagens.

1.4 Organização da Dissertação

No próximo capítulo, serão abordadas as etapas de um sistema de reconhecimento de padrões em imagens de caracteres, com ênfase para a descrição das técnicas empregadas pelo sistema NeuroTexto. Ainda no capítulo 2, é feita uma revisão sobre as técnicas atualmente empregadas no reconhecimento.

No capítulo 3 são discutidas algumas abordagens para a implementação de uma combinação das técnicas de reconhecimento e exemplos encontrados na literatura. Há ainda uma discussão sobre o uso de métricas que auxiliem na escolha *a priori* dos agentes que melhor contribuiriam para o aumento do desempenho global quando combinados.

Finalmente, o capítulo 3 traz os conceitos sobre combinações de técnicas em comitê.

Como parte do processo de formulação da proposta de uma estratégia para o caso do reconhecimento e classificação de caracteres manuscritos contidos em formulários para concurso, o capítulo 4 é dedicado a descrição teórica e formulação do algoritmo sobre as quais foram realizados os estudos de casos apresentados no capítulo 5.

O capítulo 5 apresenta os experimentos e resultados no emprego de diferentes composições de comitê para o problema de reconhecimento de dígitos e letras manuscritos. É apresentada uma análise da diversidade entre os agentes que compõem o comitê com o uso de duas métricas distintas.

Ao final do capítulo 5 tem-se uma comparação feita entre o desempenho das Máquinas de Comitê criadas e o desempenho da rotulação feita por seres humanos avaliando um mesmo conjunto de caracteres de modo isolado.

O capítulo 6 traz a conclusão desta dissertação: uma avaliação sobre a proposta apresentada e a indicação de trabalhos futuros.

No Apêndice, tem-se a descrição dos experimentos realizados com redes neurais do tipo MLP (Multi-Layer-Perceptron), utilizando a massa de dados selecionada para o treinamento e testes nas implementações citadas neste trabalho.

2 ETAPAS DO PROCESSO DE RECONHECIMENTO DE CARACTERES MANUSCRITOS

Independentemente da abordagem do problema, em geral existem cinco estágios pelos quais os sistemas de reconhecimento de caracteres devem passar: pré-processamento, segmentação, extração de características, reconhecimento e pós-processamento (ARICA & VURAL, 2001).

As fases que antecedem o reconhecimento, propriamente dito, devem ser capazes de isolar o caractere e “captar” informações sobre a sua essência, a fim de descrevê-lo e diferenciá-lo dos demais, reunindo assim um conjunto de características num formato apropriado para ser utilizado na etapa seguinte.

Estudos anteriores (ROSSETO, 2001), (RODRIGUES, 2003), (SILVA, 2002), (SUEN *et al*, 2000), (VIANNA, 2000), (BARBOZA, 2002), entre outros, trazem mais detalhadamente as diferentes técnicas e aplicações referentes a cada um dos estágios que compõem todo o processo de reconhecimento quando se trata da identificação de um caractere.

Como já mencionado, a composição do sistema NeuroTexto será utilizada para ilustrar estas etapas. A figura 2.1 sintetiza este processo.

O processo tem início com a captura (digitalização) dos formulários e a extração das regiões de interesse, isto é, dos diferentes campos que compõem o formulário. Em seguida é realizada a segmentação dos caracteres que compõem os diversos campos e, finalmente, a extração das características e codificação dos vetores a serem apresentados ao módulo de reconhecimento (SILVA, 2002). As fases seguintes são a apresentação do resultado do reconhecimento ao usuário do sistema e a permissão para a interação deste usuário validando tal resultado, corrigindo-o ou aprovando-o.

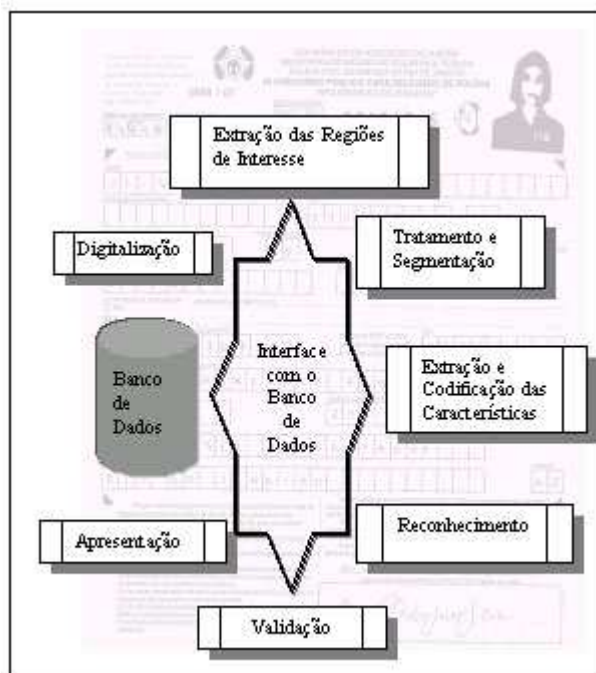


Figura 2.1: Sistema NeuroTexto

Neste capítulo, apresentam-se as etapas que compõem um sistema de reconhecimento de caracteres, porém o foco deste estudo são as técnicas utilizadas para o treinamento e reconhecimento.

2.1 Pré-processamento

Na fase de pré-processamento, a imagem do caractere é manipulada de modo que as demais fases possam operar com dados mais precisos. Os principais objetivos deste processo são a redução do ruído, a normalização do dado e a compressão da informação (ARICA & VURAL, 2001).

- **Redução do Ruído:** O ruído pode ser introduzido pelo instrumento utilizado para a captura da imagem, pela baixa qualidade do papel ou da tinta, ou até por erros no momento da escrita. Deve ser tratado de acordo com seu aspecto. Uma técnica conhecida é o uso de filtros, no domínio do espaço ou da frequência, que contribuem para a diminuição de pontos espúrios, realçando a imagem, suavizando, refinando e removendo levemente texturas ou fundos coloridos. Operações morfológicas também podem ser utilizadas a fim de suavizar os contornos do caractere, afiná-los ou definir suas bordas.

- **Normalização:** Os métodos de normalização auxiliam a remover variações na escrita e a obter uma padronização para a imagem do caractere. O estilo de escrita pode, por exemplo, ser percebido de acordo com o ângulo de inclinação dos caracteres com relação a vertical. Neste caso, a normalização de inclinação é utilizada para padronizar a direção dos caracteres. Outro processo de normalização também pode ser feito para uniformizar o tamanho dos caracteres.

- **Compressão de imagem:** a compressão para o reconhecimento de caracteres requer técnicas no domínio do espaço para que a informação da forma do caractere seja preservada. Duas técnicas conhecidas são a limiarização (thresholding) e o afinamento.
 - **Limiarização:** tem como meta reduzir a quantidade de informação a ser armazenada e aumentar a velocidade de processamento, e é geralmente aplicada na conversão de imagens em tons de cinza ou colorida para imagens binárias. O histograma dos níveis de cinza de uma imagem é constituído de dois picos, o maior corresponde à quantidade de branco (fundo) e o menor à quantidade de preto (tinta). Com isso, esta tarefa constitui-se em determinar o valor limite (acima do qual todos os pixels recebem valor referente ao branco e abaixo do qual os pixels recebem o valor correspondente ao preto) na busca em determinar este valor ótimo dentro do vale situado entre os dois picos (PLAMONDON & SRIHARI, 2000). Este é o processo de uma limiarização global. Uma limiarização local, ou adaptativa, usa diferentes valores para cada pixel de acordo com uma área local de informação.

 - **Afinamento:** provocam uma grande redução na quantidade de dado armazenado e extraem informação sobre a forma do caractere. O afinamento pode ser considerado a simulação de um traço, avaliado num sistema off-line, como se tivesse sido escrito num sistema on-line. As técnicas de afinamento são muito sensíveis ao ruído e podem deformar o caractere, por estes motivos devem ser aplicadas com cautela.

Na fase de captura da imagem para o sistema NeuroTexto utilizou-se um *scanner* simples. E, finalizado o processo de captura e a remoção das quadrículas de interesse, um pré-processamento da imagem do formulário foi feito, convertendo-a para tons de cinza e em seguida binarizando-a.

2.2 Segmentação

Os formulários capturados na fase anterior ficam armazenados numa base de dados e são compostos por vários campos de preenchimento como nome, endereço, telefone, CEP e outros. Estes campos podem ser numéricos, alfabéticos ou ainda alfanuméricos como é o caso do campo endereço.

Neste estágio a imagem do documento é separada em sub-componentes. A importância desta fase está no fato de que a separação de linhas, palavras, e caracteres, afetam diretamente as taxas de acerto obtidas pelo módulo de classificação. Existem dois tipos de segmentação: uma externa, onde são isoladas unidades de escritas tais como parágrafos, sentenças ou palavras, outra interna, onde são separados caracteres. No caso dos formulários, campos de interesse são recortados do corpo da imagem (segmentação externa) e, num segundo passo (segmentação interna), os caracteres contidos nos campos selecionados são separados um a um (ARICA & VURAL, 2001).

O Sistema NeuroTexto utiliza um método de segmentação baseado em histogramas de projeção (RODRIGUES & THOMÉ, 2000). O histograma de projeção é uma técnica que faz uso da projeção da imagem sobre cada uma das duas dimensões existentes. A cada dimensão é associado um vetor onde é armazenado o número de pixels com tonalidade de cinza acima de um certo limite (geralmente tomado como a cor de fundo). A segmentação é então realizada aplicando-se uma série de refinamentos sucessivos até se chegar a um resultado satisfatório. Três etapas distintas podem compor o emprego deste método: primeiramente faz-se uma compensação da qualidade da imagem para aprimorá-la, reduzindo ou realçando determinados detalhes como ruídos ou contrastes. Em seguida aplica-se uma segmentação, com base nas informações armazenadas nos vetores do histograma de projeção, e então, se for necessário, faz-se um conjunto de sucessivos refinamentos (RODRIGUES, 2003). A figura 2.2 mostra um campo de formulário após a utilização do método de

segmentação.

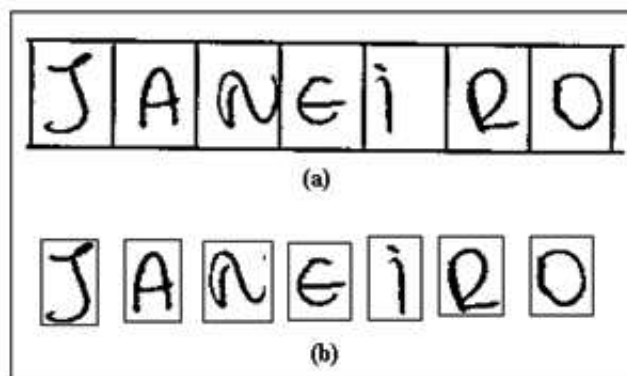


Figura 2.2: (a) Campo Extraído do Formulário (b) Caracteres Segmentados

2.3 Representação e Extração de Características

A representação das características da imagem de um caractere está altamente relacionada ao sistema de reconhecimento adotado, pois é nesta etapa que acontece a transformação de um traço gráfico em dados numéricos e/ou simbólicos que possam ser analisados a fim de classificá-lo. Para que isto aconteça, um conjunto de características é associado a cada classe, de modo que tais características carreguem informação suficiente sobre o caractere a fim de distingui-lo de outro. Vários artigos podem ser encontrados na literatura sobre este assunto, e para um bom apanhado geral cita-se (TRIER *et al*, 1996).

Segundo (ARICA & VURAL, 2001), os métodos de extração e representação de características podem ser classificados de acordo com as seguintes técnicas:

- **Transformação Global e Expansão em Séries:** deformações como translação e rotação são invariantes quando este tipo de técnica é aplicada. Algumas destas técnicas mais conhecidas são: Transformada de Fourier, Transformada de Gabour, Transformada de Wavelets e Momentos.
- **Representação Estatística:** este tipo de representação não permite a reconstrução da imagem original, porém provê uma redução de dimensão do conjunto de características. As técnicas a seguir são utilizadas visando esta vantagem: Zoneamento, Cruzamento, Distâncias e Projeções.

- **Representação Geométrica e Topológica:** capaz de representar várias propriedades do caractere tolerando distorções e variações de estilo. Várias representações deste tipo são agrupadas nas seguintes categorias: Extração e Contagem de Estruturas Topológicas, Propriedades Geométricas Medidas e Aproximadas, Coding, e Árvores e Grafos.

No sistema NeuroTexto, os vetores de características se enquadram na terceira técnica e são baseados na computação de um conjunto de distâncias tiradas a partir do contorno da imagem até um polígono de referência. O polígono é posicionado em torno da imagem do caractere, como mostrado na figura 2.3. Para maiores detalhes sobre esta técnica ver (RODRIGUES, 2003).

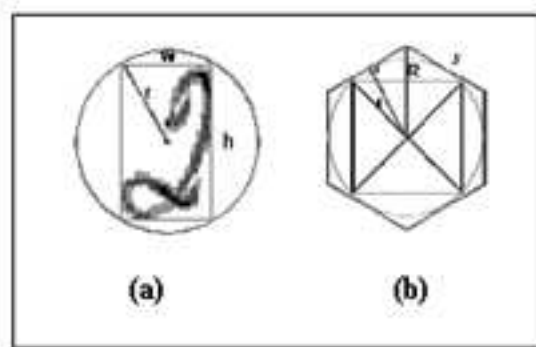


Figura 2.3: (a) círculo circunscrito à imagem (b) polígono circunscrito ao círculo. Adaptado de (SILVA, 2002)

O processo básico computa várias distâncias tomadas a partir de pontos regularmente espaçados de cada lado do polígono até o contorno da imagem e as armazena como características em um vetor de dimensão igual ao produto do número de lados do polígono adotado pelo número de pontos extraídos de cada lado. Para os experimentos realizados em (SILVA, 2002) e adotado como método de composição do vetor de características para o sistema NeuroTexto, foi acrescentada uma variante ao método de projeção poligonal utilizando o quadrado como polígono base. Com o quadrado rotacionado de 45° de sua posição original tornou-se possível considerar detalhes dos caracteres que não eram considerados com apenas o método convencional. A figura 2.4 exemplifica as varreduras realizadas. Com esta modificação, o vetor de características passou a ser composto pelas características extraídas segundo o quadrado convencional concatenadas com aquelas extraídas segundo o

quadrado rotacionado, resultando em um vetor de dimensão 256.

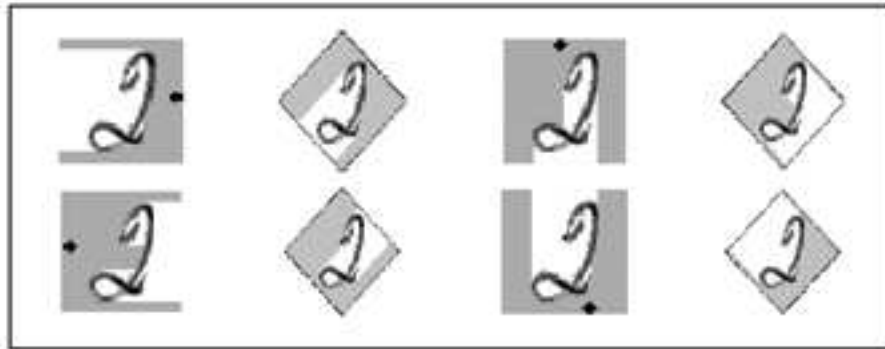


Figura 2.4: Seqüência de Projeções para o Quadrado Rotacionado. Adaptado de (SILVA, 2002)

A seção a seguir traz as técnicas conhecidas para a fase de reconhecimento e classificação, apresentadas de modo geral. No capítulo 3, algumas destas técnicas serão novamente discutidas porém com vistas ao resultado da combinação entre elas.

2.4 Reconhecimento e Classificação

As técnicas de reconhecimento têm como meta associar uma amostra a uma classe predefinida, com base na análise de apenas algumas de suas características mais relevantes. No capítulo a seguir, esta etapa será discutida trazendo as técnicas utilizadas para o treinamento e reconhecimento de caracteres manuscritos encontradas na literatura.

2.4.1 Técnicas de Reconhecimento

De acordo com (JAIN *et al*, 2000), as técnicas para reconhecimento de caracteres manuscritos podem ser distribuídas em quatro grupos maiores para reconhecimento de padrões, são eles: Cálculo de Similaridade com base em padrões (*Template Matching*), Técnicas Estatísticas, Técnicas Estruturais e Redes Neurais. Neste trabalho, no entanto, serão apresentados os três primeiros grupos citados, pois tais grupos não são independentes uns dos outros. Por exemplo, as redes neurais podem ser consideradas como uma técnica estatística.

Cálculo de Similaridade com base em padrões (*Template Matching*):

É baseado num confronto entre a imagem do caractere sendo avaliado e protótipos de caracteres armazenados num banco de imagens (ARICA & VURAL, 2001). Em geral, esta operação determina um grau de similaridade entre dois vetores (grupo de pixels, formas, curvaturas, etc) num espaço de características.

- **Confronto Direto:** compara-se uma matriz de bits, por exemplo, contendo os valores binários ou dos níveis de cinza da imagem de um caractere com um conjunto de protótipos padrões. Para a medida de similaridade (ou dissimilaridade) utiliza-se o cálculo de distância entre vetores (Euclidiana, Mahalanobis, Jaccard, etc.), combinados ou não com outras medidas como a do vizinho mais próximo.
- **Modelos Deformáveis:** a deformação de um modelo da imagem é utilizada para confrontar a imagem desconhecida com um conjunto de imagens conhecidas, devendo também ser extraídas as medidas de similaridade que serão avaliadas, como pode-se observar através da figura 2.5. Esta técnica é mais popularmente utilizada em sistemas de reconhecimento *on-line*.

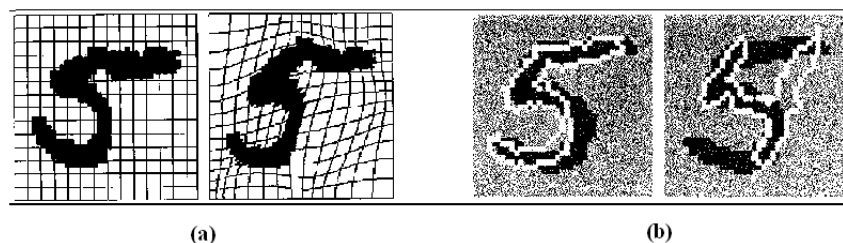


Figura 2.5: Modelos Deformáveis. (a) Deformações do Modelo de Dígitos (b) Modelo Deformado Superposto a Imagem Alvo. Adaptado de (ARICA & VURAL, 2001)

Técnicas Estatísticas

A teoria sobre a tomada de decisões através de informações estatísticas tem por base o uso de uma função de decisão estatística e um conjunto de critérios de condições ótimas, que maximizam a probabilidade de um certo padrão observado pertencer a uma dada classe.

As técnicas estatísticas mais usadas para o reconhecimento de caracteres estão listadas a seguir:

- **Redes Neurais:** realiza um mapeamento não linear, construído automaticamente, e de forma adaptativa, entre as características que representam cada um dos padrões de entrada e suas respectivas classes. Existe uma vasta bibliografia que aborda os conceitos e os vários modelos de redes neurais existentes e suas aplicações. Uma abordagem mais detalhada encontra-se em (FREIXINHO, 2002), (BRAGA *et al*, 2000), (HAYKIN, 2001), (BISHOP, 1995) e (ZURADA, 1992).
- **Categorização:** o agrupamento em classes distintas pode ser feito através de algoritmos “aglomerativos” ou “divisivos”. O primeiro trabalha passo-a-passo agregando cada amostra a grupos inicialmente pequenos através de algum critério de distância. O outro algoritmo começa incluindo todas as amostras numa grande e única classe, e a cada passo subdivide em classes menores mantendo num mesmo grupo amostras próximas de acordo com alguma regra. O objetivo das sucessivas aplicações do algoritmo é o de que cada grupo resultante represente um caractere (ARICA & VURAL, 2001) e (FRANCO, 2002).
- **Modelagem através da Cadeia de Markov (HMM - *Hidden Markov Modeling*):** é uma técnica estatística geralmente utilizada para problemas de classificação de caracteres cursivos. Segundo (ARICA & VURAL, 2001) pode ser definida como um processo estocástico gerado por dois mecanismos inter-relacionados: a Cadeia de Markov com um número finito de estados e um conjunto de funções randômicas, cada qual associada a um estado da cadeia. No reconhecimento de uma palavra ou um caractere, o sistema passa de um estado a outro emitindo uma saída de acordo com algumas probabilidades até que a palavra ou caractere seja identificado. Existem duas abordagens para os sistemas que utilizam HMM:
 - **HMM Modelo-Discriminante:** um modelo é construído para cada classe (caractere, palavra ou unidade segmentada) na fase do treinamento. Os estados representam os centros de massa para os grupos de características. O objetivo da classificação é decidir, através dos modelos, qual produz a seqüência até então desconhecida. Ver figura 2.6.

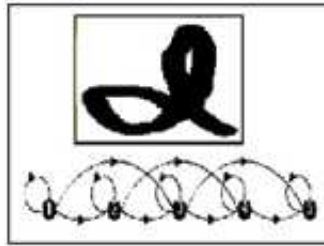


Figura 2.6: HMM Modelo-Discriminante de cinco estágios do caractere “d”. Adaptado de (ARICA & VURAL, 2001)

- **HMM Caminho-Discriminante:** um único HMM é construído para toda uma linguagem ou contexto. Cada estado pode significar um caractere completo, uma parte do caractere ou uma junção de caracteres. O reconhecimento consiste na estimativa de um caminho ótimo para cada classe usando o algoritmo de Viterbi, baseado em programação dinâmica.
- **Lógica Nebulosa:** esta técnica emprega conjuntos de elementos nebulosos na descrição das características dos caracteres. Conjuntos nebulosos produzem resultados mais realísticos quando não se têm informações a priori sobre os dados, e por isso, probabilidades não podem ser calculadas (ARICA & VURAL, 2001). Quando os caracteres são comparados com padrões de referência, medidas de similaridades nebulosas são extraídas, e de acordo com estas medidas os caracteres são associados a uma determinada classe. Para maior detalhamento desta técnica: (KOSKO, 1994), (TSOULAKAS & UHIG, 1997), (COX, 1995) e (FRANCO, 2002).

Técnicas Estruturais:

A idéia principal é a de se fazer uma descrição recursiva de um padrão complexo em termos de padrões simples baseados na forma do caractere. Tais padrões são utilizados para descrever e classificar o caractere. Segundo (ARICA & VURAL, 2001) podem ser subdivididas entre Métodos Gramaticais e Métodos Gráficos, descritos a seguir:

- **Métodos Gramaticais:** onde uma produção de regras é feita para se reconstruir o caractere a partir de um conjunto de primitivas, através de uma

gramática formal. Estes métodos podem combinar algum tipo de característica topológica e estatística, sob alguma regra sintática e/ou semântica (PAVIDILIS, 1993). Neste método, o treinamento é feito descrevendo-se cada caractere por uma gramática G. Na fase do reconhecimento, a string, árvore ou grafo de alguma unidade de escrita (caractere, palavra ou sentença) é analisado para que seja decidido a qual padrão gramatical pertence.

- **Métodos Gráficos:** unidades de escrita são representadas por árvores ou grafos. As primitivas de um caractere são selecionadas através de uma avaliação estrutural. Para cada classe, um grafo ou uma árvore é formado no estágio de treinamento para representar caracteres, palavras ou combinação de caracteres. O estágio de reconhecimento inicializa um grafo desconhecido para uma das classes pelo uso de uma medida gráfica de similaridade (ARICA & VURAL, 2001).

2.5 Pós-Processamento

Como foi visto, a fase de pré-processamento visa tornar o documento o mais claro possível para facilitar o sucesso das operações seguintes, no entanto, informações importantes sobre o caractere sendo avaliado podem ser perdidas. Desta forma, surge a necessidade de agregar ao processo de reconhecimento informações sobre o contexto ao qual pertence o caractere (ARICA & VURAL, 2001), fase esta chamada de pós-processamento. Esta fase pode aumentar consideravelmente o desempenho do reconhecimento, visto que os seres humanos utilizam-se muito do auxílio do contexto do documento para lerem seus textos.

Uma maneira de fazer isto é a utilização de um dicionário para correção de erros de reconhecimento. Por exemplo, numa leitura automática de cheques bancários, inconsistências entre valores válidos e inválidos podem ser detectadas a tempo de serem corrigidas.

O contexto onde uma palavra está inserida também pode ser utilizado para eliminar possíveis ambigüidades. No caso dos CEP's por exemplo, que são números com regras fixas de formação, pode-se usar uma base de dados de números válidos para verificar a acurácia do reconhecimento (SILVA, 2002).

Para o caso do Sistema NeuroTexto inclui-se as etapas de validação e apresentação neste processo de pós-processamento. Tais módulos não foram concluídos e incentivam trabalhos de pesquisa futuros devido a importância dos mesmos para o aumento das taxas de acerto do sistema como um todo.

3 COMBINAÇÕES DE TÉCNICAS DE RECONHECIMENTO

Nas últimas décadas, diferentes métodos de extração de características e técnicas de classificação foram propostos na literatura. Avaliando-se o emprego de técnicas de reconhecimento de padrões de modo individual, pelo fato de cada uma possuir vantagens e desvantagens, conclui-se que não constituem uma solução definitiva para o problema. As técnicas para reconhecimento de caracteres, por exemplo, não precisam, necessariamente, ser empregadas isoladamente umas das outras, podendo uma delas compor com a outra. Neste contexto, a fim de aperfeiçoar o processo de reconhecimento e classificação de um sistema, surge a seguinte questão: de que maneira estes métodos poderiam ser combinados de modo que o resultado do reconhecimento melhore significativamente? No intuito de responder a esta pergunta, algumas estratégias de combinação destas técnicas têm sido estudadas, apresentando resultados promissores (OLIVEIRA *et al*, 2002), (OLIVEIRA *et al*, 2004), (SHRIDHAR & KIMURA, 1991), (ARICA & VURAL, 2001) e (OH & SUEN, 2002). Em geral, os resultados de classificadores individuais, denominados agentes neste texto, são combinados, acreditando-se que estes diferentes agentes possam oferecer informações complementares sobre os padrões, podendo, desta forma, melhorar o desempenho do sistema como um todo.

A seguir, diferentes métodos de combinação de agentes de reconhecimento serão apresentados seguindo a classificação adotada pela literatura, bem como um estudo sobre algumas métricas que podem ser empregadas para auxiliar na escolha dos agentes a serem usados na combinação.

3.1 Métodos de Combinação de Agentes

Os agentes de reconhecimento de padrões podem ser agrupados de modo homogêneo ou heterogêneo. No primeiro caso, todos os agentes escolhidos para compor a combinação empregam a mesma tecnologia. Por outro lado, uma combinação heterogênea é aquela formada por agentes que aplicam tecnologias distintas sobre as características do padrão apresentado.

Outra maneira de classificar uma combinação dos agentes de reconhecimento é sob a forma ou arquitetura dos mesmos (ARICA & VURAL, 2001). De acordo com este último método de classificação, existem dois tipos de arquiteturas de combinação de agentes classificadores: Serial ou Paralela.

- **Arquitetura em Série:** a saída de um agente alimenta o próximo agente. Pode-se citar três disposições que representam uma arquitetura em série: seqüencial ou seletiva.
 - *Seqüencial:* a cada estágio o objetivo é o de reduzir o número de possíveis classes as quais um padrão desconhecido pode pertencer. No início, o padrão desconhecido pode pertencer a qualquer uma das classes possíveis e, a cada estágio, o número de prováveis classes as quais o padrão poderia pertencer diminui, até que no estágio final o padrão recebe um rótulo. Ver figura 3.1.

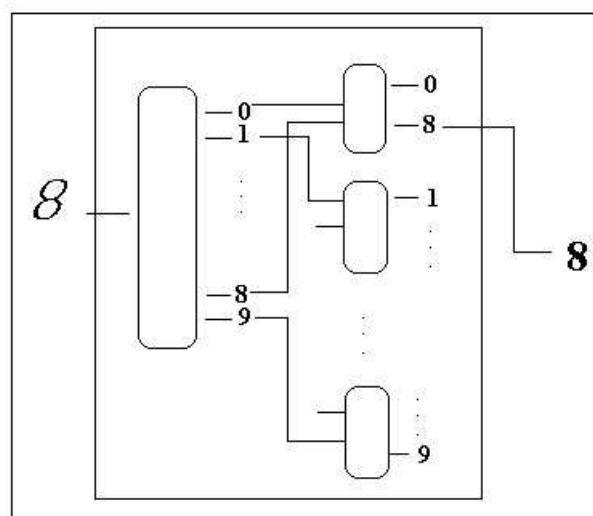


Figura 3.1: Exemplo de Arquitetura em Série do Tipo Seqüencial

- **Seletiva**: Inicialmente o agente associa o novo padrão apresentado a um grupo com padrões similares entre si. Num estágio posterior, os grupos formados são classificados hierarquicamente como uma árvore. Cada nó filho desta árvore tem semelhanças com o nó pai, de acordo com alguma medida de comparação adotada. O exemplo mostrado na figura 3.5 contém, nos dois primeiros estágios do sistema, uma distribuição do tipo seletiva.
- **Arquitetura em Paralelo**: combina o resultado de mais de um agente independente, ver figura 3.2. Tais agentes podem seguir técnicas diferentes de classificação, bem como serem treinados com conjuntos de dados diferentes uns dos outros. A combinação empregada pode obedecer a uma de várias estratégias de combinação existentes na literatura, dentre estas pode-se citar: voto, bayesiana e mistura ponderada de agentes.

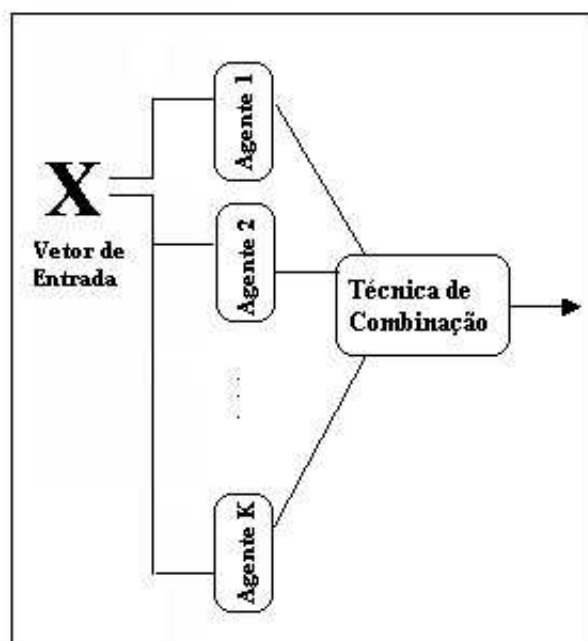


Figura 3.2: Exemplo de Arquitetura em Paralelo

- **Voto**: baseada em “a opinião da maioria vence”, tratando os classificadores de igual modo, sem considerar o erro de cada classificador de forma

independente para a indicação do resultado final. Por exemplo, atribui-se o valor 1 para a saída de maior valor para cada agente, e o valor 0 para as demais saídas. Como resultado final, a classe eleita em consenso para rotular o padrão de entrada, é aquela com maior número de “votos” dados pelos agentes individuais. Pode-se admitir que a classe seja considerada inexistente caso haja empate entre as opiniões dos agentes (MATOS, 2004).

- **Bayesiana**: usa a fórmula de Bayes para integrar as decisões dos classificadores. Para o caso especial onde todos os K agentes são classificadores bayesianos, têm-se que para um padrão $x \in X$,

$$P_k(x \in C_i/x), i = 1, \dots, M \text{ e } k = 1, \dots, K$$

Onde $x \in C_i$ denota que x pertence a classe C_i , das M classes existentes no problema, rotulada pelo agente k . Então, para uma combinação E segundo a estratégia de Bayes, têm-se a seguinte formulação para a probabilidade média utilizando as probabilidades individuais de cada agente (XU *et al*, 1992):

$$P_E(x \in C_i/x) = \frac{\sum_{k=1}^K P_k(x \in C_i/x)}{K}, \quad i = 1, \dots, M$$

- **Mistura Ponderada de Agentes**: a saída do modelo é dada pela soma ponderada das saídas dos agentes. Esta estratégia, em particular, será discutida na seção 3.3.

De acordo com a classificação descrita acima, pode-se entender que um time de agentes é o mesmo que uma combinação de técnicas segundo uma arquitetura em série, e um comitê de agentes uma combinação de técnicas com uma arquitetura em paralelo.

Encontram-se na literatura diversos exemplos do emprego da combinação de técnicas de reconhecimento de padrões, em especial para o reconhecimento de caracteres manuscritos. Na próxima seção apresentam-se alguns destes exemplos.

3.1.1 Exemplos de Técnicas Combinadas

Time de Redes Neurais:

Um exemplo de arquitetura do tipo sequencial e homogênea é o trabalho realizado por (SILVA, 2002), que apresenta uma estratégia de combinação de técnicas para o reconhecimento de caracteres manuscritos denominada Time de Redes Neurais. Foram utilizadas exclusivamente redes neurais segundo o modelo MLP para o reconhecimento de letras manuscritas, provenientes de formulários para concurso.

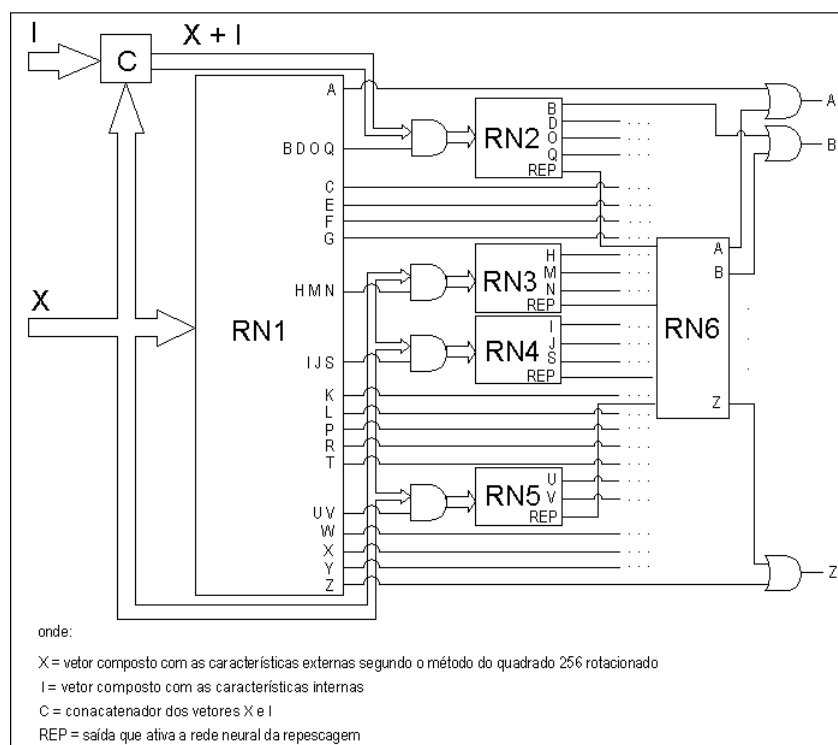


Figura 3.3: Exemplo de Arquitetura em Série do Tipo Sequencial: Time de Redes. Adaptado de (SILVA, 2002)

Na figura 3.3 pode-se ver o modelo adotado para o reconhecimento de letras manuscritas (26 classes). Foram formados quatro grupos constituídos pelas seguintes letras: B - D - O - Q, H - M - N, I - J - S e U - V, agrupadas num primeiro estágio da classificação, para receber um tratamento mais especializado num segundo estágio.

De modo geral, a cada estágio, no processo de reconhecimento de um padrão apresentado ao modelo, de acordo com a saída resultante do estágio anterior, apenas

uma das redes especializadas (agente) é ativada. Esta, por sua vez, é alimentada pelo vetor de características. Com exceção da RN2, nas redes RN3, RN4 e RN5 foram utilizados vetores de entrada compostos com as características extraídas segundo o método do quadrado rotacionado. Na RN2, os vetores foram compostos com as características extraídas segundo o método do quadrado rotacionado concatenadas às características internas.

Como uma última tentativa de corrigir possíveis erros de classificação cometidos pelo modelo, num terceiro estágio, para o caso das letras pertencentes aos grupos citados, existe uma rede denominada de “reescapagem”.

Classe-Modular:

Um exemplo de modelagem de forma paralela utilizando a técnica de voto para combinar as saídas dos agentes está descrito em (OH & SUEN, 2002), no qual os autores propõem o conceito de modularidade em classes utilizando redes neurais MLP. Os estudos feitos incluem o reconhecimento de dígitos manuscritos (10 classes), letras maiúsculas (26 classes), pares de dígitos conectados (100 classes) e caracteres do código postal coreano (352 classes).

Na arquitetura classe-modular (*Class-Modular*), K redes neurais são especializadas em sub tarefas referentes as K classes existentes no problema. Ou seja, o problema original de classificar K classes torna-se em subproblemas de classificação entre 2 classes. As saídas das redes especialistas, ou agentes, são restritas a responderem se o padrão apresentado pertence ou não à classe representada pelo mesmo, discriminando a respectiva classe das $k - 1$ classes restantes. A estratégia de combinação para a chegada a um resultado final é feita com base no algoritmo “o vencedor leva tudo”, alimentada com as saídas das K redes especializadas (ou agentes, como denominamos neste texto).

Na figura 3.4 pode-se ver a arquitetura classe-modular. O vetor de características referente ao padrão X é apresentado a cada um dos módulos M_i , com $i = 1, \dots, K$, onde cada módulo efetua o processo de *forward* utilizando os pesos já fixados para cada rede, produzindo o vetor de saída $D = (o_0, o_1)$.

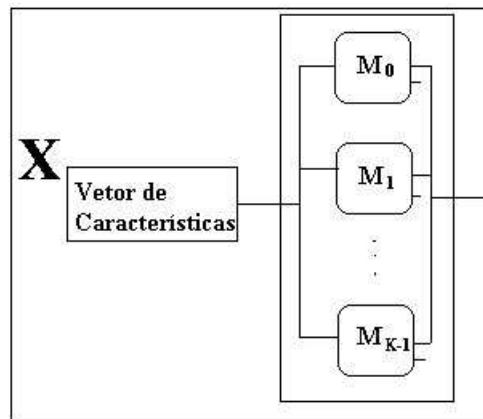


Figura 3.4: Arquitetura de uma Combinação de Técnicas de Reconhecimento Classe-Modular

Two Stage Multi-Network:

Um bom exemplo de arquitetura serial seletiva e heterogênea é o método proposto em (GOPISETTY *et al*, 1996), no qual um grupo de pesquisadores da IBM, combinam Redes Neurais e Template Matching (Cálculo por Similaridade) num sistema de reconhecimento de caracteres denominado TSMN (Two Stage Multi-Network), mostrado na figura 3.5.

Consiste num conjunto de redes neurais especializadas em um tipo de caractere, subdividido em: letras maiúsculas, minúsculas, dígitos e caracteres especiais. Um pré-classificador e uma rede seletora são responsáveis por ativar uma das redes especializadas. Num passo posterior, a técnica de Template Matching é empregada, para comparar com o padrão de entrada, apenas os modelos de caracteres pertencentes a categoria selecionada pelo sistema TSMN. Esta última técnica é acionada somente se o grau de incerteza sobre o resultado final do TSMN estiver acima de um limiar pré-determinado pelo sistema.

3.2 Medidas de Eficácia Numa Combinação de Técnicas

Tentativas de avaliação do desempenho de um MCS (Multiple Classifiers System), ou Sistema Multi-Agentes, como será chamado neste texto, têm incentivado o estudo sobre medidas de eficácia. Por exemplo, o cálculo do intervalo de confiança. Após submeter ao modelo diferentes conjuntos de dados, calcula-se os valores da mé-

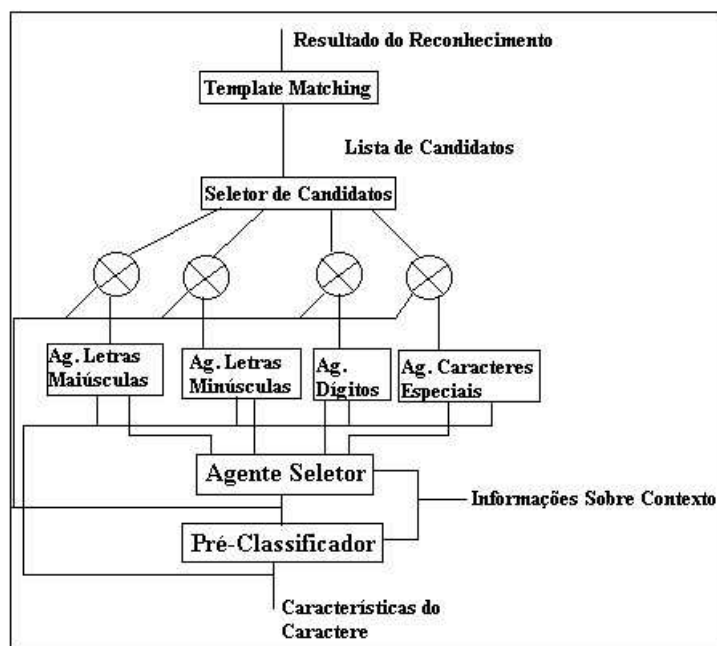


Figura 3.5: Arquitetura de uma Combinação de Técnicas de Reconhecimento do Tipo Seletiva e Heterogênea

dia, desvio padrão, variância e coeficiente de variância com relação aos percentuais de acerto alcançados, possibilitando uma avaliação sobre a robustez do modelo.

No entanto, surge o questionamento sobre a possibilidade de se medir a eficiência da estratégia de combinação selecionada, de modo *a priori*, ou seja, o uso de um mecanismo de predição que oriente a escolha de uma combinação específica entre agentes possam melhorar o desempenho do Sistema de Multi-Agentes.

Uma abordagem recente (KUNCHEVA & WHITAKER, 2003) estuda medidas de diversidade entre os agentes que compõem um sistema e a relação com a exatidão dos resultados deste sistema, como uma maneira de escolher que agentes devem compor a combinação.

As próximas seções trazem uma discussão sobre esta relação, definindo ambos os termos e citando algumas maneiras de determinar a diversidade existente entre dois ou mais agentes.

3.2.1 Relação entre Diversidade e Exatidão

Segundo as pesquisas em (KUNCHEVA & WHITAKER, 2003), o grau de similaridade e de discordância nas respostas produzidas por diferentes agentes, de

modo individual, possibilita prever de que maneira os agentes devem ser combinados para produzirem um resultado final satisfatório, aumentando desta forma, a exatidão global do sistema. Pode-se entender como exatidão de um sistema ou de um agente individual como sendo a capacidade de classificar corretamente um padrão apresentado para reconhecimento.

Segundo (ADEVA *et al*, 2006), a exatidão resultante da combinação de um conjunto de técnicas depende principalmente dos seguintes fatores: a exatidão individual de cada agente, a complexidade do problema de classificação e a qualidade da decomposição. Como qualidade da decomposição, estes autores definem como sendo um indicador único capaz de representar uma combinação dos valores obtidos entre as diferentes medidas de diversidade.

No entanto, como explicado em (HADJITODOROV *et al*, 2006), para alcançar níveis mais altos de exatidão, a diversidade entre os classificadores que compõem o conjunto deve diminuir de forma a esperar-se um equilíbrio entre diversidade e exatidão. Estes autores afirmam que nenhuma teoria convincente ou estudo experimental já foi feito com o objetivo de sugerir alguma medida que pudesse prever de modo eficaz o erro de generalização de um conjunto. Contudo, sabe-se através de outros autores, (WINDEATT, 2005) e (KUNCHEVA, 2005), que é necessário achar um ponto de equilíbrio entre diversidade e exatidão. Em outras palavras, estas são as medidas que devem ser avaliadas para se fazer uma previsão sobre a eficiência de uma combinação de técnicas num Sistema Multi-Agente.

Através da proposta feita por (HADJITODOROV *et al*, 2006), onde afirma que: “A seleção do conjunto através da diversidade mediana permite obter um resultado melhor que a seleção randômica do conjunto ou seleção do conjunto com base na discordância máxima”, conclui-se que conjuntos muito diversos são menos exatos que os conjuntos menos diversos. Ou seja, para agrupar-se um conjunto de agentes com respostas muito diferentes entre si, conseqüentemente seria necessário selecionar agentes que não apresentassem bom desempenho quando avaliados individualmente. Desta forma, utilizando a máxima discordância, o resultado global do sistema Multi-Agente não seria melhor que o uso de apenas um agente para solução do problema. Portanto, esta hipótese, chamada de regra da correlação média ou *soft-correlation*, sugere que sejam selecionados agentes com discordância média entre si.

3.2.2 Medidas de Diversidade

Segundo (KUNCHEVA & WHITAKER, 2003), a chave do sucesso de alguns Sistemas Multi-Agentes é que estes são compostos por um conjunto de agentes diversos. Embora ainda não haja uma descrição formal para o termo diversidade, em (ADEVA *et al*, 2006) ela é definida como uma medida intuitiva da relação entre os agentes. Neste contexto, pode-se avaliar o grau de similaridade ou de discordância entre eles. Entende-se que dois agentes são similares entre si quando, construídos sob as mesmas condições ou não, apresentam respostas individuais de classificação muito parecidas quando apresentados aos mesmos padrões de entrada. Ao contrário, um alto grau de discordância entre agentes é verificado quando apresentam respostas distintas sobre a classificação de um mesmo padrão apresentado. Com relação às condições de construção dos agentes, pode-se citar, por exemplo, o conjunto de dados utilizado para o treinamento, a técnica de representação dos dados, ou ainda a estratégia de reconhecimento empregada.

As medidas de diversidade podem ser calculadas de duas formas (KUNCHEVA & WHITAKER, 2003):

- *pair-wise*: nesta abordagem calcula-se a média da distância medida entre todas as combinações dos agentes 2 a 2. A distância calculada é usada para determinar a diversidade medida. Como exemplos deste tipo de medida pode-se citar: a matriz de probabilidade de discordância e a discordância através das matrizes de confusão.
- *non-pair-wise*: calcula a correlação de cada agente com a saída média do conjunto de agentes que farão parte da combinação. Algumas medidas deste tipo são: entropia e variância de Kohavi-Wolpert (KOHAVI & WOLPERT, 1996).

As seções seguintes apresentam algumas medidas *pair-wise* que serão utilizadas para a realização de uma avaliação dos classificadores que compõem a proposta deste trabalho. Tais avaliações serão apresentadas no capítulo 5.

3.2.3 Matriz de Probabilidade de Discordância

Em (DUIN *et al*, 2004) aplica-se o conceito de discordância para medir a diferença entre dois agentes A_a e A_b construídos para resolver um problema de classificação

P representado por um conjunto de N amostras. A discordância pontual entre dois agentes quaisquer é dada por:

$$d_j(A_a, A_b) = \text{Prob}(A_a(x_j) \neq A_b(x_j) | x_j \in P) \quad (3.1)$$

Na qual $A_i(x)$ retorna a rotulação dada a um padrão x qualquer pelo agente A_i . A discordância sobre o conjunto de amostras do problema é calculada conforme a equação 3.2.

$$D(A_a, A_b) = \frac{\sum_{j=1}^N d_j(A_a, A_b)}{N} \quad (3.2)$$

M classificadores constituem uma matriz $[M \times M]$ de discordâncias D para o problema P , com elementos $D(m, n) = D(A_m, A_n)$.

3.2.4 Discordâncias Através das Matrizes de Confusão

Em (AIRES, 2003), a informação contida nas matrizes de confusão de cada agente individual foi utilizada para computar as distâncias que representam as discordâncias entre eles. Denomina-se esta abordagem de Discordância Baseada no Critério da Distância (*DD-based*).

A Matriz de Confusão é uma representação quantitativa do desempenho obtido para cada classificador em termos do reconhecimento de cada classe. Os termos de uma matriz de confusão devem ser interpretados como:

$$\mathbf{MT} = \begin{vmatrix} R_{1,1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & R_{2,2} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \dots & \dots & R_{M,M} \end{vmatrix}$$

Tabela 3.1: MT - Matriz de Targets

Onde $R_{i,i}$ é o número de amostras do problema pertencentes a classe “i”.

Onde $S_{i,j}$ é o número de amostras do problema pertencentes a classe “j” porém classificadas como pertencentes a classe “i”. E, observando-se a matriz de confusão conclui-se que os valores, por exemplo, $S_{i,i}$ representam as amostras corretamente classificadas na classe “i”; $S_{i,k}$, $k \neq i$, representam os erros do tipo falso positivo; e $S_{k,i}$, $k \neq i$, representam os erros do tipo falso negativo para a classe em questão.

$$\mathbf{MA}_a = \begin{vmatrix} S_{1,1} & S_{1,2} & \dots & S_{1,M} \\ S_{2,1} & S_{2,2} & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ S_{M,1} & \dots & \dots & S_{M,M} \end{vmatrix}$$

Tabela 3.2: MA_a - Matriz de Confusão Gerada pelo Agente “a”

Através do traço da matriz de confusão pode-se calcular o desempenho global do classificador, como indicado na equação 3.3.

$$TR_{i,i} = \frac{\sum_{i=1}^M S_{i,i}}{N} \quad (3.3)$$

A abordagem *DD-based* utiliza a informação contida nas matrizes de confusão, onde para cada par de agentes são computadas as distâncias que representam as discordâncias entre os mesmos. Tais distâncias podem ser obtidas considerando que todas as matrizes de confusão possuem o mesmo tamanho como definido nas equações 3.4 e 3.5.

Sendo MA_a e MA_b as matrizes de confusão geradas pelos agentes a e b respectivamente, a diferença entre elas resulta na matriz MCD^{A_a, A_b} . Esta, por sua vez, é usada no cálculo da distância entre a combinação dos agentes.

$$MCD^{A_a, A_b} = MA_a - MA_b \quad (3.4)$$

$$D^{A_a, A_b} = \sum_{i=1}^M \sum_{j=1}^M |MCD_{i,j}^{A_a, A_b}| \quad (3.5)$$

3.3 Combinação de Classificadores em Comitês

Com base no princípio freqüentemente usado em engenharia: “dividir para conquistar”, a abordagem apresentada nesta seção propõe uma subdivisão da tarefa de classificar um conjunto de padrões em tarefas menores.

Segundo Simon Haykin (HAYKIN, 2001), as máquinas de comitê do tipo Mistura Ponderada podem ser classificadas em duas categorias:

1. **Estruturas Estáticas:** neste tipo de máquina de comitê, as repostas de vários especialistas devem ser combinadas de modo que não envolvam o sinal

de entrada, por isso a denominação estática. Neste modelo, a saída final é produzida pela combinação das saídas dos diferentes agentes especialistas. Como pode ser visualizado na figura 3.6, uma rede neural do tipo Adaline foi utilizada para realizar a combinação entre as saídas dos agentes.

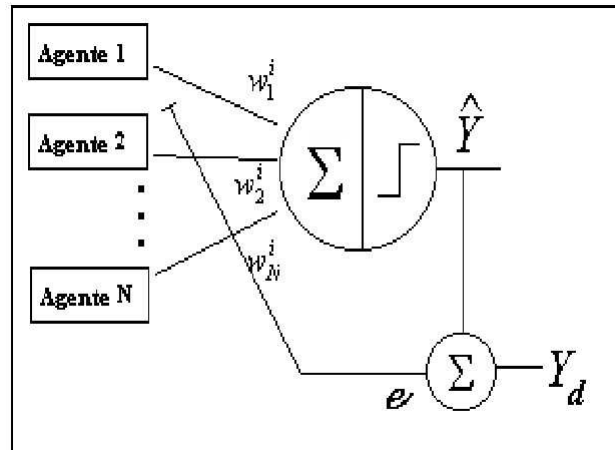


Figura 3.6: Estruturas Estáticas: rede do tipo Adaline

Um dos motivos para a utilização deste tipo de máquina de comitê é o fato de que se fosse substituída por uma única rede neural, o número de parâmetros ajustáveis desta rede seria muito grande, além do tempo que seria gasto para o seu treinamento ser maior.

2. **Estruturas Dinâmicas:** nesta categoria de máquinas de comitê, o sinal de entrada está relacionado ao mecanismo utilizado para integrar as saídas dos especialistas individuais. Podemos citar dois tipos de estruturas dinâmicas:

- *Comitê de Agentes Especialistas:* com base em uma rede de passagem, combina não linearmente, as repostas dos especialistas.
- *Comitê Hierárquico de Agentes Especialistas:* realizam a combinação não linear das repostas individuais dos especialistas com base em várias redes de passagem dispostas de maneira hierárquica.

3.3.1 Comitê de Agentes Especialistas

Consiste em K módulos supervisionados chamados de agentes especialistas e uma unidade integradora chamada de rede de passagem, que tem a função de um

mediador entre os agentes especialistas. A figura 3.7 traz o esquema que representa o modelo.

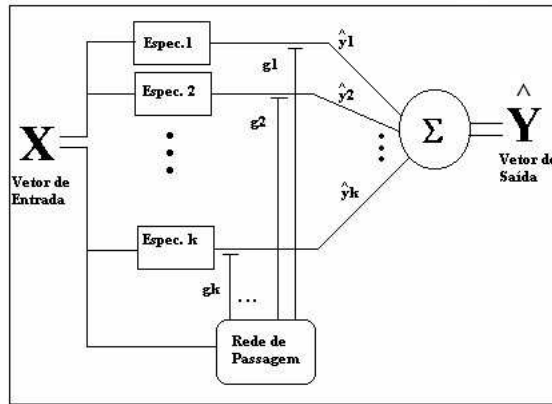


Figura 3.7: Diagrama em blocos do modelo: Comitê com Estrutura Dinâmica

Onde:

- X = vetor amostra de dimensão p
- Y = vetor alvo (resposta desejada) de dimensão q
- \hat{y}_k = vetor resposta do agente k
- \hat{Y} = vetor resposta do comitê

A rede de passagem é formada por K neurônios em sua camada de saída, onde cada neurônio é atribuído a um especialista específico. Em (HAYKIN, 2001) sugere-se que os neurônios tenham funções de propagação definidas por:

$$g_k = \frac{e^{(u_k)}}{\sum_{j=1}^K e^{(u_j)}}, k = 1, 2, \dots, K \quad (3.6)$$

A figura 3.8 mostra o grafo arquitetural da rede de passagem e o grafo do fluxo de sinal do neurônio k naquela rede, onde u_k é a saída da função de ativação dada pelo produto interno do vetor de entrada x como vetor de peso sináptico a_k , isto é:

$$u_k = a_k^T x, k = 1, 2, \dots, K$$

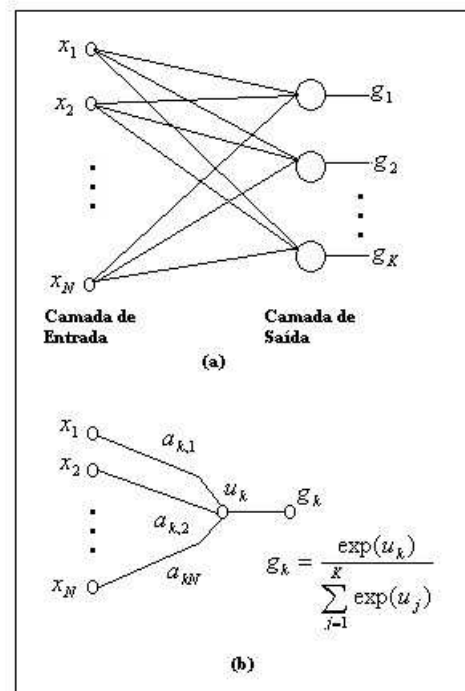


Figura 3.8: (a)Grafo arquitetural da rede de passagem.(b)Grafo do fluxo de sinal do neurônio k

Também segundo (HAYKIN, 2001), a transformação exponencial “normalizada” da equação 3.6 pode ser considerada como uma generalização diferenciável da operação “o vencedor leva tudo” de escolha do valor máximo, e por isso, a função de propagação da equação 3.6 é chamada como *softmax*. Esta função de garante que:

$$0 \leq g_k \leq 1 \quad \forall k$$

e

$$\sum_{k=1}^K g_k = 1$$

O vetor de amostra X é apresentado aos agentes especialistas e à rede de passagem simultaneamente. O vetor de resposta do comitê é calculado pela equação 3.7:

$$\hat{Y} = \sum_{i=1}^K g_i \hat{y}_i \quad (3.7)$$

Dois aspectos importantes do modelo mostrado na figura 3.7, assumindo que cada especialista tenha sido devidamente treinado separadamente, são:

1. A saída \hat{y}_k do agente especialista k fornece uma estimativa da variável aleatória que representa a resposta desejada Y , dado X .
2. A saída g_k da rede de passagem define um conjunto de coeficientes respectivos a cada agente especialista k , baseado no conhecimento obtido a partir de X .

3.3.2 Comitê Hierárquico de Agentes Especialistas

O modelo da figura 3.7 divide o espaço de entrada em subespaços, com base em uma única rede de passagem responsável pela distribuição da informação (proveniente da amostra X) para os vários agentes especialistas. Já o modelo de mistura hierárquica ponderada de agentes especialistas, observado na figura 3.9, é uma extensão do modelo de mistura ponderada de especialistas. Sua arquitetura é similar a uma árvore, na qual as redes de passagem estão em pontos não-terminais e os agentes especialistas encontram-se nas folhas. A diferença entre os modelos hierárquico e não hierárquico está no fato de que, para o primeiro, o espaço de entrada é dividido em conjuntos aninhados de subespaços, onde a informação é combinada e redistribuída entre os especialistas sob o controle das redes de passagem arranjadas de maneira hierárquica (HAYKIN, 2001).

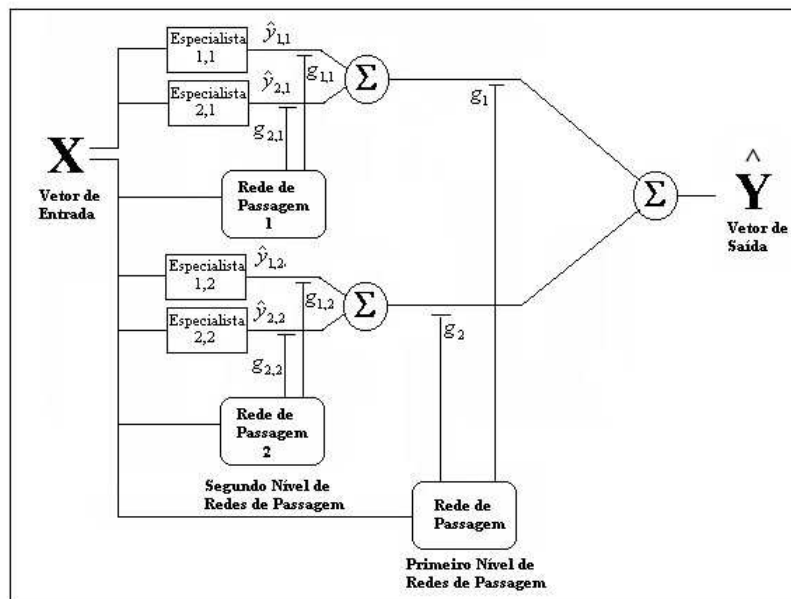


Figura 3.9: Modelo de Comitê Hierárquico com dois níveis de hierarquia.

Modelos do tipo máquinas de comitê baseadas numa mistura ponderada de agentes especialistas foram escolhidos como proposta de resolução do problema em estudo neste trabalho.

No próximo capítulo apresentam-se deduções matemáticas para a obtenção de uma formulação teórica adaptada aos estudos de caso previstos e no capítulo seguinte a descrição dos experimentos feitos.

4 PROPOSTA DE UM COMITÊ DE AGENTES NEURAIS

No intuito de compor um modelo baseado numa combinação de técnicas de reconhecimento conhecida como Máquina de Comitê baseada em uma Mistura Ponderada de Agentes Especialistas, descrita no capítulo anterior, as próximas seções dedicam-se às deduções das fórmulas que orientaram a implementação da proposta deste trabalho.

4.1 Formulação Teórica para o Modelo

4.1.1 Comitê Dinâmico com Uma Camada: Não Linear

Primeiramente idealizou-se o esquema observado na figura 4.1. Uma mistura de especialistas, composta por K agentes treinados e uma rede de passagem com o ajuste dos pesos feito através de uma adequação do algoritmo *backpropagation* ao sistema proposto.

Considerando que a rede de passagem tenha apenas uma camada de neurônios não-lineares, com função de propagação logarítmica sigmoideal definida por:

$$u_k = \text{logsig}(a_k) = \frac{1}{1 + e^{-a_k}} \quad (4.1)$$

Convencionando X a variável que representa o conjunto de entrada, e \hat{Y} a resposta do modelo para cada padrão apresentado, seja dada pela equação 4.2:

$$\hat{Y} = \sum_{i=1}^K \hat{y}_i * g_i \quad (4.2)$$

E, para a fase forward da rede de passagem, a saída do neurônio “i” é dada por

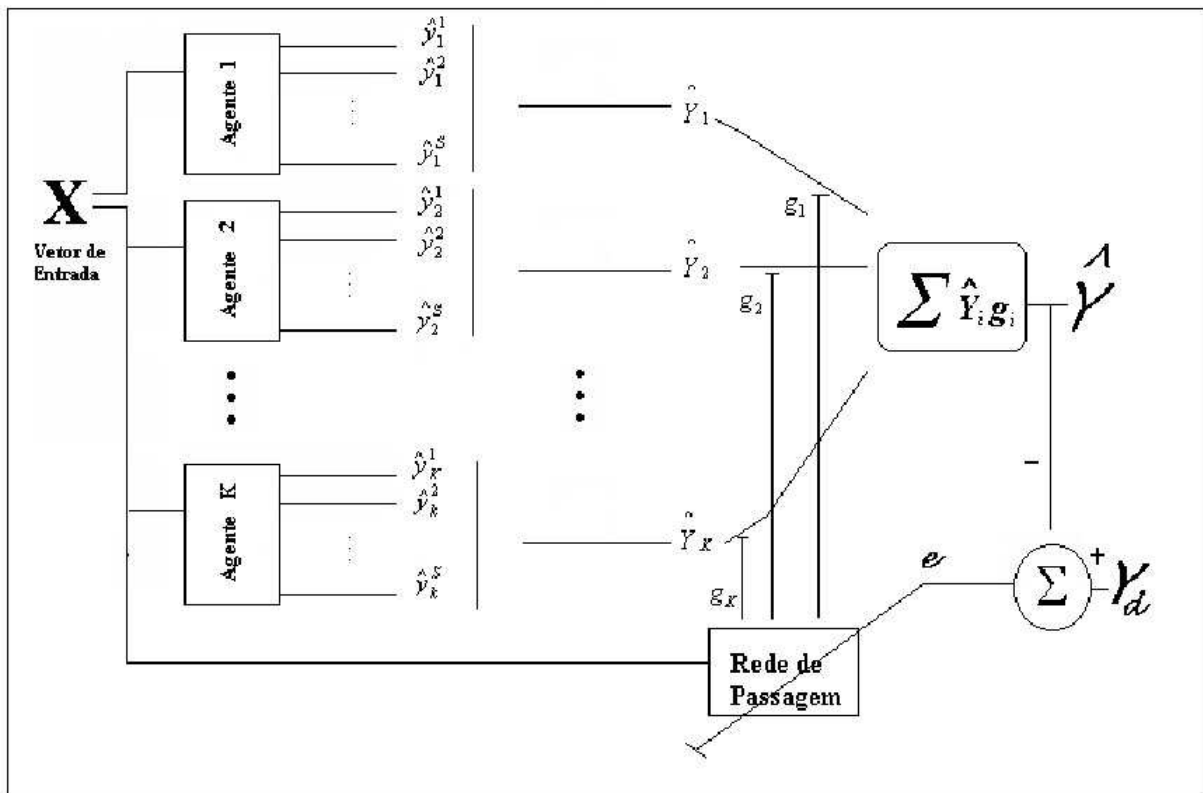


Figura 4.1: Modelo de um Comitê Dinâmico composto por Redes Neurais do tipo MLP

g_i , normalizando a saída da rede, de acordo com a figura 4.2 e a equação 4.3:

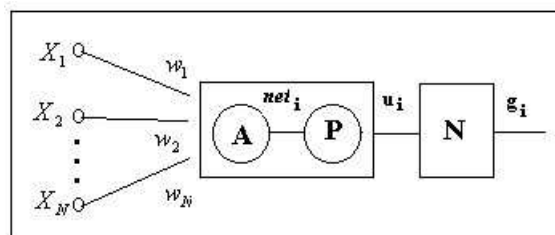


Figura 4.2: Representação de um Nodo Não-Linear da Rede de Passagem

$$g_i = \frac{u_i}{\sum_{i=1}^K u_i} \quad (4.3)$$

Onde K é o número de agentes especialistas utilizados no sistema.

Pode-se observar através da figura 4.1 a disposição dos agentes e a composição da saída \hat{Y}_s , onde S é o número de neurônios da camada de saída dos agentes, e $s = 1, \dots, S$.

O aprendizado da rede de passagem foi feito através da retropropagação do erro, E_p , relativo a contribuição à resposta do comitê dada pelo agente k para a entrada l do padrão p . O erro E_p para cada padrão p apresentado a rede é obtido pela média quadrática da subtração entre o resultado desejado, Y e o resultado alcançado pelo comitê, \hat{Y} , como mostra a equação 4.4.

$$E_p = \frac{1}{2} \sum_{s=1}^S (Y_s - \hat{Y}_s)^2 \quad (4.4)$$

Cada neurônio da camada de saída da rede de passagem, ver figura 4.2, é relativo a um agente k , e o ajuste dos seus pesos sinápticos, ΔW , é dado pela equação 4.5. Onde $k = 1, \dots, K$, $l = 1, \dots, N$, e N é o número de entradas das redes que compõem o modelo, tanto os agentes como a rede de passagem.

$$\Delta W_{lk} = -\lambda \left(\frac{\partial E_p}{\partial W_{lk}} \right) \quad (4.5)$$

Donde, através da regra da cadeia, obtem-se:

$$\Delta W_{lk} = -\lambda \left(\frac{\partial E_p}{\partial \hat{Y}_p} \right) \left(\frac{\partial \hat{Y}_p}{\partial g_k} \right) \left(\frac{\partial g_k}{\partial net_k} \right) \left(\frac{\partial net_k}{\partial w_{lk}} \right) \quad (4.6)$$

E, resolvendo as derivadas parciais da fórmula 4.6, considerando as funções envolvidas no sistema, tem-se:

$$\frac{\partial E_p}{\partial \hat{y}_p} = -\sum_{i=1}^S (Y_i - \hat{Y}_i) \quad (4.7)$$

$$\frac{\partial \hat{Y}_p}{\partial g_k} = \hat{y}_k \quad (4.8)$$

$$\frac{\partial g_k}{\partial net_k} = f'(net_k) \quad (4.9)$$

$$\frac{\partial net_k}{\partial W_{lk}} = X_l \quad (4.10)$$

Finalizando, a equação para o ajuste dos pesos sinápticos da camada de saída da rede de passagem, é feita através da equação 4.11:

$$\Delta W_{lk} = \lambda \sum_{s=1}^S ((Y_s - \hat{Y}_s) \hat{y}_l) f'(net_k) X_l \quad (4.11)$$

O ajuste é feito a cada rodada (época) do algoritmo de treinamento da seguinte maneira:

$$W_{lk}^{n+1} = W_{lk}^n + \Delta W_{lk}^n \quad (4.12)$$

4.1.2 Algoritmo de Treinamento

De posse das equações concluídas na seção anterior, apresenta-se a seguir o algoritmo de treinamento utilizado para adaptar os pesos sinápticos da rede de passagem. Configurando a rede de passagem como uma rede neural do tipo MLP com uma camada de entrada de dimensão L e uma camada de saída de dimensão K , e função de propagação do erro não linear, neste caso, logarítmica sigmoideal (equação 4.1).

1. Salvar os resultados de classificação feita pelos agentes: \hat{y}_i
2. Inicializar os pesos da rede de passagem
3. A cada época $n + 1$:

(a) Fase Forward

- Cálculo do net para cada neurônio k :

$$net_k = W_{kl} * X_l$$

- Cálculo de u_k :

$$u_k = f'(net)$$

- Cálculo de g_k utilizando a equação 4.3

(b) Fase Backward

- Cálculo da Saída Final do Modelo:

$$\hat{Y} = \sum_{i=1}^K \hat{y}_i * g_i$$

- Cálculo do Erro utilizando a equação 4.4
- Atualização dos Pesos da Rede de Passagem com base nas equações 4.11 e 4.12

4.1.3 Comitê Dinâmico com Duas Camadas: Não Linear

Em seguida, foram feitos estudos considerando uma rede de passagem contendo duas camadas de neurônios não-lineares, com funções de propagação também logarítmica sigmoideal (equação 4.1). O esquema com a arquitetura do modelo com a rede de passagem com duas camadas pode ser visto na figura 4.3. Seja a entrada X_p para cada padrão e a saída do neurônio i da camada intermediária convencionalizada como H_i .

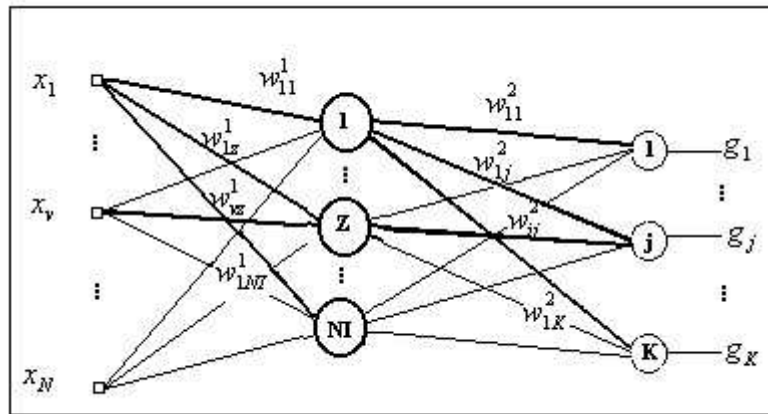


Figura 4.3: Diagrama Esquemático dos Nodos da Rede de Passagem com uma Camada Intermediária

O ajuste dos pesos para a camada de saída é feito através da equação 4.13:

$$\Delta W_{ij}^0 = -\lambda \left(\frac{\partial E_p}{\partial W_{ij}^0} \right) \quad (4.13)$$

onde $i = 1, \dots, NI$, $j = 1, \dots, K$. Considerando que NI é o número de neurônios na camada intermediária, K o número de neurônios da camada de saída da rede de passagem, S o número de neurônios da camada de saída de cada agente e o índice numérico superior serve para diferenciar as camadas, sendo 0 para camada de saída e 1 para camada intermediária.

Resolvendo através da regra da cadeia, obtém-se a equação 4.14:

$$\Delta W_{ij}^0 = -\lambda \sum_{s=1}^S \left(\left(\frac{\partial E_p}{\partial \hat{Y}_s} \right) \left(\frac{\partial \hat{Y}_s}{\partial g_j} \right) \left(\frac{\partial g_j}{\partial net_j^0} \right) \left(\frac{\partial net_j^0}{\partial W_{ij}^0} \right) \right) \quad (4.14)$$

E, resolvendo as derivadas parciais da fórmula 4.14, considerando as funções envolvidas no sistema, tem-se:

$$\frac{\partial E_p}{\partial \hat{y}_p} = -\sum_1^S (Y_d - \hat{Y}) \quad (4.15)$$

$$\frac{\partial \hat{Y}_p}{\partial g_j} = \hat{y}_j \quad (4.16)$$

$$\frac{\partial \vec{g}_j}{\partial net_j^0} = f'(net_j^0) \quad (4.17)$$

$$\frac{\partial net_j^0}{\partial W_{ij}^0} = H_i \quad (4.18)$$

Finalizando, a equação para o ajuste dos pesos sinápticos da camada de saída da rede de passagem, é feita através da equação 4.19:

$$\Delta W_{ij}^0 = \lambda \sum_{s=1}^S ((Y_{ds} - \hat{Y}_s) \hat{y}_j) f'(net_j^0) H_i \quad (4.19)$$

O ajuste é feito a cada rodada $n + 1$ do algoritmo de treinamento da seguinte maneira:

$$W_{ij}^0(n + 1) = W_{ij}^0(n) + \Delta W_{ij}^0 \quad (4.20)$$

Já para o ajuste dos pesos da camada escondida de neurônios, partindo do esquema mostrado na figura 4.3 e da equação 4.21:

$$\Delta W_{vz}^1 = -\lambda \left(\frac{\partial E_p}{\partial W_{vz}^1} \right) \quad (4.21)$$

Onde $v = 1, \dots, N$ e $z = 1, \dots, NI$. Considerando que N significa o número de entradas, NI o número de neurônios na camada intermediária e o índice numérico superior serve para diferenciar as camadas, sendo 0 para camada de saída e 1 para camada intermediária.

E, utilizando a regra da cadeia, obtem-se:

$$\Delta W_{vz}^1 = -\lambda \sum_{s=1}^S \left(\frac{\partial E_p}{\partial \hat{Y}_s} \right) \sum_{i=1}^K \left(\left(\frac{\partial \hat{Y}}{\partial g_i} \right) \left(\frac{\partial g_i}{\partial net_i^0} \right) \left(\frac{\partial net_i^0}{\partial H_z} \right) \right) \left(\frac{\partial H_z}{\partial net_z^1} \right) \left(\frac{\partial net_z^1}{\partial W_{vz}^1} \right) \quad (4.22)$$

E, resolvendo as derivadas parciais da fórmula 4.22, considerando as funções envolvidas no sistema, tem-se:

$$\frac{\partial E_p}{\partial \hat{Y}_p} = -\sum_1^S (Y_d - \hat{Y}) \quad (4.23)$$

$$\frac{\partial \hat{Y}_p}{\partial g_i} = \hat{Y}_i^0 \quad (4.24)$$

$$\frac{\partial g_i}{\partial net_i^0} = f'(net_i^0) \quad (4.25)$$

$$\frac{\partial net_i^0}{\partial H_z} = W_{zi}^0 \quad (4.26)$$

$$\frac{\partial H_v}{\partial net_z^1} = f'(net_z^1) \quad (4.27)$$

$$\frac{\partial net_z^1}{\partial W_{vz}^1} = X_v^1 \quad (4.28)$$

Finalizando, a equação para o ajuste dos pesos sinápticos da camada intermediária da rede de passagem, é feita através da equação 4.29:

$$\Delta W_{vz}^1 = \lambda \sum_{s=1}^S (Y_{ds} - \hat{Y}_s) \left(\sum_{i=1}^K Y_i f'(net_i^0) W_{zi}^0 f'(net_z^1) X_v^1 \right) \quad (4.29)$$

O ajuste é feito a cada rodada $n+1$ do algoritmo de treinamento da seguinte maneira:

$$W_{vz}^1(n+1) = W_{vz}^1(n) + \Delta W_{vz}^1 \quad (4.30)$$

4.1.4 Algoritmo de Treinamento

De posse das equações concluídas na seção anterior, apresenta-se a seguir o algoritmo de treinamento utilizado para adaptar os pesos sinápticos da rede de passagem. Configurando a rede de passagem como uma rede neural do tipo MLP com uma camada de entrada de dimensão N , uma camada intermediária com NI nodos, e uma camada de saída de dimensão K . Neste caso, a função de propagação do erro das camadas intermediária e de saída são não-lineares, fazendo uso da função logaritmica sigmoideal (equação 4.1).

1. Salvar os resultados de classificação feita pelos agentes: \hat{y}_j
2. Inicializar os pesos da rede de passagem

3. A cada época $n + 1$:

(a) **Fase Forward**

Camada Intermediária

- Cálculo do net_z^1 para cada neurônio z :

$$net_z^1 = W_{vz}^1 * X_v$$

- Cálculo da saída da camada intermediária H_z :

$$H_z = f'(net_z^1)$$

Camada de Saída

- Cálculo do net_j^0 para cada neurônio j :

$$net_j^0 = W_{ij}^0 * H_i$$

- Cálculo de u_j :

$$u_j = f'(net_j^0)$$

- Cálculo de g_j utilizando a equação 4.3

(b) **Fase Backward**

- Cálculo da Saída Final do Modelo:

$$\hat{Y} = \sum_{j=1}^K \hat{y}_j * g_j$$

- Cálculo do Erro utilizando a equação 4.23

- Atualização dos Pesos da Rede de Passagem com base nas equações 4.29 e 4.30

4.1.5 Comitê Dinâmico com Duas Camadas: Linear

Outra abordagem estudada considera que a rede de passagem possua uma camada de entrada, uma camada intermediária e uma camada de saída. A diferença entre esta abordagem e a descrita na seção anterior é que os neurônios da camada

de saída, neste caso, são lineares com função de propagação dada pela equação 4.31, e a camada intermediária continua não-linear com função de propagação com base na equação 4.1.

$$f(x) = x \quad (4.31)$$

Com relação às deduções das fórmulas, a diferença para a abordagem anterior está no momento da derivação da função de propagação f com relação a (net_i^0) ou seja, considerando novamente a equação 4.22, o resultado de $\frac{\partial g_i}{\partial net_i^0}$ é igual a 1.

Então a equação final de ajuste dos pesos da camada de saída fica:

$$\Delta W_{ij}^0 = \lambda \sum_{s=1}^S ((Y_{ds} - \hat{Y}_s) \hat{Y}_j) H_i \quad (4.32)$$

e o ajuste dos pesos da camada intermediária passa a ser:

$$\Delta W_{vz}^1 = \lambda \sum_{s=1}^S (Y_{ds} - Y_s) \left(\sum_{i=1}^K Y_i W_{zi}^0 f'(net_z^1) X_v^1 \right) \quad (4.33)$$

4.1.6 Algoritmo de Treinamento

De posse das equações concluídas na seção anterior, apresenta-se a seguir o algoritmo de treinamento utilizado para adaptar os pesos sinápticos da rede de passagem. Configurando a rede de passagem como uma rede neural do tipo MLP com uma camada de entrada de dimensão N , uma camada intermediária com NI nodos, e uma camada de saída de dimensão K . Neste caso, a função de propagação do erro das camadas intermediária é não linear, fazendo uso da função logarítmica sigmoideal (equação 4.1) e para a camada de de saída é linear, equação 4.31.

1. Salvar os resultados de classificação feita pelos agentes: \hat{y}_j
2. Inicializar os pesos da rede de passagem
3. A cada época $n + 1$:

(a) Fase Forward

Camada Intermediária

- Cálculo do net_z^1 para cada neurônio z :

$$net_z^1 = W_{vz}^1 * X_v$$

- Cálculo da saída da camada intermediária H_z :

$$H_z = f'(net_z^1)$$

Camada de Saída

- Cálculo do net_j^0 para cada neurônio j :

$$net_j^0 = W_{ij}^0 * H_i$$

- Cálculo de u_j :

$$u_j = f'(net_j^0)$$

- Cálculo de g_j utilizando a equação 4.3

(b) Fase Backward

- Cálculo da Saída Final do Modelo:

$$\hat{Y} = \sum_{j=1}^K \hat{y}_j * g_j$$

- Cálculo do Erro utilizando a equação 4.23

- Atualização dos Pesos da Rede de Passagem com base nas equações 4.32 e 4.33

Desta forma, utilizando as equações para ajuste de pesos da rede de passagem relativas as diferentes abordagens apresentadas, o próximo capítulo traz os detalhes sobre a implementação e os testes destes modelos com base no estudo de caracteres manuscritos avaliados isoladamente, para o caso da classificação de dígitos e letras maiúsculas extraídos de formulários para concursos.

5 ESTUDO DE CASOS E AVALIAÇÃO DOS RESULTADOS

De acordo com as deduções feitas no capítulo 4, implementações e testes foram feitos no intuito de compor uma máquina de comitê dinâmica capaz de proporcionar desempenho na classificação dos dígitos e letras manuscritas melhor que o já obtido com o Time de Redes (SILVA, 2002). Nas discussões sobre os resultados dos experimentos realizados avaliações são feitas sobre o fato de o emprego de uma máquina de comitê ser vantajoso ou não quando comparado ao uso de apenas uma rede mlp para resolver o problema.

Pode-se avaliar os percentuais obtidos em outros trabalhos disponíveis na literatura, porém fazer uma comparação entre resultados torna-se uma tarefa difícil, pois os experimentos descritos normalmente são realizados em circunstâncias diferentes. No entanto, é possível fazer-se comparações entre este trabalho e os experimentos com o Time de Redes (SILVA, 2002) sob o ponto de vista dos dados e métodos de extração das características, pois são os mesmos utilizados neste trabalho.

Os conjunto de dados, dígitos (10 classes) e letras maiúsculas (26 classes), estavam disponíveis, a partir dos experimentos realizados por (SILVA, 2002). A quantidade total de amostras de dígitos era de 8820 e de letras 20543.

Além dos conjuntos de dados citados no parágrafo acima, um novo conjunto foi gerado visando o aumento da quantidade de amostras de cada padrão estudado e a utilização destes para a composição de um banco de dados mais robusto para o sistema. A base estendida foi montada a partir de uma nova remessa de formulários, dos quais 38728 imagens de dígitos e 55186 imagens de letras foram segmentadas e submetidas ao processo de extração de características segundo a técnica do quadrado

rotacionado, mencionada no capítulo 2.

As próximas seções trazem os experimentos feitos com os dígitos e as letras manuscritas. Por possuírem características particulares, as implementações e testes com estes dois grupos de caracteres foram feitos separadamente.

Para o caso dos dígitos, três tentativas de composição de agentes especializados foram feitas. Na primeira tentativa, os agentes, redes neurais do tipo MLP com as melhores taxas de acerto alcançadas, foram combinados a uma rede de passagem formando um comitê. Definiu-se empiricamente que, a princípio, três agentes seriam suficientes para a composição do comitê. Os três agentes selecionados foram treinados com o uso de um mesmo conjunto de dados (chamado de “Treino Ag.”, descrito na tabela 5.1). Já na segunda tentativa com os dígitos, a idéia foi construir agentes mais especializados em diferenciar determinados grupos de dígitos que apresentaram maiores índices de confusão. Avaliando as matrizes de confusão do melhor agente da tentativa um, constatou-se que as maiores confusões aconteciam entre os dígitos 0 e 8 e entre os dígitos 2 e 7. Desta forma, o conjunto de dados foi dividido de tal maneira que o agente que se especializaria em distinguir 0 e 8 foi treinado com um conjunto de dados com uma quantidade grande destes dígitos, e pequena dos demais. Da mesma forma foi feito com relação ao agente que seria especializado em distinguir os dígitos 2 e 7. O comitê foi composto pelos agentes criados e a rede de passagem e, numa segunda implementação desta tentativa, ao comitê foi acrescido o agente de melhor desempenho da tentativa anterior, denominado de “AgenteG”.

Uma terceira tentativa foi implementada. Desta vez, o particionamento do espaço de entrada ficou a cargo do algoritmo Kmeans, empregado sobre a massa de dados para treinamento dos agentes. O conjunto de dados foi particionado primeiramente em dois grupos e agentes foram treinados respectivamente com cada grupo gerado. Comitês foram compostos através de combinações dois a dois entre os agentes criados e o “AgenteG”, e um último comitê foi composto pelos três agentes em questão.

Em seguida, ainda na terceira tentativa com dígitos, todo conjunto foi novamente particionado, desta vez em três grupos. Comitês foram compostos através de combinações três a três entre os agentes criados e o “AgenteG” e um último comitê foi composto pelos quatro agentes em questão.

5.1 Experimentos com Dígitos Manuscritos

Decidiu-se que, para a construção dos agentes especialistas e para o treinamento do comitê, diferentes conjuntos de dados deveriam ser utilizados. Com isso, para a realização dos treinamentos e testes, toda a massa de dados, aqueles já utilizados anteriormente por (SILVA, 2002) e os resultantes da nova coleta, foram subdivididos em grupos, apresentados na tabela 5.1. A subdivisão do grupo de dados destinado ao teste do comitê em quatro subgrupos teria dois objetivos. O primeiro, a tentativa de diminuir-se a dispersão em tais testes observada através de uma análise do desvio padrão sobre a distribuição dos percentuais de acerto obtidos. O segundo objetivo, separando-se o grupo denominado “TTime”, conjunto este que fora utilizado para testar a proposta em (SILVA, 2002), foi comparar-se os resultados alcançados pelas tentativas de composição do comitê e a proposta citada.

Dígito	Treino Ag.	Teste Ag.	Treino Co.	Teste1	Teste2	Teste3	TTime
0	2727	550	909	550	120	400	385
1	2340	550	780	550	120	400	416
2	2700	550	900	550	120	400	492
3	1884	550	628	550	120	400	69
4	1860	550	620	550	120	400	259
5	1782	550	594	550	120	400	299
6	1767	550	589	550	120	400	313
7	2544	550	848	550	120	400	412
8	1725	550	575	550	120	400	665
9	1572	550	524	550	120	400	410
Total	20901	5500	6967	5500	1200	4000	3720

Tabela 5.1: Grupos de Dígitos

Onde “Treino Ag.” e “Teste Ag” são respectivamente os conjuntos para treinamento e teste dos agentes especialistas. “Treino Co.” é o conjunto utilizado para o treinamento do comitê. O conjunto designado para testar o comitê é formado por cinco grupos. O primeiro é o mesmo conjunto utilizado para testar os agentes individualmente, para que fosse possível uma comparação entre o emprego de uma única

rede MLP e um comitê na classificação dos dígitos. Os demais, “Teste1”, “Teste2” e “Teste3” corroboram para a verificação da estabilidade dos modelos de comitê experimentados. E por fim, “TTime”, é o mesmo conjunto utilizado em (SILVA, 2002), para a proposta apresentada naquele trabalho.

5.1.1 Treinamento dos Agentes - Primeira Tentativa

Para a construção dos agentes especialistas em dígitos realizou-se o treinamento e teste de redes neurais do tipo MLP utilizando os respectivos conjuntos de dados para treino e teste dos agentes.

Para a escolha das características das redes em treinamento, estudos detalhados sobre este assunto foram feitos antes do início dos experimentos com comitês. Os referidos estudos e suas conclusões encontram-se descritos no Anexo desta dissertação.

Após avaliarem-se os índices de acerto com o conjunto de testes, as redes treinadas que apresentaram os três melhores resultados foram selecionadas e chamadas de agentes especialistas. Os resultados alcançados estão relacionados na tabela 5.2.

Agentes	Resultados
1	94.27%
2	94.04%
3	94.07%

Tabela 5.2: Resultados dos Agentes Individuais

O comitê foi composto pelos três agentes selecionados e mais uma rede de passagem, conforme o algoritmo relativo a abordagem de um comitê composto com uma rede de passagem com uma camada não-linear de neurônios de saída visto no capítulo anterior e com estrutura mostrada na figura 5.1. As características do treinamento do comitê estão dispostas na tabela 5.3.

Em Estatística, o coeficiente de variação é uma medida de dispersão que se presta

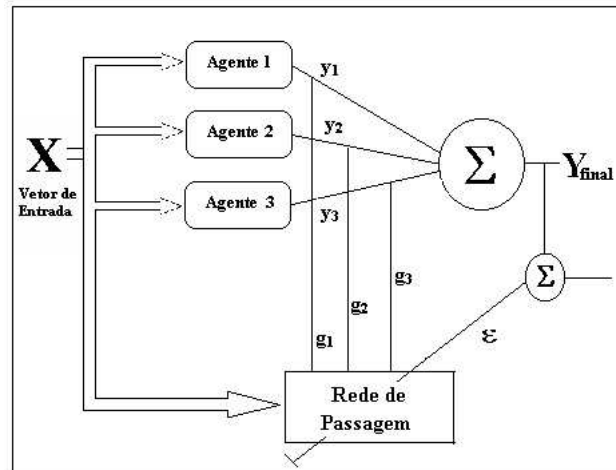


Figura 5.1: Modelo do Comitê para Classificação de Dígitos - Primeira Tentativa

Método de Extração de Características:	Quadrado Rotacionado 128
Número de Entradas da Rede:	217
Número de Saídas da Rede:	3
Número de Camadas da Rede:	1
Função de Propagação:	Logaritmica Sigmoidal
Taxa de Aprendizado:	0.05
Número de épocas:	1000
Número de épocas realizados:	1000

Tabela 5.3: Características do Treinamento da Rede de Passagem no Comitê Preliminar - Dígitos

para a comparação de distribuições diferentes. O desvio-padrão, uma medida de dispersão, é relativo à média e, como duas distribuições podem ter médias diferentes, o desvio dessas duas distribuições não é comparável. A solução é usar o coeficiente de variação, que é igual ao desvio-padrão dividido pela média. A tabela 5.4 mostra os percentuais atingidos, a média do desempenho, o desvio padrão, a variância e o coeficiente de variância para cada agente e para o comitê submetidos aos diferentes conjuntos de testes, já mencionados.

Onde:

- **Ag.1, Ag.2, Ag.3** = Agentes individuais selecionados

- **Comitê** = Comitê composto
- **T.Ag., T1, T2 e T3** = Conjuntos de Teste dos Agentes e Comitê
- **TTime** = Conjunto de Teste utilizado pelo Time de Redes Neurais
- **Md** = Média dos Índices de Acerto
- **DP** = Desvio Padrão
- **Va** = Variância
- **CV** = Coeficiente de Variância

	Ag.1	Ag.2	Ag.3	Comitê
T.Ag.	94,27%	94,04%	94,07%	94,47%
T1	94,00%	93,76%	93,62%	94,35%
T2	94,08%	93,92%	93,83%	94,08%
T3	94,40%	94,30%	94,17%	94,63%
TTime	95,73%	95,67%	95,67%	96,02%
Md	94,50%	94,34%	94,30%	94,71%
DP	0,71	0,77	0,81	0,76
Va	0,5	0,59	0,66	0,58
CV	0,0075	0,0082	0,0086	0,008

Tabela 5.4: Resultado nos Testes da Primeira Tentativa para Classificação dos Dígitos

Através da tabela 5.4 verifica-se que o melhor resultado no teste com o comitê, utilizando-se a mesma massa de teste aplicada aos agentes de modo individual, foi de 94.47%, não superando, em muito, o índice alcançado pelo Agente 1, que é de 94,27%.

Considerando o desvio padrão de 0,76, obtido com a distribuição dos índices de acerto alcançados nos testes com o comitê, conclui-se que a utilização de uma máquina de comitê não apresenta vantagem sobre o uso de apenas um agente de modo individual.

A eficácia, tanto dos agentes de modo individual quanto do comitê, sobre a massa de testes “TTime” supera o índice de 94,57% obtido em (SILVA, 2002) quando fora

utilizado um time de redes neurais para a classificação deste mesmo conjunto de dígitos. O melhor desempenho sobre esta massa de teste foi alcançado pelo comitê, chegando a 96,02%, valor este que está acima da margem de erro calculada para os índices obtidos.

No entanto, dada às condições de treinamento de cada agente e os resultados de acerto alcançados por cada um, suspeitou-se que o grau de similaridade entre os agentes selecionados estivesse alto e, como discutido no capítulo 3, isto fosse a causa de um resultado global de acerto não muito melhor quando comparado ao índice alcançado por um agente único. Desta forma, a tabela 5.5, que representa a Matriz de Probabilidade de Discordâncias, e a tabela 5.6, que representa a Discordância Através das Matrizes de Confusão foram geradas a fim de se comprovar tal hipótese.

	Ag.1	AG.2	Ag.3
Ag.1	0%	3,16%	3,33%
Ag.2	3,16%	0%	3,51%
Ag.3	3,33%	3,51%	0%

Tabela 5.5: Matriz de Probabilidade de Discordância

Combinação	Distância
Ag.1 e 2	0,0225
Ag.1 e 3	0,0258
Ag.2 e 3	0,0258

Tabela 5.6: Discordância Através das Matrizes de Confusão

Os três agentes são muito similares entre si quanto à exatidão. Uma análise dos percentuais contidos na tabela 5.5 mostra que o percentual de darem respostas diferentes para um mesmo caractere é pequeno, afirmando que os agentes em questão possuem grau de discordância entre si muito baixos. A tabela 5.6 confirma este fato, trazendo valores muito pequenos relativos as distâncias entre os agentes.

Neste ponto ficou claro que, na combinação de um conjunto de agentes com alto grau de similaridade entre si, ainda que a exatidão individual dos especialistas seja a melhor possível, não resulta em aumento significativo de exatidão na classificação

do comitê.

Ainda neste contexto, decidiu-se fazer uma avaliação sobre as matrizes de confusão resultantes dos testes feitos com os agentes individualmente, a fim de que fossem observadas os dígitos em que ocorrem as maiores incidências de confusão. Desta análise, uma outra tentativa de compor o comitê foi implementada, com agentes especializados em alguns dígitos específicos. Esperava-se assim ter um grau de diversidade maior entre os agentes, com a expectativa de melhora nos resultados de classificação global com o comitê.

5.1.2 Treinamento dos Agentes - Segunda Tentativa

Seguindo o raciocínio derivado da seção anterior, observou-se a matriz de confusão relativa ao agente número 1, ver tabela 5.7, que apresenta o agente com maior percentual de acerto. Nota-se que as maiores confusões acontecem entre os dígitos 0 e 8 e entre os dígitos 2 e 7.

	0	1	2	3	4	5	6	7	8	9
0	530	0	3	2	1	2	5	0	29	8
1	0	531	3	5	10	0	0	8	4	5
2	2	5	523	9	1	4	1	10	13	1
3	1	3	5	508	1	3	0	3	5	3
4	2	2	2	4	517	4	0	4	1	8
5	2	0	2	5	3	526	4	2	1	3
6	5	0	1	0	4	7	535	0	2	1
7	0	5	6	8	8	0	0	514	4	6
8	7	4	2	3	1	0	5	5	488	2
9	1	0	3	6	4	4	0	4	3	513

Tabela 5.7: Matriz de Confusão do “Agente 1”

Diante destas observações, decidiu-se treinar um agente especializado em distinguir entre os dígitos 0 e 8 e outro agente especializado nos dígitos 2 e 7. Um terceiro agente, para o qual todo o conjunto de treinamento foi apresentado, seria o “Agente 1” utilizado na primeira tentativa e com melhor índice de acerto dentre os três agentes utilizados. Este último passou a ser chamado de “AgenteG”, em referência ao

fato de ser um agente treinado com toda a massa, considerado então um “Agente Geral”. A tabela 5.8 traz as respectivas quantidades de cada dígito para o tipo de agente sendo treinado.

	Ag. 0/8	Ag. 2/7
0	2000	200
1	200	200
2	200	2000
3	200	200
4	200	200
5	200	200
6	200	200
7	200	2000
8	1600	100
9	200	200

Tabela 5.8: Quantidade de Dígitos Representantes das Dez Classes em Cada Conjunto

Onde a coluna “Ag. 0/8” traz as quantidades de amostras de cada dígito a fim de que o agente seja especializado em diferenciar os dígitos 0 e 8. Para isso, foi estabelecido que a massa deveria conter uma quantidade maior de representantes de 0’s (2000) e de 8’s (1600, pois a massa total só dispunha de 1700 dígitos e 100 destes foram utilizados para compor a massa de treino para o “Ag.2/7”), e menor para os demais dígitos. Da mesma forma a coluna “Ag. 2/7” traz as quantidades de amostras de cada dígito a fim de que o agente seja especializado em distinguir os dígitos 2 e 7.

Redes neurais do tipo MLP foram treinadas um número suficiente de vezes para que pudessem ser selecionadas as redes que melhor tivessem “aprendido” a diferenciar os dígitos em questão. Tal seleção teve como critério o número de confusões entre os dígitos, que deveriam diminuir com relação ao resultado encontrado na matriz de confusão do agente 1.

A tabela 5.9, referente ao “Ag. 0/8” mostra que as confusões entre os dígitos 0 e 8 diminuíram com relação ao número de confusões entre estes mesmos dígitos no teste com o agente número 1. O total de confusões entre esta dupla de dígitos passou

de 36 para 32. Já a tabela 5.10, referente ao “Ag.2/7”, mostra que o número total de confusões entre os dígitos 2 e 7 se manteve. Mesmo sem o decréscimo esperado, decidiu-se utilizar este agente na montagem do comitê por saber-se que fora treinado de modo a “aprender” mais sobre os dígitos 2 e 7.

	0	1	2	3	4	5	6	7	8	9
0	533	19	15	2	8	3	21	5	25	18
1	0	458	5	2	32	1	1	7	3	2
2	0	14	465	14	2	5	0	9	4	3
3	2	10	13	462	2	8	0	4	1	4
4	0	6	2	4	456	3	1	4	1	8
5	1	1	0	11	3	503	7	3	0	5
6	5	7	1	0	11	6	509	0	2	0
7	0	21	10	26	14	3	0	477	2	4
8	7	14	33	11	12	11	11	39	509	29
9	2	0	6	18	10	7	0	2	3	477

Tabela 5.9: Matriz de Confusão do Agente Especialista nos Dígitos 0 e 8

	0	1	2	3	4	5	6	7	8	9
0	516	0	1	0	1	1	10	0	107	13
1	1	437	2	5	21	1	8	2	12	5
2	14	54	538	38	9	8	10	9	199	12
3	1	3	0	443	0	6	0	0	12	2
4	2	5	0	5	459	1	3	1	3	7
5	1	3	0	8	2	501	3	1	3	2
6	13	2	2	1	4	11	515	0	14	3
7	0	45	7	40	44	13	1	533	27	15
8	0	1	0	1	1	2	0	0	119	1
9	2	0	0	9	9	6	0	4	54	490

Tabela 5.10: Matriz de Confusão do Agente Especialista nos Dígitos 2 e 7

Embora tenha havido um decréscimo nas confusões entre os dígitos 0 e 8, nota-

se um aumento do número de confusões entre outros dígitos. Conseqüentemente, constatou-se uma perceptível queda no índice geral de acerto quando as redes especializadas foram submetidas a massa de teste. Verifica-se através da tabela 5.11 que, para a rede especializada em 0 e 8, o índice de acerto global foi de 88,16%, para a rede especializada em 2 e 7, foi de 82,75%, e lembrando que para o “Agente Geral” (rede neural com melhor índice de acerto treinada com toda a massa separada para o treino dos agentes) o índice de acerto foi de 94,27%.

Agente	Desempenho
Ag.0/8	88,16%
Ag.2/7	82,75%
Ag.Geral	94,27%

Tabela 5.11: Desempenho dos Agentes - Segunda Tentativa

A tabela 5.12, que representa a Matriz de Probabilidade de Discordâncias, foi gerada a fim de que fosse verificado se o método escolhido para a construção dos agentes colaborou para o aumento do grau de dissimilaridade entre os mesmos, na expectativa de melhores resultados para o sistema global.

	Ag.0/8	Ag.2/7	Ag.Geral
Ag.0/8	0%	19,05%	9,6%
Ag.2/7	19,05%	0	15,38%
Ag.Geral	9,6%	15,38%	0

Tabela 5.12: Matriz de Probabilidade de Discordância

Combinação	Distância
Ag.0/8 e Ag.2/7	0,2833
Ag.0/8 e Ag.Geral	0,1469
Ag.2/7 e Ag.Geral	0,2673

Tabela 5.13: Discordância Através das Matrizes de Confusão

A tabela 5.12 mostra que os percentuais entre os agentes são bem maiores que os avaliados na tentativa anterior, indicando um aumento na discordância que existe

entre eles. A tabela 5.13 afirma o fato de que as discordâncias entre os agentes é pequena, porém, quando comparadas as distâncias calculadas na primeira tentativa, tais valores sofreram um acréscimo.

É importante notar que os dois primeiros agentes possuem exatidão individual claramente menor que a exatidão alcançada pelo Agente Geral.

Um comitê construído com os agentes descritos acima foi composto de acordo com a figura 5.2, a fim de que os resultados pudessem confirmar as expectativas sobre o aumento ou não do desempenho global com relação aos índices alcançados individualmente pelos agentes e também com relação aos resultados da primeira tentativa.

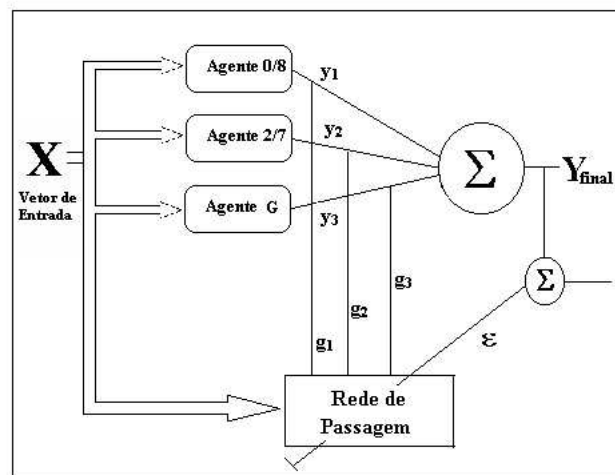


Figura 5.2: Modelo do Comitê para Classificação de Dígitos - Segunda Tentativa

O comitê foi composto pelos três agentes selecionados e mais a rede de passagem conforme o algoritmo relativo a abordagem de um comitê composto com uma rede de passagem com uma camada não-linear de neurônios de saída, visto no capítulo anterior. As características do treinamento do comitê estão dispostas na tabela 5.14.

A tabela 5.15 mostra os percentuais atingidos, a média de desempenho, desvio padrão, variância e coeficiente de variância para cada agente e para o comitê submetidos aos diferentes conjuntos de testes, já mencionados.

Onde:

- **Co2A** = Comitê composto pelos Agentes 0/8 e 2/7

Método de Extração de Características:	Quadrado Rotacionado 128
Número de Entradas da Rede:	217
Número de Saídas da Rede:	3
Número de Camadas da Rede:	1
Função de Propagação:	Logaritmica Sigmoidal
Taxa de Aprendizado:	0.05
Número de épocas:	1000
Número de épocas realizados:	1000

Tabela 5.14: Características do Treinamento da Rede De Passagem no Comitê - Segunda Tentativa

- **Co3A** = Comitê composto pelos Agentes 0/8, 2/7 e Geral
- **T.Ag.,T1, T2 e T3** = Conjuntos de Teste dos Agentes e Comitê
- **T.Time** = Conjunto de Teste utilizado pelo Time de Redes Neurais
- **Md** = Média dos Índices de Acerto
- **DP** = Desvio Padrão
- **Va** = Variância
- **CV** = Coef. de Variância

Submetido ao treinamento, o melhor resultado foi o percentual de 93,75% com a aplicação do comitê com uso dos três agentes em questão, sobre a mesma massa de teste utilizada com os agentes de modo individual e na primeira tentativa. No entanto, este valor é mais baixo que o índice atingido pelo “Ag.G” individualmente, verificando-se uma queda de desempenho no uso do comitê. Isto acontece para todos os conjuntos de teste, mostrando que, neste caso, os agentes 0/8 e 2/7, por apresentarem uma exatidão individual muito abaixo da exatidão alcançada pelo “Ag.G”, acabam por “atrapalhar” na decisão final de classificação feita pelo comitê.

Outra observação sobre o desempenho do comitê composto pelos três agentes em questão é que a média de acerto obtida quando submetido aos conjuntos de testes, foi de 88,07%, apresentando uma variância relativamente alta.

	Ag.0/8	Ag.2/7	Ag.G	Co2A	Co3A
T.Ag.	88,16%	82,75%	94,27%	90,09%	93,75%
T1	87,71%	82,04%	94,00%	90,07%	91,40%
T2	83,50%	78,67%	94,08%	85,08%	87,25%
T3	75,30%	74,13%	94,40%	78,30%	82,20%
T.Time	79,41%	78,74%	95,73%	82,34%	85,73%
Md	82,82%	79,27%	94,30%	85,17%	88,07%
DP	5,5	3,42	0,71	5,08	4,58
Va	30,27	11,71	0,66	25,85	20,99
CV	0,066	0,043	0,0075	0,196	0,052

Tabela 5.15: Resultado nos Testes da Segunda Tentativa para Classificação dos Dígitos

Contudo, é interessante notar que o comitê composto apenas pelos agentes 0/8 e 2/7 apresenta um ganho em percentual quando comparado aos índices de cada agente utilizado, reforçando que o treinamento da rede de passagem efetivamente explora a complementaridade entre os agentes e melhora o desempenho final.

Os índices obtidos para o conjunto “TTime” foram de 82,34% para o comitê com os agentes 0/8 e 2/7 e de 85,73% para o comitê composto pelos três agentes, sendo inferiores ao percentual obtido pelo Time de Redes (SILVA, 2002).

A análise do coeficiente de variância mostra-nos a diferença existente nas distribuições dos resultados de classificação.

5.1.3 Treinamento dos Agentes - Terceira Tentativa

Numa terceira tentativa de geração de agentes que fossem combinados numa estrutura em forma de comitê dinâmico, implementou-se um particionamento na massa de dados utilizada para o treino de tais agentes.

Com o particionamento do espaço de amostras, a intenção foi de agrupar caracteres próximos espacialmente, representados por vetores num espaço de entrada de 217 características. Acredita-se que, submetendo a massa de dados de treinamento dos agentes a este processo, dígitos pertencentes a classes diferentes estariam num mesmo grupo por terem sido grafados de um modo semelhante. Por exemplo, na figura 5.3 o dígito 7 a esquerda mais se parece graficamente com o dígito 4 da direita

do que com o dígito de mesma classe 7 também a direita.

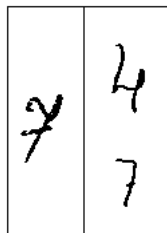


Figura 5.3: Dígitos de Classes Diferentes Próximos Espacialmente

Neste contexto, aplicou-se um algoritmo de particionamento do espaço de dados em k categorias denominado Kmeans. Tal algoritmo organiza os padrões em categorias de acordo com a localização na qual se encontram no espaço euclidiano (FRANCO, 2002).

Embora sejam 10 classes de dígitos, realizou-se testes particionando o espaço em duas e em três categorias de dígitos. Aplicou-se o algoritmo para números pequenos de partições para que, mesmo havendo concentrações diferentes de quantidade de dígitos em cada classe, cada qual contivesse um número representativo de dígitos, a fim de que o agente pudesse “ver” os 10 dígitos.

Após a implementação do algoritmo para as duas quantidades de categorias, $k = 2$ e $k = 3$, aplicou-se um mecanismo de validação denominado CS (*Compactness and Separation*) que avalia a compacidade das categorias geradas e também a qualidade de separação entre elas (FRANCO, 2002). Ainda segundo (FRANCO, 2002) quanto menor o valor de CS melhor a disposição das categorias.

Este último mostrou que a aplicação do Kmeans com a utilização de ambos os índices resultaram em boas partições do espaço com valores de CS próximos a zero.

Particionando em Dois Grupos

Com a partição do grupo de treinamento em dois grupos, a distribuição de representantes em cada classe foi feita e pode ser vista na tabela 5.16.

Observou-se que o grupo 1 concentra mais amostras de dígitos 1, 4, 7, 8 e 9 quando comparado ao grupo 2. Este, por sua vez, agrega mais representantes dos dígitos 0, 2 e 5. Analisando as imagens contidas na figura 5.4 supôs-se que os dígitos do primeiro grupo sejam semelhantes graficamente aos dígitos da esquerda da figura

	Gr.1	Gr.2
0	365	2362
1	2139	201
2	562	2138
3	1067	817
4	1584	276
5	663	1119
6	866	900
7	2160	385
8	1132	593
9	1138	434

Tabela 5.16: Quantidade de Dígitos Representantes das Dez Classes em Cada Conjunto

e os dígitos pertencentes ao grupo 2 estejam de acordo com o lado direito da mesma figura.

7	
1	5
9	0
8	2
4	
GRUPO 1	GRUPO 2

Figura 5.4: Dígitos Semelhantes.

Redes neurais do tipo MLP foram treinadas separadamente para cada grupo de dados gerado e foram submetidas ao mesmo conjunto de teste. Ressaltando que tal conjunto de teste é o mesmo utilizado para todas as redes treinadas neste trabalho, como forma de obter-se uma comparação.

Foram selecionadas as redes neurais com melhores índices de acerto no teste

realizado. Para o grupo 1, selecionou-se o Agente 1 (Ag.1) com índice de 91,75% e para o grupo 2, o Agente 2 (Ag.2), com índice de 87,51%.

Para compor o comitê também foi considerada a participação do Agente Geral (Ag.G), já utilizado nas outras tentativas, que atinge o índice de 94,27% nos testes. Estes índices estão relacionados na tabela 5.17

Agente	Desempenho
Ag.1	91,75%
Ag.2	87,51%
Ag.Geral	94,27%

Tabela 5.17: Desempenho dos Agentes - Terceira Tentativa com Dois Grupos

Avaliando a tabela 5.18, a Matriz de Probabilidade de Discordâncias entre os agentes considerados, percebe-se que existe uma a probabilidade de os agentes rotulem um mesmo caractere de modo diferente mais acentuada entre os agentes 1 e 2 e entre os agentes 2 e G.

	Ag.1	AG.2	Ag.G
Ag.1	0%	13,76%	5,87%
Ag.2	13,76%	0	10,38%
Ag.G	5,87%	10,38%	0%

Tabela 5.18: Matriz de Probabilidade de Discordância

A tabela 5.19, que traz a Discordância através das Matrizes de Confusão dos Agentes confirma o que foi discutido avaliando-se a Matriz de Probabilidade de Discordância, mostrando que a distância entre os agentes 1 e G é menor quando comparada as distâncias entre os agentes 1 e 2 e entre 2 e G.

Feita a combinação dos agentes dois a dois e também agregando os três, a tabela 5.20 mostra os percentuais atingidos, a média de desempenho, desvio padrão, variância e coeficiente de variância para cada agente e para os comitês submetidos aos diferentes conjuntos de testes, já mencionados.

Onde:

Combinação	Distância
Ag.1 e 2	0,1505
Ag.1 e G	0,0651
Ag.2 e G	0,1495

Tabela 5.19: Discordância Através das Matrizes de Confusão

- **T.Ag., T1, T2 e T3** = Conjuntos de Testes dos Agentes e Comitê
- **T.Time** = Conjunto de Teste utilizado pelo Time de Redes Neurais
- **Md** = Média dos Índices de Acerto
- **DP** = Desvio Padrão
- **Va** = Variância
- **CV** = Coef. de Variância

E onde,

- A^{12} = Comitê composto pelos Agentes 1 e 2
- A^{1G} = Comitê composto pelos Agentes 1 e G
- A^{2G} = Comitê composto pelos Agentes 2 e G
- A^{12G} = Comitê composto pelos Agentes 1, 2 e G

A primeira avaliação é feita comparando-se os resultados das diferentes arquiteturas para o conjunto de teste “T.Ag.”. O melhor desempenho obtido para este conjunto foi através do emprego de um comitê formado pelos agentes 1 e 2, um pouco melhor que o índice do “Ag.G”, apenas 0,20% a mais, porém ainda dentro da margem de erro calculada para esta combinação de 0,77 para mais ou para menos.

Nota-se que a utilização do “Ag.G” combinado aos outros dois agentes resulta em uma queda de percentual, indo de 94,27% do “Ag.G” para 93,47%. Este fato pode ser devido a uma acentuada diferença de 6,76% entre a exatidão individual dos agentes G e 2.

	Ag.1	Ag.2	Ag.G	A^{12}	A^{1G}	A^{2G}	A^{12G}
T.Ag.	91,75%	87,51%	94,27%	94,47%	94,45%	94,33%	93,47%
T1	91,84%	86,15%	94,00%	93,62%	94,15%	94,02%	93,45%
T2	91,42%	85,83%	94,08%	93,17%	94,33%	94,08%	93,50%
T3	92,20%	84,03%	94,30%	94,15%	94,50%	94,45%	94,25%
T.Time	93,84%	84,76%	95,73%	95,70%	95,83%	95,89%	95,75%
Md	92,21%	85,67%	94,30%	94,12%	94,65%	94,55%	94,08%
DP	0,95	1,34	0,81	0,77	0,67	0,77	0,99
Va	0,91	1,79	0,66	0,59	0,45	0,59	0,98
CV	0,10	0,016	0,0075	0,0082	0,0071	0,0081	0,010

Tabela 5.20: Resultado nos Testes da Terceira Tentativa com Dois Grupos

O estudo feito através das tabelas 5.18 e 5.19 não é confirmado para todos os conjuntos de teste, existindo uma variação sobre o melhor resultado de acordo com a combinação realizada.

O coeficiente de variância revela que as distribuições relativas às combinações dois a dois são semelhantes entre si e semelhantes a distribuição do “Ag.G”.

Comprova-se a eficácia do treinamento para o comitê composto somente pelos agentes 1 e 2 quando comparado o índice obtido por esta estrutura aos índices individuais dos agentes citados. Para todos os conjuntos há uma melhora no desempenho.

O melhor resultado para o conjunto “TTime” foi obtido com o emprego do comitê formado pelos agentes 2 e geral, sendo de 95,89%, superando o índice anterior de 94,54% em 1,35%, ficando acima da margem de erro calculada para esta combinação.

Particionando em Três Grupos

Devido as discussões feitas com o particionamento da massa de dados em dois grupos, decidiu-se avaliar os resultados com o particionamento em três grupos. A distribuição de representantes em cada classe foi feita e pode ser vista na tabela 5.21.

Observou-se que o grupo 1 concentrou os dígitos 1, 6 e 8. O grupo 2 apresentou mais amostras dos dígitos 1, 4, 7 e 9. E o grupo 3 agrega um número maior de dígitos 0 e 2. Em termos de semelhança gráfica pode-se supor que o grupo 2 reuniu

	Gr.1	Gr.2	Gr.3
0	516	66	2145
1	1060	1177	103
2	897	40	1763
3	782	600	502
4	308	1377	175
5	893	325	564
6	1140	37	589
7	954	1372	219
8	1097	248	380
9	301	1082	189

Tabela 5.21: Quantidade de Dígitos Representantes das Dez Classes em Cada Conjunto dígitos escritos, normalmente, com traços retos. E que os grupos 1 e 3 tenderam a reunir dígitos escritos com uso de formas arredondadas, fugindo e esta regra apenas o dígito 1, no grupo 1. A figura 5.5 traz possíveis representantes de cada grupo citado.

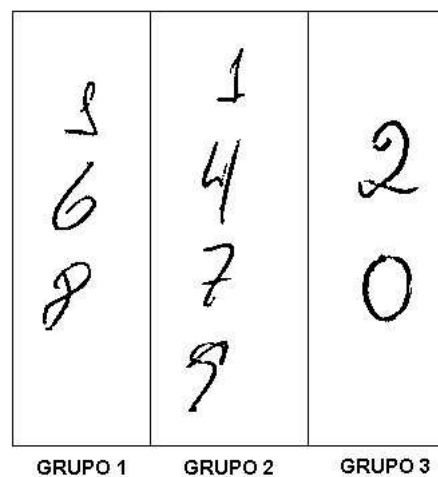


Figura 5.5: Dígitos Agrupados como Mais Semelhantes.

Avaliando as tabelas 5.22, a Matriz de Probabilidade de Discordâncias percebe-se que o grau de discordância entre os agentes é relativamente alto. É possível ver que a discordância é mais baixa entre os agentes 1 e 3, 17,38%. Entre os agentes 1

e 2 a discordância é de 20,93% e a maior discordância está entre os agentes 2 e 3, de 26,58%. Comparando-se os Agentes 1, 2 e 3 com o Agente Geral, vê-se que as discordâncias diminuem bastante. A menor discordância é vista entre os Agentes 1 e Geral, sendo de 8,15%.

	Ag.1	AG.2	Ag.3	Ag.G
Ag.1	0%	20,93%	17,38%	8,15%
Ag.2	20,93%	0	26,58%	17,53%
Ag.3	17,38%	26,58%	0%	14,25%
Ag.G	8,15%	17,53%	14,25%	0%

Tabela 5.22: Matriz de Probabilidade de Discordância

A tabela 5.23, que traz a Discordância através das Matrizes de Confusão dos Agentes mostra que existe um grau de discordância entre eles, porém não chega a ser médio em nenhuma das combinações. Nota-se que entre os Agentes 2 e 3 e entre os Agentes 2 e G os índices de discordância chegam a 0,3538 e 0,2887, respectivamente, sendo estas as maiores discordâncias entre os agentes em questão.

No entanto, comparando as relações de discordância obtidas com o particionamento em dois grupos apenas, vê-se um aumento nestes índices e uma chance de as combinações entre tais agentes possam gerar percentuais de classificação correta para o sistema maiores que as já atingidas até aqui.

Combinação	Distância
Ag.1 e 2	0,2785
Ag.1 e 3	0,1811
Ag.2 e 3	0,3538
Ag.1 e G	0,1068
Ag.2 e G	0,2887
Ag.3 e G	0,2182

Tabela 5.23: Discordância Através das Matrizes de Confusão

Decidiu-se porém apenas combinar os agentes três a três e também agregando todos os agentes ao Agente Geral. A tabela 5.24 mostra os percentuais atingidos,

a média de desempenho, desvio padrão, variância e coeficiente de variância para cada agente e para os comitês submetidos aos diferentes conjuntos de testes, já mencionados.

Onde:

- **T. Ag., T1, T2 e T3** = Conjuntos de Testes dos Agentes e Comitê
- **T.Time** = Conjunto de Teste utilizado pelo Time de Redes Neurais
- **Md** = Média dos Índices de Acerto
- **DP** = Desvio Padrão
- **Va** = Variância
- **CV** = Coef. de Variância

E onde,

- A^{123} = Comitê composto pelos Agentes 1, 2 e 3
- A^{12G} = Comitê composto pelos Agentes 1, 2 e G
- A^{13G} = Comitê composto pelos Agentes 1, 3 e G
- A^{23G} = Comitê composto pelos Agentes 2, 3 e G
- A^{123G} = Comitê composto pelos Agentes 1, 2, 3 e G

Avaliando-se os resultados para o conjunto de teste “T.Ag.”, percebe-se um pequeno acréscimo no índice de acerto quando feita a combinação dos Agentes 1, 2, 3 e o Agente Geral, de 94,38%, comparado ao índice de 94,27% atingido pelo Agente Geral de modo individual. A melhor média de desempenho também é atingida pelo comitê A^{123G} . No entanto, com um desvio padrão de 0,96 esta combinação não é vantajosa quando comparada a utilização de um único agente especializado em todo conjunto de dados, neste caso, o “Agente Geral”.

Especificamente para o conjunto “TTime” o índice alcançado com o comitê A^{123G} foi de 95,97% superando o Time de Redes em 1,43%, com desvio padrão de 0,96.

	Ag.1	Ag.2	Ag.3	Ag.G	A^{123}	A^{12G}	A^{13G}	A^{23G}	A^{123G}
T.Ag.	89,42%	80,55%	84,02%	94,27%	93,51%	92,93%	93,78%	93,98%	94,38%
T1	88,67%	79,55%	82,20%	94,00%	90,47%	88,64%	91,04%	83,15%	93,53%
T2	88,08%	82,83%	82,08%	94,08%	91,92%	90,00%	90,00%	84,17%	93,75%
T3	89,90%	83,30%	82,03%	94,30%	92,20%	90,63%	92,05%	86,65%	94,60%
T.Time	91,21%	84,17%	83,04%	95,73%	94,09%	92,31%	93,49%	87,98%	95,97%
Md	89,45%	82,08%	82,67%	94,30%	92,43%	90,90%	92,07%	87,18%	94,44%
DP	1,20	1,95	0,85	0,81	1,42	1,73	1,60	4,25	0,96
Va	1,44	3,79	0,73	0,66	2,01	3,02	2,57	18,11	0,91
CV	0,0134	0,0238	0,0102	0,0075	0,0154	0,019	0,0174	0,0487	0,0102

Tabela 5.24: Resultado nos Testes da Terceira Tentativa com Três Grupos

Porém, o emprego de somente um agente, o “Agente Geral”, para este mesmo conjunto de teste, alcança o índice de 95,73% com desvio padrão de 0,81, mostrando que a composição de uma máquina de comitê não gera vantagem real.

Como nas tentativas anteriores, vale ressaltar que o comitê composto sem a presença do “Ag.G”, alcança índices de acerto melhores que os atingidos pelos agentes de modo individual, para todos os conjuntos de testes. Mostrando o poder de combinar cada agente de acordo com a capacidade individual de acertar a classificação de grupos distintos de caracteres.

5.2 Experimentos com Letras Manuscritas

Para a realização dos treinamentos e testes com as letras, toda a massa de dados, aquela já utilizada anteriormente pelo Time de Redes Neurais e a resultante da nova coleta, foram subdivididas em subgrupos apresentados na tabela 5.25.

Neste caso, duas tentativas foram realizadas. A primeira, seguindo os passos dados no caso dos dígitos, reuniu três agentes com índices de acerto os melhores possíveis para a composição do comitê. Desta tentativa o agente número um, com o melhor índice de acerto, foi escolhido como “AgenteG”, posteriormente utilizado na segunda tentativa de implementação do comitê.

Uma segunda tentativa foi experimentada. Desta vez, fazendo a subdivisão da massa de treinamento através do algoritmo Kmeans e esperando com isso particio-

Conjuntos	Quantidade
Treinamento Agentes	31691
Teste Agentes	9150
Treinamento Comitê	11246
Teste 1	1150
Teste 2	11246
Teste 3	11246
Teste Time	8843

Tabela 5.25: Grupos de Letras

nar o espaço de forma que caracteres próximos espacialmente ficassem num mesmo grupo. O conjunto de treinamento foi particionado em três grupos e agentes foram treinados para serem especializados em cada um deles.

Nesta tentativa, quatro comitês foram compostos, variando o número de agentes, o número de camadas escondidas da rede de passagem e a função de propagação utilizada na camada de saída.

5.2.1 Treinamento dos Agentes - Primeira Tentativa

Para a construção do comitê responsável pela classificação das letras, definiu-se, como no caso dos dígitos, agentes especialistas como sendo redes neurais do tipo MLP, treinadas de modo que fossem selecionadas as três com melhores índices de acerto sobre o conjunto de teste. Os resultados alcançados estão relacionados na tabela 5.26.

Agentes	Resultados
1	86,83%
2	86,58%
3	86,51%

Tabela 5.26: Resultados dos Agentes Individuais

Desta forma, o comitê foi composto pelos três agentes selecionados e mais a rede de passagem conforme o algoritmo relativo a abordagem de um comitê composto

com uma rede de passagem com uma camada não-linear de neurônios de saída. As características do treinamento estão dispostas na tabela 5.27.

Método de Extração de Características:	Quadrado Rotacionado 128
Número de Entradas da Rede:	216
Número de Saídas da Rede:	3
Número de Camadas da Rede:	1
Função de Propagação:	Logaritmica Sigmoidal
Taxa de Aprendizado:	0.05
Número de épocas:	1000
Número de épocas realizados:	1000

Tabela 5.27: Características do Treinamento da Rede De Passagem no Comitê Preliminar - Letras

Os melhores resultados nos testes com o comitê, utilizando-se a mesma massa de teste aplicada aos agentes de modo individual, giraram em torno de 72.00%. Mediante este valor muito abaixo do alcançado peloa agentes individualmente, concluiu-se que a utilização deste modelo de comitê não produz vantagem quando comparada ao emprego de apenas uma rede MLP de modo individual para a classificação de letras manuscritas. Devido a isto os demais testes não foram experimentados e uma nova tentativa de composição de máquina de comitê foi estudada.

As tabelas 5.28 e 5.29 representam respectivamente, a Matriz de Probabilidade de Discordâncias e a Discordância através das Matrizes de Confusão. Onde:

- **Ag.1** = Agente 1
- **Ag.2** = Agente 2
- **Ag.3** = Agente 3

Uma análise da Matriz de Probabilidade de Discordância, revela que a diversidade entre os Agentes 1, 2 e 3 dois a dois é muito baixa, tendo em vista que a probabilidade de darem respostas diferentes para um mesmo caractere é pequena, em torno de 7%. A Matriz de Confusão reforça esta hipótese mostrando valores baixos e parecidos sobre as distâncias entre os agentes.

	Ag.1	AG.2	Ag.3
Ag.1	0%	7,38%	7,38%
Ag.2	7,38%	0	6,85%
Ag.3	7,38%	6,85%	0

Tabela 5.28: Matriz de Probabilidade de Discordância

	Ag.1	AG.2	Ag.3
Ag.1	0	0,0647	0,0656
Ag.2	0,0647	0	0,0619
Ag.3	0,0656	0,0619	0

Tabela 5.29: Discordância Através das Matrizes de Confusão

Questionou-se a escolha de agentes similares entre si, pois, ainda que a exatidão individual de cada especialista fosse a melhor possível, não resultou em aumento de exatidão na classificação global, pelo contrário, este percentual caiu bastante.

A partir deste ponto, assim como na terceira tentativa para o caso dos dígitos, decidiu-se fazer uma avaliação sobre o espaço de entrada de características dos dados do problema. Pensou-se na realização de mais uma etapa de pré-processamento sobre o conjunto de dados de modo a subdividi-lo em subconjuntos com um certo grau de semelhança entre si utilizando para isso o algoritmo de categorização Kmeans. O objetivo da aplicação do algoritmo foi o de organizar os padrões de entrada em categorias de acordo com a localização que possuem no espaço euclidiano.

O treinamento de uma rede neural com amostras pertencentes a um grupo específico produziria um agente especialista numa partição do espaço de entrada. Desta forma, agentes com nível de diversidade maior entre si seriam construídos, e resultariam em agentes especializados em partições diferentes de um mesmo espaço de entrada.

A próxima seção traz a descrição do procedimento adotado e os resultados de uma nova tentativa de composição de um comitê com estrutura dinâmica para classificação de letras manuscritas.

5.2.2 Treinamento dos Agentes - Segunda Tentativa

Embora o problema aborde apenas letras maiúsculas, o que definiria 26 subespaços, na realidade, pelo fato de que uma mesma letra pode ser escrita de várias maneiras diferentes, isto não acontece. Duas letras pertencentes a classes diferentes podem ter uma distância espacial, distância entre os vetores que as descrevem, menor que duas letras pertencentes a mesma classe. Por exemplo, as letras representadas na figura 5.6(a) foram escritas de maneira longelínea, aproximando-as do grupo que contém mais “I” ou “T”. Já na mesma figura 5.6(b), as mesmas letras foram escritas de modo mais arredondado, aproximando-as do grupo que contém as letras tipo “O”, ou “D”.

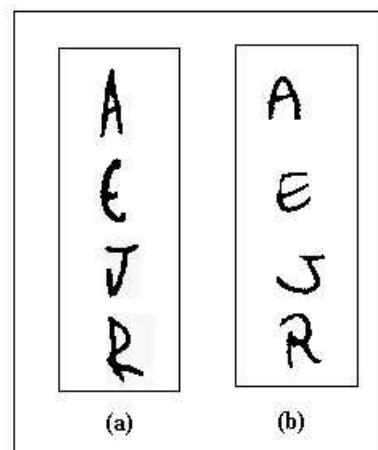


Figura 5.6: Exemplo de Letras Agrupadas: (a)longelíneas (b)arredondadas

Dois testes foram feitos para a decisão de um número ideal de categorias. O primeiro definindo k , número de categorias, igual a 3 e um segundo igual a 5.

Após a implementação do algoritmo para as duas quantidades de categorias, aplicou-se um mecanismo de validação denominado *CS (Compactness and Separation)* que avalia a compacidade das categorias geradas e também a qualidade de separação entre elas (FRANCO, 2002). Este último mostrou que a aplicação do Kmeans com a utilização de ambos os índices resultaram em boas partições do espaço.

Para o caso das letras, no entanto, a decisão sobre o número de partições deu-se

através de uma avaliação do número de representantes de cada classe nos grupos resultantes após a partição em três e em cinco grupos. Para o segundo caso, houve grupos que não continham representantes em algumas classes, e por isso escolheu-se fazer uso de $k = 3$, onde isso não acontecia. A tabela 5.30 traz o número de representantes de cada classe para cada grupo criado.

Let	Gr.1	Gr.2	Gr3	Let	Gr.1	Gr.2	Gr3
A	1536	468	1960	N	1116	137	373
B	171	65	541	O	1752	286	742
C	447	152	471	P	236	817	74
D	537	166	600	Q	168	20	196
E	764	597	971	R	1151	322	1775
F	79	568	17	S	359	258	766
G	232	59	424	T	47	1010	55
H	144	47	369	U	465	94	332
I	78	2365	160	V	170	245	259
J	292	710	567	W	306	2	67
K	112	20	234	X	101	54	246
L	144	166	778	Y	7	352	41
M	628	24	174	Z	161	30	413

Tabela 5.30: Número de Letras em Cada Grupo Particionado

Portanto, três grandes grupos de dados foram criados e utilizados para o treinamento dos agentes especializados nos respectivos grupos de letras manuscritas.

Um quarto agente considerado foi chamado de “Agente Geral” ou “Ag.G.”, constituído pela rede neural com melhor desempenho utilizada na primeira tentativa e com índice de acerto de 86,83%, listado na tabela 5.26.

Aplicou-se as medidas de diversidade apresentadas no capítulo 3 para avaliar de modo *a priori* se a combinação dos agentes criados resultaria em melhora sobre os índices de acerto já alcançados. As tabelas 5.31 e 5.32 representam respectivamente, a Matriz de Probabilidade de Discordâncias e a Discordância através das Matrizes

de Confusão. Onde:

- **Ag.1** = Agente relativo ao Grupo 1
- **Ag.2** = Agente relativo ao Grupo 2
- **Ag.3** = Agente relativo ao Grupo 3
- **Ag.G** = Agente Geral

	Ag.1	AG.2	Ag.3	Ag.G
Ag.1	0%	47,51%	31,7%	28,85%
Ag.2	47,51%	0	45,32%	35,51%
Ag.3	31,7%	45,32%	0	25,4%
Ag.G	28,85%	35,51%	25,4%	0

Tabela 5.31: Matriz de Probabilidade de Discordância

	Ag.1	AG.2	Ag.3	Ag.G
Ag.1	0	0,0059	0,0005	0,0017
Ag.2	0,0059	0	0,0064	0,0077
Ag.3	0,0005	0,0064	0	0,0012
Ag.G	0,0017	0,0077	0,0012	0

Tabela 5.32: Discordância Através das Matrizes de Confusão

Uma análise da Matriz de Probabilidade de Discordância, revela que a diversidade entre os Agentes 1, 2 e 3 dois a dois é média, e diminui quando os agentes são comparados com o Agente Geral. Este fato já era esperado tendo em vista os conjuntos distintos de caracteres nos quais os Agentes 1, 2 e 3 e o Agente Geral se especializaram.

Já a Matriz de Confusão deixa mais claro que os Agentes 1 e 3 tendem a confundir a mesma classe de caracteres pois possuem discordância baixa quando comparada a relação entre os agentes 1 e 2, e entre os agentes 2 e 3. O Agente 2, por sua vez, apresenta uma discordância maior quando comparado ao Agente Geral.

Na construção do comitê, três abordagens para a rede de passagem foram implementadas. Na primeira, a rede de passagem foi composta por apenas a camada de entrada de dados e a camada de saída, esta última com função de propagação não-linear. Na segunda abordagem, a rede de passagem foi construída com uma camada de entrada de dados, uma camada intermediária e uma camada de saída, sendo estas duas últimas não-lineares. E, por fim, a terceira na abordagem, a rede de passagem contou com uma camada de entrada de dados, uma camada intermediária e uma camada de saída, sendo que a função de propagação da camada de saída linear.

E para que uma comparação justa fosse garantida entre o comitê e o Time de Redes, um dos testes do comitê foi feito com o mesmo conjunto de dígitos utilizado para o Time. As tabelas 5.33 e 5.34 trazem os índices de acerto referentes aos Agentes e aos Comitês, respectivamente, quando submetidos as massas de teste. Onde:

- **T.Ag., T1, T2 e T3** = Conjuntos de Testes dos Agentes e Comitês
- **T.Time** = Conjunto de Teste utilizado pelo Time de Redes Neurais
- **Md** = Média dos Índices de Acerto
- **DP** = Desvio Padrão
- **Va** = Variância
- **CV** = Coef. de Variância.

E onde,

- 3^{A1^C} = Comitê composto por 3 agentes e Rede de Passagem com 1 Camada
- 4^{A1^C} = Comitê composto por 4 agentes e Rede de Passagem com 1 Camada
- $4^{A2^{CNL}}$ = Comitê composto por 4 agentes e Rede de Passagem com 2 Camadas/Não-Linear
- $4^{A2^{CL}}$ = Comitê composto por 4 agentes e Rede de Passagem com 2 Camadas/Linear

	Ag.1	Ag.2	Ag.3	Ag.G
T.Ag.	72,45%	63,77%	73,46%	86,83%
T1	77,39%	68,35%	83,65%	92,87%
T2	65,06%	65,84%	67,86%	67,4%
T3	67,74%	63,47%	67,61%	68,68%
T.Time	77,45%	63,69%	81,09%	91,55%
Md	72,02%	65,02%	74,73%	81,47%
DP	5,6	2,09	7,41	12,47
Va	31,32	4,37	54,88	155,46
CV	0,08	0,03	0,1	0,15

Tabela 5.33: Resultado Final nos Testes dos Agentes para Classificação das Letras

	3^{A_1C}	4^{A_1C}	$4^{A_2C_{NL}}$	$4^{A_2C_L}$
T,Ag,	87,91%	89,16%	88,72%	87,55%
T1	97,48%	97,57%	96,87%	95,3%
T2	84,02%	83,33%	83,7%	80,61%
T3	84,55%	83,62%	84,07%	81,65%
T.Time	92,43%	93,61%	93,21%	92,38%
Md	89,28%	89,46%	89,31%	87,5%
DP	5,08	6,22	5,74	6,45
Va	32,27	38,69	32,91	41,59
CV	0,06	0,07	0,06	0,07

Tabela 5.34: Resultado Final nos Testes dos Comitês para Classificação das Letras

Sobre o comportamento dos agentes, avaliados através da tabela 5.33, é perceptível o baixo desempenho individual dos agentes especialistas “Ag.1”, “Ag.2” e “Ag.3”, e com relação ao “Ag.G”, com excessão dos resultados com os conjuntos de teste “T1” e “TTime”, também observa-se baixos índices de acerto na classificação das letras.

Uma análise dos valores do coeficiente de variância, mostra a similaridade entre a distribuição de desempenho dos agentes “Ag.1” e “Ag.2”, e entre os agentes “Ag.3” e “Ag.G”.

Analisando a tabela 5.34, nota-se que a média de acerto alcançada em cada modelo de comitê ultrapassa os índices de acerto resultantes dos agentes de modo individual. O primeiro modelo, composto apenas pelos três agentes especializados em espaços distintos do conjunto total de padrões de entrada, alcança uma média de acerto nos testes superior em 14,55% comparando-se ao agente com maior média obtida, o “Ag.3”. Este fato reforça a hipótese de que o emprego de um comitê de agentes especializados em grupos de padrões pertencentes a espaços distintos produz um aumento considerável no índice de acerto global da combinação.

A inclusão do “Ag.G” no segundo modelo de comitê, apresenta um acréscimo de apenas 0,18% na média de acerto quando avaliados os resultados dos testes. No entanto, esta configuração de comitê com quatro agentes: “Ag.1”, “Ag.2”, “Ag.3” e “Ag.G”, e a utilização de uma rede de passagem com uma camada não-linear, resulta num índice de acerto de 93,61% para o conjunto “T.Time”. Este mesmo conjunto, quando submetido ao Time de Redes (SILVA, 2002), o percentual de acerto foi de 91,91%, verificando-se o acréscimo de 1,7% com o emprego do comitê para este caso.

Observa-se que os resultados do modelo de comitê com uso de duas camadas, sendo uma linear, “ 4^A2^{CL} ”, são sempre menores quando da utilização do modelo “ 4^A2^{CNL} ”, que utiliza a não linearidade em sua camada de saída.

Outra observação interessante é sobre os valores do coeficiente de variância, onde verifica-se a similaridade entre as distribuições de desempenho para todos os modelos de comitês testados.

5.3 Máquina *versus* Ser Humano

Com excessão dos testes comparativos feitos entre os modelos experimentados neste capítulo e o modelo de Time de Rede Neurais (SILVA, 2002), a realização de uma comparação de resultados com propostas de outros trabalhos torna-se uma tarefa impossível pelo fato de os conjuntos de dados utilizados não serem os mesmos. Com intuito de estabelecer-se uma meta de percentual de acerto que servisse como um valor de referência a ser alcançado tanto para a classificação de dígitos quanto para letras, acreditou-se que o índice de acerto resultante de uma avaliação feita por um ser humano seria um bom patamar de reconhecimento para ser adotado.

Para a coleta deste percentual, os mesmos conjuntos de caracteres submetidos

aos testes realizados neste trabalho foram disponibilizados para a avaliação humana. A mesma condição de avaliação dos caracteres foi mantida, ou seja, a imagem de cada caractere seria avaliada de modo isolado, sem o contexto do qual foi extraído. Através de uma página via internet, o participante da avaliação deveria observar a imagem do caractere e classificá-lo conforme sua interpretação. Um modelo da página disponibilizada está representado na figura 5.7

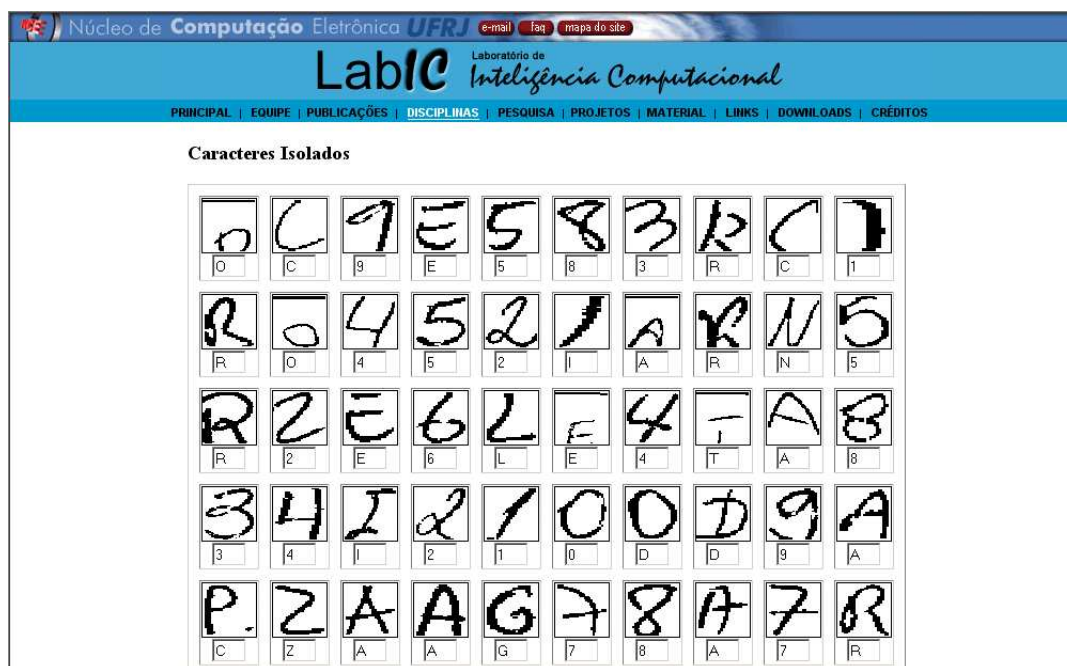


Figura 5.7: Modelo da Página de Coleta da Avaliação Humana

Num período de um mês, cerca de 100 pessoas opinaram a respeito da classificação das imagens dos caracteres. O resultado preliminar de acertos foi de 85.10% para os dígitos e de 78.98% para as letras. Porém, tal inferência humana foi realizada num ambiente alfanumérico, ou seja, além de não estar agregado a um contexto, não foi disponibilizada a informação se o caractere avaliado tratava-se de um dígito ou uma letra. Sendo assim, dentre outras confusões esperadas, casos como a letra “O”, “S”, “I” e respectivamente os dígitos “0”, “5” e “1”, por exemplo, certamente não seriam distinguidos. Desta forma, desconsiderando os casos onde houve erro por motivo desta falta de informação e uma conseqüente confusão entre dígitos e letras, os índices de acerto passaram a ser de 98,97% para os dígitos e 90.90% para as letras.

Os resultados desta pesquisa podem ser vistos e comparados às melhores médias de acerto obtidas pelos modelos apresentados ao longo deste capítulo através da

tabela 5.35. A porcentagem relativa aos dígitos provém da média nos testes obtida pelo comitê composto na primeira tentativa. E a porcentagem apresentada para as letras é o resultado da média obtida nos testes com o emprego do comitê com estrutura “4^A1^C”, ou seja, quatro agentes e uma rede de passagem com apenas uma camada não-linear de saída.

	Ser Humano	Comitê
Dígitos	98,97%	94,71%
Letras	90,90%	89,46%

Tabela 5.35: Tabela Comparativa entre Desempenho do Ser Humano e Modelos de Comitê

De posse dos resultados acima, conclui-se que o emprego da máquina representada aqui pelo Comitê Dinâmico de Agentes Neurais, alcançou índices satisfatórios quando comparada ao percentual de acerto atingido por seres humanos nas mesmas condições, principalmente no caso das letras.

6 CONCLUSÃO

O presente trabalho foi desenvolvido com o objetivo de estudar, conceber e implementar um modelo computacional para reconhecimento de caracteres manuscritos. Os experimentos foram voltados para o uso de redes neurais como solução do problema, tendo como referência os índices de acerto obtidos com um Time de Redes no trabalho feito por (SILVA, 2002).

O modelo proposto tem por base a composição de um comitê de redes neurais combinados dinamicamente através de uma rede neural adicional conhecida como “rede de passagem”.

A princípio fez-se um estudo sobre a escolha de parâmetros para o treinamento de redes neurais do tipo MLP, como o fator de corte mais adequado no emprego do PCA, a escolha da taxa de aprendizado e o termo de momento. Estas discussões estão apresentadas no Apêndice deste trabalho.

O passo seguinte foi o particionamento da massa de dados entre grupos para treinamento e teste. Utilizou-se o mesmo conjunto de dados citado por (SILVA, 2002) acrescido de uma nova amostragem coletada especificamente para esta pesquisa. Houve o cuidado de manter reservado o mesmo conjunto de teste utilizado no Time de Redes para fins de comparação com os experimentos realizados naquele trabalho.

Um estudo foi feito sobre a diversidade entre os agentes selecionados para comporem o comitê e qual a relação existente no tocante a exatidão individual dos agentes. Questionou-se a melhor maneira de combinar agentes neurais de acordo com a avaliação de algumas medidas de diversidade. O desejado seria que agentes selecionados fossem complementares entre si e que sua união potencializasse resultados melhores que os obtidos por apenas um deles.

Com a realização de experimentos com diferentes alternativas de combinações,

observou-se que a composição de comitês contendo agentes com índices relativamente altos de exatidão individual, e conseqüentemente pouco diversos entre si, gera pouco ganho no índice de acerto da classificação final. Tal resultado deve-se, possivelmente, ao fato de que os agentes possuem “opinião” muito parecida sobre os padrões apresentados, tanto nas classificações corretas quanto nas incorretas, não contribuindo, desta forma, para o aumento do índice global de acerto.

Por outro lado, agentes diversos entre si e com índices de exatidão individual muito diferentes, ou seja, a composição de um comitê com agentes com bons índices de acerto e outro(s) com índices bem menores, acarretam resultados menos satisfatórios que aqueles alcançados pelos bons agentes individualmente avaliados. Acredita-se que os agentes com índices menores acabam “atrapalhando” a decisão final do comitê.

Sendo assim, vê-se que uma boa combinação deve contar com agentes diversos, talvez especializados em grupos distintos dentre as classes envolvidas no problema, porém com índices de exatidão individual não tão diferentes entre si.

Para o caso dos dígitos, três tentativas de composição de agentes especializados foram feitas. Na primeira delas, os agentes, no caso redes neurais do tipo MLP com as melhores taxas de acerto alcançadas, foram combinados a uma rede de passagem formando um comitê. A média dos resultados obtidos com o comitê foi de 94,71%. Comparando com a maior média percentual alcançada pelos agentes de modo individual, 94,50% e considerando o desvio padrão de 0,76, obtido com a distribuição dos índices de acerto alcançados nos testes com o comitê, conclui-se que a utilização de uma máquina de comitê, neste caso, não apresentou vantagem sobre o uso de apenas um agente de modo individual.

Já na segunda tentativa, compôs-se o comitê com um agente especializado em distinguir entre 0's e 8's, outro especializado em distinguir os dígitos 2 e 7. O comitê foi composto pelos agentes criados e a rede de passagem e, numa segunda implementação desta tentativa, ao comitê foi acrescido do agente de melhor desempenho da tentativa anterior. o melhor resultado foi a média percentual de 88,07% com a aplicação do comitê composto pelos três agentes em questão, inferior à média de 94,30% obtida pelo agente de melhor desempenho da tentativa anterior testado individualmente.

Numa terceira tentativa, o conjunto de dados foi particionado primeiramente em dois grupos e agentes foram treinados respectivamente com cada grupo gerado. Comitês foram compostos através de combinações dois a dois entre os agentes criados e o agente de melhor desempenho da primeira tentativa, e um último comitê foi composto pelos três agentes em questão. Em seguida, ainda na terceira tentativa com dígitos, todo conjunto foi novamente particionado, desta vez em três grupos. Comitês foram compostos através de combinações três a três entre os agentes criados e o agente de melhor desempenho da primeira tentativa, e um último comitê foi composto pelos quatro agentes em questão. Os melhores resultados foram obtidos com o uso de um comitê formado pelo “Ag.1” e “Ag.G”, com média de 94,65% de acerto. No entanto, com o desvio padrão de 0,67, este resultado também não supera as expectativas de que o emprego desta composição de máquina de comitê tenha vantagem sobre o uso de apenas uma rede neural.

No caso das letras duas tentativas foram realizadas. A primeira reuniu três agentes com índices de acerto os melhores possíveis para a composição do comitê. Nesta etapa, o comitê formado alcançou percentuais em torno de 72,00%, bem abaixo do obtido pelos agentes de modo individual, que giraram em torno de 86,00%. Numa segunda tentativa os comitês foram compostos por agentes especializados em subespaços diferentes do espaço total de entrada. Nesta tentativa, quatro comitês foram compostos, variando o número de agentes, o número de camadas escondidas da rede de passagem e a função de propagação utilizada na camada de saída. Em termos de médias alcançadas pelas diferentes combinações de comitês, a melhor foi de 89,46%, superior à melhor média obtida com o uso de apenas um agente de modo individual, 81,47%. E, no caso do comitê composto por apenas agentes especializados em subespaços diferentes, a média de acerto foi de 89,28%, superior à melhor média alcançada pelos agentes em questão, 74,73%.

Conclui-se que, de acordo com os resultados obtidos para os dígitos, a composição de uma máquina de comitê, embora na primeira tentativa tenha atingido um índice de acerto superior aos índices individuais dos agentes em questão, não apresenta vantagem quando se avalia o desvio padrão da distribuição dos resultados sobre os conjuntos de teste. Também deve-se considerar que os resultados já obtidos por apenas uma rede neural, mostrados tanto em (SILVA, 2002) quanto neste trabalho,

giram em torno de 95,00% na classificação de padrões de dígitos. Desta forma, torna-se uma tarefa difícil superar tais índices sem o auxílio de outros recursos, como por exemplo, o contexto ao qual o dígito está inserido.

Sendo assim, uma análise sobre os resultados alcançados em cada tentativa contribui para conclusões a respeito de como melhor combinar agentes especialistas e o que esperar sobre o comportamento de uma Máquina de Comitê para este fim. No entanto, nenhum dos modelos pôde ser apontado como vantajoso quando comparado ao emprego de apenas um agente de modo individual.

Já para o caso das letras, tendo em vista os resultados obtidos por agentes neurais de modo isolado, cerca de 86,00%, a criação de um comitê composto por agentes neurais especializados em subespaços distintos do espaço total de entrada, produz resultados médios da ordem de 8% superiores à média de acerto alcançada por um único agente treinado com todo espaço de entrada, e da ordem de 14,7% quando comparado à média alcançada por um dos agentes que foram especializados.

A seguir, a tabela 6.1 traz a comparação para os dígitos entre o índice de acerto alcançado pelo Time de Redes Neurais (SILVA, 2002) e os resultados utilizando os comitês propostos neste trabalho. Sendo que, com relação à terceira tentativa, somente os melhores índices, com dois grupos e com três grupos, estão apresentados. Estes resultados foram extraídos das tabelas 5.4, 5.15, 5.20 e 5.24. Onde,

- Co_1 : Comitê Formado na Primeira Tentativa
- Co_2^2 : Comitê Formado na Segunda Tentativa com 2 Agentes
- Co_2^3 : Comitê Formado na Segunda Tentativa com 3 Agentes
- Co_3^2 : Comitê Formado na Terceira Tentativa com 2 Grupos
- Co_3^3 : Comitê Formado na Terceira Tentativa com 3 Grupos

	Time	Co_1	Co_2^2	Co_2^3	Co_3^2	Co_3^3
Desempenho	94,54%	96,02%	82,34%	85,73%	94,65%	95,97%

Tabela 6.1: Tabela Comparativa entre Desempenho do Time de Redes e Modelos de Comitê - Dígitos

Pelos resultados expostos na tabela 6.1, conclui-se que, com exceção dos comitês compostos na segunda tentativa, todas as outras implementações geraram índices de acerto melhores que o apresentado pelo trabalho de (SILVA, 2002).

O melhor desempenho foi alcançado pelo modelo composto pelos três agentes especializados mais o melhor agente selecionado da primeira tentativa, com apenas uma camada escondida. A média alcançada por este modelo foi de 89,46%.

Foi visto também que o uso de uma função não-linear na camada de saída produz resultados melhores, e que a inclusão de mais uma camada escondida na rede de passagem não resulta em aumento nos índices alcançados com apenas uma camada.

A tabela 6.2 traz a comparação para as letras entre o índice de acerto alcançado pelo Time de Redes Neurais (SILVA, 2002) e o resultado utilizando o comitê proposto neste trabalho com melhor índice de acerto para este conjunto de teste. Onde,

- Co_2 : Comitê Formado na Segunda Tentativa - com quatro agentes e rede de passagem com uma camada

	Time	Co_2
Desempenho	91,91%	93,61%

Tabela 6.2: Tabela Comparativa entre Desempenho do Time de Redes e Modelos de Comitê - Letras

Pelos resultados expostos na tabela 6.2, vê-se que o comitê proposto neste trabalho alcançou um índice de acerto maior em 1,7% que o apresentado pelo trabalho de (SILVA, 2002).

Ao final do capítulo 5, foi feito um estudo sobre o percentual de acerto de um ser humano na classificação de caracteres manuscritos vistos sem contexto, com o uso dos mesmos conjuntos de caracteres utilizados no treinamento e testes dos comitês implementados. Os índices alcançados tanto para as letras quanto para os dígitos permitiram a realização de uma comparação, em termos de capacidade de classificação, entre a Máquina de Comitê e o Ser Humano.

Concluiu-se que a ferramenta computacional, embora esteja aquém do desempenho alcançado pelo ser humano, e não era esperado que fosse ao contrário, apresentou bons resultados de classificação, vide a tabela 5.35.

6.1 Trabalhos Futuros

Para estudos posteriores, seria interessante a tentativa de composição de uma Máquina de comitê composta por Agentes especializados e um ou mais destes agentes fosse um Time de Redes, realizando a junção das duas arquiteturas.

Um segundo trabalho interessante seria o estudo sobre um modelo computacional que classificasse caracteres alfanuméricos.

Outra pesquisa necessária e interessante para completar o modelo seria a introdução de um módulo de interpretação dos resultados do comitê que leve em consideração o contexto, ou seja, regras gramaticais e a semântica das palavras.

A seguir, como uma breve demonstração da utilização do contexto no qual uma palavra está inserida, apresenta-se um texto que circulou na internet por algum tempo, que é perfeitamente compreendido pelo ser humana apesar das palavras estarem escritas com letras embaralhadas, mantendo apenas a primeira e a última na posição correta:

“Para quem sabe ler, não precisa pingar o I nem cortar o T. Vejam como é interessante nosso cérebro:

De acordo com uma pesquisa de uma universidade inglesa, não importa em qual ordem as letras de uma palavra estão, a única coisa importante é que a primeira e a última letras estejam no lugar certo.”

Como verificado através da leitura do texto acima, o contexto ao qual uma palavra pertence oferece uma grande contribuição para sua interpretação.

REFERÊNCIAS

- ADEVA, J.J.G . **Accuracy and Diversity in Ensembles of Text Categorisers**, CLEI Electronic Journal, vol 9, pp 1-12, 2005. Disponível em: <<http://www.clei.cl/cleiej/paper.php?id=109>>. Acesso em: maio de 2006.
- AIRES, S,B,K. **Reconhecimento de Caracteres Manuscritos Baseado em Regiões Perceptivas**, Dissertação (Mestrado em Informática Aplicada), PUCPR/PPGIA, Curitiba, Paraná, 82f, 2003.
- ARICA, N. ; VURAL, F. T. Y. **An Overview of Character Recognition Focused on Off-Line Handwriting**. Ieee Transactions on Systems, Man, and Cybernetics - Part C:Applications and Reviews, vol. 31, no 2, pp. 216-233, maio 2001.
- BARBOZA, J. L. L. **Estudo de Algoritmos de Categorização Visando Reconhecimento Automático de Caracteres Manuscritos**, Monografia (Projeto Final de Curso), Universidade Federal do Rio de Janeiro - UFRJ/IM/NCE, 143f., 2002.
- BISHOP, C. M . **Neural Networks For Pattern Recognition.**, Oxford University, Press, 1995
- BRAGA, A P.; *et al.* **Fundamentos de Redes Neurais Artificiais**. IN: 11A Escola de Computação, Rio de Janeiro, JULHO 1998.
- COX,E. **Fuzzy Systems Handbook**, AP Professional, 1995. ISBN 0-12-194270-8
- DUIN, A P.; *et al.* **The Characterization of Classification Problems by Classifier Disagreements**. IN: ICPR, pp.140-143, Cambridge, UK, 2004.
- FRANCO, C. R. **Novos Métodos de Classificação Nebulosa e de Validação de Categorias e Suas Aplicações a Problemas de Reconhecimento de Padrões**, Dissertação (Mestrado em Informática) Universidade Federal do Rio de Janeiro, IM/NCE, 133f, 2002.
- FREIXINHO, M. A. P. **Reconhecimento De Dígitos Manuscritos Por Redes Neurais**, Dissertação (Mestrado em Sistemas e Computação) Pontifícia Universidade Católica do Rio de Janeiro, 162f, 1996.
- GONZALEZ, R. C. WOODS R.E. **Processamento de Imagens Digitais**, Ed. Edgard Blücher Ltda, 2000.

- GOPISETTY, S. *et al.* **Automated Forms - Processing Software and Services**. IBM J. Res. Develop., V. 40, No 2, pp. 211-230, Março 1996.
- HADJITODOROV, S.T. *et al.* **Moderate Diversity for Better Cluster Ensemble**. Disponível em
<http://www.informatics.bangor.ac.uk/Kuncheva/recent_publications.htm.>
Acessado em maio de 2006.
- HANMANDLU, M. *et al.* **Unconstrained handwritten character recognition based on fuzzy logic**, Pattern Recognition, vol 36, no 3, pp 603-623, 2003.
- HAYKIN, S. **Redes Neurais: Princípios E Prática**, 2a. Ed. Porto Alegre: Bookman, 2001. 900 P. Il.
- JAIN, A.K.. *et al.* **Statistical Patern Recognition - A Review**, In IEEE, Transaction Patern Analysis Machine Intell, Vol. 22, pp.4-38, jan 2000.
- KOHAVI, R. & WOLPERT, D. **Bias Plus Variance Decomposition for Zero-One Loss Functions**, In L.Saitta, Machine Learning: Proceedings 13th International Conference, pp.275 - 283, 1996.
- KOSKO, B. . **Fuzzy Thinking**, HarperCollins Publishers, 1994. ISBN 0-00-654713-31s
- KUNCHEVA, L.I. & WHITAKER, C.J. **Measures of Diversity in classifier Ensembles and Their Relationship Whit The Ensemble Accuracy**, Machine Learning, Vol 51, No 2, pp. 181-207,2003.
- KUNCHEVA, L.I. **Measures of Diversity**, Machine Learning, Vol 51, No 2, pp. 181-207,2005.
- MATOS, L.N. **Utilização de Redes Bayesianas Como Agrupador de Classificadores Globais e Locais**, Dissertação (Doutorado em Pós Graduação em Engenharia Elétrica) Universidade Federal de Campina Grande, Campina Grande, Paraíba, 2004.
- MORITA, M. *et al* **An HMM-MLP Hybrid System to Recognize Handwriting Dates**, Neural Networks, IJCNN'02. Proceedings of the 2002, V.1, Pp. 867-872, 2002.
- OH, I. & SUEN, C.Y. **A Class-Modular Feedfoward Neural Network For Handwriting Recognition**, Pattern Recognition, V. 35, Pp. 229-244, 2002.
- OLIVEIRA, L.S., *et al.* **Automatic Recognition Of Handwritten Numerical Strings: A Recognition E Verification Strategy**. IEEE Transactions On Pattern Analysis And Machine Intelligence, V. 24, N. 11, Pag. 1438-1454, Novembro 2002.
- OLIVEIRA, L.S., *et al.* **Handwritten Month Word Recogniton Using Multiple Classifiers** XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBIGRAPI), Curitiba, Brasil, Outubro, 2004.

PAVIDILIS, T. **Algorithms for Graphics and Image Processing**, Computer Science Press, Inc., 1982.

PLAMONDON, R. & SRIHARI S. N., **On-Line And Off-Line Handwriting Recognition: A Comprehensive Survey**, Ieee Transactions On Pattern Analysis And Machine Intelligence, V. 22, N. 1, Pag. 63-84, Janeiro 2000.

RODRIGUES, J.R. & THOMÉ, A.G., **Cursive Character Recognition - A Character Segmentation Method Using Projection Profile-Based Technique**, The 6th International Conference on Information Systems, Analysis and Synthesis ISAS 2000, Orlando, August 2000, Proceedings, Vol. 5, pp. 274-279.

RODRIGUES, J.R., **Segmentação E Extração de Características para Reconhecimento Automático de Caracteres - Estudo e Propostas**, Dissertação (Mestrado Em Informática) Universidade Federal Do Rio De Janeiro, Rio De Janeiro, 169f, 2003.

ROSSETO, S., **Uma Classificação Do Conhecimento Usado Para Melhorar O Desempenho De Um Sistema De Reconhecimento De Manuscrito Em Formulário**. 2001, Dissertação (Mestrado Em Ciência Da Computação) Universidade Federal Do Espírito Santo, Vitória, 2001.

SRIHARI S. N. & LAM, S W., **Character Recognition**. Disponível em: <<http://www.cedar.buffalo.edu/papers/pubs1995.html>> Acesso em: 30 de julho de 2006.

SHRIDHAR, M.; KIMURA, F., **Handwritten Numerical Recognition Based On Multiple Algorithms**, Pattern Recognit., Vol 24, No. 10, pp. 969-983, 1991.

SILVA, E., **Reconhecimento Inteligente de Caracteres Manuscritos**, Dissertação (Mestrado em Sistemas e Computação) - Instituto Militar de Engenharia, Rio de Janeiro, 168f, 2002.

SUEN, C.Y., *et al*, **Automatic Recognition of Handprinted Characters - The State Of The Art**. Proc. IEEE, Vol. 68, pp. 469-487, Abril, 1980.

SUEN, C.Y., *et al*, **Handwriting Recognition The Last Frontiers**, Proc. Int. Conf. Pattern Recognition, Vol. 4, pp. 1-10, Barcelona, Setembro 2000.

TRIER, O.D., *et al*. **Feature Extraction Method for Character Recognition - A Survey**, Pattern Recognition, Vol. 29, no. 4. pp. 641-662, 1996.

TSOULAKAS, L.H. & UHIG, R.E. **Fuzzy and Neural Approaches in Engineering**, John Wiley and Sons, Inc, 1997, ISBN 0-471-16003-2

VERMA, B., *et al*, **Recent Achievements In Off-Line Handwriting Recognition Systems**, International Conference On Computational Intelligence And Multimedia Applications (ICCIMA98), pp. 27-33, 1998.

VIANNA, G. K., **Sistemas Inteligentes Para Reconhecimento De Padrões**, Dissertação (Mestrado Em Informática) Universidade Federal Do Rio De Janeiro, Rio De Janeiro, 156f, 2000.

WINDEATT, T., **Diversity Measures for Multiple Classifier System Analysis and Design** Information Fusion, Vol 6, No 1, pp. 21-36, 2005

XU, L. *et al*, **Methods Of Combine Multiple Classifiers And Their Applications To Handwriting Recognition**. Ieee Transactions On Systems, Man, And Cybernetics, V. 22, N 3, PP. 418-435, Maio 1992.

ZURADA, J., **Introduction To Artificial Neural Systems**, West Publishing Company, 1992.

7 APÊNDICE - EXPERIMENTOS COM REDES MLP

7.1 Introdução

O uso de redes neurais como técnica para reconhecimento de padrões, certamente já tem sido difundido devido ao enorme poder de transformar um espaço n dimensional de vetores de características em um número bem definido de classes, que representam os diferentes tipos de objetos apresentados à rede. No entanto, explorar todo o poder de discriminação de uma rede neural requer, através de exaustivas tentativas, a busca do melhor ajuste dos parâmetros envolvidos nos cálculos de treinamento desta rede.

Com o objetivo de estudarmos a melhor forma de configurar uma rede neural do tipo MLP para o reconhecimento de caracteres manuscritos, foram realizados experimentos que visavam à avaliação dos resultados obtidos em diferentes abordagens relativas tanto aos parâmetros e arquitetura de uma rede neural para este fim, quanto à apresentação da massa de dados a rede.

Todo o treinamento de redes MLP foi feito com massa de dados de caracteres manuscritos, escritos em quadrículas de formulários, segmentados de modo isolado, e cujas características foram extraídas através da técnica do Quadrado Rotacionado. Este conjunto de dados foi o mesmo utilizado em (SILVA, 2002), para o treinamento e teste de um Time de Redes.

A princípio, os vetores estavam compostos por 256 características, porém, após uma avaliação de todas as colunas da matriz de padrões e características, identificação e eliminação de colunas zeradas (sem informação), restaram 232 atributos.

Do experimento constaram as quatro etapas abaixo definidas:

- 1ª etapa: O objetivo nesta fase foi o de treinar redes MLP com diferentes parâmetros de treinamento e arquiteturas a fim de que a melhor configuração fosse apontada para posterior utilização na próxima fase deste experimento.
- 2ª etapa: Nesta etapa, a rede com melhor taxa de acerto nos testes foi utilizada para a realização de uma análise de sensibilidade quanto à importância das variáveis que representam as características dos padrões apresentados a rede.
- 3ª etapa: Sem a retirada de nenhuma característica dos padrões apresentados à rede, uma Análise dos Principais Componentes (PCA), foi feita sobre o conjunto de dados a fim de que, através do descorrelacionamento das variáveis em estudo, a rede pudesse ser treinada e testada com um número reduzido de variáveis e, conseqüentemente, de pesos sinápticos, visando uma economia

quanto ao tempo necessário para tais execuções. Antes da busca de uma melhor rede aplicando-se o PCA, foi feito um estudo sobre qual o fator de corte seria mais adequado ao conjunto de dados avaliado.

- 4ª etapa: Nesta fase, foi feita uma análise de sensibilidade sobre as variáveis resultantes da aplicação do PCA. As quatro etapas foram executadas tanto para o reconhecimento de dígitos manuscritos, quanto para letras.

7.2 Experimentos para Reconhecimento de Dígitos Manuscritos

7.2.1 Descrição dos Experimentos e Resultados

A seguir, o relato dos resultados obtidos em cada etapa do experimento com uso de redes neurais para reconhecimento de dígitos manuscritos, relativos a cada etapa realizada.

1ª etapa Treinamento de Redes MLPs

Para as fases de execução das redes neurais foram utilizadas as seguintes quantidades de dados descritos na tabela 7.1:

Fase de Execução da Rede	Nº de Padrões
Treinamento	4000
Validação	500
Teste	3720

Tabela 7.1: Quantidade de dígitos para o treinamento, teste e validação das Redes Neurais

Os dados foram apresentados à rede, normalizados pela técnica de normalização conhecida como Zscore. Para isso, fez-se uso das funções do Matlab 6.5, r13: “prestd” e “trastd”.

Após alguns treinamentos utilizando-se Taxa de Aprendizado=0,7 e Momentum=0,3 a melhor rede obteve taxa de acerto no teste de 93,60%, parando sempre o treinamento no número de épocas. Ver características da rede neural implementada na tabela 7.2

No intuito de tentar melhorar as taxas de acerto no teste até então obtidas, os valores da taxa de aprendizado e momentum foram alterados para 0,3 e 0,5, respectivamente. Com isso, observou-se uma notável alteração no comportamento das redes. A rede começou a convergir atingindo o alvo de erro mínimo, utilizando cerca de um quarto do número de épocas até então utilizadas. O melhor resultado está descrito na tabela 7.3 e sua matriz de confusão representada pela tabela 7.4.

Portanto, a rede selecionada para realização dos testes para próxima fase foi a de número 11, com taxa de acerto de 95,65%.

2ª Etapa Testes para realização de análise de sensibilidade

Método de Extração de Características	Quadrado Rotacionado 128
Número de Entradas da Rede	232
Número de Saídas da Rede	10
Número de Camadas da Rede	2
Neurônios na Camada Escondida	48
Função de Treinamento	traingdx
Taxa de Aprendizado	0,70
Momentum	0,30
Número de épocas	2000

Tabela 7.2: Parâmetros de Treinamento das Redes Neurais

NÚMERO DA REDE	11
Número de Épocas Realizadas	384
Classificações Corretas	3558
Classificações Incorretas	162
Rejeições	0
Porcent. de Classificações Corretas	95,65
Porcent. de Classificações Incorretas	4,35
Porcent. de Rejeições	0.00
Número de épocas	2000

Tabela 7.3: Resultados da Melhor Rede Neural Obtida - Rede nº11

	0	1	2	3	4	5	6	7	8	9
0	377	0	0	0	0	0	1	0	20	0
1	1	402	4	1	5	0	1	1	3	4
2	0	2	477	0	2	1	0	2	2	0
3	0	1	2	63	0	2	0	0	3	3
4	2	2	0	0	244	0	4	11	0	2
5	0	1	0	2	0	291	2	0	3	3
6	0	0	0	0	2	4	303	0	3	0
7	0	1	3	2	1	0	0	388	3	3
8	4	2	4	1	0	1	4	0	624	7
9	2	2	1	1	4	0	0	9	5	389

Tabela 7.4: Matriz de Confusão Resultante da Rede Neural Selecionada - Rede nº11

Executou-se 232 vezes (número de atributos da massa de dados) o teste da análise de sensibilidade, utilizando a melhor rede obtida na etapa 1. O teste consistiu em retirar-se uma variável de entrada a cada rodada, (toda a linha relativa a variável em questão recebeu o valor zero), e observar-se o desempenho da rede. A tabela 7.5 apresenta a lista das vinte variáveis menos relevantes e as vinte que mais relevantes no desempenho da rede. Com base nestas porcentagens obtidas, concluiu-se que todas as variáveis têm um impacto relativamente igual no resultado final, visto que estes valores pouco são alterados quando comparados uns aos outros. A diferença da taxa de acerto está na casa dos décimos e centésimos.

(a)		(b)	
Nº variável	% acerto	Nº variável	% acerto
50	95,22	179	95,67
75	95,22	187	95,67
216	95,22	189	95,67
96	95,24	218	95,67
103	95,24	226	95,67
67	95,27	11	95,70
95	95,27	12	95,70
102	95,30	26	95,70
138	95,30	33	95,70
46	95,32	112	95,70
69	95,32	157	95,70
104	95,32	181	95,70
48	95,35	202	95,70
49	95,35	18	95,73
94	95,35	28	95,73
230	95,35	58	95,73
78	95,38	64	95,73
79	95,38	19	95,75
97	95,38	24	95,75
105	95,38	116	95,78

Tabela 7.5: Na tabela a) as variáveis mais relevantes e, em b) as variáveis menos relevantes no desempenho da rede neural - Dígitos

3ª Etapa Aplicação de PCA

As funções `prepca` e `trapca` (Matlab 6.5, r13) foram utilizadas para a realização de uma “Análise dos Principais Componentes”, PCA, com relação aos atributos considerados sobre as características dos dígitos. Vale lembrar que, antes da aplicação do PCA, o número de atributos era de 232.

Primeiramente, a fim de se definir qual o valor do fator de corte que proporcionasse a melhor representatividade das variáveis descorrelacionadas, verificou-se os resultados da taxa de acerto no teste para valores diferentes de fatores de corte de PCA, a saber:

Fator de corte	Número de Atributos Restantes	Melhor %Teste	Pior %Teste
0,001	90	95,24%	94,54%
0,0005	119	95,24%	95,05%
0,0001	182	95,67%	95,27%
0,00005	199	95,70%	95,13%
0,00004	205	95,83%	95,43%
0,00001	222	95,48%	95,22%

Tabela 7.6: Taxas de Acerto Relativas ao Fator de Corte Utilizado - Dígitos

Através de uma análise dos melhores resultados relativos a cada valor (vide tabela 7.6), empiricamente escolhido para o fator de corte, decidiu-se pelo uso de um fator igual a 0.00004.

4ª Etapa Análise De Sensibilidade Após a Aplicação De PCA

Foi feita uma análise de sensibilidade, agora com atributos resultantes da aplicação do PCA. Os resultados revelaram que, após o descorrelacionamento das variáveis, feito pela aplicação do PCA, as características representadas pelas “novas” variáveis são de extrema importância para a convergência da rede e taxa de acerto nos testes. A seguir, as listas das variáveis que menos influenciam, porém o impacto de sua retirada do conjunto de características é alto comparando a taxa de acerto alcançada sem a retirada de nenhuma variável, que foi de 95,83%. Veja a tabela 7.7

(a)		(b)	
Nº variável	% acerto	Nº variável	% acerto
1	91,29	190	25,89
3	90,40	186	25,86
2	88,01	185	25,83
4	86,40	191	25,83
5	79,89	188	25,75
6	73,39	189	25,67
7	69,76	197	25,43
8	67,63	199	25,35
9	59,73	196	25,30
10	56,21	202	25,30
11	54,03	201	25,24
12	51,91	200	25,19
13	48,39	203	25,19
14	47,34	192	25,16
15	45,46	198	25,16
16	44,92	205	25,16
17	44,35	204	25,08
18	43,92	195	24,97
19	43,49	194	24,92
20	42,96	193	24,81

Tabela 7.7: (a) as variáveis menos relevantes e, (b) as variáveis mais relevantes no resultado da análise de sensibilidade após aplicação de PCA - Dígitos.

7.2.2 Experimento com Dígitos Manuscritos - Conclusões

Para o reconhecimento de dígitos manuscritos, uma rede do tipo MLP foi selecionada, atingindo taxa de acerto de 95,83%, fazendo uso de PCA, com fator de corte relativamente pequeno, 0.00004. Nas tabelas 7.8 e 7.9, as características do treinamento e teste da melhor rede após uso do PCA e a matriz de confusão respectiva:

A princípio, acreditava-se que a diferença nos resultados obtidos devia-se a uma escolha inadequada para a taxa de aprendizado inicial, 0.7, visto que utilizava uma

Método de Extração de Características	Quadrado Rotacionado 128
Número de Entradas da Rede	205
Número de Saídas da Rede	10
Número de Camadas da Rede	2
Neurônios na Camada Escondida	45
Função de Treinamento	traingdx
Taxa de Aprendizado	0.30
Momentum	0.50
Número de épocas	1000
Número de Épocas Realizadas	449
Classificações Corretas	3565
Classificações Incorretas	155
Rejeições	0
Porcent de Classificações Corretas	95,83
Porcent de Classificações Incorretas	4,17
Porcent de Rejeições	0,00

Tabela 7.8: Resultados da Melhor Rede Neural Obtida Após PCA - Dígitos

	0	1	2	3	4	5	6	7	8	9
0	382	1	0	0	0	0	1	0	18	0
1	1	398	4	1	5	0	1	1	2	3
2	0	5	479	0	1	0	0	2	4	0
3	0	1	0	65	0	2	0	1	3	2
4	2	3	0	0	243	0	2	7	0	3
5	0	1	0	1	0	293	2	0	4	0
6	0	1	0	0	2	2	303	0	5	0
7	0	1	3	0	1	1	0	389	1	2
8	1	0	4	2	1	0	6	2	621	9
9	0	2	1	1	5	1	0	9	8	392

Tabela 7.9: Matriz de Confusão Resultante da Melhor Rede Neural Após PCA - Dígitos

função de otimização no treinamento “traingdx” (Matlab 6.5, r13) Gradiente Descendente com taxa de Aprendizado adaptativa. Porém quando alterado o valor da taxa de aprendizagem inicial para 0.3, e do termo de momento, no Time de Redes (SILVA, 2002) 0.3, para 0.5, a convergência no treinamento foi muito melhor com relação ao erro mínimo alcançado, número de épocas necessárias e, conseqüentemente, o tempo utilizado. Desta forma, durante o treinamento da rede houve um aproveitamento melhor da potencialidade do Gradiente Descendente com taxa de Aprendizado adaptativa associado ao uso do termo de momento maior.

7.3 Experimento para Reconhecimento de Letras Manuscritas

7.3.1 Descrição dos Experimentos e Resultados

A seguir, o relato dos resultados obtidos em cada etapa do experimento com uso de redes neurais para reconhecimento de letras manuscritas, relativos a cada etapa realizada.

1ª etapa Treinamento de Redes MLPs

Para as fases de execução das redes neurais foram utilizadas as seguintes quantidades de dados listadas na tabela 7.10:

Fase de Execução da Rede	Nº de Padrões
Treinamento	10400
Validação	1300
Teste	8843

Tabela 7.10: Quantidade de Letras para o Treinamento, Teste e Validação das Redes Neurais

Os dados foram apresentados à rede, normalizados pela técnica de normalização conhecida como Zscore. Para isso, fez-se uso das funções do Matlab 6.5, `r13: prestd` e `trastd`.

Para a definição dos parâmetros das redes tipo MLP que seriam treinadas, foram feitos dois tipos de análises. Na primeira, avaliou-se qual o número de neurônios na camada escondida melhor se adequaria, e conseqüentemente, produziria melhores taxas de acerto no treinamento e teste das redes neurais para o caso do reconhecimento de letras manuscritas.

Estudos anteriores, (BRAGA *et al*, 2000), (SILVA, 2002), sugerem o cálculo deste número através de uma média geométrica entre o número entradas e saídas da rede. A fim de comparação de resultados, foram treinadas redes com número de neurônios na camada escondida calculados através, tanto da média geométrica (77 neurônios), como da média aritmética (129 neurônios) entre o número de entradas e saídas. E, para um terceiro teste com u número menor de neurônios, calculados como duas vezes o número de saídas (52 neurônios). A escolha destes métodos de cálculo foi empírica, embora tenha-se procurado valores maiores e menores do que o número obtido quando comumente aplicada a média geométrica. Ver tabela 7.11.

Método de Cálculo	Melhor Taxa de Acerto nos Testes
2x Número de Saídas	91,68%
Média Geométrica	87,91%
Média Aritmética	92,69%

Tabela 7.11: Resultados no Cálculo do Número de Neurônios na Camada Escondida - Letras

Através da análise feita, concluiu-se que o uso da média aritmética, como método para calcular o número de neurônios utilizados na camada escondida, foi o

mais adequado ao problema em questão. A tabela 7.12 mostra as características e resultados da melhor rede obtida. A figura 7.1 mostra a matriz de confusão relativa a rede selecionada que alcançou o desempenho de acerto global de 92.69%.

Método de Extração de Características	Quadrado Rotacionado 128
Número de Entradas da Rede	232
Número de Saídas da Rede	26
Número de Camadas da Rede	2
Neurônios na Camada Escondida	129
Função de Treinamento	traingdx
Taxa de Aprendizado	0.30
Momentum	0.50
Número de épocas	1000
Número de Épocas Realizadas	758
Classificações Corretas	8197
Classificações Incorretas	646
Rejeições	0
Porcent de Classificações Corretas	92,69
Porcent de Classificações Incorretas	7,31
Porcent de Rejeições	0,00

Tabela 7.12: Resultados da Melhor Rede Neural Obtida - Letras

2ª Etapa Testes para realização da análise de sensibilidade

Utilizou-se a rede com melhor taxa de acerto no teste para a realização da análise de sensibilidade para o caso das variáveis relativas as letras. Após a retirada de um atributo por vez, a massa de dados era submetida ao teste com a rede neural escolhida, a taxa de acerto calculada, e na próxima rodada o atributo antes retirado, era recolocado e o atributo seguinte retirado. Analisando o quadro a seguir, vemos as variáveis de menos relevância a esquerda e na direita, aquelas que, quando retiradas causam um considerável queda na taxa de acerto nos testes.

3ª Etapa Aplicação de PCA

As funções “prepc” e “trapca” (MATLAB) foram utilizadas para a realização de uma Análise dos Principais Componentes, PCA, com relação aos atributos considerados sobre as características dos dígitos. Primeiramente, a fim de se definir qual o valor do fator de corte que proporcionasse a melhor representatividade das variáveis decorrelacionadas, verificou-se os resultados da taxa de acerto no teste para valores diferentes de fatores de corte de PCA, a saber:

Através de uma análise dos melhores resultados relativos a cada valor (vide tabela 7.14), escolhido para o fator de corte, decidiu-se pelo uso de um fator igual a 0.00004. Desta forma, os testes a seguir foram realizados com o uso de 217 atributos relativos aos padrões de entrada para as redes neurais.

4ª Etapa Análise De Sensibilidade Após a Aplicação De PCA

ESPERADO \ ESTIMADO	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	W	V	X	Y	Z	
A	772	0	0	0	0	0	1	1	0	0	0	0	5	1	1	0	0	10	0	0	0	0	0	1	0	0	0
B	5	237	0	11	3	0	2	0	1	1	1	0	0	0	10	0	1	5	1	0	0	0	0	0	0	0	3
C	0	0	203	0	10	0	0	0	1	0	0	2	0	0	1	0	0	9	2	0	0	0	0	0	0	0	3
D	2	12	0	253	0	0	0	0	0	0	0	0	0	0	13	3	2	1	0	0	0	0	0	1	0	3	0
E	2	0	1	0	288	0	4	0	12	0	2	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	5
F	0	0	0	0	0	88	0	0	1	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	2
G	0	1	1	0	14	0	244	0	0	1	1	1	0	0	3	0	1	0	2	0	0	0	0	0	0	0	0
H	4	0	0	0	0	0	1	246	0	0	1	0	23	11	0	0	0	0	0	0	0	0	1	1	0	0	0
I	0	0	0	0	0	0	0	1	266	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	25	272	0	0	0	0	0	0	0	0	4	0	0	0	0	0	3	4
K	0	1	0	0	1	0	0	1	0	0	304	1	4	0	0	0	0	9	0	0	0	0	0	0	9	0	4
L	3	0	3	0	0	0	1	0	0	0	0	337	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1
M	15	0	0	0	0	0	0	17	0	0	5	0	305	4	1	0	0	2	0	0	0	0	0	2	0	0	0
N	2	0	0	0	1	0	0	4	0	1	0	0	10	391	0	0	0	2	0	0	0	0	0	4	0	0	0
O	1	5	0	13	0	0	2	0	0	0	0	0	0	0	412	0	10	0	0	0	0	0	0	0	0	0	0
P	0	0	0	2	0	0	0	0	1	0	0	0	1	0	2	296	0	4	0	0	0	0	0	2	0	0	1
Q	5	6	1	1	1	0	1	2	0	0	0	0	0	0	14	1	248	11	0	0	0	0	0	0	0	0	2
R	4	0	0	0	1	1	0	0	1	0	4	0	1	0	0	0	6	501	0	0	0	0	0	0	1	0	4
S	0	0	0	0	0	0	0	0	2	6	0	0	0	0	1	0	0	0	378	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	3	0	1	0	89	0	0	0	0	2	1	0
U	0	0	0	0	0	0	1	0	0	1	0	0	0	4	3	0	1	0	0	0	210	9	3	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	282	2	1	1	0	0
V	1	1	0	0	0	0	0	0	0	1	1	0	0	15	1	0	0	0	0	0	1	1	346	0	0	0	0
X	0	0	0	0	0	0	0	0	13	2	8	0	0	0	0	0	0	1	0	0	0	0	0	0	334	0	4
Y	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	6	0	4	373	0	0
Z	0	0	0	0	1	0	0	0	17	2	0	0	0	0	1	0	1	5	1	0	0	0	0	0	0	0	522

Figura 7.1: Matriz de Confusão da Rede Neural Selecionada - Letras

Foi feita uma análise de sensibilidade, agora com atributos resultantes da aplicação do PCA. Os resultados revelaram que, após o decorrelacionamento das variáveis, feito pela aplicação do PCA, as características representadas pelas “novas” variáveis são de extrema importância para a convergência da rede e taxa de acerto nos testes. Ver tabela 7.15

Analisando a tabela 7.15, observa-se as variáveis de menos relevância a esquerda, porém o impacto de sua retirada do conjunto de características é alto comparando a taxa de acerto alcançada sem a retirada de nenhuma variável, que foi de 92,85%. À direita, as variáveis que, quando retiradas causam uma considerável queda na taxa de acerto nos testes.

7.3.2 Experimento com Letras Manuscritas - Conclusões

Para o reconhecimento de letras manuscritas, uma rede do tipo MLP foi selecionada, atingindo taxa de acerto de 92,85%, fazendo uso de PCA, com fator de corte relativamente pequeno, 0,00004. A tabela 7.16 traz as características do treinamento e teste da melhor rede apontada. A figura 7.2 mostra a matriz de confusão gerada.

(a)		(b)	
Nº variável	% acerto	Nº variável	% acerto
1	92,81	210	9,62
3	92,79	211	9,53
2	92,77	215	8,71
4	92,66	216	7,59
5	92,45	228	7,09
6	92,31	217	6,81
7	92,30	218	4,49
8	92,02	230	3,91
9	91,58	231	3,86
10	91,29	229	3,78
11	91,04	219	3,61
12	90,52	227	3,61
13	89,18	220	3,36
14	87,11	226	3,20
15	84,76	232	3,08
16	82,61	221	2,97
17	81,09	222	2,87
18	80,79	223	2,85
19	80,35	224	2,85
20	79,86	225	2,60

Tabela 7.13: (a) as variáveis menos relevantes e, (b) as variáveis mais relevantes no resultado da análise de sensibilidade - Letras.

Fator de corte	Número de Atributos Restantes	Pior %Teste	Melhor %Teste
0,005	29	80,96	87,31
0,001	92	91,06	91,84
0,0005	126	91,79	92,19
0,0001	199	92,00	92,80
0,00004	217	92,15	92,85
0,00001	229	92,27	92,71

Tabela 7.14: Taxas de Acerto Relativas ao Fator de Corte Utilizado - Letras

7.4 Experimento Complementar

Uma das dificuldades no uso do algoritmo backpropagation é a convergência para mínimos locais, pontos na superfície de erro que apresentam soluções estáveis, embora não sejam a saída correta. A fim de minimizar esta dificuldade e acelerar o algoritmo backpropagation, algumas técnicas podem ser adotadas, tais como: a introdução do termo de momento, a adição de nós intermediários, a inclusão de ruído aos dados ou a utilização de uma taxa de aprendizado adaptativa (BRAGA *et al*, 2000).

Este estudo visa analisar a influência das taxas de aprendizado e momento no treinamento e conseqüentes taxas de acerto nos testes.

(a)		(b)	
Nº variável	% acerto	Nº variável	% acerto
1	84,08	200	3,27
2	73,99	193	3,26
3	69,60	199	3,26
4	67,14	201	3,26
5	57,92	202	3,21
6	48,93	203	3,21
7	42,45	204	3,21
8	38,10	205	3,17
9	34,03	206	3,17
10	33,29	207	3,17
11	30,14	208	3,17
12	29,50	209	3,13
13	27,82	210	3,11
14	27,19	211	3,11
15	25,35	216	3,10
16	23,26	212	3,09
18	22,39	213	3,09
17	22,28	214	3,09
19	22,09	215	3,09
20	20,16	217	3,08

Tabela 7.15: (a) as variáveis menos relevantes e, (b) as variáveis mais relevantes no resultado da análise de sensibilidade após aplicação de PCA - Letras.

A massa de dados de entrada utilizada para este experimento continha 232 atributos para cada padrão apresentado a rede. Com a aplicação do PCA (usando as funções do Matlab 6.5, r13: `prepca` e `trapca`), utilizando um fator de corte igual a 0.00004, o número de atributos foi reduzido para 205.

Os dados foram apresentados à rede, normalizados pela técnica de normalização conhecida como Zscore. Para isso, fez-se uso das funções do Matlab 6.5, r13: `prestd` e `trastd`.

O experimento consistiu da criação de uma rede neural MLP, contendo uma camada escondida. Os padrões de treinamento foram apresentados a rede somente uma vez e o treinamento interrompido. Os pesos das camadas de entrada e intermediária foram congelados, a fim de serem utilizados como pesos iniciais para todos os treinamentos utilizando diferentes combinações de valores para as taxas de aprendizado e momento. Cada rede neural tinha como critério de parada o erro médio quadrático no valor de 0.001 ou um número máximo de 1000 épocas. Outra característica importante do treinamento das redes neurais foi o uso da função do Matlab “`traingdx`”, que faz a busca na superfície de erro através da técnica de otimização do Gradiente descendente com a taxa de aprendizado adaptativa e um valor para o termo de momento.

Para cada rodada, variando os valores das taxas de aprendizado e momento, foram salvos o tempo de treinamento e a porcentagem de acerto no teste. A tabela 7.17 traz estes resultados:

Método de Extração de Características	Quadrado Rotacionado 128
Número de Entradas da Rede	217
Número de Saídas da Rede	26
Número de Camadas da Rede	2
Neurônios na Camada Escondida	121
Função de Treinamento	traingdx
Taxa de Aprendizado	0.30
Momentum	0.50
Número de Épocas	1000
Número de Épocas Realizadas	790
Classificações Corretas	8211
Classificações Incorretas	632
Rejeições	0
Porcent de Classificações Corretas	92,85
Porcent de Classificações Incorretas	7,15
Porcent de Rejeições	0,00

Tabela 7.16: Resultados da Melhor Rede Neural Obtida Após PCA - Letras

ESPERADO \ ESTIMADO	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	W	V	X	Y	Z
A	772	0	0	0	1	0	1	3	0	0	2	0	3	0	0	0	0	13	0	0	0	0	0	1	0	0
B	5	241	0	7	2	0	1	1	0	1	0	0	0	0	9	0	2	5	0	0	0	0	0	0	0	4
C	0	0	201	0	8	0	0	0	0	0	0	2	0	0	1	0	0	9	4	0	0	0	0	0	0	4
D	1	11	0	248	0	0	0	0	0	0	0	0	0	0	15	2	1	2	0	0	0	0	0	0	2	0
E	3	0	2	0	288	0	0	0	5	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	7
F	2	0	0	0	0	87	0	0	2	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
G	1	0	2	0	13	0	249	0	0	0	0	1	0	0	3	0	1	0	1	0	0	0	0	0	0	1
H	2	0	0	0	0	0	0	248	0	0	2	0	18	15	0	0	0	0	0	0	0	0	0	2	0	0
I	0	0	0	0	0	0	0	0	266	1	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	24	276	0	0	0	0	0	0	0	0	7	0	0	0	0	0	1	7
K	0	1	0	0	1	0	0	2	0	0	308	0	3	1	0	0	0	6	0	0	0	0	0	9	0	5
L	3	0	3	0	1	0	1	0	0	0	0	338	0	0	0	0	1	1	0	0	0	0	0	0	0	1
M	13	0	0	0	0	0	0	16	0	0	5	0	317	5	0	0	0	3	0	0	0	1	0	0	0	0
N	1	0	0	2	0	0	0	2	0	0	0	0	7	380	0	0	0	2	1	0	0	0	3	0	0	0
O	2	7	0	19	0	0	4	0	0	0	0	0	0	0	414	0	7	0	0	0	0	0	1	0	0	0
P	0	0	0	1	0	1	0	0	2	0	0	0	0	0	2	296	1	1	0	1	0	1	0	0	0	1
Q	5	3	1	1	0	0	0	0	0	0	0	0	0	0	12	1	250	10	0	0	0	0	0	0	0	3
R	5	0	0	0	1	1	0	0	1	0	3	0	1	0	0	0	6	506	0	0	0	0	0	0	0	4
S	0	0	0	1	0	0	0	0	5	5	0	0	0	0	0	0	0	0	373	0	0	0	0	0	0	0
T	0	0	0	0	1	0	0	0	1	3	0	0	0	0	0	3	0	1	0	87	0	0	0	0	2	0
U	0	0	0	0	0	0	1	0	0	1	0	0	0	7	3	0	1	0	0	0	212	10	4	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	11	282	5	1	4	0
V	0	0	0	1	0	0	0	0	0	1	0	0	0	12	0	0	0	0	0	0	2	0	346	0	0	0
X	1	0	0	0	1	0	0	0	15	1	7	0	0	0	0	0	0	1	0	0	0	0	0	335	0	1
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	6	1	3	373	0
Z	0	0	0	0	3	0	0	0	19	3	0	0	0	0	2	1	1	7	0	0	0	0	0	0	0	518

Figura 7.2: Matriz de Confusão da Rede Neural Seleccionada Após PCA - Letras

Uma avaliação da tabela 7.17 leva-nos a concluir que, para o problema de classificação em questão, independentemente da taxa de aprendizado inicial adotada no treinamento, o uso de taxas de momento altas, 0,7 e 0,9, faz com que o tempo

Tx.Aprendizado	Momento	%Acerto	Tempo Treino
0,1	0,1	95,65	15m 47s
0,1	0,3	95,67	12m 42s
0,1	0,5	95,70	10m 28s
0,1	0,7	95,81	8m 27s
0,1	0,9	95,73	5m 1s
0,3	0,1	95,54	12m 24s
0,3	0,3	95,62	12m 15s
0,3	0,5	95,70	9m 43s
0,3	0,7	95,75	7m 55s
0,3	0,9	95,83	4m 35s
0,5	0,1	95,59	13m 12s
0,5	0,3	95,65	12m 2s
0,5	0,5	95,70	9m 39s
0,5	0,7	95,73	7m 53s
0,5	0,9	95,83	4m 51s
0,7	0,1	95,65	14m 44s
0,7	0,3	95,65	11m 46s
0,7	0,5	95,75	9m 36s
0,7	0,7	95,78	7m 47s
0,7	0,9	95,81	5m 0s
0,9	0,1	95,70	14m 48s
0,9	0,3	95,70	11m 50s
0,9	0,5	95,73	9m 28s
0,9	0,7	95,78	8m 35s
0,9	0,9	95,81	4m 22s

Tabela 7.17: Resultados sobre Taxa de Aprendizado e Momento

de treinamento seja reduzido e as taxas de acerto nos testes aumentadas. Com o aumento do valor do termo de momento, percebemos um aumento na velocidade do aprendizado, reduzindo o perigo de instabilidade. É de grande importância uma avaliação como esta sobre a superfície de erro do problema em estudo, pois possibilita um melhor aproveitamento da potencialidade de classificação das redes neurais implementadas.