

WALTER TONON JÚNIOR

**RECONHECIMENTO DE SÍMBOLOS
ELÉTRICOS**

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

CURITIBA

2004

WALTER TONON JÚNIOR

**RECONHECIMENTO DE SÍMBOLOS
ELÉTRICOS**

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

Área de Concentração: *Metodologias e Técnicas de Computação*

Orientador : Prof. Dr. Flávio Bortolozzi

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

CURITIBA

2004

Tonon Jr, Walter

Reconhecimento de símbolos elétricos. Curitiba, 2001. 99p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

1. Engenharia elétrica 2. Transformadas 3. Rede Neural 4. Reconhecimento.
I.Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de
Tecnologia. Programa de Pós-Graduação em Informática Aplicada II-t

Página reservada à ata de defesa e termo de aprovação.

Dedico este trabalho aos meus pais, que me capacitaram a chegar até aqui, e também à minha esposa pelo companheirismo e compreensão pelas horas ausentes.

Agradecimentos

Agradeço a PUC-PR, especialmente aos professores Cinthia O. de A. Freitas, que além de orientadora, uma amiga motivadora com idéias que muito agregaram à composição deste trabalho, e Jacques Facon, um amigo instigador de pesquisas, e a ambos pela possibilidade de efetuar o curso de mestrado nesta instituição, orientando-me desde os procedimentos iniciais até a sua conclusão.

Agradeço também a empresa Automat Engenharia Ltda por possibilitar a execução do curso de mestrado, com dispensas das atividades profissionais no decorrer do curso, através de seus diretores João Luis Amazonas Pimpão e Carlos Eduardo Kaiut. Atitude esta, infelizmente bastante incomum nos setores da iniciativa privada.

Sumário

Agradecimentos	vii
Sumário	ix
Lista de Figuras	xiii
Lista de Tabelas	xvii
Lista de Símbolos	xx
Lista de Abreviaturas	xxi
Resumo	xxiii
Abstract	xxv
Capítulo 1	
Introdução	27
1.1. Desafio	28
1.2. Motivação	28
1.3. Proposta	30
1.4. Contribuição	31
1.5. Organização	32
Capítulo 2	
Fundamentação teórica	33
2.1 Introdução	33
2.2. Morfologia matemática	34
2.3. Manipulação de documentos técnicos	36
2.4. Extração de características	41
2.5. Classificadores	43

2.6. Classificadores neurais	45
2.6.1. Uma breve revisão sobre redes neurais	45
2.6.2. Arquiteturas típicas	47
2.7. Análises de confiabilidade	49
2.8. Comentários finais	51

Capítulo 3

Metodologia	54
3.1. Introdução	54
3.2. Agrupamento dos dados: Protocolo	54
3.3. Redução de dimensionalidades	55
3.3.1. Uma breve análise	55
3.3.2. A Maldição da Dimensionalidade e o fenômeno do espcoco vazio	56
3.3.3. Avaliação dos métodos	57
3.4. Extração de características	58
3.4.1. Transformada de Chebyshev	60
3.4.2. Transformada DCT (<i>Discrete Cosine Transform</i>)	62
3.4.3. Transformada de Fourier	64
3.4.4. Transformada de Radon	64
3.4.5. Transformada de Zernike	66
3.5. Etapas de processamento	67
3.6. Fluxo dos dados	69
3.7. Redes Multi-Layer Perceptron com Backpropagation (MLP-BP)	71
3.7.1. Rede Neural: Classificador primário	73
3.8. Redes Hebbianas	74
3.9. Preparação da imagem	75
3.10. Normalização dos dados	75
3.10.1. Normalização para Chebyshev	76
3.10.2. Normalização para DCT	77
3.10.3. Normalização para Fourier	79
3.10.4. Normalização para Radon	80
3.10.5. Normalização para Zernike	81

3.10.6. Composição das normalizações para Zernike e Chebyshev	83
3.10.7. Composição das normalizações para Zernike e Fourier	84
3.10.8. Exemplo numérico das normalizações	85
3.11. Metodologia para a rejeição (<i>outliers</i>)	88
3.12. Comentários finais	89

Capítulo 4

Experimentos e Análise de Resultados	91
4.1. Introdução	91
4.2. Redes Neurais: Classificador Primário	92
4.3. Redes Neurais: Classificadores Secundários	95
4.4. Transformada de Chebyshev	98
4.5. Transformada DCT	100
4.6. Transformada de Fourier	101
4.7. Transformada de Radon	103
4.8. Transformada de Zernike	105
4.9. Transformadas de Zernike e Chebyshev	106
4.9.1. Classificador primário	107
4.9.2. Classificadores secundários	108
4.10. Transformadas de Zernike e Fourier	112
4.10.1. Classificador primário	113
4.10.2. Classificadores secundários	114
4.11. Comentários finais	118

Capítulo 5

Conclusão	121
------------------	------------

Referências	125
--------------------	------------

Apêndice A

Base de dados utilizada	129
--------------------------------	------------

Apêndice B

Curvas de treinamento e testes de validação para a Transformada de Chebyshev	138
---	------------

Apêndice C

Curvas de treinamento e testes de validação para a Transformada Discreta do Coseno (DCT)	147
---	------------

Apêndice D

Curvas de treinamento e testes de validação para a Transformada de Fourier	156
---	------------

Apêndice E

Curvas de treinamento e testes de validação para a Transformada de Radon	165
---	------------

Apêndice F

Curvas de treinamento e testes de validação para a Transformada de Zernike	174
---	------------

Apêndice G

Curvas de treinamento e testes de validação para as Transformadas de Zernike e Chebyshev	183
---	------------

Apêndice H

Curvas de treinamento e testes de validação para as Transformadas de Zernike e Fourier	192
---	------------

Lista de Figuras

Figura 1.1	Diagrama unifilar elétrico industrial	29
Figura 1.2	Exemplo de representação de um único símbolo em diferentes escalas e orientações	31
Figura 2.1	Seqüência de manipulação dos dados para análises	34
Figura 2.2	Exemplo de interligação entre neurônios	47
Figura 2.3	Exemplo de Rede Neural Artificial com duas camadas	49
Figura 3.1	Comprimento dos terminais em relação á altura do componente	55
Figura 3.2	Visualização da transformada de Chebyshev	61
Figura 3.3	Modelo de funcionamento da Transformada do coseno	63
Figura 3.4	Interpretação geométrica de Radon	65
Figura 3.5	Interpretação geométrica de Zernike	66
Figura 3.6	Etapas utilizadas no reconhecimento dos símbolos com uma transformada	68
Figura 3.7	Etapas utilizadas no reconhecimento dos símbolos com duas transformadas	68
Figura 3.8	Fluxo dos dados na estrutura classificatória	70
Figura 3.9	Exemplo do agrupamento dos dados	71
Figura 3.10	Variâncias não normalizadas obtidas pela Transformada de Chebyshev	76
Figura 3.11	Variâncias normalizadas obtidas pela Transformada de Chebyshev	77
Figura 3.12	Variâncias não normalizadas obtidas pela Transformada do Coseno	78
Figura 3.13	Variâncias normalizadas obtidas pela Transformada do Coseno	78
Figura 3.14	Variâncias não normalizadas obtidas pela Transformada de Fourier	79
Figura 3.15	Variâncias normalizadas obtidas pela Transformada de Fourier	80
Figura 3.16	Variâncias não normalizadas obtidas pela Transformada de Radon	80
Figura 3.17	Variâncias normalizadas obtidas pela Transformada de Radon	81
Figura 3.18	Variâncias não normalizadas obtidas pela Transformada de Zernike	82
Figura 3.19	Variâncias normalizadas obtidas pela Transformada de Zernike	83
Figura 3.20	Variâncias não normalizadas obtidas pelas Transformadas de Zernike e Chebyshev	83
Figura 3.21	Variâncias normalizadas obtidas pelas Transformadas de Zernike e Chebyshev	84
Figura 3.22	Variâncias não normalizadas obtidas pelas Transformadas de Zernike e Fourier	84
Figura 3.23	Variâncias normalizadas obtidas pelas Transformadas de Zernike e Fourier	85
Figura 3.24	Símbolo utilizado como exemplo para visualizar as normalizações	85
Figura 4.1	Topologia da Rede Neural utilizada para os experimentos no classificador primário	93
Figura 4.2	Topologia da Rede Neural utilizada para os testes com a transformada de Zernike	93
Figura 4.3	Demonstração de como realizaram-se os testes com as RNAs	94
Figura 4.4	Classificador secundário para a 1 ^a metaclasse	95
Figura 4.5	Classificador secundário para a 2 ^a metaclasse	96

Figura 4.6	Classificador secundário para a 3 ^a metaclasse	96
Figura 4.7	Classificador secundário para a 4 ^a metaclasse	97
Figura 4.8	Classificador secundário para a 5 ^a metaclasse	97
Figura 4.9	Chebyshev: Curva de aprendizagem apresentando o Erro x Época	99
Figura 4.10	DCT: Curva de aprendizagem apresentando o Erro x Época	101
Figura 4.11	Fourier: Curva de aprendizagem apresentando o Erro x Época	102
Figura 4.12	Radon: Curva de aprendizagem apresentando o Erro x Época	104
Figura 4.13	Zernike: Curva de aprendizagem apresentando o Erro x Época	106
Figura 4.14	Zernike/Chebyshev: Curva de aprendizagem apresentando o Erro x Época	108
Figura 4.15	Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 2	110
Figura 4.16	Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 3	111
Figura 4.17	Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 4	111
Figura 4.18	Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 5	112
Figura 4.19	Zernike/Fourier: Curva de aprendizagem apresentando o Erro x Época	114
Figura 4.20	Modelo substitutivo para o classificador secundário – metaclasse 1	115
Figura 4.21	Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 1	116
Figura 4.22	Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 2	116
Figura 4.23	Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 3	117
Figura 4.24	Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 4	117
Figura 4.25	Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 5	118
Figura A.1	Metaclasse I	130
Figura A.2	Metaclasse II	131
Figura A.3	Metaclasse III	132
Figura A.4	Metaclasse IV	133
Figura A.5	Metaclasse V	134
Figura A.6	Símbolos <i>outliers</i> para a Metaclasse I	135
Figura A.7	Símbolos <i>outliers</i> para a Metaclasse II	136
Figura A.8	Símbolos <i>outliers</i> para a Metaclasse III	136
Figura A.9	Símbolos <i>outliers</i> para a Metaclasse IV	137
Figura A.10	Símbolos <i>outliers</i> para a Metaclasse V	137
Figura B.1	Chebyshev: Curva de erro para o 1 ^o conjunto de testes	138
Figura B.2	Chebyshev: Curva de erro para o 2 ^o conjunto de testes	139
Figura B.3	Chebyshev: Curva de erro para o 3 ^o conjunto de testes	140
Figura B.4	Chebyshev: Curva de erro para o 4 ^o conjunto de testes	141
Figura B.5	Chebyshev: Curva de erro para o 5 ^o conjunto de testes	142
Figura B.6	Chebyshev: Curva de erro para o 6 ^o conjunto de testes	142
Figura B.7	Chebyshev: Curva de erro para o 7 ^o conjunto de testes	143
Figura B.8	Chebyshev: Curva de erro para o 8 ^o conjunto de testes	144

Figura B.9	Chebyshev: Curva de erro para o 9 ^o conjunto de testes	145
Figura B.10	Chebyshev: Curva de erro para o 10 ^o conjunto de testes	146
Figura C.1	DCT: Curva de erro para o 1 ^o conjunto de testes	147
Figura C.2	DCT: Curva de erro para o 2 ^o conjunto de testes	148
Figura C.3	DCT: Curva de erro para o 3 ^o conjunto de testes	149
Figura C.4	DCT: Curva de erro para o 4 ^o conjunto de testes	150
Figura C.5	DCT: Curva de erro para o 5 ^o conjunto de testes	151
Figura C.6	DCT: Curva de erro para o 6 ^o conjunto de testes	151
Figura C.7	DCT: Curva de erro para o 7 ^o conjunto de testes	152
Figura C.8	DCT: Curva de erro para o 8 ^o conjunto de testes	153
Figura C.9	DCT: Curva de erro para o 9 ^o conjunto de testes	154
Figura C.10	DCT: Curva de erro para o 10 ^o conjunto de testes	155
Figura D.1	Fourier: Curva de erro para o 1 ^o conjunto de testes	156
Figura D.2	Fourier: Curva de erro para o 2 ^o conjunto de testes	157
Figura D.3	Fourier: Curva de erro para o 3 ^o conjunto de testes	158
Figura D.4	Fourier: Curva de erro para o 4 ^o conjunto de testes	159
Figura D.5	Fourier: Curva de erro para o 5 ^o conjunto de testes	160
Figura D.6	Fourier: Curva de erro para o 6 ^o conjunto de testes	160
Figura D.7	Fourier: Curva de erro para o 7 ^o conjunto de testes	161
Figura D.8	Fourier: Curva de erro para o 8 ^o conjunto de testes	162
Figura D.9	Fourier: Curva de erro para o 9 ^o conjunto de testes	163
Figura D.10	Fourier: Curva de erro para o 10 ^o conjunto de testes	164
Figura E.1	Radon: Curva de erro para o 1 ^o conjunto de testes	165
Figura E.2	Radon: Curva de erro para o 2 ^o conjunto de testes	166
Figura E.3	Radon: Curva de erro para o 3 ^o conjunto de testes	167
Figura E.4	Radon: Curva de erro para o 4 ^o conjunto de testes	168
Figura E.5	Radon: Curva de erro para o 5 ^o conjunto de testes	169
Figura E.6	Radon: Curva de erro para o 6 ^o conjunto de testes	170
Figura E.7	Radon: Curva de erro para o 7 ^o conjunto de testes	170
Figura E.8	Radon: Curva de erro para o 8 ^o conjunto de testes	171
Figura E.9	Radon: Curva de erro para o 9 ^o conjunto de testes	172
Figura E.10	Radon: Curva de erro para o 10 ^o conjunto de testes	173
Figura F.1	Zernike: Curva de erro para o 1 ^o conjunto de testes	174
Figura F.2	Zernike: Curva de erro para o 2 ^o conjunto de testes	175
Figura F.3	Zernike: Curva de erro para o 3 ^o conjunto de testes	176
Figura F.4	Zernike: Curva de erro para o 4 ^o conjunto de testes	177
Figura F.5	Zernike: Curva de erro para o 5 ^o conjunto de testes	178
Figura F.6	Zernike: Curva de erro para o 6 ^o conjunto de testes	178
Figura F.7	Zernike: Curva de erro para o 7 ^o conjunto de testes	179
Figura F.8	Zernike: Curva de erro para o 8 ^o conjunto de testes	180
Figura F.9	Zernike: Curva de erro para o 9 ^o conjunto de testes	181
Figura F.10	Zernike: Curva de erro para o 10 ^o conjunto de testes	182
Figura G.1	Zernike/Chebyshev: Curva de erro para o 1 ^o conjunto de testes	183
Figura G.2	Zernike/Chebyshev: Curva de erro para o 2 ^o conjunto de testes	184
Figura G.3	Zernike/Chebyshev: Curva de erro para o 3 ^o conjunto de testes	185
Figura G.4	Zernike/Chebyshev: Curva de erro para o 4 ^o conjunto de testes	186
Figura G.5	Zernike/Chebyshev: Curva de erro para o 5 ^o conjunto de testes	187
Figura G.6	Zernike/Chebyshev: Curva de erro para o 6 ^o conjunto de testes	187

Figura G.7	Zernike/Chebyshev: Curva de erro para o 7 ^o conjunto de testes	188
Figura G.8	Zernike/Chebyshev: Curva de erro para o 8 ^o conjunto de testes	189
Figura G.9	Zernike/Chebyshev: Curva de erro para o 9 ^o conjunto de testes	190
Figura G.10	Zernike/Chebyshev: Curva de erro para o 10 ^o conjunto de testes	191
Figura H.1	Zernike/Fourier: Curva de erro para o 1 ^o conjunto de testes	192
Figura H.2	Zernike/Fourier: Curva de erro para o 2 ^o conjunto de testes	193
Figura H.3	Zernike/Fourier: Curva de erro para o 3 ^o conjunto de testes	194
Figura H.4	Zernike/Fourier: Curva de erro para o 4 ^o conjunto de testes	195
Figura H.5	Zernike/Fourier: Curva de erro para o 5 ^o conjunto de testes	196
Figura H.6	Zernike/Fourier: Curva de erro para o 6 ^o conjunto de testes	196
Figura H.7	Zernike/Fourier: Curva de erro para o 7 ^o conjunto de testes	197
Figura H.8	Zernike/Fourier: Curva de erro para o 8 ^o conjunto de testes	198
Figura H.9	Zernike/Fourier: Curva de erro para o 9 ^o conjunto de testes	199
Figura H.10	Zernike/Fourier: Curva de erro para o 10 ^o conjunto de testes	200

Lista de Tabelas

Tabela 2.1	Exemplos de funções de ativação	48
Tabela 3.1	Resultado obtido pela aplicação das transformadas	86
Tabela 3.2	Distribuição dos símbolos <i>outliers</i> por metaclasse	89
Tabela 4.1	Configuração da aprendizagem/épocas por transformada	93
Tabela 4.2	Chebyshev: Matriz de confusão (%)	98
Tabela 4.3	Chebyshev: Testes com <i>outliers</i>	99
Tabela 4.3	DCT: Matriz de confusão (%)	100
Tabela 4.5	DCT: Testes com <i>outliers</i>	101
Tabela 4.6	Fourier: Matriz de confusão (%)	102
Tabela 4.7	Fourier: Testes com <i>outliers</i>	103
Tabela 4.8	Radon: Matriz de confusão (%)	104
Tabela 4.9	Radon: Testes com <i>outliers</i>	105
Tabela 4.10	Zernike: Matriz de confusão (%)	105
Tabela 4.11	Zernike: Testes com <i>outliers</i>	106
Tabela 4.12	Zernike/Chebyshev: Matriz de confusão (%)	107
Tabela 4.13	Zernike/Chebyshev: Testes com <i>outliers</i>	108
Tabela 4.14	Zernike/Chebyshev: Relação de quantidade de símbolos por metaclasse	109
Tabela 4.15	Zernike/Chebyshev: Distribuição dos elementos por metaclasse	109
Tabela 4.16	Zernike/Chebyshev: Saída do classificador secundário 1, (%)	109
Tabela 4.17	Zernike/Chebyshev: Saída do classificador secundário 2, (%)	110
Tabela 4.18	Zernike/Chebyshev: Saída do classificador secundário 3, (%)	111
Tabela 4.19	Zernike/Chebyshev: Saída do classificador secundário 4, (%)	112
Tabela 4.20	Zernike/Chebyshev: Saída do classificador secundário 5, (%)	112
Tabela 4.21	Zernike/Fourier: Matriz de confusão (%)	113
Tabela 4.22	Zernike/Fourier: Testes com <i>outliers</i>	114
Tabela 4.23	Zernike/Fourier: Saída do classificador secundário 1, (%)	116
Tabela 4.24	Zernike/Fourier: Saída do classificador secundário 2, (%)	116
Tabela 4.25	Zernike/Fourier: Saída do classificador secundário 3, (%)	117
Tabela 4.26	Zernike/Fourier: Saída do classificador secundário 4, (%)	117
Tabela 4.27	Zernike/Fourier: Saída do classificador secundário 5, (%)	118
Tabela 4.28	Comparação com os resultados obtidos para <i>outliers</i> , (%)	118
Tabela 4.29	Comparação com os resultados globais obtidos pelas transformadas combinadas	119
Tabela B.1	Chebyshev: Reconhecimento do 1º conjunto de teste/validação	139
Tabela B.2	Chebyshev: Reconhecimento do 2º conjunto de teste/validação	140
Tabela B.3	Chebyshev: Reconhecimento do 3º conjunto de teste/validação	140
Tabela B.4	Chebyshev: Reconhecimento do 4º conjunto de teste/validação	141
Tabela B.5	Chebyshev: Reconhecimento do 5º conjunto de teste/validação	142

Tabela B.6	Chebyshev: Reconhecimento do 6 ^o conjunto de teste/validação	143
Tabela B.7	Chebyshev: Reconhecimento do 7 ^o conjunto de teste/validação	144
Tabela B.8	Chebyshev: Reconhecimento do 8 ^o conjunto de teste/validação	145
Tabela B.9	Chebyshev: Reconhecimento do 9 ^o conjunto de teste/validação	145
Tabela B.10	Chebyshev: Reconhecimento do 10 ^o conjunto de teste/validação	146
Tabela C.1	DCT: Reconhecimento do 1 ^o conjunto de teste/validação	148
Tabela C.2	DCT: Reconhecimento do 2 ^o conjunto de teste/validação	149
Tabela C.3	DCT: Reconhecimento do 3 ^o conjunto de teste/validação	149
Tabela C.4	DCT: Reconhecimento do 4 ^o conjunto de teste/validação	150
Tabela C.5	DCT: Reconhecimento do 5 ^o conjunto de teste/validação	151
Tabela C.6	DCT: Reconhecimento do 6 ^o conjunto de teste/validação	152
Tabela C.7	DCT: Reconhecimento do 7 ^o conjunto de teste/validação	153
Tabela C.8	DCT: Reconhecimento do 8 ^o conjunto de teste/validação	154
Tabela C.9	DCT: Reconhecimento do 9 ^o conjunto de teste/validação	154
Tabela C.10	DCT: Reconhecimento do 10 ^o conjunto de teste/validação	155
Tabela D.1	Fourier: Reconhecimento do 1 ^o conjunto de teste/validação	157
Tabela D.2	Fourier: Reconhecimento do 2 ^o conjunto de teste/validação	158
Tabela D.3	Fourier: Reconhecimento do 3 ^o conjunto de teste/validação	158
Tabela D.4	Fourier: Reconhecimento do 4 ^o conjunto de teste/validação	159
Tabela D.5	Fourier: Reconhecimento do 5 ^o conjunto de teste/validação	160
Tabela D.6	Fourier: Reconhecimento do 6 ^o conjunto de teste/validação	161
Tabela D.7	Fourier: Reconhecimento do 7 ^o conjunto de teste/validação	162
Tabela D.8	Fourier: Reconhecimento do 8 ^o conjunto de teste/validação	163
Tabela D.9	Fourier: Reconhecimento do 9 ^o conjunto de teste/validação	163
Tabela D.10	Fourier: Reconhecimento do 10 ^o conjunto de teste/validação	164
Tabela E.1	Radon: Reconhecimento do 1 ^o conjunto de teste/validação	166
Tabela E.2	Radon: Reconhecimento do 2 ^o conjunto de teste/validação	167
Tabela E.3	Radon: Reconhecimento do 3 ^o conjunto de teste/validação	167
Tabela E.4	Radon: Reconhecimento do 4 ^o conjunto de teste/validação	168
Tabela E.5	Radon: Reconhecimento do 5 ^o conjunto de teste/validação	169
Tabela E.6	Radon: Reconhecimento do 6 ^o conjunto de teste/validação	170
Tabela E.7	Radon: Reconhecimento do 7 ^o conjunto de teste/validação	171
Tabela E.8	Radon: Reconhecimento do 8 ^o conjunto de teste/validação	172
Tabela E.9	Radon: Reconhecimento do 9 ^o conjunto de teste/validação	172
Tabela E.10	Radon: Reconhecimento do 10 ^o conjunto de teste/validação	173
Tabela F.1	Zernike: Reconhecimento do 1 ^o conjunto de teste/validação	175
Tabela F.2	Zernike: Reconhecimento do 2 ^o conjunto de teste/validação	176
Tabela F.3	Zernike: Reconhecimento do 3 ^o conjunto de teste/validação	176
Tabela F.4	Zernike: Reconhecimento do 4 ^o conjunto de teste/validação	177
Tabela F.5	Zernike: Reconhecimento do 5 ^o conjunto de teste/validação	178
Tabela F.6	Zernike: Reconhecimento do 6 ^o conjunto de teste/validação	179
Tabela F.7	Zernike: Reconhecimento do 7 ^o conjunto de teste/validação	180
Tabela F.8	Zernike: Reconhecimento do 8 ^o conjunto de teste/validação	181
Tabela F.9	Zernike: Reconhecimento do 9 ^o conjunto de teste/validação	181
Tabela F.10	Zernike: Reconhecimento do 10 ^o conjunto de teste/validação	182
Tabela G.1	Zernike/Chebyshev: Reconhecimento do 1 ^o conjunto de teste/validação	184
Tabela G.2	Zernike/Chebyshev: Reconhecimento do 2 ^o conjunto de teste/validação	185
Tabela G.3	Zernike/Chebyshev: Reconhecimento do 3 ^o conjunto de teste/validação	185

Tabela G.4	Zernike/Chebyshev: Reconhecimento do 4 ^o conjunto de teste/validação	186
Tabela G.5	Zernike/Chebyshev: Reconhecimento do 5 ^o conjunto de teste/validação	187
Tabela G.6	Zernike/Chebyshev: Reconhecimento do 6 ^o conjunto de teste/validação	188
Tabela G.7	Zernike/Chebyshev: Reconhecimento do 7 ^o conjunto de teste/validação	189
Tabela G.8	Zernike/Chebyshev: Reconhecimento do 8 ^o conjunto de teste/validação	190
Tabela G.9	Zernike/Chebyshev: Reconhecimento do 9 ^o conjunto de teste/validação	190
Tabela G.10	Zernike/Chebyshev: Reconhecimento do 10 ^o conjunto de teste/validação	191
Tabela H.1	Zernike/Fourier: Reconhecimento do 1 ^o conjunto de teste/validação	193
Tabela H.2	Zernike/Fourier: Reconhecimento do 2 ^o conjunto de teste/validação	194
Tabela H.3	Zernike/Fourier: Reconhecimento do 3 ^o conjunto de teste/validação	194
Tabela H.4	Zernike/Fourier: Reconhecimento do 4 ^o conjunto de teste/validação	195
Tabela H.5	Zernike/Fourier: Reconhecimento do 5 ^o conjunto de teste/validação	196
Tabela H.6	Zernike/Fourier: Reconhecimento do 6 ^o conjunto de teste/validação	197
Tabela H.7	Zernike/Fourier: Reconhecimento do 7 ^o conjunto de teste/validação	198
Tabela H.8	Zernike/Fourier: Reconhecimento do 8 ^o conjunto de teste/validação	199
Tabela H.9	Zernike/Fourier: Reconhecimento do 9 ^o conjunto de teste/validação	199
Tabela H.10	Zernike/Fourier: Reconhecimento do 10 ^o conjunto de teste/validação	200

Lista de Símbolos

$A[i,j]$	Elemento da coordenada linha ' i ' pela coluna ' j ' da matriz A
A_{mn}	Matriz resultante da Transformada de Zernike, salvo especificado em contrário
C	Matriz resultante da Transformada do Coseno
DP	Desvio padrão
\mathfrak{F}	Matriz resultante da Transformada de Fourier
f_{ativ}	Função de ativação em uma célula neuronal
δ_k	Erro de retropropagação do neurônio k
G	Matriz resultante da Transformada Discreta de Fourier
N	Número de amostras de um conjunto
N_i	Identificação do neurônio ' i '.
η	Passo (gradiente) de aprendizagem para redes neurais
Γ	Matriz resultante da Transformada de Chebyshev
$P(x,y)$	Pixel normalizado de uma imagem na posição (x,y)
π	Constante numérica pi 3,14159
R	Matriz resultante da Transformada de Radon
S	Saída do neurônio (rede neural), com o valor resultante pela sua função de ativação
w_{ij}	Peso atribuído à ligação do neurônio ' i ' com o neurônio ' j '.

Lista de Abreviaturas

ABNT	Associação Brasileira de Normas Técnicas
CFS	<i>Correlation on based Feature Selection</i> (Seleção Característica baseada na Correlação), algoritmo de <i>wrapper</i>
DCT	<i>Discrete Cosine Transform</i> (Transformada Discreta do Coseno)
DFT	<i>Discrete Fourier Transform</i> (Transformada Discreta de Fourier)
EM	<i>Expectation Maximization</i> (Maximização Esperada)
EPP	Exploratory Projection Pursuit (Perseguição exploratória da Projeção), algoritmo de redução de dimensionalidades
k-means	Algoritmo para realizar agrupamentos nas k-médias
KNN	<i>K-Nearest Neighbor</i> ou K-Vizinhos mais Próximos
MLP	<i>Multi-Layered Perceptron</i> (<i>Perceptron</i> Multi Camadas), Rede Neural
MLP-BP	<i>Multi-Layered Perceptron with Backpropagation</i> (<i>Perceptron</i> Multi Camadas com algoritmo de treinamento de Retropropagação), Rede Neural
MSE	Erro médio quadrático
NN	<i>Neural Networks</i> , (Redes Neurais)
SSE	Somatória dos quadrados dos erros residuais
PCA	<i>Principal Component Analysis</i> (Análise pela Componente Principal)
PPR	<i>Projection Pursuit Regression</i> (Perseguição da Projeção pela Regressão), algoritmo de redução de dimensionalidades
RNA	Rede Neural Artificial

Resumo

Este trabalho aborda o reconhecimento de símbolos em engenharia elétrica manualmente segmentados, (dentro das normas da ABNT), através de métodos estatísticos utilizando-se transformadas numéricas (ou momentos matemáticos) e uma rede neural hierárquica. Os símbolos foram extraídos de desenhos técnicos unifilares em engenharia elétrica, formando assim uma base de dados. As transformadas, foram utilizadas individualmente e em paralelo, na tentativa de obter-se as características mais acentuadas de cada modelo matemático. Após extraídas as primitivas, os vetores obtidos das transformadas são concatenados (quando trabalhando com duas transformadas combinadas), e aplicados a uma Rede Neural funcionando como um classificador primário, identificando a que grupo de símbolos (metaclassa) pertence o símbolo sob análise, e assim ativando a rede neural da camada subsequente para o seu reconhecimento final. Foram analisadas as transformadas de Chebyshev, DCT (Transformada Discreta do Coseno), Fourier, Radon e Zernike. O trabalho faz também um comparativo no desempenho das transformadas operando isoladas e em paralelo, apresentando um desempenho superior das transformadas de Zernike/Fourier e Zernike/Chebyshev. As taxas de reconhecimento obtidas para o classificador primário foram de 63,1% para a Transformada de Chebyshev, 65,2% para a Transformada Discreta do Coseno, 63,5% para a Transformada de Fourier, 63,4% para a Transformada de Radon, 78,6% para a Transformada de Zernike, 100% para as Transformadas de Zernike e Chebyshev associadas e 96,8% para as Transformadas de Zernike e Fourier associadas. A Rede Neural Artificial possui dois níveis: o primário seleciona a que classe pertence, e o secundário identifica o símbolo. Durante a classificação total (primária mais secundária), as Transformadas de Zernike e Fourier associadas apresentaram uma perda de reconhecimento de 15 símbolos, com um desempenho superior, contra 33 símbolos das Transformadas de Zernike e Chebyshev associadas.

Palavras-Chave: Símbolos Elétricos, Transformadas, Redes Neurais, Reconhecimento.

Abstract

This work deal with the recognition of symbols segmented in electric engineering, (according to Brazilian Standards), through statistical methods using mathematical moments (or transforms) and by the use of a hierarchical Neural Net (NN). The symbols were extracted from technical uni-wired drawings (unifilar) in electric engineering, to form this database. Applying the moments, individually or in parallel, in the attempt of obtaining the accentuated characteristics of each mathematical model. After extracted the primitives, the vectors obtained from those moments are concatenated (when working with the join of two transforms), and applied in a Neural Network working as a primary classifier, identifying the one that group of symbols (metaclass) belongs the symbol under analysis, and this first Neural Network active the corresponding Neural Network of the subsequent final recognition. They were analyzed moments of Chebyshev, DCT (Transform Discreet of Coseno), Fourier, Radon and Zernike. The work also makes a comparative in the performance of those transformed operating isolated and in parallel, presenting a superior performance of those transformed of Zernike/Fourier and Zernike/Chebyshev. The recognition rates obtained for the primary classifier were of 63,1% for Chebyshev Moments, 65,2% for DCT, 63,5% for Fourier Moments, 63,4% for Radon Moments, 78,6% Zernike Moments, 100% for associated Zernike and Chebyshev Moments and 96,8% for associated Zernike and Fourier Moments. The hierarchical Neural Net has two levels: the primary selecting a group belongs and the secondary identifying the symbol. During the total classification (primary more secondary), The associated Zernike and Fourier Moments presented a loss of recognition of 15 symbols, with a superior performance, against 33 symbols of those associated Zernike and Chebyshev Moments.

Keywords: Electric Symbols, transforms, neural networks, recognition.

Capítulo 1

Introdução

Devido à crescente demanda de documentos, sentiu-se a necessidade de organizá-los, de forma que seja simples e rápida a sua recuperação. Surgiu então, o armazenamento de documentos por meios eletrônicos. Atualmente, através da disseminação de documentos eletrônicos, tornou-se popular o uso de computadores para gerenciamento de papéis. O mesmo ocorre com desenhos em engenharia, embora ainda hoje não exista um padrão estabelecido para arquivamento e distribuição de desenhos técnicos. Observa-se então, que desenhos de engenharia carecem do mesmo procedimento (de padronização e organização), pois são geradas várias pranchas por projeto, cada prancha é encaminhada à área pertinente para análise e/ou execução do projeto e, também, é encaminhada para a equipe de manutenção. Com desenhos em engenharia elétrica, existem diversos aplicativos que traçam, interligam e representam os componentes, mas não os interpreta.

Existem trabalhos já realizados [KAS90] e em andamento no processo de digitalização de desenhos [TOM96] feitos em papel para os meios digitais, mas com o enfoque na vetorização dos traçados. Atualmente, não se utilizam mais desenhos traçados manualmente em papel, só por meios eletrônicos, devido a sua facilidade de manipulação e arquivamento.

Embora seja fácil manipular, gerenciar, modificar e mesclar desenhos de engenharia elétrica no formato eletrônico, a manipulação ainda exige a intervenção humana. Atualmente, urge a necessidade de manipulações automáticas destes desenhos em plantas industriais.

O problema de interpretações automáticas de imagens de desenhos de engenharia, ou reconhecimento gráfico, tem sido estudado por um longo tempo [KAS90]. Nos anos recentes, vários métodos de vetorização foram propostos [CHH00], embora ainda não exista um padrão

efetivo de mercado. Por isso, os símbolos aqui serão analisados na estrutura de mapa de bit (bitmap).

O enfoque deste trabalho visa o reconhecimento automático de símbolos elétricos em desenhos de engenharia elétrica, já segmentados, através de até dois métodos estatísticos de extração de características operando em paralelo e aplicados a uma rede neural discriminante em 5 metaclasses. Cada uma destas metaclasses contém um conjunto de elementos ao qual o símbolo analisado deverá pertencer.

Os métodos estatísticos analisados são as transformadas de Chebyshev, do Coseno, Fourier, Radon e Zernike, e com normalizações dos dados de entrada para as Redes Neurais Artificiais. As Redes Neurais Artificiais utilizadas para identificação das metaclasses são redes MLP-BP (*Multi-Layer Perceptron with Backpropagation*) e redes Hebbianas.

1.1. Desafio

O contínuo aumento do tamanho e complexidade dos sistemas de potência tem levado à necessidade de ferramentas computacionais sofisticadas para resolver problemas no planejamento, operação, diagnósticos e projetos destes sistemas [DIL96].

Portanto, o desafio está centrado no setor de energia elétrica, em desenhos unifilares da área de Engenharia Elétrica em plantas industriais, como mostra a Figura 1.1 (um desenho unifilar de uma planta de uma subestação elétrica). Não há impedimentos de que, futuramente, a identificação de desenhos analisados também aborde esquemáticos bifilares e trifilares (polifásicos), bastando para tal, a implementação de uma expansão da base de dados.

1.2. Motivação

Técnicas baseadas em otimização para processamento de imagens tem sido amplamente utilizadas nos campos de recuperações de imagens, pré-processamento, segmentação e reconhecimento [KEI00]. O objetivo aqui é aplicar este conhecimento na identificação automática de símbolos de desenhos elétricos.

Uma característica desejável para o sistema de reconhecimento de símbolos está na capacidade de funcionamento com o mínimo de intervenção humana (no ideal, sem intervenção alguma). Cheng et al. [CHE93], propôs um sistema de reconhecimento de símbolos voltado a circuitos eletrônicos, já segmentados e definidos em 4 metaclasses,

demonstrando ser possível a aplicação de técnicas de reconhecimento neste contexto. Novos testes realizados por Tonon Jr. et al. [TON03b] mostraram a possibilidade de se estender este trabalho para a área de engenharia elétrica industrial.

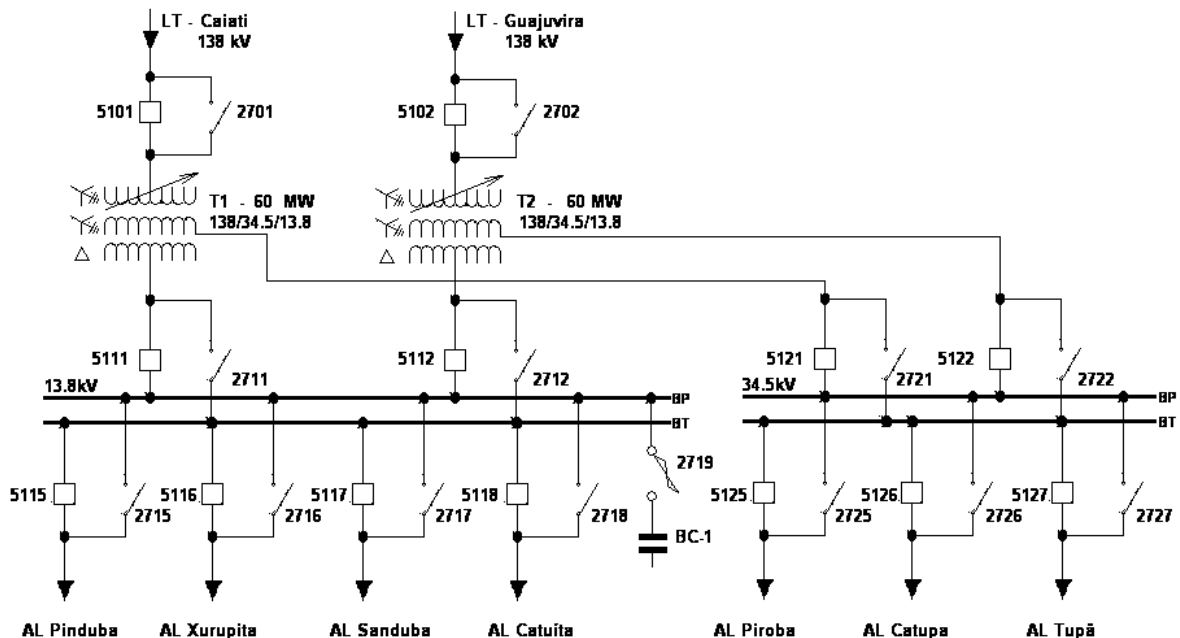


Figura 1.1 - Diagrama unifilar elétrico industrial.

O sistema neste trabalho é impulsionado pelas possibilidades de atividades geradas pelo reconhecimento automático de símbolos, de forma eficaz e confiável, reduzindo custos de pessoal para montagem de topologias em plantas elétricas industriais, localização de equipamentos em diagramas técnicos (por exemplo, com fins de manutenção), e auxílio de análise da planta com fins de estudo de capacidade da unidade (capacidade de carregamento).

A topologia da unidade, tem por função identificar qual equipamento está conectado com quem, e assim por diante. Tem também por função, identificar a ordem de proteção da unidade, intertravamento dos equipamentos (para segurança do pessoal autorizado a trabalhar na unidade ou para segurança do próprio equipamento instalado), e a seqüência de operação dos equipamentos (procedimento), para um correto funcionamento da unidade (evitando danos na carga consumidora, como por exemplo, queima de equipamentos no final de linha em redes elétricas). A partir da topologia, torna-se possível efetuar um estudo sobre a capacidade de carregamento da unidade, redefinindo o correto dimensionamento da planta e

evitando custos por interrupções indevidas, causadas por sobrecargas na unidade (por exemplo, atuações por ajustes de proteções).

A localização automática dos componentes em desenhos, destina-se a programação de manutenções, sem o custo de intervenções humanas, aumentando a vida útil do equipamento e otimizando os onerosos recursos de deslocamento das equipes de manutenção.

1.3. Proposta

A proposta consiste na classificação automática de símbolos gráficos utilizados em projetos de engenharia elétrica [TON03a].

A proposta deste trabalho tem por base o reconhecimento de desenhos em arquivos gerados por aplicativos CAD, tal qual apresentado por Kasturi & Triverdi [KAS90] – sobre digitalização de desenhos técnicos, e Tombre [TOM96] – sobre reconhecimento de documentos técnicos, adaptados à segmentação e extração de primitivas de símbolos em engenharia elétrica.

Uma vez tendo os símbolos segmentados, aplica-se um processo de extração de características (composto de um ou mais momentos matemáticos combinados entre si, por concatenação). Os processos de extração de características escolhidos devem ser capazes de minimizar os efeitos de escala e rotação do símbolo a ser analisado. Por isso, a abordagem tem por base o uso da Estatística. Esta escolha, deve-se ao fato de que em desenhos elétricos (na área eletrotécnica, principalmente), não há um tamanho padrão para se representar um único símbolo, ou seja: o mesmo símbolo pode estar representado em tamanhos diferentes no mesmo desenho. O mesmo problema ocorre nos casos de orientação do traçado (desenho do símbolo), pois não há um único sentido para a sua representação. Isto descaracteriza técnicas estruturais como por exemplo, *Template Matching* e Métodos de *Zoning* para montagem de características a serem aplicadas a um classificador. A Figura 1.2 mostra um exemplo de desenho de engenharia elétrica aonde não há uma representação única de orientação e nem tamanho de um único símbolo (para este caso, chaves fusíveis e transformadores de potência da unidade, como identificados no desenho). Observa-se também, que, para os projetistas, normalmente o tamanho do símbolo representa a sua potência elétrica ou o grau de importância dentro da unidade esquematizada, para facilitar a visualização na planta. Observam-se as diferentes escalas para os transformadores de 2 enrolamentos (destinados aos transformadores do serviço auxiliar da unidade e do transformador de aterramento na barra

principal), assim como as chaves fusíveis para o banco de capacitores e as chaves para os transformadores do sistema de serviço auxiliar.

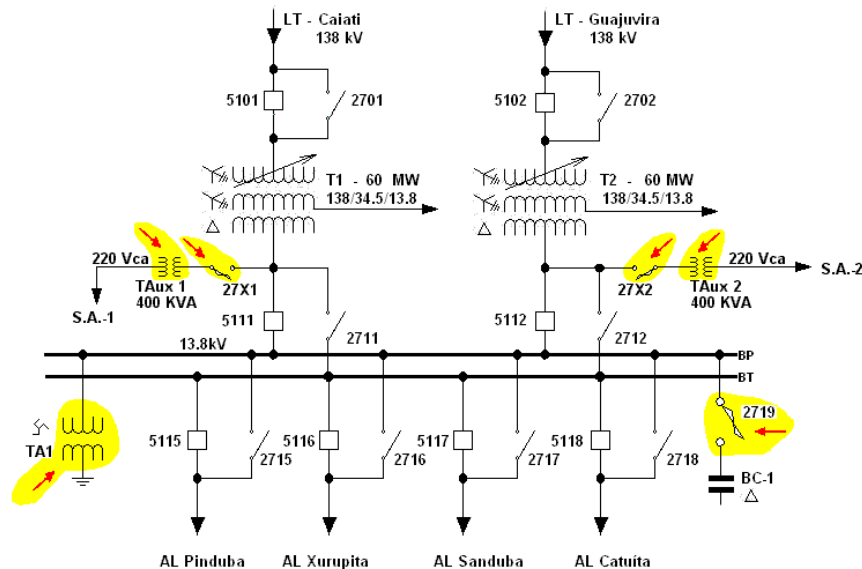


Figura 1.2: Exemplo de representação de um único símbolo em diferentes escalas e orientações.

Como classificador para se efetuar o reconhecimento, implementou-se uma Rede Neural, devido a sua versatilidade de adaptação e facilidade de implementação.

Classificadores não supervisionados podem identificar símbolos parecidos, mas não idênticos funcionalmente, sem garantir o correto agrupamento dos dados. Observando-se os símbolos, percebe-se imediatamente a inviabilidade de utilizar-se um classificador não supervisionado, conforme descrito no Capítulo 3 (Metodologia).

1.4. Contribuição

Este trabalho espera contribuir com uma ferramenta inicial para reconhecimento de símbolos gráficos utilizados em desenhos de engenharia elétrica, a partir de diagramas unifilares, previamente segmentados. Esta é a primeira etapa para se montar um analisador de diagramas esquemáticos, com o objetivo de identificar e interpretar os componentes para estudos de interligações, proteções, intertravamentos de topologia e condições de operação da planta.

As principais contribuições podem ser citadas como:

- Criação de uma base de dados,

- Comparação de 5 momentos matemáticos de extração de primitivas,
- Comparação de 2 momentos matemáticos operando de forma conjunta,
- Hierarquização do problema em metaclasses,
- Dois níveis de classificação (classificador primário e classificadores secundários).

1.5. Organização

Nos próximos capítulos deste projeto, são abordados elementos de reconhecimento de padrões por técnicas estatísticas, baseadas em Redes Neurais Artificiais. Também são mostradas técnicas de normalização dos dados para serem introduzidos em uma Rede Neural para um melhor desempenho discriminatório.

Este trabalho está organizado em capítulos separando as atividades, No Capítulo 2, está descrita a fundamentação teórica, as justificativas e os critérios de escolha dos modelos matemáticos das transformadas e redes neurais, descrevendo-se os procedimentos e fundamentação matemática de como extrair as características, e as possibilidades de variação de métodos de extração das características a serem repassados a rede neural.

No Capítulo 3 está descrito o tratamento dos dados, suas normalizações e os registros das variâncias, mostrando o porquê das normalizações e preparativos para os testes, assim como uma descrição sobre o problema das variantes de tamanho, orientação e complexidade de representação do símbolo, e também uma descrição de porquê as técnicas de normalização foram utilizadas, e também, uma descrição da base de dados utilizada.

No Capítulo 4, apresentam-se as topologias dos classificadores utilizados, os gráficos com os erros de treinamento, descrevem-se os resultados obtidos e realiza-se uma análise dos experimentos realizados.

O Capítulo 5 descreve as conclusões obtidas com o presente trabalho, bem como relacionam-se os possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Introdução

Este capítulo apresenta as fundamentações necessárias para a execução do projeto apresentado no Capítulo 1, com abordagens sobre a morfologia matemática na imagem como um pré-tratamento antes da sua manipulação para que tenha as suas primitivas extraídas e reconhecidas.

Existem muitas técnicas e tipos de representações de conhecimento usadas por pessoas nos campos de Inteligência Artificial e Técnicas Sintáticas ou Técnicas Estatísticas, como por exemplo: em reconhecimentos e localizações de padrões, e pesquisas sobre uma determinada base de dados. Optou-se por métodos estatísticos e utilização de Inteligência Artificial por diferentes razões [FER86]:

- a) A independência de um conhecimento prévio dos símbolos e do contexto da sua utilização, implicando em uma estrutura dimensional de agrupamento mais simples.
- b) As aplicações de reconhecimento de padrões, estão concentradas onde não há limites nas técnicas e tipos das representações que possam ser aplicadas, possibilitando uma vasta gama de opções para extração de características e formas de reconhecimento, como a combinação das Técnicas Estatísticas.
- c) Eliminação de sub-grafos, obtidos com Técnicas Sintáticas, obtidas pelas composições gramaticais, que tendem a dificultar a formação de agrupamentos, caso não se obtenham informações adicionais sobre o objeto sob análise.

Estes argumentos justificam a utilização das técnicas que serão abordadas neste trabalho para resolução de problemas de rotação e escalas. As técnicas estatísticas aqui utilizadas servem como redutores de dimensionalidades, diminuindo a quantidade de informações a serem introduzidas nos classificadores.

Neste capítulo, retratar-se-á os trabalhos realizados por outros autores, trabalhos tais que forneceram subsídio para a pesquisa, e cuja seqüência de análises está mostrada na Figura 2.1. Os dados. Devem ser agrupados em suas classes, ter as suas características extraídas, posteriormente classificados e dispostos em regiões de decisão. Para que seja possível esta etapa, abordar-se-á inicialmente os métodos de extração de características (para reduzir a quantidade de informações para o classificador, reduzindo a dimensionalidade), gerando uma nova base de dados mais compacta e de simples análise, e finalmente uma abordagem sobre os classificadores.

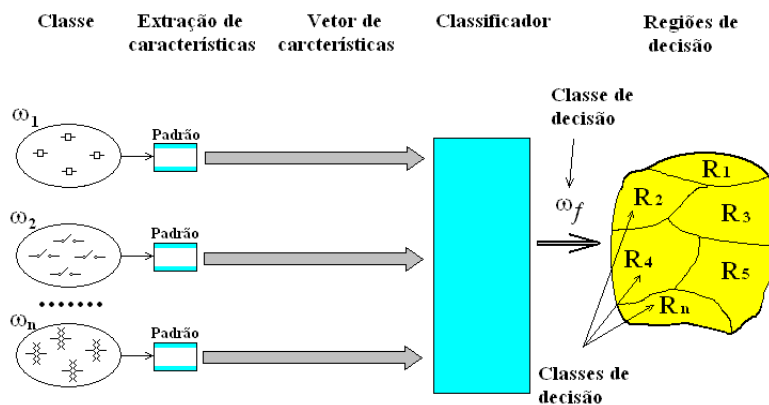


Figura 2.1: Seqüência de manipulação dos dados para análises.

2.1. Morfologia matemática

O primeiro item a considerar em processamento de imagens, base de todos os trabalhos e pesquisas, consiste na morfologia da imagem. A Morfologia consiste em manipular e preparar a imagem para um pós-processamento. Facon [FAC96], em seu livro apresenta as definições e técnicas de manipulação de uma imagem com os trabalhos de limiarização, erosão e dilatação em imagens binárias.

O histograma de uma imagem, como define Facon [FAC02], é a somatória da quantidade dos níveis de cinza por nível, ou seja, cria-se um vetor com todos os níveis de cinza e contabiliza-se a quantidade de elementos por nível.

A limiarização, segundo a definição de Facon [FAC02], é baseada no histograma da imagem: Todos os pixels considerados abaixo de um valor de referência, são convertidos para níveis de preto, e os acima, convertidos para níveis de branco. Assim, obtém-se a partir de uma imagem com tons de cinza, uma nova com duas cores. Após limiarizada a imagem, para reduzir o tamanho e/ou os seus valores numéricos de 0 (preto) e 255 (branco), pode-se binarizá-la, convertendo os valores de 255 para 1, representando a cor branca. Obtém-se assim uma representação mais compacta da imagem.

A operação de afinamento, como definido por [FAC96], define-se como: dado um conjunto X de pontos, consiste em retirar dele pontos que correspondam a uma configuração dada. Operações de afinamento representam processos homotópicos¹, portanto não causam destruição das propriedades de conectividade e preservam as suas propriedades de similaridades. Caso o resultado tenha sido alcançado, os processamentos adicionais não destruirão o resultado obtido. Neste nível, as transformações morfológicas de afinamento tornam-se idempotentes².

Uma avaliação dos métodos de limiarização para aplicações em testes não destrutivos é feita por Sezgin & Sankur [SEZ01], comparando 41 algoritmos. Estes estão agrupados em seis categorias baseadas nas informações a serem exploradas, tais como forma do histograma, atributos do objeto ou procedimento de agrupamento. As suas avaliações de desempenho estão baseadas na combinação ponderada de objetivas métricas complementares, as quais são:

- a) Informação a partir da forma do histograma, com detecção de vales e planícies,
- b) Informação da entropia do histograma, com a maximização da soma das distribuições dos níveis de cinza do objeto a analisar,
- c) Agrupamento das informações dos níveis de cinza, através do algoritmo de Otsu,
- d) Informações do atributo da imagem, aproximando a borda dos campos a partir do nível de cinza da imagem original,
- e) Informação espacial do contexto, pela sua probabilidade de co-ocorrência, e,

¹ Que significa *homós* do grego *igual, semelhante* e da palavra latina *tópica*, originária do grego que significa *doutrina que estuda os tópicos, em retórica*: homotópicos significa estudo dos tópicos semelhantes.

f) Adaptação local, ou seja, determinando-se a variação do nível de cinza em função da sua variação local.

Baseados nos resultados de tais testes não destrutivos, imagens incompletas como as térmicas (por infravermelho), por ultra-sonografias, por contorno de borda, etc., que utilizam os algoritmos de limiarização de forma eficiente, empregam na maioria dos casos de uma mistura de técnicas limiarizadoras [SEZ01]. Os critérios para avaliação quantitativa utilizados pelos autores foram os seguintes:

- a) Erro por má classificação, informando o “quanto a mais foi classificado”.
- b) Casamento por borda, utilizando-se das perdas obtidas pelas diferenças nas bordas da imagem.
- c) Região de não uniformidade, baseando-se na variância total da imagem, na variância da imagem de “*foreground*” e na probabilidade da classe de “*foreground*”.
- d) Erro relativo da área de “*foreground*”, apurando-se medidas como áreas e formas. As comparações feitas com imagens segmentadas com respeito à imagem de referência fornecem medidas discrepantes.
- e) Medida da avaliação pela lógica “*Fuzzy*”, baseada na conjectura de que a fidelidade da borda, a região de uniformidade e o erro da área relativa de “*foreground*” combinados, possam apresentar um resultado qualitativo.

2.2. Manipulações de documentos técnicos

Um sistema automático de reconhecimento gráfico que pode gerar uma descrição sucinta de várias primitivas gráficas e as suas relações espaciais têm muitas aplicações, conforme Kasturi & Triverdi [KAS90]. O seu trabalho refere-se à conversões de desenhos baseados em papel para integração em um ambiente CAD (Projeto Auxiliado por Computador), como por exemplo: um armazenamento eficiente, possibilidade de transmissão de dados de textos/gráficos misturados, e criação de versões atualizadas de documentos existentes.

Um diagrama gráfico típico é composto por linhas, polígonos, círculos, e outras curvas de ordens mais altas. Estas entidades têm vários atributos associados a eles. Exemplos de tais atributos são espessuras de linhas e preenchimentos de polígonos com detalhes, etc. Frequentemente, estas

² Proveniente do latim, que tem a mesma consideração.

entidades se sobrepõem e obscurecem algumas das linhas. Os gráficos são enfocados tipicamente com seus textos.

Documentos baseados em papel, que contêm dados misturados de textos e gráficos, são digitalizados em resolução de 120 pixels por centímetro. Para se obter um arquivo de descrição gráfica no nível mais alto possível com interação mínima do operador, as seguintes operações são executadas na imagem binária:

- a) Separação dos textos dos gráficos,
- b) Identificação de segmentos de linha e os seus atributos,
- c) Reconhecimento das primitivas gráficas e os seus atributos, e
- d) Descrição de relações de espaço entre primitivas.

Kasturi & Triverdi [KAS90], então propuseram o desenvolvimento de um algoritmo robusto para segmentar a imagem em texto e gráfico, com processamentos independentes para gráficos e textos, onde destacam-se:

a) As imagens de gráficos são processadas através de rotinas de processamento de imagem de baixo nível para identificar vários tipos de linhas. Entidades afinadas como linhas são representadas pelos seus traços em si, e entidades espessas (como por exemplo, polígonos completamente preenchidos), são representadas pelos seus contornos. Também é tratado um algoritmo de detecção de linha. Os segmentos de linha são ordenados em laços simples e complexos, usando um número de regras heurísticas. São reconhecidas formas simples como polígonos regulares. Os laços complexos são identificados como polígonos simples com sobreposição de objetos, sempre que possível. Identificam-se vários padrões sombreados com tracejados dentro de laços fechados. São encontradas as relações espaciais entre várias primitivas. Um grande número de primitivas pequenas, se inseridas em objeto maior, é identificado como preenchimento (textura). Linhas curvas são processadas para identificar círculos. Linhas curvas que não são segmentos de círculos são descritas como arcos circulares concatenados. O arquivo de descrição gerado pelo sistema contém toda a informação para reconstruir os dados originais. Claramente, o ponto de partida principal deste sistema proveniente de sistemas de compressão de dados convencionais é que geram uma descrição dos conteúdos dos gráficos e não simplesmente dados em uma forma comprimida para armazenamento,

b) As imagens de textos são processadas por um sistema de reconhecimento de caractere. O método para reconhecimento de caracteres e símbolos em um desenho consiste em 2 passos: 1) Detectar-se áreas dentro do desenho que contenham caracteres ou símbolos, e 2) Determinando-se tipos de caracteres ou símbolos em cada área detectada.

No seu livro, Kasturi & Triverdi [KAS90] abordam a conversão de diagramas de circuitos lógicos (em circuitos eletrônicos), que normalmente consistem em três elementos principais: símbolos que representam os componentes do circuito, linhas que representam ligações e conexões, e caracteres e números que representam identificação de componentes e atributos relacionados. O reconhecimento de tal circuito de esquema lógico pode ser executado pela seguinte seqüência: Primeiro separam-se caracteres e números de símbolos de componentes e linhas de ligações, baseado em uma característica de relação e tamanho que assume que caracteres e números são figuras pequenas, isoladas. Logo, executa-se a localização do contorno para a imagem de caracter e número isolados. Uma análise da estrutura do grupo resultante de coordenadas de pontos que compõem o contorno é então executada para reconhecer os caracteres e números.

A parte da imagem que contém os símbolos de componentes restantes e linhas de fiação são afinadas, e as linhas medianas resultantes são convertidas a um grupo de coordenadas de ponto. Aplica-se uma operação de afinamento à imagem de símbolos e linhas restantes. Disto, pontos de característica como pontos terminais, pontos de “*jumpers*”³, pontos de cruzamento e pontos de canto são extraídos e analisados, resultando nas informações de conexão para as ligações. Laços também podem ser extraídos dos dados de linha medianos. Analisando a forma do laço, são reconhecidos tipos de componente em uma das 4 orientações nas direções horizontais e verticais.

A partir deste trabalho, Kasturi & Triverdi sugerem uma metodologia para reconhecimento de componentes e caracteres e a sua substituição por símbolos pré-escritos, fontes com formas padrões, e linhas reconhecidas a serem endireitadas para estarem na horizontal e/ou vertical. Outras produções finais na tarefa de reconhecimento de desenho incluem uma tabela de componentes e uma tabela de informações de ligações elétricas entre cada componente.

Como uma orientação para a montagem da base de dados, e criar uma interface com metodologias em segmentação, o método proposto por Likforman-Sulem et al. [LIK95] utiliza a Transformada de Hough para detectar linhas de textos em páginas manuscritas, rasuras ou

anotações orientadas em diversas direções. O método possui uma estratégia de validação a qual é iterativamente ativada até que o fim da segmentação seja atingido. A cada estágio do processo, a melhor hipótese de linha/texto é gerada pelo domínio de Hough, obtidas pela contabilização das flutuações dos componentes de linha/texto. Subseqüentemente, a validade da linha é verificada no domínio da imagem usando o critério da proximidade o qual analisa o contexto em que é percebida a hipótese do alinhamento. Componentes ambíguos entre as linhas são também marcados, para análises.

Uma metodologia para extração de símbolos em desenhos baseada em multi-agentes foi proposta por Keiji [KEI00]. Todos os símbolos devem ser previamente cadastrados, formando um gabarito para pesquisa. As pesquisas percorrem os caminhos longitudinais descendentes e ascendentes na imagem. Os agentes são definidos como:

- a) Espécie, informando a natureza do objeto a extrair,
- b) Tipo, informando a família ao qual o objeto pertence,
- c) Plausibilidade, medições que avaliam as restrições qualitativas,
- d) Relacionamento, com as restrições simbólicas entre pai e filho,
- e) Área de pesquisa, definindo aonde os agentes deverão atuar.

Esta proposta assume que cada agente encarrega-se da tarefa de extrair um objeto de uma imagem, e através dos relacionamentos entre agentes, determina-se o símbolo a segmentado.

Uma tentativa para reconhecimento de componentes em desenhos de CAD voltada para circuitos eletrônicos foi feita por Cheng et al. [CHE93], utilizando-se de uma rede neural artificial hierárquica e servindo como protótipo de um sistema de reconhecimento de símbolos para processamento automático de desenhos de engenharia elétrica. Esta abordagem obteve uma significativa redução do envolvimento humano no modelo da codificação do símbolo e melhora nos processos de reconhecimento, em contraste às técnicas tradicionais baseadas em afinamentos (técnicas de erosão), percurso de linhas e outras técnicas estruturais. Este trabalho pode ser resumido como a criação de 4 metaclasses (ou grupos) para a divisão dos símbolos, e a utilização de momentos elípticos aplicados aos símbolos para determinar-se a extração de características. Estes vetores característicos (de 6 dimensões) são aplicados à uma rede neural artificial hierárquica, dividida em dois níveis de classificação (um classificador primário que identifica a que grupo o

³ Definição técnica utilizada para interligações externas ao circuito.

símbolo pertence e ativa uma outra rede neural artificial para a tarefa de reconhecimento do símbolo dentro daquele grupo). Cheng et al. também avalia a sua estrutura de redes quanto ao desempenho e precisão, definindo-a como eficiente.

A partir de desenhos feitos em papel, Jisheng [JIS00] faz um estudo sobre a digitalização de desenhos a partir de um OCR (*Optical Character Recognition* – Reconhecimento Óptico de caracteres) e propõe uma metodologia de reconhecimento de símbolos especiais. Este trabalho baseia-se no reconhecimento de símbolos gregos e/ou matemáticos misturados em documentos de texto. A proposta baseia-se na utilização de uma função trabalhando em paralelo à metodologia de reconhecimento, analisando se o símbolo pertence ou não ao conjunto das exceções, consideradas como os símbolos gregos e matemáticos.

O problema de interpretação automática de imagens de desenho em engenharia ou reconhecimento gráfico vem sendo estudado por um longo tempo, conforme Chhabra & Phillips [CHH00]. Nos recentes anos, vários novos métodos de vetorização foram propostos, e muitos sistemas comerciais foram desenvolvidos [CHH00]. Neste artigo, Chhabra & Phillips propõem uma métrica de avaliação de desempenho de algoritmos que fazem a função de converter desenhos técnicos em engenharia na forma de arquivos vetoriais, utilizando como base de dados desenhos em engenharia mecânica e arquitetônica. A partir de arquivos gerados pelo Autocad⁴, cria-se um padrão de saída, salvando o arquivo na forma *vectorial container* (Arquivos no formato “.VEC”) e realiza-se a conversão para o formato TIFF, sendo este o seu padrão de armazenamento dos dados.

Uma abordagem de reconhecimento de símbolos utilizando algoritmos com sub-grafos de isomorfismos tolerantes a erros, é feita por Lladós et al. [LLA01], trabalhando com grafos de regiões de adjacências. Nestes ensaios, utilizou-se uma base de dados de desenhos poligonais com formas irregulares e com variações de níveis de distorção. Lladós et al. definem isomorfismo como a diferença matemática entre os grafos registrados durante as análises, como por exemplo, o mapeamento bijetivo que associam os nós entre as imagens sob análise e a de referência. Se um dos grafos envolvidos no processo de casamento (*matching*) for maior que o outro, em termos de número de nós, então o casamento é realizado por um isomorfismo de sub-grafo. O isomorfismo de sub-grafo entre os grupos de análise e o de referência, significa que entre dois grupos de grafos G_1 (identificando o grafo a analisar) estiver contido G_{ref} (grafo de referência), deve-se encontrar um

grafo G_2 (identificando o sub grafo de G_1) que seja igual a G_{ref} , determinando-se que G_1 seja isomórfico a G_2 . O método em si possui as desvantagens inerentes da complexidade de casamento de grafos e a demanda de carga computacional para a sua execução, mas como vantagem, a possibilidade de identificar a rotação do símbolo e variações de grandeza de escalas. Junto a abordagem dos métodos de grafos, faz-se também um comparativo de desempenho dos métodos de casamento de grafos com complexidade computacional quanto a isomorfismos de grafos, sub-grafos, tolerância a erros, otimizações e níveis de complexidade.

Como um trabalho direcionado para manuscritos, Madhvanath & Govindaraju [MAD01] direcionam os seus esforços no reconhecimento, utilizando métodos no domínio *online* e *offline* (como laços em palavras) e através da forma (através de uma envoltória, definindo figuras típicas) das palavras referenciadas. Madhvanath e Govindaraju categorizam como métodos holísticos:

- a) O Domínio, *online* e *offline*,
- b) Uso léxico, estático e dinâmico com aplicações em reduções e reduções léxicas,
- c) Características, de baixo nível, médio nível e alto nível,
- d) Representação característica: Vetores, inteiros, seqüências e grafos,
- e) Métodos híbridos, que explicitamente utilizam-se de uma combinação de metodologias.

A abordagem holística conseguiu melhores resultados de segmentação em documentos com falhas de escrita em relação aos métodos analíticos. Contudo, o seu tratamento de palavras léxicas como classes de padrões distintos têm sido tradicionalmente limitadas em aplicações de reconhecimento em cenários envolvendo léxicos pequenos e estáticos.

2.3. Extração de características

Uma abordagem geral sobre os métodos de extração de características de métodos *off-line* de reconhecimento de caracteres segmentados (isolados) é feita por Trier et al. [TRI95]. Nesta definição, o método de extração de características é possivelmente o mais importante fator para se obter um desempenho eficiente nos sistemas de reconhecimento de caracteres. O trabalho aborda diferentes representações de caracteres, tais como caracteres binários sólidos, contornos de caracteres, caracteres afinados ou caracteres com níveis de cinza de cada caractere individual. Os métodos são discutidos em termos de propriedades de invariantes, capacidade de reconstrução,

⁴ Programa comercializado pela Autodesk Inc.

faixas aceitáveis de distorções e variabilidade de caracteres. Os autores também debatem sobre o problema dos métodos de escolha da extração de característica e escolha do classificador, afetando o desempenho do sistema como um todo. Os métodos utilizados são:

- a) Para imagens com níveis de cinza, utilizou técnicas de *Template Matching* (Casamento de Pares), *Deformable Templates* (Gabarito Deformável), Transformadas Unitárias, *Zoning* (Zoneamento), Momentos Geométricos e Transformada de Zernike,
- b) Para imagens binárias, utilizou em caracteres sólidos as técnicas de *Template Matching*, Transformadas Unitárias, Projeção de Histogramas, *Zoning*, Momentos Geométricos e a Transformada de Zernike. Para contornos de caracteres, utilizou a técnica de Perfil de Contorno, *Zoning*, Curvas *Spline* e Descritores de Fourier,
- c) Para vetores ou imagens esqueletizadas (afinadas), utilizou *Template Matching*, *Deformable Templates*, Descritores de Grafos, Características Discretas, *Zoning* e Descritores de Fourier.

Mukundan et al. [MUK01], citam que funções de momentos em imagens bidimensionais são usadas em uma variedade de aplicações, como descritores. Os momentos (ou transformadas) que são invariantes quanto à escala, translação e rotação têm aplicações em áreas tais como reconhecimento de padrões, identificação de objetos e casamento de padrões. Mukundan et al. também citam que momentos ortogonais⁵ tem propriedades adicionais se forem mais robustos quanto a presença de ruídos na imagem, e têm uma redundância quase nula no espaço característico⁶. Os autores também comentam sobre os momentos de Zernike pela sua capacidade de representação das características de imagens com a sua base polinomial ortogonal radial e pelos polinômios cartesianos de Legendre. Os autores fazem experimentos com aproximações discretas (realizando aproximações numéricas, como por exemplo, com os polinômios radiais de Zernike) e com momentos ortogonais discretos (com os polinômios cartesianos de Legendre e Chebyshev).

Os padrões sintáticos, (obtidos através da elaboração de cadeias de códigos), tendem ao desuso devido à capacidade do tratamento formal de gramáticas ser limitado, principalmente pelas

⁵ Na geometria, orthogonal significa “envolvendo ângulos corretos”, (do grego *ortho* significando *certo* e *gon* que significa *angular*). O termo tem se estendido a usos gerais, significando a característica de serem independentes (relativo a algo). Pode também significar: não redundante, não sobreposta ou irrelevante.

⁶ O espaço característico ou vetorial é qualquer conjunto de objetos que possam ser somados entre si e multiplicados por escalares.

dificuldades de análises gramaticais, embora possa ser atenuado por pesquisas na linguagem formal [SCH92].

2.4. Classificadores

Um exemplo de classificação de maçãs é apresentado por Schalkoff [SCH92], dividindo o grupo em duas famílias, as das maçãs mais rubras e das maçãs mais pálidas. O autor define o mapeamento das maçãs como sendo as características (ou primitivas) as informações utilizadas para distinguir o grupo. Na sua expressão “de que o mapeamento define a escolha do sistema de medição”, o autor refere-se que para cada modelo de extração de característica, usa-se uma métrica distinta. Esta métrica envolve o sistema de extração de características e o modelo de agrupamento dos dados. Quanto melhor definido os grupos de dados, mais simples é a tarefa de classificação: no exemplo utilizado, quanto menos maçãs com cores mistas entre pálidas e rubras existir, menos complexa será a classificação das maçãs.

Schalkoff define que existem duas famílias de classificadores: Os que precisam de supervisão de aprendizagem (associando as primitivas obtidas aos modelos de referência para formar os respectivos agrupamentos) e os não supervisionados (em que o próprio classificador define os agrupamentos). Em ambos os casos, estes agrupamentos recebem o nome de espaço característico em classes definidas como regiões de decisões. Através desta perspectiva, a classificação do vetor característico torna-se mais simples pela determinação da região no espaço dimensional a que ele pertence, partindo de uma métrica a ser escolhida. Contudo, se a estratégia de classificação for direta, a decisão da região de decisão será um desafio. O autor também define que a aprendizagem supervisionada melhora o desempenho do classificador, porque um sistema de aprendizagem consegue adaptar a sua estrutura interna para obter a melhor resposta, a partir da técnica de correção baseada no erro. A medida de desempenho obtida pode ser a diferença entre a saída do sistema atual e o desejado [SCH92]. Aplicando-se iterativamente a técnica de correção de erro, a medida de desempenho define a curva de aprendizagem para cada iteração de aprendizagem.

Uma definição do classificador não supervisionado KNN (*K-Nearest Neighbor* ou *K-vizinhos mais Próximos*) é feita por Gose et al. [GOS96], como uma técnica para desviar do problema das densidades de probabilidades, classificando uma amostra desconhecida como pertencente à mesma classe ao mais similar (ou próximo) conjunto de dados no grupo de

treinamento, os quais são chamados de conjuntos de referências. O classificador utiliza a distância Euclidiana para avaliação dos conjuntos de referências. Os autores definem a contabilização do erro como desigual por agrupamento e dependente somente da real semelhança com a classe sem a confiabilidade de que esteja correta, portanto o número de vizinhos encontrados em cada classe deve ser ponderado por perdas de má classificação de uma amostra a partir da classe para minimizar o risco.

As Cadeias de Markov, como definidas por Davis [DAV93], são modelos com propriedades estocásticas e suficientemente genéricas para resolverem uma grande variedade de problemas categorizados como casos especiais. As Cadeias de Markov consistem de uma mistura de movimentos determinísticos e saltos aleatórios gerando estados.

As Cadeias de Markov são classificadores, conforme abordam Silva & Muntz [SIL92], afirmando que necessitam menos tempo para sua construção, são menos susceptíveis a erros, mas modelos realísticos frequentemente possuem um espaço de estados cuja cardinalidade torna proibitiva a solução. O problema da explosão do espaço de estados, como citam os autores, pode ser incrementado de tal forma que ultrapasse as limitações de memória, tornando-se a maior restrição do uso abrangente das Cadeias de Markov.

Simulated Annealing, surgiu através de problemas estatísticos da mecânica, baseados entre a analogia da têmpera de sólidos e otimização de problemas de solução combinacional, conforme abordam os autores Karaboga et al. [KAR00]. Os autores definem o algoritmo como uma seqüência de iterações. Cada iteração consiste de randomicamente mudar-se o resultado para criar uma nova solução na vizinhança da solução atual. A vizinhança é definida como a escolha do mecanismo de geração. Uma vez que a nova solução seja criada, a correspondente troca na função de perda seja computada à decidir quando a nova solução produzida seja aceita como a solução atual [KAR00]. Se a troca com a função de perda for negativa, a nova solução obtida passará a ser a atual. Caso contrário será aceita em função do critério de *Metropolis*, baseado na probabilidade de Boltzman. O autor define o critério de *Metropolis* como sendo a diferença entre a solução da função de perda corrente e da nova solução ser igual ou maior que zero, para que se gere um número aleatório (entre 0 e 1) a partir de uma distribuição uniforme para que a nova solução seja aceita como a atual.

O paradigma de reconhecimento é definido como 4 passos a se seguir, pelas definições de Rabiner e Juang [RAB93]:

Passo 1: Avaliação da característica, na qual a seqüência de medições a ser feita com o padrão de entrada recebe a designação de padrões de testes,

Passo 2: Padrões de treinamento, no qual um ou mais padrões correspondentes da mesma classe são usados para criar um padrão representativo das características daquela classe,

Passo 3: Padrões de classificação, no qual um padrão desconhecido de teste é comparado com cada padrão de referência de classe, e são medidas as similaridades (distâncias) com os padrões de referências,

Passo 4: Lógica de decisão, na qual as métricas das similaridades aos padrões de referências são usadas para decidir a qual padrão de referência (ou possivelmente a seqüência de padrões de referências) melhor se aproxima(m) do padrão de teste sob avaliação.

2.5. Classificadores neurais

Técnicas de Redes de Neurais Artificiais - RNA (*Artificial Neural Network - ANN*) [PRI99, SCH92, FRE92] foram desenvolvidas para simular sistemas de aprendizagem biológicos. De nosso conhecimento, para este caso, RNA provêm um grau mais alto de robustez e adaptação que classificadores não neurais. Além disso, metodologias baseadas em aproximação de rede de neural permitem um certo nível de ruído na entrada e podem ser adaptadas para aplicação em grandes domínios com mínimas modificações significativas. RNA facilitam extensões com mínimas perturbações ao modelo já codificado. Estas características, além das referenciadas por [PER97] para utilização de transformadas, conduzem a rede neural a ser o melhor candidato do classificador para a metodologia estudada. Vários modelos de RNA para resolução de problemas de reconhecimento de padrões foram propostos e aplicados satisfatoriamente [SCH92].

Para o caso de treinamento da rede neural, devido à pequena base de dados a ser utilizada, o treinamento realizou-se com validação cruzada. A base inicialmente gerada, contém 304 elementos gráficos, com os símbolos dos componentes representados dentro dos graus de liberdade tolerados pelas normas ABNT [ABNTa-d], e nos casos em que a norma não contempla, as variantes utilizadas (em senso comum) pelas equipes de projetistas.

2.5.1. Uma breve revisão sobre redes neurais

Como os computadores tornam-se a cada dia mais poderosos, os cientistas e pesquisadores são desafiados a encarregar às máquinas tarefas que são simples para os humanos. Para os seres humanos, baseados em exemplos, e junto com algumas realimentações a partir de um “professor”, pode-se facilmente reconhecer a letra “A” ou distinguir um gato de um rato. Quanto mais experiência do observador, melhor será o refinamento e melhor o desempenho de discriminação.

Pode-se definir então Redes Neurais Artificiais como um sistema de processamento de informação que tenha certo desempenho em comum com redes neurais biológicas. Redes Neurais Artificiais foram desenvolvidas como generalizações de modelos matemáticos da cognição humana ou biologia neural, baseadas nas seguintes pressuposições [FAU94]:

- a) O processamento das informações ocorre em elementos mais simples chamados neurônios,
- b) As informações são passadas aos neurônios como ligações de conexão,
- c) Cada ligação de conexão tem um peso associado, o qual define-se como uma típica rede neural, multiplicando o sinal transmitido,
- d) Cada neurônio aplica uma função de ativação (normalmente não linear) à sua malha de entrada (somando os sinais com as entradas ponderadas pelos seus pesos), para determinar o sinal de saída.

Uma Rede Neural Artificial é caracterizada pelos seus padrões de conexões entre os neurônios (chamada arquitetura), seus métodos de determinação dos pesos nas conexões (chamada de treinamento ou algoritmo de aprendizagem) e a sua função de ativação.

Uma Rede Neural Artificial consiste em um grande número de simples elementos processantes chamados neurônios, unidades, células ou nós. Cada neurônio é conectado a outros por ligações diretas, com cada ligação associada a um peso. Os pesos representam as informações que são utilizadas pela rede para resolver um problema. As Redes Neurais Artificiais podem ser aplicadas em uma variedade de problemas, tais como armazenamentos (memórias), classificação de padrões, realizar mapeamentos ou agrupamentos dos padrões de entrada ou ainda encontrar soluções a problemas com restrições às otimizações.

Cada neurônio tem um estado interno, chamado de ativação ou nível de atividade, o qual é uma função das entradas recebidas. Tipicamente, um neurônio envia sua ativação como um sinal a vários outros neurônios (sinais S_1 , S_2 e S_3 na Figura 2.2 – no caso somente um neurônio). É

importante notar que um neurônio pode enviar somente um sinal ao mesmo tempo, contudo este sinal é difundido a todos os neurônios a ele conectados.

Como exemplo, considerando-se o neurônio ‘a’ ilustrado na Figura 2.2, que recebe como entradas as saídas dos neurônios 1, 2 e 3. As ativações (sinais de saída) destes neurônios são S_1 , S_2 e S_3 respectivamente. Os pesos das conexões a partir dos neurônios 1, 2 e 3 para o neurônio ‘a’ são w_1 , w_2 e w_3 . A malha de entrada para o neurônio ‘a’ é a soma dos sinais $S_{1,2,3}$ dos neurônios 1, 2 e 3 multiplicados pelos seus respectivos pesos.

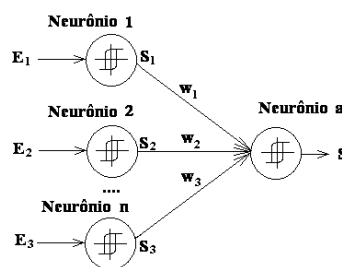


Figura 2.2: Exemplo de interligação entre neurônios

Assim, podemos escrever a saída S do neurônio ‘a’ como sendo a equação 2.1, sendo f_{ativ} é a função de ativação do neurônio

$$S = f_{ativ}(w_1 S_1 + w_2 S_2 + w_3 S_3) \quad (2.1)$$

A função de ativação pode ser uma composição linear, sigmoidal, tangencial, limiar (por patamar) ou qualquer outra escolhida para exercer a sua função, desde que haja convergência, conforme mostrado na Tabela 2.1.

2.5.2. Arquiteturas típicas

Freqüentemente, torna-se conveniente visualizar os neurônios arranjados em camadas. Tipicamente, neurônios na mesma camada se comportam de forma semelhante. Os fatores chaves na determinação do comportamento de um neurônio, são a sua função de ativação e o mesmo padrão de pesos nas conexões para quem envia e recebe sinais pelas suas conexões. Dentro de cada

camada, os neurônios normalmente têm a mesma função de ativação e o mesmo padrão de conexões que outros neurônios. Para ser mais específico, em muitas Redes Neurais Artificiais, os neurônios dentro de uma camada podem estar totalmente interconectados ou não totalmente interconectados. Se qualquer neurônio na camada (para a instância, a camada escondida) estiver conectado a um neurônio em outra camada (diga-se camada de saída), então cada unidade da camada escondida está conectada a cada neurônio de saída [PRI99].

Tabela 2.1: Exemplos de funções de ativação [PRI99]

Função de ativação	Modelo matemático
Identidade	$f(x_i) = x_i$
Tangente hiperbólica	$f(x_i) = \tanh(\alpha \cdot x_i)$
Sigmoidal	$f(x_i) = \frac{1}{1 + e^{-\alpha x_i}}$
Sigmoidal bipolar	$f(x_i) = \frac{2}{1 + e^{-\alpha x_i}} - 1$
Logarítmica	$f(x_i) = \log(1+x_i)$ para $x_i < 0$, e $f(x_i) = -\log(1-x_i)$ para $x_i > 0$
Limiar	$f(x_i) = -1$ para rede ≤ 0 , e $f(x_i) = +1$ para rede > 0
Gaussiana	$f(x_i) = \frac{1}{\sqrt{2\pi\delta}} e^{-\frac{(x_i-u)^2}{2\delta^2}}$ onde $u = \frac{1}{N} \sum_{i=1}^N x_i$ e $\delta^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - u)^2$

A disposição dos neurônios em camadas e os padrões de conexão intra e extra-camadas é chamada **Arquitetura da Rede**. Muitas redes neurais têm uma camada de entrada com a função de ativação identidade (com a saída igual a entrada).

As redes neurais são sempre classificadas como camada simples (“*single layer*”) ou multicamadas (“*multilayer*”). Na determinação do número de camadas, os elementos de entrada não contam como uma camada, porque eles não realizam processamento algum. Equivalentemente, o número de camadas na rede pode ser definido como um número de camadas de ligações com pesos interconectados entre os grupos de neurônios. Esta perspectiva é motivada pelo fato de que os pesos

da rede contém informações extremamente importantes. A rede mostrada na Figura 2.3 contém duas camadas de interligações com pesos, portanto uma rede multicamadas. A Figura 2.2 é um exemplo de uma rede de camada simples ou perceptron.

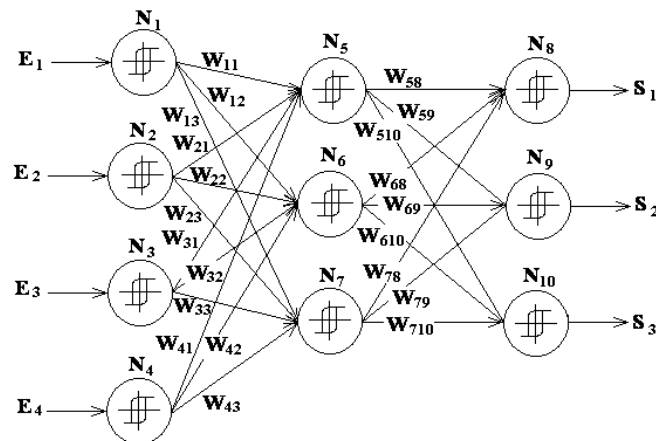


Figura 2.3: Exemplo de Rede Neural Artificial com duas camadas

Para as redes, há ainda outro elemento, chamado de “*bias*”, que é um “*offset*” (deslocamento de valores) adicionado aos neurônios. Ele pode ser definido como um neurônio de valor constante, agindo com os pesos em suas conexões, auxiliando no treinamento. O “*bias*”, normalmente não é representado na arquitetura da rede.

2.6. Análises de confiabilidade

Desde que redes neurais convencionais foram utilizadas em reconhecimento de caracteres para classificar um grande número de classes, a complexidade de determinação dos contornos ideais das regiões de decisão inevitavelmente aumentou e passou a envolver um espaço característico de maior dimensionalidade, de acordo com Oh & Suen [OH_02]. Neste artigo, os autores introduzem o conceito de modularidade de classe, decompondo o problema das regiões de contorno em subproblemas, com menores complexidades e como uso de grandes grupos de classificações para testar a capacidade de reconhecimento do número de classes.

A maioria dos algoritmos de aprendizagem operam de forma estável em seus domínios com informações relevantes, e seus desempenhos degradam com a presença de informações irrelevantes

ou redundantes, como descrito por Barabidharan & Ioerger [BAN03]. A proposta dos autores está na aprendizagem seletiva ou focada para a solução deste problema. Os autores apresentam duas componentes de aprendizagem seletiva que são: a observação seletiva (pela extração de características) e a utilização seletiva (seleção de exemplos). Estes métodos são combinados para se avaliar a relevância da característica para se determinar a construção de hipóteses consistentes, melhorar a precisão da predição e reforçar o conceito de informação relevante.

Um estudo sobre avaliação de desempenho em alguns classificadores com reconhecimento de dígitos manuscritos é descrito por Liu et al. [LIU02]. Os estudos mediram a precisão de classificação, sensibilidade ao tamanho das amostras de treinamento, rejeição de ambigüidades e resistência a símbolos de rejeição. Para estes casos, os autores propõem o uso de elementos modulares de testes de rejeição, chamados de *outliers*. Os símbolos *outliers* podem ser definidos como modelos de discordância para a classe (símbolos de indução a erro), de onde a extração de suas características não pertença à distribuição do espaço característico da classe. Os autores reforçam que para manuscritos, as redes neurais MLP são inadequadas para símbolos *outliers*, mas podem melhorar os seus desempenhos se forem treinadas com os símbolos *outliers*. Os classificadores avaliados foram os estatísticos (Bayesianos), Redes Neurais Artificiais e baseados no classificador LVQ (*Learning Vector Quantization* – Aprendizagem de Vetor Quantizado), aonde os classificadores supervisionados apresentaram um desempenho superior.

Define-se a precisão do classificador como a probabilidade condicional de que um padrão seja corretamente classificado, conforme mencionam Fumera et al. [FUM00]. Os autores investigam o erro estimado na regra de Chow (regra que fornece uma relação otimizada de rejeição) e propõe o uso de múltiplos limiares relacionados às classes de dados. Os resultados experimentais obtidos mostram que tais classes relacionadas aos limiares apresentam um desempenho superior à regra isolada de Chow. Os autores sugerem um equilíbrio entre o reconhecimento e a rejeição com taxas variando entre 0 e 20%, desde que as taxas obtidas sejam relevantes à aplicação do seu processo.

Dois importantes conjuntos de ferramentas são descritos por Moody [MOO94] para estimar e modelar o risco de predição em Redes Neurais Artificiais. O autor define o risco de predição como o desempenho esperado de um estimador a definir novos valores a partir de uma observação. Estes valores podem ser utilizados para se determinar a qualidade do modelo em uso e a seleção do

modelo, ambos especialmente importantes para quantidade limitada de dados. O autor sugere o uso de validação cruzada para experimentos com limitações na quantidade de dados que formam a base de dados e também o uso da Predição de Erro Quadrático Generalizada.

2.7. Comentários finais

Apresenta-se aqui um resumo dos trabalhos relevantes para elaboração dos estudos de reconhecimento de símbolos gráficos em projetos de engenharia elétrica a partir de desenhos realizados em arquivos de imagem.

Metodologias e algoritmos para conversão de desenhos adquiridos por digitalizadores em arquivos vetoriais, separando o processamento de traçados gráficos do processamento de cadeia de textos são apresentados por Kasturi e Triverdi [KAS90]. Também é proposta a idéia para desenvolvimento de um algoritmo que execute a digitalização de componentes eletrônicos, efetue o seu reconhecimento e monte as tabelas de interligações.

Os métodos apresentados por Facon [FAC02, FAC96] e Sezgin e Sankur [SEZ01], fornecem um subsídio para a manipulação das informações em uma base de dados a ser utilizada, permitindo uma seqüência de experimentos em imagens binárias, reduzindo a faixa de domínio numérico a se manipular.

A Transformada de Hough foi utilizada por Likforman-Sulem et al. [LIK95] para determinar segmentos de retas e/ou linhas de textos. Esta metodologia permite detectar e extrair traços do desenho, independentes das suas orientações.

O sistema proposto por Keiji [KEI00] consegue obter resultados que satisfaçam restrições simbólicas nos relacionamentos entre os objetos definidos na base de conhecimento dos respectivos agentes. Uma característica interessante deste sistema é a sua versatilidade quanto á configuração dos símbolos a segmentar.

Um critério para avaliações de desempenho de novas metodologias que realizam vetorizações de desenhos, comparando-os à uma métrica estabelecida como referência foi fornecido por Chhabra et al. [CHH00].

Um uso de metodologias operando em paralelo para reconhecimento de caracteres e símbolos especiais foram propostos por Jisheng et al. [JIS00]. Esta idéia pode ser adaptada para acompanhamento de metodologias de extração de características a operarem em paralelo.

O método proposto por Lladós et al. [LLA01] fazem o reconhecimento de símbolos com tolerância à rotação e escala a partir de um padrão de imagem, junto com um acompanhamento do desempenho dos classificadores quanto aos isomorfismos de grafos, otimização e complexidade da classe, fornecendo uma referência de análise dos métodos.

Sistemas holísticos, conforme Madhvanath & Govindaraju [MAD01], são capazes de distinguir modelos de sub-palavras na hierarquia por processamento visual devido a sua relativa baixa complexidade, contudo esta distinção é subjetiva [MAD01]. Normalmente, em reconhecimento de palavras, utiliza-se um método de extração de características ou um descritor estrutural para reconhecimento das palavras. Esta técnica foi ponderada para o direcionamento do reconhecimento de símbolos por métodos estruturais, mas desmotivada, considerando-se a devida análise da complexidade gráfica.

Vários métodos para extração de características foram avaliados por Trier et al. [TRI95]. Os autores alertam para a escolha do classificador a se utilizar, sob pena de perdas globais de desempenho. Também salientam que antes de se escolher um método específico de extração de características, deve-se estudar bem o sistema no qual o processo deverá operar, prevendo situações de rotação, escala e distorção na imagem a se analisar. Trier et al. utilizou como exemplo a avaliação da funcionalidade da extração de características com a Transformada de Zernike em um mapa hidrográfico.

Os problemas relacionados à implementação de momentos ortogonais contínuos, tais como erros de aproximações e a necessidade de transformação a partir da coordenada espacial de uma imagem para um espaço de coordenadas normalizadas da imagem foram discutidos por Mukundan et al. [MUK01]. O autor comparou os momentos de Legendre e Chebyshev. Chebyshev apresentou uma capacidade de desempenho superior em relação aos momentos contínuos em seus trabalhos de reconstrução de imagem.

A definição do classificador não supervisionado KNN que utiliza a distância Euclidiana para determinar dissimilaridade (ou distância) do dado foi apresentado por Gose et al. [GOS96]. Os autores sugerem que se execute diversos experimentos com vários conjuntos de agrupamentos para que se possa determinar a dimensão do conjunto de agrupamento mais eficiente para cada aplicação.

Os processos estocásticos de Markov apresentando exemplos, análises de distribuições, controles e relaxações foram discutidos por Davis [DAV93]. A análise do autor introduz que a topologia do controle da classe deve ser compacta e as trajetórias das soluções dependem continuamente de seus controles.

A abordagem sobre o aumento de demanda de recursos exigidos pelas Cadeias de Markov durante o treinamento, e também sobre a facilidade de construção das cadeias em relação aos demais classificadores foi realizada por Silva & Muntz [SIL92].

O algoritmo de *Simulated Annealing*, é apresentado por Karaboga et al. [KAR00] para a representação das soluções, definições das funções de perdas (erros) e mecanismos de vizinhanças.

Por fim dos trabalhos de classificação, Rabiner [RAB93] define as 4 etapas para avaliação dos classificadores, como a avaliação do vetor característico, padrões de treinamento e classificação e a lógica de decisão.

As abordagens de modularidade descritas por Oh & Suen [OH_02] e Barabidharan & Ioeger [BAN03], separam o conjunto de dados em sub-conjuntos de menor complexidade, como tentativa de melhorar a confiabilidade do sistema classificatório.

As abordagens seletivas combinadas como tentativa de se melhorar a confiabilidade do sistema classificatório identificando-se a relevância da característica diferenciada pelo método empregado foram descritas por Barabidharan & Ioerger [BAN03].

A introdução de símbolos *outliers* no seu trabalho de avaliação do desempenho de vários classificadores, reforçando que os classificadores apresentam um desempenho superior quando treinados com elementos de rejeição foram feitas por Liu et al. [LIU02].

As opções de rejeição com vários limiares, demonstrando resultados iguais ou superiores na precisão da tarefa de classificação, propondo um equilíbrio entre resposta aceitável e restrição da rejeição foram apresentadas por Fumera et al. [FUM00].

O uso de validação cruzada para a realização dos experimentos com modelagens não paramétricas e quantidades limitadas de dados em suas bases, assim como a utilização de modelos de Predição de Erro Quadrático Generalizado foram descritos por Moody [MOO94].

Todos estes autores subsidiaram os estudos neste trabalho, contribuindo com as idéias e os protocolos de experimentos a serem utilizados. No próximo capítulo, apresenta-se a metodologia utilizada para a realização dos experimentos.

Capítulo 3

Metodologia

Introdução

Este capítulo apresenta a técnica de manipulação dos dados, a sua preparação e como foram disponibilizados para testes, explicando de que forma os resultados foram atingidos. Observa-se que a base de dados é convertida para a forma binária, porque não existem gradientes de cores em desenhos técnicos unifilares utilizados em diagramas de engenharia elétrica.

3.1. Agrupamento dos dados: Protocolo

Para a área elétrica industrial, os recursos ainda estão incipientes, e este trabalho baseia-se na identificação de símbolos a partir de 5 metaclasses [TON03b], todos já segmentados, e com seus terminais de conexão já definidos com tamanho padrão, proporcional a sua altura.

Os símbolos foram extraídos de desenhos em engenharia feitos por CAD (Autocad), convertidos para o formato *bitmap* (no padrão MS-Windows), e segmentados manualmente, com o cuidado de delimitar o comprimento dos terminais proporcional a 10% da largura do componente, para garantir a um padrão de segmentação e manter o símbolo identificável visualmente. Este valor foi assim definido por questão visual, onde em desenhos de engenharia, os símbolos preservam uma distância mínima próxima a metade do seu tamanho, evitando assim uma confusão visual aos técnicos que deles se utilizem. Poderia ser fixada uma outra relação. Então, para esta heurística, a distância mínima entre os componentes é de 20% das alturas, conforme Figura 3.1.

Esta reserva é para garantir trabalhos futuros em segmentação dos componentes e montagens de topologias a partir de desenhos unifilares. A Figura 3.1 mostra o porquê destes valores.

Os símbolos então são convertidos para a forma binária e têm sua imagem invertida (com o preto como cor de fundo), a fim de minimizar os tetos numéricos atingidos pelas transformadas, conseqüentemente menores valores numéricos a se manipular.

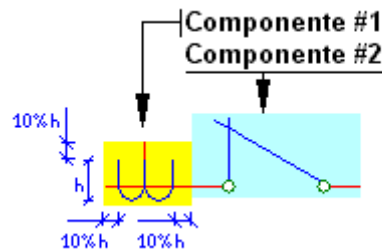


Figura 3.1: Comprimento dos terminais em relação á altura do componente

3.2. Redução de dimensionalidades

A técnica de redução de dimensionalidade foi introduzido como um modo de superar a maldição da dimensionalidade (*curse of the dimensionality*) quanto ao procedimento com vetores de dados em espaços de alta dimensionalidade [PER97]. Esta técnica é definida como a pesquisa para múltiplos espaços de baixa dimensão que abrangem o espaço de alta dimensionalidade. Neste trabalho seguem-se várias técnicas de análises de redução de dimensionalidades, e também técnicas referentes a análises destas dimensionalidades.

Redução da dimensionalidade pode ser sempre utilizada em agrupamentos, classificação, aprendizagem computacional e *data mining* [PER97].

3.2.1. Uma breve análise

Algoritmos de agrupamentos são uma ponte entre o subespaço dimensional e o espaço original, fornecendo flexibilidade e facilidade de implementação. Análises de agrupamentos realizadas em Gaussianas extremamente sobrepostas (como análises de códigos genéticos de DNA e processamento de imagens) demonstram a eficiência da técnica de redução de dimensionalidades [DIN02]. Porém, há o risco de, se for mal escolhido, afetar o sucesso da aprendizagem da máquina,

causando o fracasso do projeto. Além desta opção de escolha, a qualidade da base de dados a se analisar é importante, pois se houverem informações irrelevantes ou redundantes, e/ou ainda com muito ruído e pouca confiabilidade, implicam em uma dificuldade maior de reconhecimento e classificação.

Para que o processo de reconhecimento seja eficiente, deve-se adotar um procedimento para criar os subespaços com as características dos objetos a se analisar, retirando as informações desnecessárias.

3.2.2. A Maldição da Dimensionalidade e o fenômeno do espaço vazio

A Maldição da Dimensionalidade (termo concebido por Bellman em 1961) [PER97], refere-se ao fato de que, na ausência de garantias que possam simplificar, o tamanho da amostra necessária para estimar a função de várias variáveis para um dado grau de precisão, cresce exponencialmente com o número de variáveis.

Por exemplo, conforme Perpiñán [PER97], a maioria dos estimadores de densidade baseiam-se nas suas estimativas de observação em algum local médio das suas vizinhanças. Mas em ordem para localizar vizinhos suficientes em espaços de alta dimensionalidade, estimadores com multivariáveis devem estender-se até o ponto mais distante, perdendo sua característica local.

Um modo de evitar a Maldição da Dimensionalidade é reduzir a dimensão de entrada da função a ser estimada; isto é a base para o uso de funções objetivas locais, dependentes de um pequeno número de variáveis, em métodos não supervisionados.

Um fato relacionado, responsável pela Maldição da Dimensionalidade, é o fenômeno do espaço vazio: Espaços de alta dimensionalidade são inerentemente esparsos. Por exemplo, Perpiñán [PER97] retrata sobre a probabilidade de que um ponto distribuído uniformemente em uma unidade esférica de 10 dimensões situar-se a uma distância de 0,9 ou menos a partir do centro é somente 0,35. Este é um difícil problema em estimações de densidades multivariadas, pois regiões de relativas densidades baixas podem conter uma parte considerável da distribuição, considerando que regiões de aparentemente altas densidades podem ser completamente destituídas de observações em amostras de tamanho moderado [PER97]. Por exemplo, para um desvio padrão (p. ex.: no intervalo $[-1,1]$); em uma hiper-esfera $\mathcal{N}(0,1)$, esta contém somente 0,02% da massa e outra hiper-esfera, no mesmo centro, que possua o raio de 3 vezes o desvio padrão contém 70% da massa. Contudo, e

contrariamente a nossa intuição, em distribuições de alta dimensionalidade, os detalhes são muito mais importantes do que em outros espaços de menores dimensionalidades.

Outro problema causado pela Maldição da Dimensionalidade é que, se há correlações lineares nos dados (muito comum em altas dimensões), a média ideal integra o erro quadrático quando estimar a densidade de dados que deverá ser grande sempre se o tamanho da amostra for arbitrariamente grande.

3.2.3. Avaliação dos métodos

Algoritmos para extração de características estão divididos em 2 famílias: *wrappers* que utilizam o próprio algoritmo de aprendizagem para avaliar a utilidade das características obtidas e *filtros* que avaliam as características conforme as heurísticas baseadas nas características gerais dos dados [HAL00]. Para aplicação em grandes bases de dados, os filtros apresentam-se mais eficientes que os *wrappers*, porque são muito mais rápidos. Porém a maioria dos algoritmos existentes que operam como filtros trabalham com problemas de classificação discreta. Hall, em seu trabalho [HAL00] propõe o uso dos algoritmos de *wrapper* (como o CFS – *Correlation on based Feature Selection*) e estimador ReliefF, baseado no método de Naive Bayes, instanciado em aprendizagem, com árvore de decisão e regressão local ponderada, com a desvantagem de reduzir a dimensionalidade em somente 50%.

Sabe-se que para agrupamentos de dados com altas dimensionalidades, algoritmos padrões como EM (*Expectation Maximization*) e K-means sempre caem em armadilhas de mínimos locais [DIN02]. Muitos métodos de inicialização foram propostos para solucionar este problema, mas com resultados limitados. Ding [DIN02], apresenta um método adaptativo para utilização destes agrupadores como uma tentativa de redução de dimensionalidades, de forma dinâmica. A desvantagem deste método é a de agrupar símbolos de classes diferentes, não suportando elementos não separáveis linearmente.

Perpiñán, em seu trabalho [PER97], apresenta uma coletânea de métodos para redução de dimensionalidades, entre eles análises pela PCA (*Principal Component Analysis* – Análise pela Componente Principal), EPP (*Exploratory Projection Pursuit* – Perseguição exploratória da Projeção), PPR (*Projection Pursuit Regression* – Perseguição da Projeção pela Regressão), NN (*Neural Networks* – Redes Neurais), PPDA, PPDE, *Backfitting*, MARS, etc... Neste trabalho, há

menções sobre índices baseados em momentos polinomiais, aproximando as funções de densidade por uma série truncada de polinômios ortogonais (Legendre, Hermite, Fourier, etc...), possibilitando um cálculo relativamente simples dos momentos, e definindo-os como:

- a) Eficientes computacionalmente,
- b) Não necessitam ser recalculados para cada passo do procedimento numérico, uma vez que podem ser derivados de cada direção de projeção a partir de estatísticas provenientes do grupo de dados,
- c) Parte de uma ênfase pesada a partir da normalidade da distribuição dos valores, tornando-se mais sensível a erros (*outliers*); como resultado, seu desempenho enfraquece. Alguns métodos tentam resolver este problema de várias formas, como a utilização de transformações não lineares e utilização de funções sigmóides em redes neurais.

3.3. Extração de características

Com as tecnologias computacionais e de administração de bases de dados avançando rapidamente, nós seres humanos estamos atribuindo às máquinas as tarefas de acumular dados, processar e manipular informações [LIU03]. Por isso, necessita-se elaborar formas compactas para se extrair características que possam identificar a informação a se analisar. Com os novos avanços na área de *data mining*, várias ferramentas para manipular dados e estabelecer padrões estão se tornando acessíveis, impulsionando os experimentos científicos, melhorando a inteligência comercial, expandindo as coordenações empresariais e aperfeiçoando a produtividade e a rentabilidade. Pesquisadores e cientistas já perceberam que para se utilizar corretamente estas ferramentas, devem-se pré-processar as informações antes de utilizá-las para treinamentos, reconhecimentos ou ainda algoritmos visualizadores. Uma chave para esta operação, é encontrar subgrupos da população que tenham similaridade suficiente para merecerem uma análise. Este tipo de tarefa torna-se necessariamente importante e pertinente em aplicações, tanto em *data mining* como para as demais áreas que tentam representar o mundo real. Contudo, a redução de dados é um passo frequentemente necessário, e esta técnica chama-se **Extração de Características**. A extração de características, seleção e construção são aproximações efetivas de reduções de dados entre outros tais como seleção de instâncias e seleção de dados [LIU03]. O objetivo da extração de características é:

- a) Reduzir a quantidade dados,
- b) Centrar a atenção em informações relevantes,
- c) Melhorar a qualidade dos dados, e conseqüentemente o desempenho dos algoritmos que manipularão estes dados.

Para que se atinja estes objetivos, necessita-se considerar a confiança nos algoritmos de extração de características, os quais devem ser robustos, e ainda observar-se às condições de pré-processamento das informações.

Então, define-se a Extração de Características como sendo o processo que extrai um grupo de dados através de um método funcional de mapeamento. Assumindo-se que há n características ou atributos A_1, A_2, \dots, A_n , após a extração de características, têm-se outro grupo de características B_1, B_2, \dots, B_m com $m < n$ e $B_i = F_i(A_1, A_2, \dots, A_n)$ e F_i sendo a função de mapeamento. Para a instância, um mapeamento pode ser $B_1 = c_1A_1 + c_2A_2$, onde c_1 e c_2 são coeficientes. Pesquisas intensivas são normalmente necessárias para se obter boas transformações. O objetivo da extração de características é o de localizar um grupo mínimo de novas características via alguma transformação de acordo com alguma medição de desempenho (investigando o que é mais satisfatório na avaliação de características extraídas) [LIU03].

A extração de característica, então é aplicada a imagens, sendo o processamento em imagens bidimensionais (2D). A diferença com a extração unidimensional (1D) está na quantidade de informações, no fato de haver menos recursos matemáticos completos para aplicações 2D (como equações diferenciais parciais) e ainda na ausência de teoremas fundamentais de álgebra polinomial 2D [JAE88], como por exemplo, a diferença entre processamento de voz e de uma imagem, que gera muito mais pontos de informações para análises.

Para os trabalhos de extração de características, Perpiñán [PER97] apresenta um estudo de utilização de modelos estatísticos com classificadores, inclusive com Redes Neurais Artificiais. Para os modelos de extração de características, escolheu-se as transformadas de Zernike (citadas por Trier [TRI95] com sua característica de invariabilidade à rotação), e Chebyshev (citadas por Liang [LIA00] e Mukundan [MUK01] pelas suas características cartesianas). Também foram escolhidas as Transformadas de Fourier (devido a quantidade de informações disponíveis em pesquisas bibliográficas e pela sua capacidade de análise e detecção quanto à rotação do símbolo), e para a Transformada Discreta do Coseno (utilizada para compressão de imagens) e Radon (pela sua

característica de multi-histogramas), realizou-se um experimento para avaliação quanto ao seu desempenho de extração de característica. Estudou-se a viabilidade de utilização das transformadas de Laplace, Butterworth e Mellin.

Escolheu-se a classificação com Redes Neurais Artificiais com treinamento supervisionado devido a sua capacidade de discriminar elementos não linearmente separáveis, (que exigiria um trabalho muito mais complexo para classificadores não supervisionados), e pela disponibilidade de documentações para utilização, ante a um classificador, como por exemplo HMM, considerando-se no contexto do presente trabalho.

3.3.1. Transformada de Chebyshev

A Transformada de Chebyshev trabalha com entropia plana [BOY00][MUK02]: independe da posição relativa em relação às bordas da imagem. Os momentos de Chebyshev eliminam os problemas associados com os momentos ortogonais de imagem, tais como os momentos de Legendre e Zernike, embora seja derivada dos Polinômios de Legendre. São menos sensíveis a ruídos quando comparados a momentos geométricos, e possuem uma melhor representação de suas características. As informações redundantes são mínimas para um conjunto de momentos ortogonais. A Figura 3.2 mostra uma simplificação da influência da transformada de Chebyshev. Conclui-se que é uma transformação linear, gerando nos conjuntos dos reais, valores equidistantes ao ponto de início da transformação. Esta transformada pode também ser ajustada para se ter um centro, semelhante a Zernike.

A definição da Transformada de Chebyshev é uma composição de integrais, como mostra a Equação 3.1, mas por questões práticas, converte-se os limites das integrais às dimensões da imagem em análise, variando de pixel a pixel, definindo assim os valores para $dx = 1$ e $dy = 1$.

$$\Gamma[p, q] = \frac{1}{\rho_p(N)\rho_q(n)} \int_0^{\infty} \int_0^{\infty} t_p(x) t_q(y) f(x, y) dx dy \quad p, q = 0, 1, 2, 3, 4, \dots, N-1 \quad (3.1)$$

sendo

$$\rho_p(N) = \frac{N}{2p+1} \prod_{i=1}^p \left(1 - \frac{i^2}{N^2}\right)$$

$$\rho_q(N) = \frac{N}{2q+1} \prod_{i=1}^q \left(1 - \frac{i^2}{N^2}\right)$$

e os valores de escala do polinômio são:

$$t_p(x) = 1 \quad \text{para } p = 0$$

$$t_p(x) = \frac{2x+1-N}{N} \quad \text{para } p = 1$$

$$t_p(x) = \frac{(2p-1) \cdot t_1(x) \cdot t_{p-1}(x) - (p-1) \cdot \left\{1 - \frac{(p-1)^2}{N^2}\right\} \cdot t_{p-2}(x)}{p} \quad \text{para } p > 1$$

e finalmente, N é a dimensão máxima da imagem a analisar

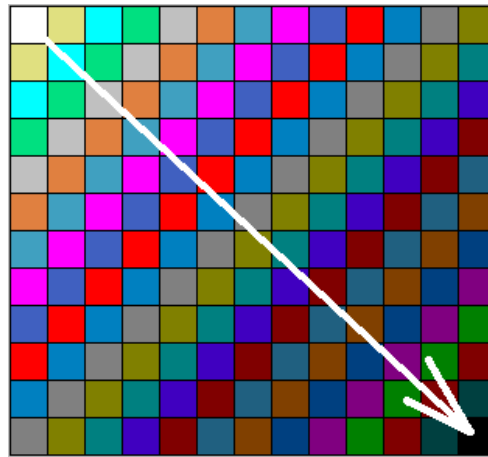


Figura 3.2: Visualização da transformada de Chebyshev

Assim, a equação principal da reescrita pode ser representada pela Equação 3.2, que está na forma de somatória, sem perdas das propriedades essenciais associadas com a ortogonalidade, mantendo o seu *kernel*.

$$\Gamma[p, q] = \frac{1}{\rho_p(N)\rho_q(N)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x)t_q(y)f(x, y) \quad p, q = 0, 1, 2, 3, 4, \dots, N-1 \quad (3.2)$$

Há duas formas de se trabalhar com esta transformada para solucionar o problema de rotação, devido as normatizações de traçado de desenho definidas pela norma ABNT [ABNTa-d]:

1 - Deve-se extrair as características da imagem a cada rotação de 90° , gerando 4 vetores característicos por imagem,

2 – Determina-se o centro geométrico da imagem, extrai-se então 4 transformadas a **partir** do seu centro geométrico. Para cada elemento $T(m,n)$ da transformada resultante passa a ser a média matemática das suas respectivas transformadas parciais $T_1(m,n)$, $T_2(m,n)$, $T_3(m,n)$, $T_4(m,n)$, como mostra a Equação 3.3.

$$T(m,n) = \frac{T_1(m,n) + T_2(m,n) + T_3(m,n) + T_4(m,n)}{4} \quad (3.3)$$

3.3.2. Transformada DCT (*Discrete Cosine Transform*)

Recentemente, a Transformada do Coseno se popularizou devido a ser um método de compressão de imagem [SMI97] (utilizado em arquivos *mpeg*) e pela sua característica de redução de dimensionalidade. Aqui, a característica de redução de dimensionalidade, torna-se relevante diante do fato de que o resultado da transformada será a entrada para o classificador neural. A transformada do coseno é na realidade um momento matemático de entrelaçamento de funções cosenoidais (portanto, periódicas).

A definição teórica da transformada é dada pelo produto interno (ou escalar) de dois vetores: Seja $V=C[-\pi,\pi]$ o espaço vetorial de funções contínuas definidas no intervalo $[-\pi,\pi]$. Existe um produto escalar para este espaço vetorial. Se $f(t)$ e $g(t)$ são duas funções em V , então o seu produto vetorial é definido como uma integral de $-\pi$ a π de $f(t)g(t)$ [LAY02], como mostra a Equação 3.4.

$$C(u,v) = \alpha(u)\alpha(v) \int_0^\infty \int_0^\infty f(x,y)T(u,v,x,y,N) dx dy \quad (3.4)$$

sendo

$$T(u, v, x, y, N) = \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

$$\alpha(u) = \sqrt{\frac{1}{N}} \quad \text{para } u = 0$$

$$\alpha(u) = \sqrt{\frac{2}{N}} \quad \text{para } u = 1, 2, \dots, N-1$$

e finalmente, N é a dimensão máxima da imagem a se analisar.

A sua implementação pode ser executada na forma de somatória, como mostrado na Equação 3.5.

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) T(u, v, x, y, N) \quad (3.5)$$

A Figura 3.3 mostra uma idéia de funcionamento da transformada, aonde as quadrículas representam os pesos definidos pelo *kernel* para os cálculos dos momentos das dimensões horizontais e verticais. O interesse recai no uso desta transformada em processos de reconhecimento de padrões. Esta transformação muda o domínio de representação da imagem e não introduz perdas consideráveis de qualidade, permitindo uma representação mais compacta [SMI97].

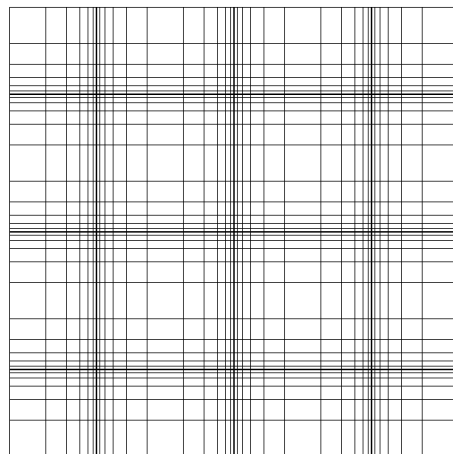


Figura 3.3: Modelo de funcionamento da Transformada do Coseno

3.3.3. Transformada de Fourier

Esta é a transformada mais rica em documentação, e a sua implementação no modelo do domínio contínuo [JAE88] pode ser definida pela Equação 3.6. Este modelo apresenta alguns problemas de demanda computacional, e para extração das características utilizou-se a DFT (*Discrete Fourier Transform* – Transformada Discreta de Fourier).

$$\mathfrak{F}(f(x, y)) = F(m, n) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(xu + yv)} dx dy \quad (3.6)$$

Para a Transformada Discreta de Fourier, utilizou-se a Equação 3.7, com modelo discreto, obtido em [CAS01]. A Transformada bidimensional de Fourier possui uma propriedade interessante a ressaltar: se houver rotação a imagem sob análise, também haverá rotação do seu espectro, ou seja, o módulo do espectro permanece o mesmo, embora a fase se altere. Assim, para evitar problemas de rotação na imagem, deve-se trabalhar somente com o módulo da transformada. Vale a pena lembrar que a componente $F(m, n)$ é complexa, e composta de duas componentes: uma real, e outra complexa.

$$F(m, n) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x, y) e^{-i2\pi\left(m\frac{x}{N} + n\frac{y}{N}\right)} \quad (3.7)$$

Em [SMI97], há uma consideração para a ordem ‘N’ dos elementos, devendo ser uma potência de 2 (p. ex.: 64, 128, 256, etc...). Caso o tamanho da imagem a ser analisada não coincida com as potências de 2, a matriz bidimensional deverá ser completada com zeros.

Uma característica da transformada de Fourier é a sua capacidade de análise por borda, devido às mesmas conterem várias frequências nas suas linhas que separam objetos ou regiões, [SMI97].

3.3.4. Transformada de Radon

A transformada de Radon surgiu motivada pela análise de problemas em mecânica do contínuo [BAR03], entretanto têm sido utilizada para muitos problemas práticos, como por exemplo

a tomografia. Uma característica desta transformada, é a sua possibilidade de implementação para tornar-se mais robusto à escala, desde que o centro da matriz de Radon coincida com o centro da imagem a analisar [KIN03], como mostra a Figura 3.4, e com o passo angular (P) a se executar de forma que consiga abranger todos os pixels da imagem, conforme mostra a Equação 3.8. Pode-se considerar como sendo n histogramas da imagem, executado em sucessivas perspectivas angulares [DIM00], definidos como sendo a ordem ‘X’ de transformação. Este método de tratamento das informações permite uma reconstrução da imagem original a partir da sua transformada, chamado de *Reconstruction by Backpropagation*.

$$\begin{aligned}
 R(b, m) &= \sum_{x=0}^{p-1} f[x, (b + mx)_p] && m = 1, 2, \dots, P-1 \\
 R(b, \updownarrow) &= \sum_{y=0}^{p-1} f(b, y) && \text{vertical} \\
 R(b, \leftrightarrow) &= \sum_{x=0}^{p-1} f(x, b) && \text{horizontal}
 \end{aligned} \tag{3.8}$$

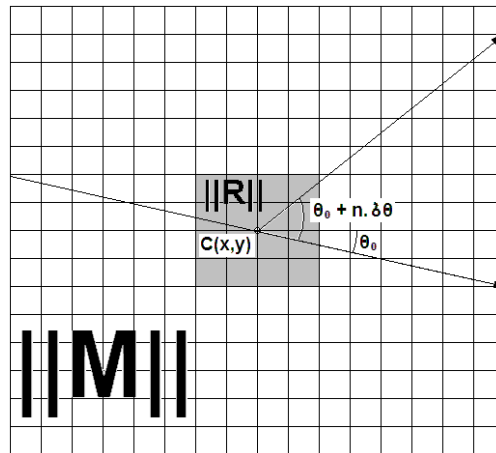


Figura 3.4: Interpretação geométrica de Radon

A transformada é uma matriz de ordem de grandeza X, Y , onde a ordem X define os passos angulares (P) de transformação (p. ex: $30^\circ, 60^\circ, 90^\circ$, etc...) para cada elemento X , e Y define a altura do “feixe de varredura” do histograma.

3.3.5. Transformada de Zernike

A Transformada de Zernike trabalha com momento radial [POU00]: Quanto mais distante do centro, maior a influência. Por ser radial, torna-se imune à rotação do símbolo. É derivada da interferometria óptica de Fresnel [HUS02], e seu funcionamento pode ser descrito como a seguir:

A imagem deve ter seu centro geométrico (x_c, y_c) definido, ou seja, coincidir com o centro geométrico da imagem. Estes valores podem ser definidos como 50% da largura da imagem (x_c) e 50% da altura da imagem (y_c). Esta demonstração pode ser vista pelo círculo representado na Figura 3.5 e pela Equação 3.9 (no domínio contínuo) de implementação da transformada.

Por ter seu centro definido, é traçada uma componente radial para todos os pixels da imagem (inclusive os seus cantos extremos). Para esta componente, é aplicado um peso proporcional à distância do centro (x_c, y_c) . Isto prova que, a transformada de Zernike é imune a rotação, já que o peso aplicado é uma função da distância radial, e não do valor angular. Esta demonstração pode ser vista pela Equação 3.10 de implementação da transformada, utilizando a definição conceitual de cálculo integral [GRAN92]. Para que seja feita a integração radial normalizada (entre 0 e 1), determina-se o raio máximo (r_{max}), que é a distância Euclidiana do centro geométrico da imagem até a coordenada inicial (0,0). Para realizar a integração, dividiram-se as integrais em 100 passos cada (dr e $d\theta$) para garantir a precisão da imagem analisada e simultaneamente para auto-ajuste das escalas, normalizou-se pela divisão de r_{max} , definindo-se os raios geométricos á uma escala entre [0,1].

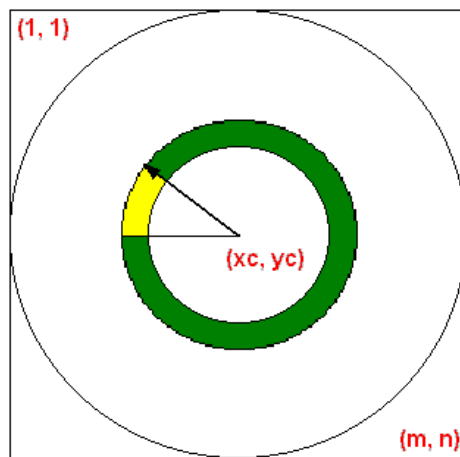


Figura 3.5: Interpretação geométrica de Zernike

$$A_{mn} = A(m, n) = \frac{m+1}{\pi} \int_0^{2\pi} \int_0^1 V^*(n, l, r, \theta) f(r, \theta) r \cdot dr \cdot d\theta \quad (3.9)$$

sendo

$$V^*(n, l, r, \theta) = \text{conj}(V(n, l, r, \theta)) = \text{conj}(R_{mn}(r) \cdot e^{i\theta}) = R_{mn}(r) (\cos(\theta) - i \cdot \sin(\theta))$$

e $R_{mn}(r)$ podendo ser expresso como:

$$R_{mn}(r) = \sum_{s=0}^{\frac{m-|l|}{2}} \left[(-1)^s \cdot \frac{(m-s)!}{s! \cdot \left(\frac{m+|l|}{2} - s\right)! \cdot \left(\frac{m-|l|}{2} - s\right)!} \cdot r^{(n-2s)} \right]$$

para $n - |l|$ ser par, e $|l| \leq n$

caso contrário, $R_{mn}(r) = 0$

satisfazendo as condições $r_{n \pm |l|}(1) = 1$ e $r_{n \pm |l|}(r) = r_{n - |l|}(r)$.

$$A_{mn} = \frac{m+1}{\pi r_{\max}} \sum_{\theta=0}^{2\pi} \sum_{r=0}^{r_{\max}} V^*(n, l, r, \theta) f(r, \theta) \cdot r \cdot dr \cdot d\theta \quad (3.10)$$

sendo $r_{\max} = \sqrt{x_c^2 + y_c^2}$, $dr = r / 100$, $d\theta = \theta / 100$,

$$r = \sqrt{x^2 + y^2} \quad \text{e} \quad \theta = \tan^{-1} \left(\frac{x_c - x}{y_c - y} \right), \text{ com } 0 \leq \theta < 2\pi$$

3.4. Etapas de processamento

As etapas do processamento podem ser divididas em duas fases:

Fase 1: Com o processamento isolado das transformadas, ou seja, realizaram-se os testes de reconhecimento utilizando somente uma transformada por teste de reconhecimento.

Fase 2: Com a concatenação das transformadas, ou seja, realizaram-se os testes utilizando as transformadas combinadas (associadas).

Testou-se inicialmente com as transformadas isoladas, para análise de convergência e capacidade discriminatória dos símbolos em teste. Este processo pode ser visualizado na Figura 3.6, que mostra as etapas de processamento dos símbolos. Para cada transformada, realizou-se um teste de reconhecimento (por validação cruzada), consistindo de treinamento, validação e testes.

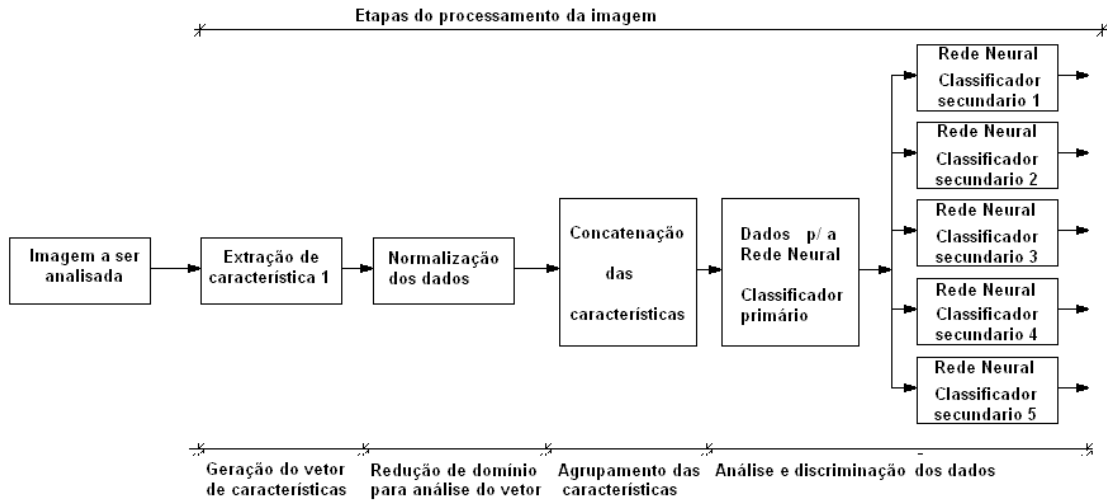


Figura 3.6: Etapas utilizadas no reconhecimento dos símbolos com uma transformada

Após realizados os experimentos com as transformadas isoladas, avaliou-se o desempenho da metodologia com as transformadas associadas (concatenando os seus resultados), para análise de convergência e capacidade discriminatória dos símbolos em teste. Este processo pode ser visualizado na Figura 3.7, que mostra as etapas de processamento dos símbolos. As ordens de grandeza das transformadas foram reduzidas para se obter o mesmo tamanho final do vetor característico. Os experimentos de reconhecimento foram realizados por validação cruzada, consistindo de conjuntos distintos de treinamento, validação e testes.

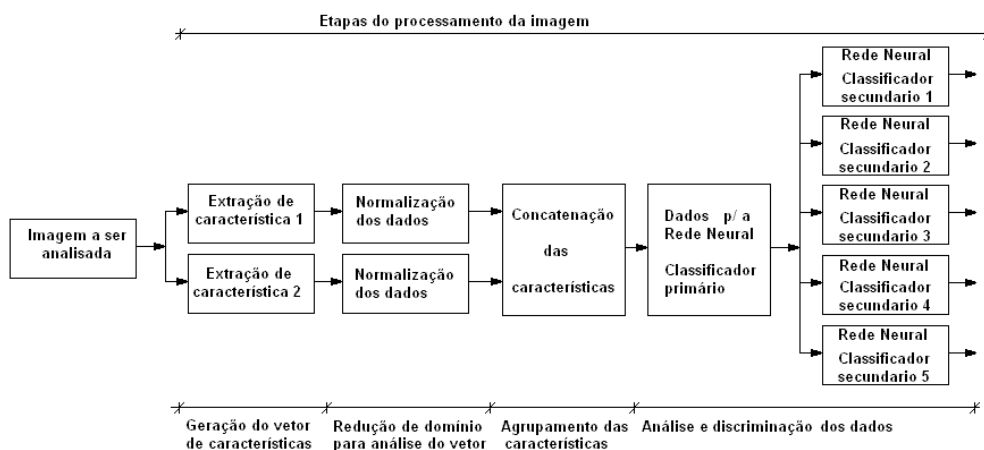


Figura 3.7: Etapas utilizadas no reconhecimento dos símbolos com duas transformadas

Escolheu-se a topologia de rede MLP (*Multi Layer Perceptron*) com BP (*Backpropagation*) para o classificador principal da rede neural. Esta opção foi adotada por ser a de concepção mais simples e com capacidade de resolver os problemas comuns do mundo real [FRE92].

Para os classificadores secundários, utilizou-se tanto a rede Hebbiana quanto a rede MLP como memória e com a função de discriminar o símbolo pela maior semelhança (através da menor distância Euclidiana, entre suas características).

As transformadas devolvem matrizes bidimensionais (de ordem $M \times N$), e podem ser consideradas como vetores unidimensionais (de ordem $M \cdot N$), e a concatenação das transformadas segue um padrão de alinhamento de vetores. Estes vetores são, depois de serem normalizados, aplicados à entrada de uma rede neural, conforme a Equação 3.11.

$$\|A\| \cdot \|B\| \Rightarrow \|A\| \text{ concatenada com } \|B\| \Rightarrow \|AB\| \quad (3.11)$$

3.5. Fluxo dos dados

Os dados, após extraídas as suas características, (e concatenados quando possuírem as transformadas associadas), formam um vetor característico, que são aplicados a uma rede neural hierárquica. Através da Figura 3.8, observa-se que o modelo possui 2 níveis de rede, sendo o nível superior um classificador primário, cuja função é a de identificar a que grupo o símbolo pertence, e no nível inferior, as redes secundárias que identificam o símbolo dentro da sua metaclassa, realizando uma classificação mais refinada.

Como os símbolos foram agrupados em função de suas semelhanças gráficas, necessitou-se adaptar o número de saídas para cada agrupamento. Conseqüentemente, as saídas para as Redes Neurais Artificiais dos classificadores secundários foram divididas da seguinte forma:

- O Primeiro classificador identifica os símbolos do tipo autotransformadores e indutores, com e sem controles de regulação de tensão,
- O Segundo classificador identifica uma parcela dos componentes com estruturas gráficas semelhantes, incluindo alguns tipos de disjuntores,

- O Terceiro classificador identifica os dispositivos de função (equipamentos que possuem funções) e algumas representações de disjuntores (cuja representação gráfica é semelhante á alguns dispositivos – mais informações, ver Apêndice A – Base de dados utilizada),
- O Quarto classificador identifica os transformadores com 2 enrolamentos, com e sem funções de ajuste de potencial e os transformadores de corrente (utilizados para medição de corrente elétrica).
- O Quinto classificador identifica os transformadores com 3 enrolamentos, com e sem funções de ajuste de potencial. Esta classe permaneceu isolada devido á sua característica gráfica peculiar.

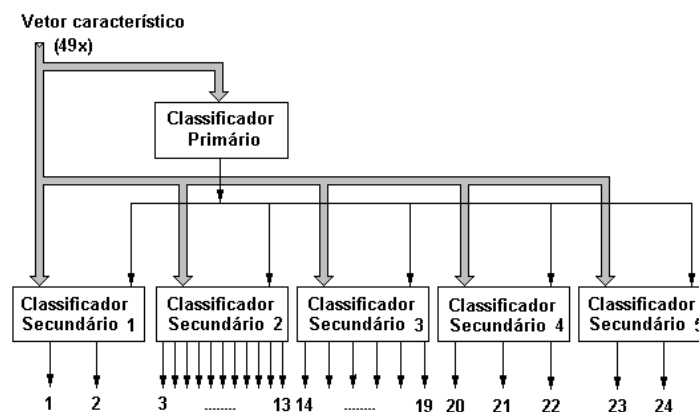


Figura 3.8: Fluxo dos dados na estrutura classificatória

Como etapa seguinte, os dados foram agrupados em função da sua semelhança gráfica, em termos de primitivas gráficas utilizadas em projetos CAD. Assim deixa-se para o classificador primário a função de identificar a similaridade gráfica (de mais simples obtenção devido aos processos estatísticos) e para o classificador secundário a tarefa mais especializada de identificação (classificação fina), como mostra a Figura 3.9 (mais detalhes, ver Apêndice A – Base de Dados).

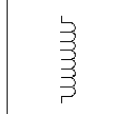
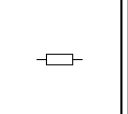
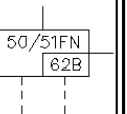
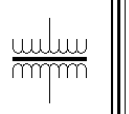
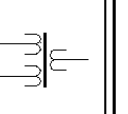
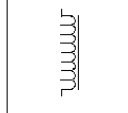
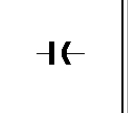
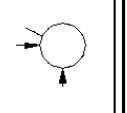
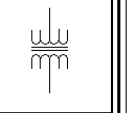
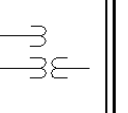
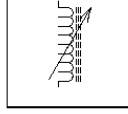
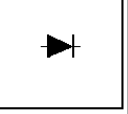
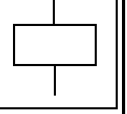
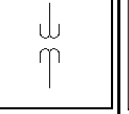
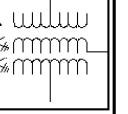
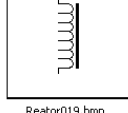
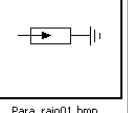
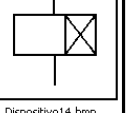
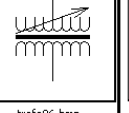
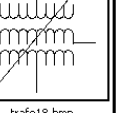
G - I	G - II	G - III	G - IV	G - V
 Reator001.bmp	 Resistor01.bmp	 Dispositivo01.bmp	 trafo03.bmp	 trafo06.bmp
 Reator002.bmp	 Capacitor13.bmp	 Dispositivo08.bmp	 trafo22.bmp	 trafo07.bmp
 Reator013.bmp	 Diodo01.bmp	 Dispositivo13.bmp	 trafo10.bmp	 trafo12.bmp
 Reator019.bmp	 Para_raio01.bmp	 Dispositivo14.bmp	 trafo06.bmp	 trafo18.bmp

Figura 3.9: Exemplo do agrupamento dos dados

3.6. Redes *Multi-Layer Perceptron* com *Backpropagation* (MLP-BP)

O treinamento da rede por retro-propagação (backpropagation) envolve três estágios [FAU94]: O avanço das informações do padrão de treinamento, a retro-propagação do erro associado e o ajuste dos pesos.

Durante a fase de avanço das informações, cada unidade N_1 , N_2 e N_3 (Figura 2.3) recebe como entrada um sinal e difunde-o a cada unidade escondida (N_5 , N_6 e N_7). Cada unidade escondida então calcula a sua ativação e envia seus sinais para cada unidade de saída (N_8 , N_9 e N_{10}), que calculam as suas saídas pelas respectivas funções de ativação para formar a resposta da rede para um dado padrão de entrada.

Durante o treinamento, cada saída da rede é comparada com o resultado a ser treinado (usado como referência) para determinar-se o erro associado ao padrão de entrada. Baseado neste erro, calcula-se o fator δ_k ($k=1,2,\dots,n$). O fator δ_k é usado para retroagir o erro da sua unidade de saída correspondente para os demais neurônios (que estão com as suas saídas interligadas a estes neurônios). Este fator δ_k também é utilizado para uma posterior atualização dos pesos entre as

saídas e a camada escondida. De maneira similar, o fator δ_j ($j=1,2,\dots,p$) é calculado para cada neurônio escondido (N_5, N_6 e N_7). Não é necessário propagar o erro até a camada de entrada formada pelos neurônios N_1, N_2 e N_3 , mas δ_j é usado para atualizar os pesos entre a camada escondida e a camada de entrada.

Após todos os fatores δ serem determinados, atualizam-se todos os pesos de todas as camadas simultaneamente. As atualizações dos pesos $w_{i,j}$ (a partir da camada escondida até as saídas) são baseados no fator δ_k e nas funções de ativação respectivas a cada neurônio da camada escondida para os neurônios de saída, e para os neurônios da camada escondida utiliza-se δ_j e os valores das funções de ativação dos neurônios da camada de entrada.

O algoritmo de treinamento *backpropagation* previamente descrito pode ser visualizado na Listagem 3.1 [FAU94].

Listagem 3.1 – Algoritmo de treinamento para a rede MLP-BP

Início

Passo 0: Inicialize os pesos (com pequenos valores radômicos)

Passo 1: Enquanto a condição de parada não for válida, faça os passos 2-9

 Passo 2: Para cada par em treinamento, faça os passos 3-8

 Avanço das informações:

 Passo 3: Para cada entrada E (Figura 2.7), recebe-se um valor e propaga-o para todas os neurônios da camada escondida

 Passo 4: Para cada neurônio da camada escondida, somam-se suas entradas (as saídas dos neurônios antecessores multiplicados pelos seus pesos).

$$N = bias + \sum_{i=1}^{N \text{ Entradas}} (SaidaNeurônio_i \times PesoNeurônio_i)$$

 Calcula-se a função de ativação do respectivo neurônio

$$N_i = função_ativação(N), \text{ e envia-o para as camadas subseqüentes}$$

 Passo 5: Para cada unidade de saída, somam-se os valores de entrada com os seus respectivos pesos

$$N = bias + \sum_{i=1}^{N \text{ Entradas}} (SaidaNeurônio_i \times PesoNeurônio_i)$$

 Calcula-se a função de ativação do respectivo neurônio

$$N_i = função_ativação(N), \text{ e envia-o para as camadas subseqüentes}$$

 Retro-propagação do erro

 Passo 6: Cada neurônio de saída recebe um padrão correspondente ao padrão de treinamento de entrada, calculando assim o erro:

$$\delta_k = (\text{ValorSaída} - \text{função_ativação}) \frac{\partial(\text{função_ativação})}{\partial t}$$

calcula-se então a sua correção do peso, $\Delta w_{\text{EntreNeurônios}} = \alpha \delta_k N_i$,

e a correção do peso do *bias* $\Delta w_{\text{BiasNeurônio}} = \alpha \delta_k$,

onde α é o passo de aprendizagem

Passo 7: Para cada neurônio da camada escondida, somam-se as δ_k , provenientes dos neurônios sucessores, multiplicando-se pela derivada da função de ativação do neurônio

$$\delta_i = \left(\sum_{k=1}^m (\delta_k w_{\text{EntreNeurônios}}) \right) \times \frac{\partial(\text{função_ativação})}{\partial t}$$

calcula-se assim a correção do peso $\Delta w_{\text{EntreNeurônios}} = \alpha \delta_j N_i$,

e a correção do peso do *bias* $\Delta w_{\text{BiasNeurônio}} = \alpha \delta_j$

Atualização dos pesos e *bias*

Passo 8: Para cada neurônio, a partir da saída, atualizam-se os pesos

$$w_{\text{EntreNeurônios}} = \Delta w_{\text{EntreNeurônios}} + w_{\text{EntreNeurônios}} \text{ e também do } \textit{bias}$$

$$w_{\text{BiasNeurônio}} = \Delta w_{\text{BiasNeurônio}} + w_{\text{BiasNeurônio}}$$

Fim do Passo 2

Passo 9: Teste condição de parada

Fim do Passo 1 (Enquanto)

Fim

3..6.1. Rede Neural: Classificador primário

A topologia escolhida para a Rede Neural foi a MLP (*Multi Layer Perceptron*) com algoritmo de aprendizagem *Backpropagation*, nas redes com algoritmos de extração de características utilizando as transformadas associadas, DCT, Fourier, Radon e Zernike e com o algoritmo de aprendizagem *Backpropagation with Momentum* para as redes com algoritmo de extração de características utilizando a transformada de Chebyshev. A escolha da rede MLP/BP deve-se a sua capacidade de resolver a maioria dos problemas do mundo real [FRE92], pela sua popularidade e capacidade de tratar eventuais problemas de não linearidade que possam a ocorrer.

Para os testes, utilizou-se a versão do simulador JavaNNS [ZEL94] para montagem da rede, treinamento e testes de reconhecimento.

Utilizou-se para a montagem da rede, treinamento e os testes de reconhecimento a topologia MLP/BP (*Multi-Layer Perceptron with BackPropagation*), sem ligações competitivas. Para a camada de entrada, definiram-se 49 células (50 para os testes isolados com a transformada de Zernike), para a camada escondida, (*hidden*), 27 células definidas para o JavaNNS com a função de

ativação “*Act_Perceptron*”, e função de saída “*Threshold*”. E, finalmente para a camada de saída, com 5 células definidas com a função de ativação como “*Act_Perceptron*”, e função de saída “*Threshold*”. Escolheu-se 27 células na camada intermediária devido a sua melhor relação de quantidade mínima de neurônios x eficiência de reconhecimento.

3.7. Redes Hebbianas

No final da década de 1940, o neurofisiologista Donald Hebb (1949) enunciou um princípio que tornou-se muito influente em neurocomputação. Pelos estudos da comunicação entre neurônios, Hebb verificou que se um neurônio repetidamente excita outro neurônio, o limiar de excitação entre os neurônios decresce. Isto significa que repetidas excitações diminuem os limiares, ou equivalentemente que o efeito das excitações do primeiro neurônio é amplificado.

Estendendo-se facilmente isto para sistemas neurais artificiais, o princípio de Hebb deverá aumentar o peso comum w_{ij} , quando houver atividade entre os neurônios j -ésimo e i -ésimo. Deve-se considerar também que a aprendizagem Hebbiana é intrinsecamente instável [PRI99], produzindo pesos com valores positivos ou negativos muito grandes. Na biologia, não há este problema porque há não linearidades naturais que limitam a eficácia das sinapses (deplexão química, faixa dinâmica, etc...). Computacionalmente, este problema pode ser resolvido com a Regra de Oja [PRI99], por exemplo.

Há vários modos de se implementar a regra de treinamento de Hebb. O algoritmo de treinamento *Hebbiano* descrito aqui pode ser visualizado na Listagem 3.2 [FAU94], utilizando-se da técnica de associação de padrões (onde a saída é um vetor).

Listagem 3.2 – Algoritmo de treinamento para Redes Hebbianas

Início: Inicialize todos os pesos para os n neurônios: $w_i = 0$, para $i = 1 \dots n$

Passo 1: Para cada vetor de treinamento, com o seus pares de entrada (E) e saída (S), faça os passos 2-5:

 Passo 2: Ajuste as ativações (x_i) para as unidades de entrada (e_i):
 $x_i = e_i$

 Passo 3: Ajuste as ativações (y) para as unidades de saída (s) de cada neurônio:
 $y = s$

 Passo 4: Ajuste os pesos de todos os neurônios para
 $w_i(\text{novo}) = w_i(\text{velho}) + \eta \cdot x_i \cdot y$,
 onde η é o passo de aprendizagem.

Passo 5: Normalize os pesos de todos os neurônios para

$$w_i(novo) \leftarrow \frac{w_i(novo)}{\sum_{j=0}^L w_j(novo)}, \text{ onde } w_i \text{ é o peso do neurônio que possui } L \text{ ligações.}$$

Fim do Passo 1

Fim

Para um classificador secundário, utilizou-se a Rede Neural Hebbiana, utilizada como memória com a função de discriminar o símbolo pela maior semelhança (por exemplo, através da menor distância Euclidiana, entre suas características) [PRI99].

3.8. Preparação da imagem

Necessitou-se preparar uma nova imagem \mathbf{P}^* de ordem x,y antes de aplicar a transformada, para evitar que houvessem manipulações com elevados valores numéricos. A preparação seguiu um padrão de definição do pixel em binário (0 e 1), dividindo o pixel da imagem pelo seu valor numérico máximo (para imagens com níveis de cinza, dividiu-se por 255, o correspondente a um pixel de cor branca). Após realizada esta divisão, subtrai-se este valor de 1 (um), obtendo uma inversão das cores da imagem normalizada, como mostrado na Equação 3.12.

$$P_{x,y}^* = p^*(x,y) = 1 - \frac{p(x,y)}{255} \quad \text{para } \forall x, y \in \|\mathbf{P}\| \quad (3.12)$$

3.9. Normalização dos dados

Para todas as transformadas, necessitou-se normalizar os dados, diminuindo as suas faixas de domínios (variâncias) para evitar erro numérico ou expandindo-as, conforme o caso para que o algoritmo de treinamento dos classificadores não opere com valores muito próximos. Para a variância, considera-se como sendo a variação residual após contabilizar os fatores aleatórios [MCC01]. Se os dados estão normalmente distribuídos, deve-se definir o erro residual e atribuí-lo a uma distribuição normal.

Para esta etapa, a metodologia deve detectar a melhor faixa de valores nas dimensões dos vetores obtidos a partir da base de dados (vetores característicos). A verificação consiste em se

obter valores de variâncias das dimensões em toda a base de dados antes e depois de normalizados, com o objetivo de justificar a técnica de normalização. A Equação 3.13 mostra como pode ser obtida a variância para cada dimensão no espaço vetorial formado pelas transformadas. Os dados podem ser agrupados como N vetores, formando uma matriz de ordem $M \times N$, permitindo assim que suas variâncias (δ) sejam calculadas.

$$\delta_{m,n} = \sqrt{\frac{\sum_{n=0}^N (x_{m,n} - \bar{x}_m)^2}{N-1}} \quad \text{para todos } m \in M, | 0 \leq m \leq M \quad (3.13)$$

sendo: N é o número de elementos da base de dados,
 M é a dimensão do conjunto de vetores característicos,
 m é a coluna dos vetores da base de dados,
 n é o número do vetor sob cálculo e,

x_m é o valor médio da coluna m , obtido por $\bar{x}_m = \frac{\sum_{n=0}^N x_{m,n}}{N}$

3.9.1. Normalização para Chebyshev

Para a transformada de Chebyshev, a normalização torna-se obrigatória, pois a resultante da transformada devolve valores numericamente grandes, como mostra a Figura 3.10, mostrando as variâncias encontradas na base de dados por dimensão.

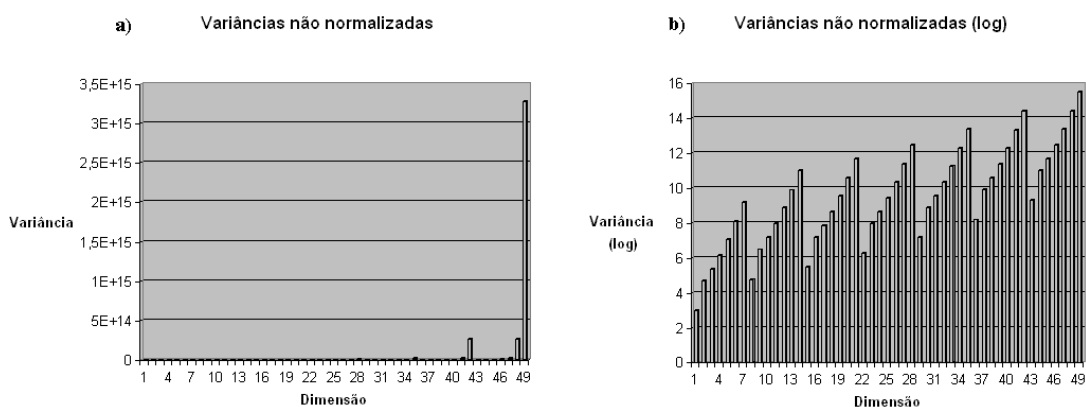


Figura 3.10: Variâncias não normalizadas obtidas pela Transformada de Chebyshev: a) Escala normal, b) Escala logarítmica, para melhor visualização

A técnica de normalização da Equação 3.14, foi utilizada devido a transformada devolver valores negativos, impossibilitando a utilização de equações logarítmicas. Esta implementação resolve o problema da manipulação de valores negativos e reduz os valores numéricos a se manipular, visto que a transformada devolve valores com potências de 10 a cada índice na matriz de Chebyshev.

$$a[i, j] \leftarrow \frac{a[i, j]}{10^{i*j}} \quad (3.14)$$

onde $a[i, j]$ é o elemento na posição i, j na matriz \mathbf{A} de Chebyshev

Obtém-se assim a nova faixa de variâncias para o conjunto de dados, que é mostrada na Figura 3.11. Esta normalização atenua o problema de escalas, preservando pela PCA (*Principal Component Analysis*) a identificação básica do símbolo. Durante os treinamentos das redes, necessitou-se dividir cada símbolo em 4 quadrantes, todos centrados no centro geométrico da imagem, somando-se as suas respectivas coordenadas $A[i, j]$ e aplicando-os à Rede Neural para que fosse possível contornar os problemas de rotação.

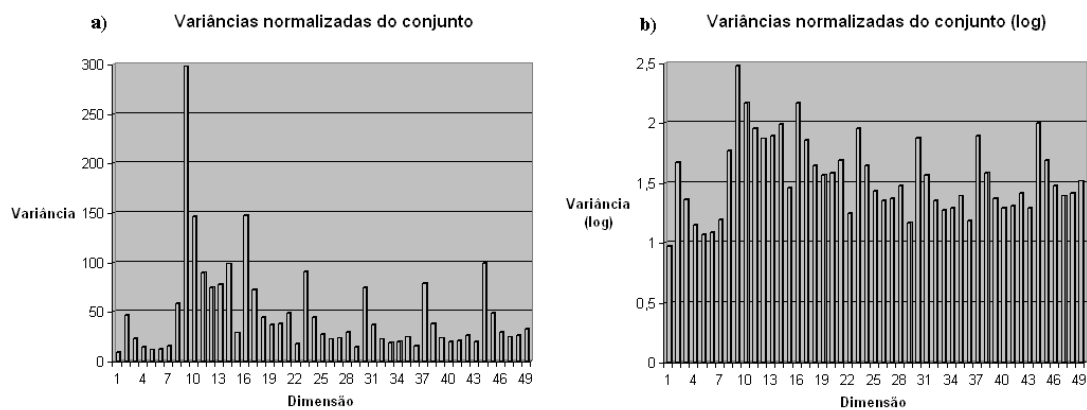


Figura 3.11: Variâncias normalizadas obtidas pela Transformada de Chebyshev a) Escala normal, b) Escala logarítmica, para melhor visualização

3.9.2. Normalização para DCT

Para a transformada de DCT (Transformada Discreta do Coseno), observa-se a presença de uma faixa ampla na variância para a primeira dimensão (como mostra a Figura 3.12) e pequenas gamas de variâncias para as demais dimensões. Por isso, necessita-se normalizá-la, para corrigir estas faixas de domínio.

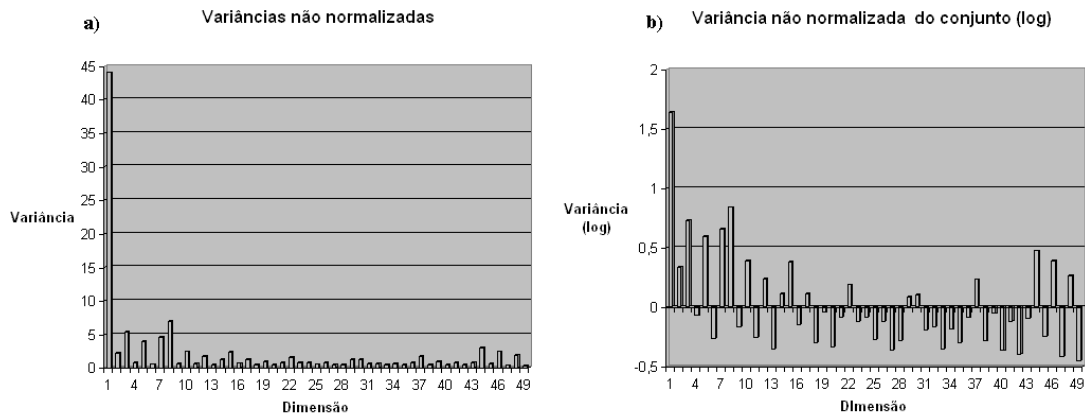


Figura 3.12: Variâncias não normalizadas obtidas pela Transformada do Coseno: a) Escala normal, b) Escala logarítmica, para melhor visualização

A normalização para a Transformada do Coseno, foi dividida em duas partes, como mostra a Equação 3.15, devido a componente $a[0,0]$ da resultante da transformada ser um módulo. Esta transformação não resolve problemas de rotação, somente escalas, por isso a rede necessita ser treinada 4 vezes com o símbolo rotacionado de 90° . A Figura 3.13 apresenta a nova faixa de variâncias obtidas a partir desta técnica de normalização.

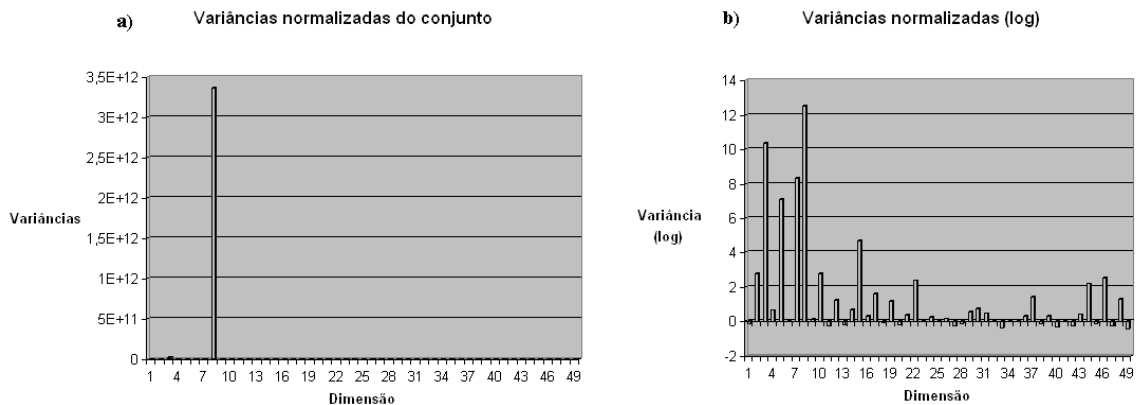


Figura 3.13: Variâncias normalizadas obtidas pela Transformada do Coseno: a) Escala normal, b) Escala logarítmica, para melhor visualização

$$\begin{aligned} a[0,0] &\Leftarrow \ln(a[0,0]) && \text{para } i = 0 \text{ e } j = 0 \\ a[i,j] &\Leftarrow e^{a[i,j]} && \text{para } i \neq 0 \text{ e } j \neq 0 \end{aligned} \quad (3.15)$$

sendo $a[i,j]$ o elemento da posição i,j na matriz **A** do Coseno

3.9.3. Normalização para Fourier

Para a transformada de Fourier, as faixas de variâncias obtidas podem ser vistas na Figura 3.14, observando-se que a transformada é imune a rotação se e somente se trabalhar com o módulo das componentes complexas obtidas, esquecendo-se da componente fasorial para este caso em particular (aonde a rotação não interessa à aplicação).

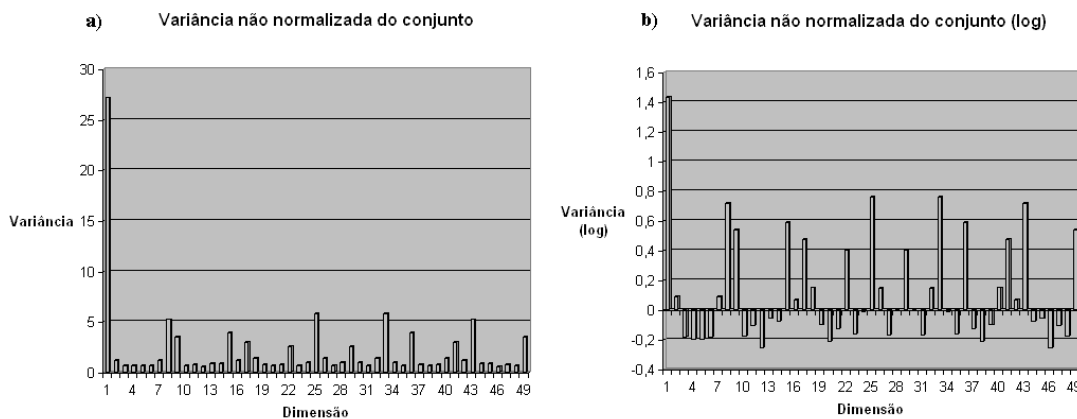


Figura 3.14: Variâncias não normalizadas obtidas pela Transformada de Fourier: a) Escala normal, b) Escala logarítmica, para melhor visualização

A normalização faz-se necessária somente por causa dos problemas de escala, conforme mostra a Equação 3.16, conseguindo-se a nova faixa de variâncias como mostra a Figura 3.15. A matriz de Fourier, é então dissociada em seu módulo e fase. A partir do seu módulo, calcula-se o logaritmo neperiano para determinar o novo vetor característico de dados (matriz, no caso).

$$b[i,j] = \ln(\text{abs}(a[i,j])) \quad (3.16)$$

sendo $a[i,j]$ é o elemento complexo da posição i,j na Matriz **A** de Fourier, e $b[i,j]$ é o elemento real da matriz **B**, resultante da normalização de **A**.

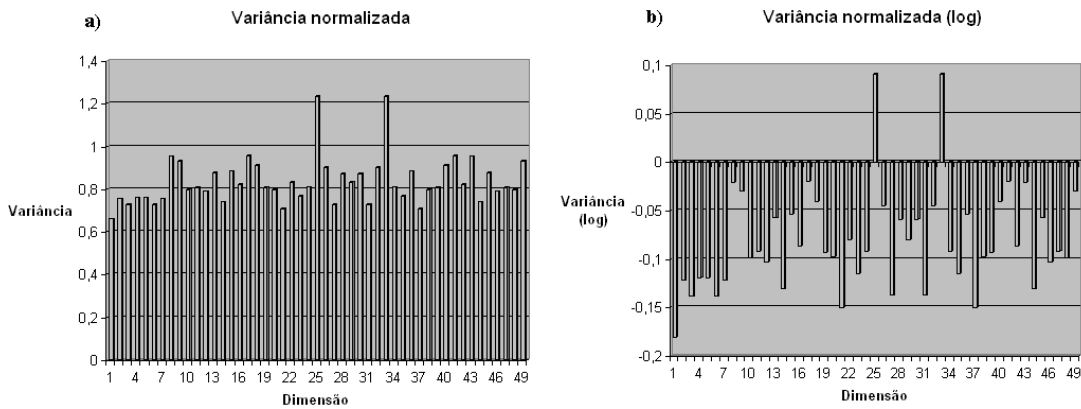


Figura 3.15: Variâncias normalizadas obtidas pela Transformada de Fourier: a) Escala normal, b) Escala logarítmica, para melhor visualização

3.9.4. Normalização para Radon

Para a transformada de Radon, obteve-se a faixa de variâncias como mostra a Figura 3.16, mas a transformada é susceptível a variações de escala e rotação. A matriz de Radon foi composta de uma concatenação de vários histogramas formando uma matriz de N elementos, e posteriormente convertida em um vetor unidimensional aplicável à Rede Neural Artificial.

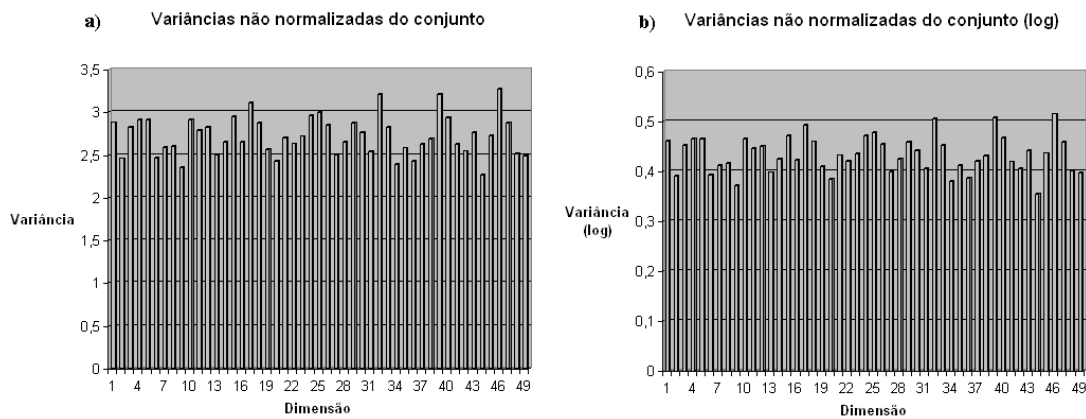


Figura 3.16: Variâncias não normalizadas obtidas pela Transformada de Radon: a) Escala normal, b) Escala logarítmica, para melhor visualização

Para atenuar os problemas de escala, faz-se a normalização de acordo com a Equação 3.17, que é na realidade uma conversão da quantidade de pixels de cada histograma em uma proporção de

pixels do histograma pela quantidade de pixels da imagem, gerando o gráfico de variâncias como mostrado na Figura 3.17. Nota-se que no gráfico logarítmico (Figura 3.17-b) ocorre um nivelamento das variâncias (onde valor igual a zero no eixo das abscissas representam uma variância de 1), demonstrando que piorou a qualidade dos dados a serem analisados, em prol da capacidade de resolução de problemas de escala. Para o treinamento da Rede Neural Artificial, há a necessidade de se rotacionar nas 4 direções a imagem (0° , 90° , 180° , 270°) durante o treinamento para capacitar a rede a reconhecer o símbolo rotacionado.

$$a[i, j] \Leftarrow \log \left(\frac{a[i, j] + 10^{-10}}{N} \right) \quad (3.17)$$

sendo $a[i, j]$ é o elemento da posição i, j na Matriz \mathbf{A} de Radon, e

N é a quantidade de pixels da imagem sob análise.

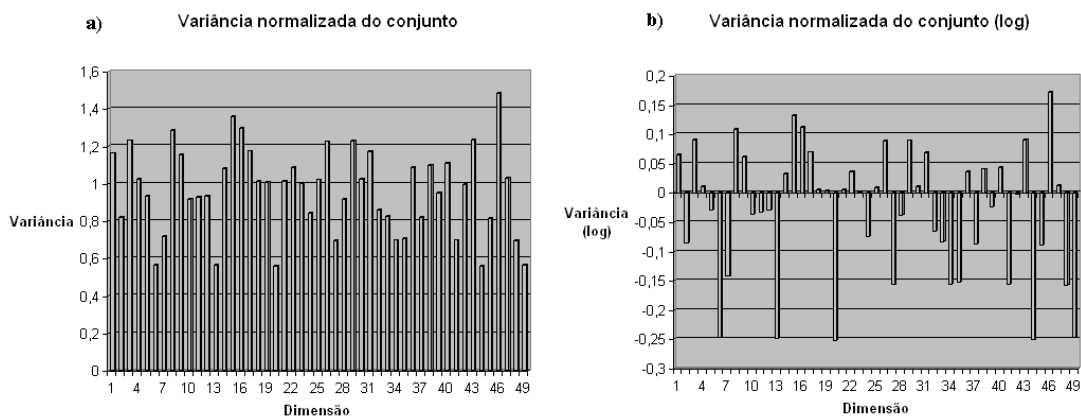


Figura 3.17: Variâncias normalizadas obtidas pela Transformada de Radon: a) Escala normal, b) Escala logarítmica, para melhor visualização

3.9.5. Normalização para Zernike

Para a transformada de Zernike, o conjunto de variâncias obtido é apresentado na Figura 3.18. Observa-se que a composição da matriz de saída preserva os seus valores complexos, não somente o módulo, mas a fase também. A fase obtida pela Transformada de Zernike não representa a rotação da imagem, mas a composição do momento sobre os pixels existente no círculo da imagem em análise. A composição da matriz resultante então é feita como mostra a Equação 3.18.

$$\begin{aligned}
 b[2*i, j] &\Leftarrow \text{Componente Real de } a[i, j] \\
 b[2*i+1, j] &\Leftarrow \text{Componente Imaginária de } a[i, j]
 \end{aligned}
 \tag{3.18}$$

sendo $a[i, j]$ é o elemento complexo da posição i, j na Matriz **A** de Zernike, e $b[2i, j]$ e $b[2i+1, j]$ são os elementos da matriz **B** resultante do alinhamento de **A**.

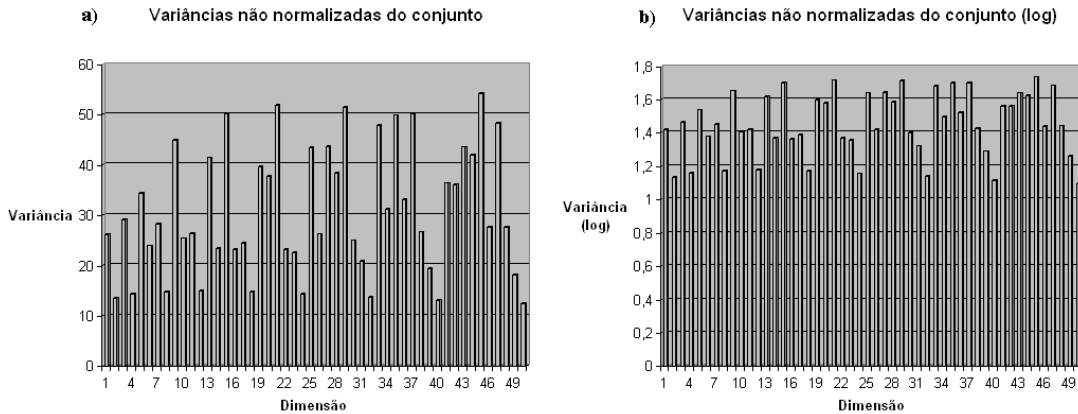


Figura 3.18: Variâncias não normalizadas obtidas pela Transformada de Zernike: a) Escala normal, b) Escala logarítmica, para melhor visualização

A normalização dos dados para a Transformada de Zernike é mostrada na Equação 3.19, lembrando que a operação é feita no domínio dos números complexos e convertida para o domínio dos números reais. Esta normalização diminui os problemas causados pela escala da imagem em análise, preservando a PCA (Principal Component Analysis). A montagem da matriz final normalizada de Zernike, Figura 3.19, segue a mesma regra da não normalizada.

$$\begin{aligned}
 a[i, j] &\Leftarrow \ln(a[i, j]) \\
 b[2*i, j] &\Leftarrow \text{Componente Real de } a[i, j] \\
 b[2*i+1, j] &\Leftarrow \text{Componente Imaginária de } a[i, j]
 \end{aligned}
 \tag{3.19}$$

sendo $a[i, j]$ é o elemento complexo da posição i, j na Matriz de Zernike, e $b[2i, j]$ e $b[2i+1, j]$ são os elementos da matriz **B** resultante do alinhamento de **A**.

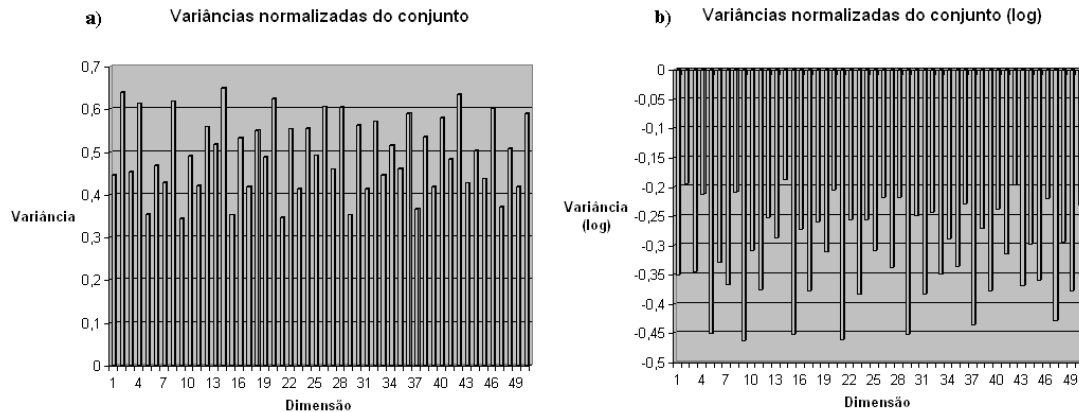


Figura 3.19: Variâncias normalizadas obtidas pela Transformada de Zernike: a) Escala normal, b) Escala logarítmica, para melhor visualização

3.9.6. Composição das normalizações para Zernike e Chebyshev

Para as transformadas de Zernike e Chebyshev, operando em paralelo, o conjunto de variâncias obtido é apresentado na Figura 3.20. Observa-se que a composição da matriz de saída apresenta duas faixas nítidas de valores. Esta composição foi escolhida para testes com uma transformada complexa associada com uma real [TON03a, TON03b], para análise comparativa de desempenho.

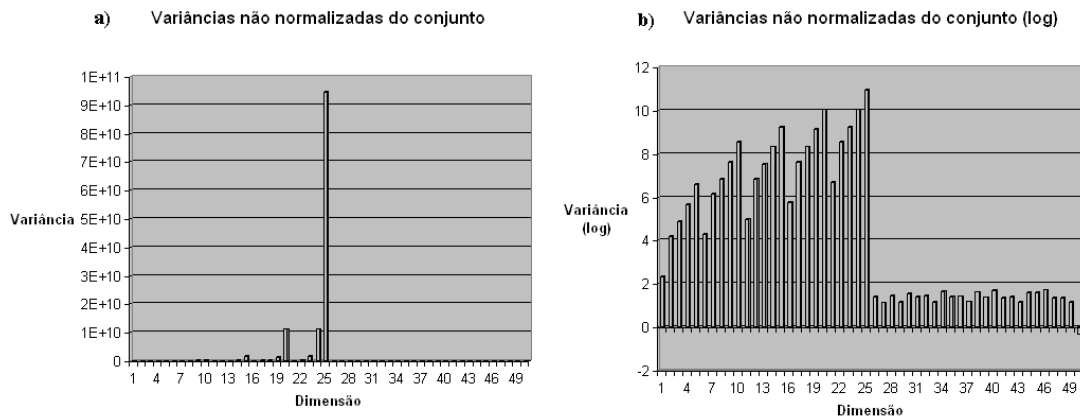


Figura 3.20: Variâncias não normalizadas obtidas pelas Transformadas de Zernike e Chebyshev: a) Escala normal, b) Escala logarítmica, para melhor visualização

As normalizações seguiram o mesmo padrão das transformadas individuais, preservando-se a técnica de normalização utilizada para cada transformada. As variâncias da nova base de dados podem ser observadas pela Figura 3.21.

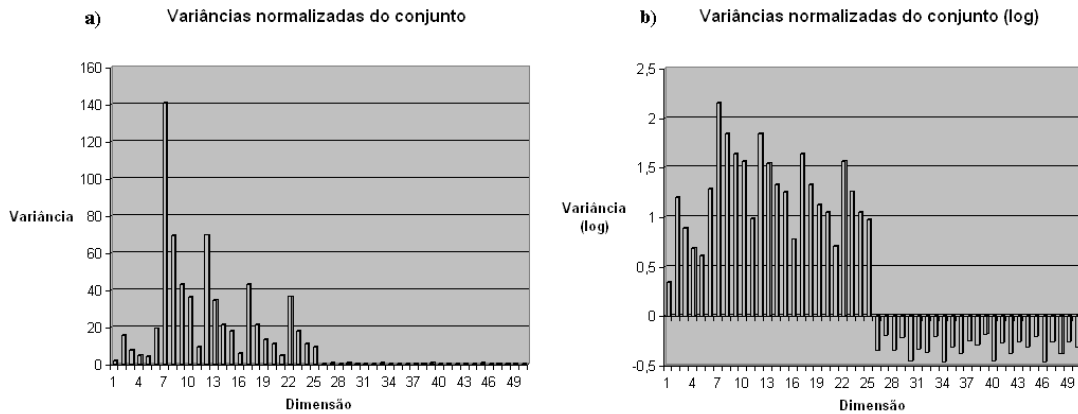


Figura 3.21: Variâncias normalizadas obtidas pelas Transformadas de Zernike e Chebyshev
a) Escala normal, b) Escala logarítmica, para melhor visualização

3.9.7. Composição das normalizações para Zernike e Fourier

Para as transformadas de Zernike e Fourier, operando em paralelo, o conjunto de variâncias obtido é apresentado na Figura 3.22. Observa-se que a composição da matriz de saída apresenta duas faixas nítidas de valores. Esta composição foi escolhida porque durante os testes de treinamento ambas houve a convergência do SSE (Erro Quadrático) a zero, como demonstrado nos resultados citados no Capítulo 4.

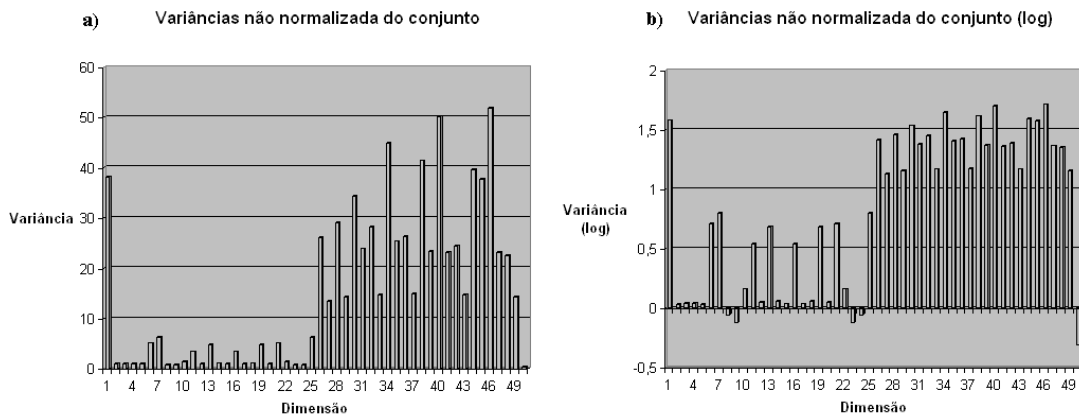


Figura 3.22: Variâncias não normalizadas obtidas pelas Transformadas de Zernike e Fourier:
a) Escala normal, b) Escala logarítmica, para melhor visualização

As normalizações seguem o mesmo padrão das transformadas individuais, cuidando-se para evitar misturas de procedimentos com os dados. As variâncias da nova base de dados podem ser observadas pela Figura 3.23.

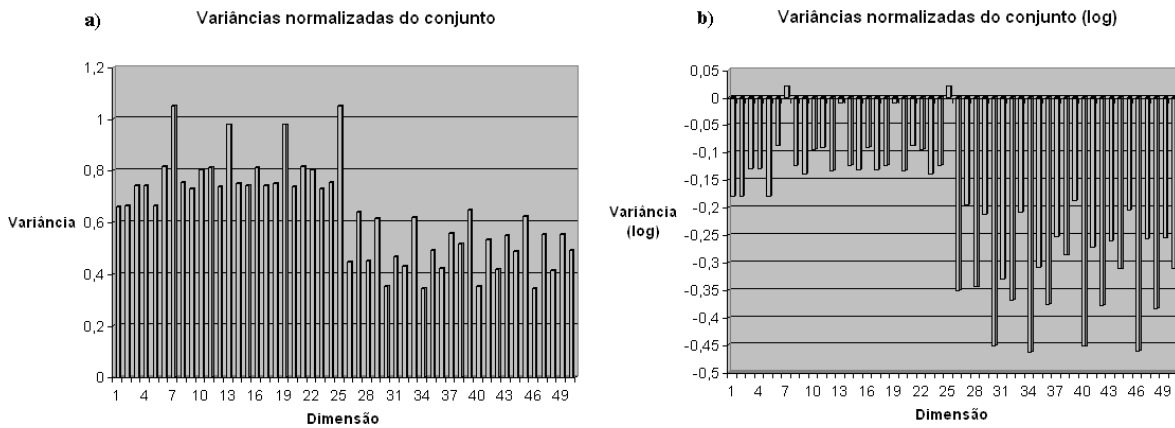


Figura 3.23: Variâncias normalizadas obtidas pela Transformadas de Zernike e Fourier: a) Escala normal, b) Escala logarítmica, para melhor visualização

3.9.8. Exemplo numérico das normalizações

Utilizou-se como exemplo o símbolo da Figura 3.24 (extraído do Apêndice A – Grupo IV, Transformador de 2 enrolamentos, símbolo Trafo05) para apresentar os resultados obtidos das transformadas, com e sem as suas normalizações.

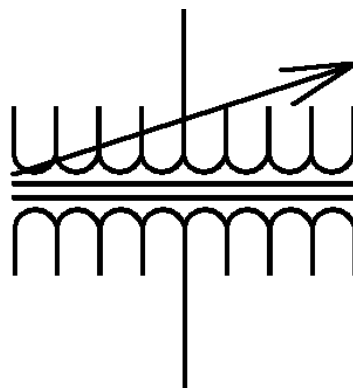


Figura 3.24: Símbolo utilizado como exemplo para visualizar as normalizações

Os resultados das transformadas do símbolo da Figura 3.24 são apresentados na Tabela 3.1, com a intenção puramente ilustrativa de se apresentar o resultado obtido antes e depois da normalização.

Tabela 3.1 – Resultado obtido pela aplicação das transformadas.

Transformada	Resultado
Chebyshev	<p>[+3.1127E+03 +8.6723E+04 +4.1837E+05 +2.5569E+06 +2.1168E+07 +2.2202E+08 +2.8114E+09] [+9.0077E+04 +2.4630E+06 +1.1874E+07 +7.2560E+07 +6.0070E+08 +6.3005E+09 +7.9782E+10] [+4.3514E+05 +1.1890E+07 +5.7319E+07 +3.5028E+08 +2.8998E+09 +3.0415E+10 +3.8514E+11] [+2.6599E+06 +7.2675E+07 +3.5035E+08 +2.1410E+09 +1.7724E+10 +1.8590E+11 +2.3541E+12] [+2.2021E+07 +6.0166E+08 +2.9005E+09 +1.7725E+10 +1.4674E+11 +1.5391E+12 +1.9489E+13] [+2.3097E+08 +6.3106E+09 +3.0422E+10 +1.8591E+11 +1.5391E+12 +1.6143E+13 +2.0441E+14] [+2.9247E+09 +7.9910E+10 +3.8522E+11 +2.3541E+12 +1.9489E+13 +2.0441E+14 +2.5884E+15]] Chebyshev não Normalizada, decomposta em 49 membros</p> <p>[+3.0929E+01 +8.5418E+01 +4.1194E+01 +2.5175E+01 +2.0842E+01 +2.1860E+01 +2.7681E+01] [+8.9967E+01 +2.4557E+02 +1.1838E+02 +7.2341E+01 +5.9888E+01 +6.2815E+01 +7.9541E+01] [+4.3468E+01 +1.1860E+02 +5.7172E+01 +3.4937E+01 +2.8923E+01 +3.0337E+01 +3.8415E+01] [+2.6572E+01 +7.2497E+01 +3.4947E+01 +2.1356E+01 +1.7680E+01 +1.8544E+01 +2.3481E+01] [+2.1999E+01 +6.0019E+01 +2.8932E+01 +1.7680E+01 +1.4637E+01 +1.5352E+01 +1.9440E+01] [+2.3074E+01 +6.2952E+01 +3.0346E+01 +1.8544E+01 +1.5352E+01 +1.6102E+01 +2.0390E+01] [+2.9218E+01 +7.9714E+01 +3.8426E+01 +2.3482E+01 +1.9440E+01 +2.0390E+01 +2.5819E+01]] Chebyshev Normalizada, decomposta em 49 membros</p>
DCT	<p>[+1.1714E+01 -6.3954E+01 +8.1007E+02 -3.1285E+02 +2.2698E+01 -1.3630E+01 -5.6060E+02] [+3.5492E+01 -3.1470E+01 +2.5754E+00 +1.0653E+02 +1.1363E+00 -3.4590E+02 +2.7584E+02] [-2.2698E+01 +1.1850E+01 -1.7642E+02 +2.5913E+01 +6.4705E+01 +3.5584E+01 +1.0322E+01] [+7.0295E+02 +8.3637E+02 -7.9714E+01 -2.3938E+02 -1.7367E+01 -4.2227E+01 -7.3564E+01] [+5.6058E+02 -1.3124E+00 -2.0014E+01 -7.4261E+02 +1.0322E+01 +4.3010E+01 -7.8050E+01] [-1.0931E+01 -5.6707E+01 +2.0126E+01 -8.8178E+01 +4.2800E+01 +3.3864E+01 -4.3642E+02] [+8.1007E+02 -4.7434E+01 +1.5226E+01 -5.4860E+01 +1.7643E+02 +1.1936E+00 +2.0014E+01]] DCT não Normalizada, decomposta em 49 membros</p> <p>[+2.4608E+00 +5.2753E+01 +1.0844E+00 +9.6920E+01 +1.2548E+00 +8.7258E+01 +9.4548E+01] [+1.4261E+00 +7.3001E+01 +1.3137E+01 +1.0107E+00 +3.1153E+00 +9.6600E+01 +1.0280E+00] [+7.9694E+01 +1.1258E+00 +9.8251E+01 +1.2958E+00 +1.9099E+00 +1.4274E+00 +1.1087E+00] [+1.0728E+00 +1.0872E+00 +4.5062E+01 +9.7635E+01 +8.4057E+01 +6.5555E+01 +4.7920E+01] [+1.0577E+00 +2.6917E+01 +8.1862E+01 +9.2843E+01 +1.1087E+00 +1.5374E+00 +4.5818E+01] [+8.9646E+01 +5.6718E+01 +1.2229E+00 +4.1405E+01 +1.5342E+00 +1.4030E+00 +9.5730E+01] [+1.0844E+00 +6.2229E+01 +1.1645E+00 +5.7776E+01 +1.0178E+00 +3.2989E+00 +1.2216E+00]] DCT Normalizada, decomposta em 49 membros</p>
Fourier	<p>[+9.7000E+01 +5.1405E+00 +3.8644E+01 +1.1329E+00 +1.1329E+00 +3.8644E+01 +5.1405E+00] [+1.0459E+01 +1.9980E+00 +7.3582E+01 +2.8058E+00 +7.5511E+01 +8.5800E+01 +1.4511E+00] [+1.6337E+01 +1.4098E+00 +3.9499E+00 +6.1069E+00 +1.2541E+00 +1.6217E+00 +2.2518E+00] [+4.0360E+00 +1.2978E+00 +7.9258E+01 +1.4696E+01 +1.9947E+00 +1.0068E+00 +5.8578E+00] [+4.0360E+00 +5.8578E+00 +1.0068E+00 +1.9947E+00 +1.4696E+01 +7.9258E+01 +1.2978E+00] [+1.6337E+01 +2.2518E+00 +1.6217E+00 +1.2541E+00 +6.1069E+00 +3.9499E+00 +1.4098E+00] [+1.0459E+01 +1.4511E+00 +8.5800E+01 +7.5511E+01 +2.8058E+00 +7.3582E+01 +1.9980E+00]] Fourier não Normalizada, decomposta em 49 membros</p>

	<p>[</p> <p>[+4.5747E+00 +1.6371E+00 -9.5079E+01 +1.2478E+01 +1.2478E+01 -9.5078E+01 +1.6371E+00]</p> <p>[+2.3475E+00 +6.9217E+01 -3.0677E+01 +1.0317E+00 -2.8089E+01 -1.5315E+01 +3.7232E+01]</p> <p>[+2.7934E+00 +3.4347E+01 +1.3737E+00 +1.8094E+00 +2.2643E+01 +4.8345E+01 +8.1172E+01]</p> <p>[+1.3953E+00 +2.6067E+01 -2.3246E+01 +2.6876E+00 +6.9049E+01 +6.7506E+03 +1.7678E+00]</p> <p>[+1.3953E+00 +1.7678E+00 +6.7529E+03 +6.9050E+01 +2.6876E+00 -2.3246E+01 +2.6066E+01]</p> <p>[+2.7934E+00 +8.1172E+01 +4.8345E+01 +2.2642E+01 +1.8094E+00 +1.3737E+00 +3.4348E+01]</p> <p>[+2.3475E+00 +3.7231E+01 -1.5315E+01 -2.8089E+01 +1.0317E+00 -3.0677E+01 +6.9216E+01]]</p> <p>Fourier Normalizada, decomposta em 49 membros</p>																																																																																																												
Radon	<p>[</p> <p>[+7.0000E+00 +9.0000E+00 +8.0000E+00 +7.0000E+00 +1.6000E+01 +8.0000E+00 +9.0000E+00]</p> <p>[+8.0000E+00 +9.0000E+00 +1.0000E+01 +7.0000E+00 +1.6000E+01 +8.0000E+00 +9.0000E+00]</p> <p>[+8.0000E+00 +7.0000E+00 +8.0000E+00 +6.0000E+00 +1.3000E+01 +9.0000E+00 +6.0000E+00]</p> <p>[+8.0000E+00 +1.1000E+01 +7.0000E+00 +7.0000E+00 +1.4000E+01 +1.1000E+01 +8.0000E+00]</p> <p>[+9.0000E+00 +8.0000E+00 +7.0000E+00 +7.0000E+00 +1.0000E+01 +1.2000E+01 +9.0000E+00]</p> <p>[+7.0000E+00 +1.0000E+01 +7.0000E+00 +9.0000E+00 +6.0000E+00 +1.2000E+01 +1.0000E+01]</p> <p>[+9.0000E+00 +6.0000E+00 +7.0000E+00 +1.1000E+01 +4.0000E+00 +1.2000E+01 +1.1000E+01]]</p> <p>Radon não Normalizada, decomposta em 49 membros</p> <p>[</p> <p>[-2.9267E+00 -2.8176E+00 -2.8687E+00 -2.9267E+00 -2.5677E+00 -2.8687E+00 -2.8176E+00]</p> <p>[-2.8687E+00 -2.8176E+00 -2.7718E+00 -2.9267E+00 -2.5677E+00 -2.8687E+00 -2.8176E+00]</p> <p>[-2.8687E+00 -2.9267E+00 -2.8687E+00 -2.9937E+00 -2.6579E+00 -2.8176E+00 -2.9937E+00]</p> <p>[-2.8687E+00 -2.7304E+00 -2.9267E+00 -2.9267E+00 -2.6257E+00 -2.7304E+00 -2.8687E+00]</p> <p>[-2.8176E+00 -2.8687E+00 -2.9267E+00 -2.9267E+00 -2.7718E+00 -2.6926E+00 -2.8176E+00]</p> <p>[-2.9267E+00 -2.7718E+00 -2.9267E+00 -2.8176E+00 -2.9937E+00 -2.6926E+00 -2.7718E+00]</p> <p>[-2.8176E+00 -2.9937E+00 -2.9267E+00 -2.7304E+00 -3.1697E+00 -2.6926E+00 -2.7304E+00]]</p> <p>Radon Normalizada, decomposta em 49 membros</p>																																																																																																												
Zernike	<p>[</p> <table> <tr> <td>[(-62.764, -13.135j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+106.62, +16.424j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(-46.133, +1.2532j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(-5.3354, -13.303j)</td> <td>(+0.0000, +0.0000j)</td> <td>(-19.225, +1.9247j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(-53.106, -13.346j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+119.83, +24.695j)</td> <td>(+0.0000, +0.0000j)</td> <td>(-103.44, -16.144j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+39.633, -2.9845j)</td> <td>(+0.0000, +0.0000j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(-40.836, -11.490j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+110.40, +25.468j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(-129.72, -26.489j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+79.201, +8.2911j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(-30.863, -9.5637j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+94.665, +23.296j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(-137.05, -31.435j)</td> <td>(+0.0000, +0.0000j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(-23.447, -7.9959j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+78.321, +20.168j)</td> <td>(+0.0000, +0.0000j)</td> <td>(-132.95, -32.879j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(-18.102, -6.8172j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+63.523, +16.921j)</td> <td>(+0.0000, +0.0000j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(-14.301, -5.9673j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+50.954, +13.928j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(-11.614, -5.3739j)</td> <td>(+0.0000, +0.0000j)</td> <td>]</td> </tr> <tr> <td>[(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td></td> </tr> <tr> <td>(+0.0000, +0.0000j)</td> <td>(+0.0000, +0.0000j)</td> <td>(-9.7194, -4.9723j)</td> <td>]]</td> </tr> </table> <p>Zernike não Normalizada, decomposta em 50 membros</p>	[(-62.764, -13.135j)	(+0.0000, +0.0000j)	(+106.62, +16.424j)		(+0.0000, +0.0000j)	(-46.133, +1.2532j)	(+0.0000, +0.0000j)		(-5.3354, -13.303j)	(+0.0000, +0.0000j)	(-19.225, +1.9247j)]	[(+0.0000, +0.0000j)	(-53.106, -13.346j)	(+0.0000, +0.0000j)		(+119.83, +24.695j)	(+0.0000, +0.0000j)	(-103.44, -16.144j)		(+0.0000, +0.0000j)	(+39.633, -2.9845j)	(+0.0000, +0.0000j)]	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(-40.836, -11.490j)		(+0.0000, +0.0000j)	(+110.40, +25.468j)	(+0.0000, +0.0000j)		(-129.72, -26.489j)	(+0.0000, +0.0000j)	(+79.201, +8.2911j)]	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(-30.863, -9.5637j)	(+0.0000, +0.0000j)	(+94.665, +23.296j)		(+0.0000, +0.0000j)	(-137.05, -31.435j)	(+0.0000, +0.0000j)]	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(+0.0000, +0.0000j)	(-23.447, -7.9959j)	(+0.0000, +0.0000j)		(+78.321, +20.168j)	(+0.0000, +0.0000j)	(-132.95, -32.879j)]	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(-18.102, -6.8172j)		(+0.0000, +0.0000j)	(+63.523, +16.921j)	(+0.0000, +0.0000j)]	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(-14.301, -5.9673j)	(+0.0000, +0.0000j)	(+50.954, +13.928j)]	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(+0.0000, +0.0000j)	(-11.614, -5.3739j)	(+0.0000, +0.0000j)]	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)		(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(-9.7194, -4.9723j)]]
[(-62.764, -13.135j)	(+0.0000, +0.0000j)	(+106.62, +16.424j)																																																																																																											
(+0.0000, +0.0000j)	(-46.133, +1.2532j)	(+0.0000, +0.0000j)																																																																																																											
(-5.3354, -13.303j)	(+0.0000, +0.0000j)	(-19.225, +1.9247j)]																																																																																																										
[(+0.0000, +0.0000j)	(-53.106, -13.346j)	(+0.0000, +0.0000j)																																																																																																											
(+119.83, +24.695j)	(+0.0000, +0.0000j)	(-103.44, -16.144j)																																																																																																											
(+0.0000, +0.0000j)	(+39.633, -2.9845j)	(+0.0000, +0.0000j)]																																																																																																										
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(-40.836, -11.490j)																																																																																																											
(+0.0000, +0.0000j)	(+110.40, +25.468j)	(+0.0000, +0.0000j)																																																																																																											
(-129.72, -26.489j)	(+0.0000, +0.0000j)	(+79.201, +8.2911j)]																																																																																																										
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(-30.863, -9.5637j)	(+0.0000, +0.0000j)	(+94.665, +23.296j)																																																																																																											
(+0.0000, +0.0000j)	(-137.05, -31.435j)	(+0.0000, +0.0000j)]																																																																																																										
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(+0.0000, +0.0000j)	(-23.447, -7.9959j)	(+0.0000, +0.0000j)																																																																																																											
(+78.321, +20.168j)	(+0.0000, +0.0000j)	(-132.95, -32.879j)]																																																																																																										
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(-18.102, -6.8172j)																																																																																																											
(+0.0000, +0.0000j)	(+63.523, +16.921j)	(+0.0000, +0.0000j)]																																																																																																										
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(-14.301, -5.9673j)	(+0.0000, +0.0000j)	(+50.954, +13.928j)]																																																																																																										
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(+0.0000, +0.0000j)	(-11.614, -5.3739j)	(+0.0000, +0.0000j)]																																																																																																										
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)																																																																																																											
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(-9.7194, -4.9723j)]]																																																																																																										

Zernike	Zernike =		
	[
	[(+1.9031, +15.252j)	(+0.0000, +0.0000j)	(+1.7433, -43.621j)
	(+0.0000, +0.0000j)	(+1.1213, -78.675j)	(+0.0000, +0.0000j)
	(+1.6607, +73.964j)	(+0.0000, +0.0000j)	(+87.064, -62.044j)
	[(+0.0000, +0.0000j)	(+1.8568, +19.826j)	(+0.0000, +0.0000j)
	(+1.7893, -44.638j)	(+0.0000, +0.0000j)	(+2.0431, +353.10j)
	(+0.0000, +0.0000j)	(+1.0720, -76.762j)	(+0.0000, +0.0000j)
	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+1.7757, +26.107j)
	(+0.0000, +0.0000j)	(+1.7706, -42.334j)	(+0.0000, +0.0000j)
	(+2.1238, +2049.4j)	(+0.0000, +0.0000j)	(+1.6327, -38.966j)
	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)
	(+1.6843, +32.809j)	(+0.0000, +0.0000j)	(+1.7283, -38.794j)
	(+0.0000, +0.0000j)	(+2.1475, -2694.7j)	(+0.0000, +0.0000j)
	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)
	(+0.0000, +0.0000j)	(+1.5889, +39.625j)	(+0.0000, +0.0000j)
	(+1.6739, -34.607j)	(+0.0000, +0.0000j)	(+2.1440, +7981.7j)
	[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)
	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+1.4919, +46.432j)
	(+0.0000, +0.0000j)	(+1.6122, -30.051j)	(+0.0000, +0.0000j)
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	
(+1.3943, +53.179j)	(+0.0000, +0.0000j)	(+1.5456, -25.279j)	
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	
(+0.0000, +0.0000j)	(+1.2964, +59.845j)	(+0.0000, +0.0000j)	
[(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	
(+0.0000, +0.0000j)	(+0.0000, +0.0000j)	(+1.1986, +66.426j)	
Zernike Normalizada, decomposta em 50 membros			

3.10. Metodologia para a rejeição (*outliers*)

A metodologia de rejeição considera a avaliação da RNA já treinada com a base de dados de treinamento, pela aplicação de símbolos representando má segmentação ou símbolos desconexos (que não pertencem ao contexto do trabalho), visando analisar a capacidade do modelo para detecção de erros e confiabilidade [FUM00]. Esta metodologia foi aplicada no classificador primário (Rede Neural MLP). Para o classificador secundário, não se aplica devido á topologia da rede escolhida funcionar como memória.

Os símbolos a serem rejeitados aparecem para o classificador como uma condição de anormalidade, com o objetivo de se testar a estabilidade do modelo. Fato que, algumas vezes pode ocultar os agrupamentos ou apresentar uma outra estrutura interessante de dados. Também a matriz

de covariância⁷ é fortemente influenciada por símbolos *outliers* com medidas fora do desvio padrão. Conseqüentemente, não devem ser utilizados os métodos confiantes na matriz de covariância (p. ex. através do agrupamento dos dados em esferas, como kmeans) porquê não são robustos a *outliers* [PER97].

O procedimento de avaliação realizada foi o de criar-se um quarto grupo de testes (após os grupos de treinamento, teste e validação) com 102 elementos (aproximadamente 30% da base de dados). A RNA não foi retreinada para os símbolos *outliers* porque não se pode garantir que os símbolos *outliers* estejam dentro de um padrão simulando futuros erros de segmentação. A distribuição dos símbolos por metaclasses é apresentada na Tabela 3.2.

Tabela 3.2 – Distribuição dos símbolos *outliers* por metaclasses

Metaclasses	I	II	III	IV	V
Símbolos <i>outliers</i>	39	14	28	15	6

3.11. Comentários finais

Este capítulo apresentou a metodologia proposta para o reconhecimento de símbolos gráficos no contexto de projetos CAD unifilares de engenharia elétrica. A ênfase do Capítulo está na abordagem hierárquica do problema e na preparação dos dados, extraídos com as transformadas para as Redes Neurais Artificiais. Conseqüentemente, a normalização dos dados deve ser tratada com muita atenção, sob o risco de tornar a base de dados não linearmente separável implicando em um esforço muito maior dos classificadores para o reconhecimento dos símbolos. Vale observar que as técnicas de normalização permaneceram as mesmas com as transformadas combinadas, pois a normalização é individualizada.

De todos os métodos analisados, observou-se também a sua praticidade de utilização, avaliando-se a base de dados, de como os símbolos estão dispostos e como serão introduzidos para análise. Todos os métodos propostos, sugerem uma redução de dimensionalidades, para otimizar as informações sobre as regiões de decisão de onde estarão dispostos os símbolos. Dos problemas práticos, observaram-se variações de escala e rotação, ainda que dos algoritmos de redução de dimensionalidades que menos demandariam recursos com estas variações são as transformadas

⁷ Ou matrix de confusão, obtida através dos experimentos de reconhecimento.

numéricas, (ou momentos matemáticos), além da eficiência computacional. Mas, conforme informações coletadas, a melhor proposta para trabalhar em conjunto com as transformadas é a de utilizar um classificador baseado em rede neural, com funções de ativação sigmoideal e/ou logarítmica.

Os algoritmos estatísticos de extração de características apresentam uma peculiaridade interessante: conforme forem as suas implementações, definindo-se os centros geométricos das imagens, e manipulando-as a partir deste ponto, e extraíndo-se as características, pode-se desprezar a rotação da imagem, como por exemplo nas implementações de Chebyshev, Fourier e Zernike.

Para a Rede Neural MLP-BP, o formato dos dados é extremamente importante para a escolha do modelo apropriado para a função de ativação. Observa-se que, por causa do simples relacionamento entre os valores da função e suas derivativas, adicionalmente, nenhuma avaliação de exponencial é necessária durante a fase de retro-propagação do algoritmo.

Uma aplicação típica para as Redes Hebbianas é a de funcionarem como memórias associativas (não lineares). As memórias associativas (artificiais) funcionam exatamente como a humana, desde que contenham ligações e pesos associados de interconexão (e um padrão de atividade). Embora tenham capacidades limitadas de armazenamento, são robustas à presença de ruídos [PRI99].

A seguir apresenta-se a topologia das RNAs utilizadas, a composição das metaclasses em função de suas formas geométricas, descrição e análise dos resultados obtidos pelos experimentos realizados e uma avaliação dos métodos utilizados.

Capítulo 4

Experimentos e Análises de Resultados

Introdução

Neste capítulo, estão descritos os resultados obtidos com os experimentos de extração de características e reconhecimento através da RNA proposta, sem a utilização dos recursos de *prunning*, que permitem a otimização da rede neural, descartando células não utilizadas ou com influências pouco consideráveis no decorrer do processamento.

Para as transformadas isoladas, testou-se somente o classificador principal, com o objetivo de avaliar o desempenho das transformadas. Para as transformadas associadas, por apresentarem resultados mais significativos, testou-se até o nível das redes secundárias (Hebbianas), para as combinações Zernike-Chebyshev e Zernike-Fourier.

A combinação Zernike-Chebyshev foi testada devido as características de imunidade rotacional da Transformada de Zernike e a forma de implementação da transformada de Chebyshev (em 4 quadrantes), para tornar-se imune a rotação da imagem em ângulos de 90°.

Para a combinação Zernike-Fourier, além das características invariantes à rotação, levou-se em consideração as suas respectivas convergências durante o treinamento das redes, com os valores de SSE tendendo a zero.

Para se determinar os resultados de reconhecimento médio ($\bar{R} \pm DP$) e do seu desvio padrão, utilizou-se da Equação 4.1 para determinação do reconhecimento médio complementada pela Equação 4.2 referente ao desvio padrão.

$$\bar{R} = \frac{1}{N} \sum_{i=1}^N R_i \quad (4.1)$$

sendo \bar{R} = Reconhecimento médio da transformada (%),
 R_i = Reconhecimento do teste (em %) da validação cruzada i , e
 $N = 10$, indicando o número de testes realizados com validação cruzada

E para o desvio padrão:

$$DP = \sqrt{\frac{\sum_{i=1}^N (\bar{R} - R_i)^2}{N}} \quad (4.2)$$

sendo DP = Desvio padrão obtido a partir dos reconhecimentos, em %
 \bar{R} = Reconhecimento médio da transformada, em %
 R_i = Reconhecimento do teste (em %) de validação cruzada i
 $N = 10$, indicando o número de testes realizados com validação cruzada.

4.1. Redes Neurais: Classificador Primário

A camada de entrada da Rede Neural é composta de 49 elementos (salvo Zernike que necessitou de 50 elementos, devido ao desmembramento dos valores complexos), uma camada escondida de 27 elementos (com função de ativação definida como mostra a Tabela 3.1) e 5 neurônios de saída (com função de ativação por *threshold*, com nível de ativação de 0,5), como mostra a Figura 4.1. A diferença existente com a rede utilizada com a Transformada de Zernike consiste na presença de mais um neurônio na camada de entrada, assim como as suas interligações com os demais neurônios da camada intermediária, como mostra a Figura 4.2, devido ao seu vetor característico ser de 50 elementos (25 valores complexos), exigindo redes classificatórias diferenciadas na capacidade de entrada para que se possa abranger os seus vetores.

O tamanho do vetor de entrada de 49 elementos para a rede neural deve-se à proximidade das dimensões dos vetores resultantes obtidos para os momentos (como por exemplo, 25 elementos para Chebyshev e 24 elementos para Zernike). Esse vetor de 49 elementos de entrada é o menor vetor aonde se consegue uma aproximação de metade das características para cada transformada.

Para treinar a rede, utilizou-se para o JavaNNS, uma distribuição randômica de pesos durante a inicialização, com atualização, pesos, h (passo de aprendizagem) e épocas definidos na Tabela 4.1, com todos os modelos utilizando-se da técnica de validação cruzada fator 10.

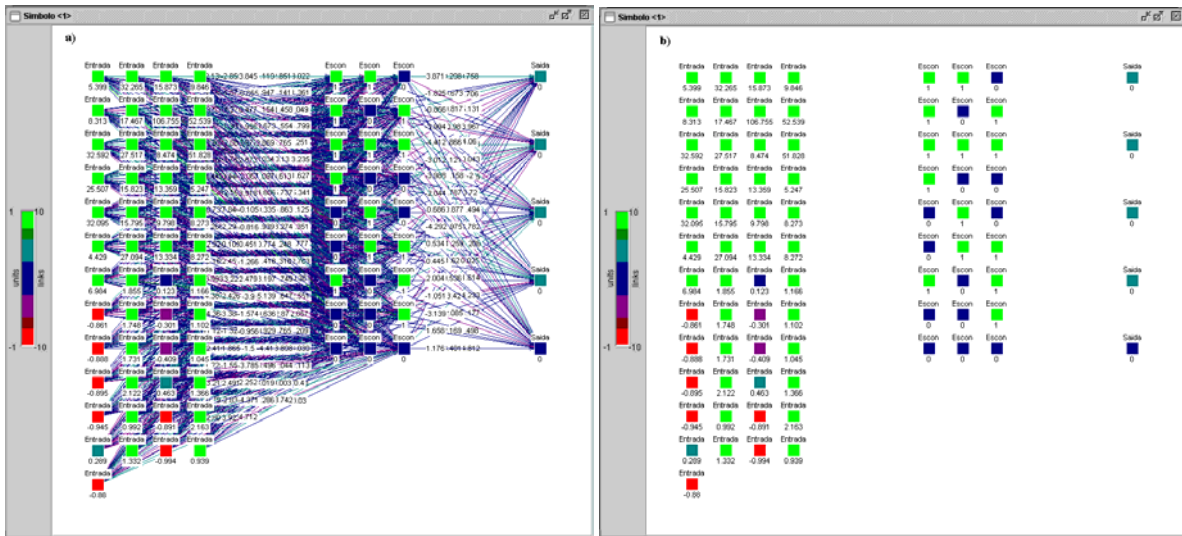


Figura 4.1: Topologia da Rede Neural utilizada para os experimentos no classificador primário: a) Mostrando as interligações b) Apresentando as células

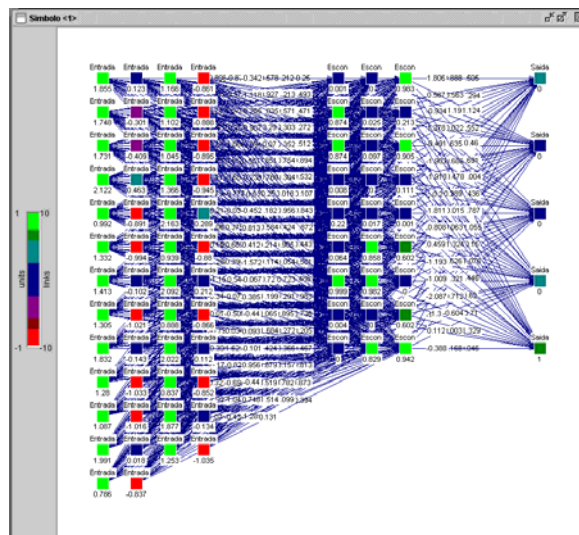


Figura 4.2: Topologia da Rede Neural utilizada para os testes com a transformada de Zernike

Tabela 4.1: Configuração da aprendizagem/épocas por transformada

Transformada	Passo (h)	Épocas	Atualização	F. Ativ.
Zernike+Chebyshev	0,2	5000	Topológica	Sigmoidal
Zernike+Fourier	0,2	1000	Topológica	Sigmoidal
Chebyshev	0,2	200	Ordem serial	Linear
DCT	0,2	5000	Topológica	Sigmoidal
Fourier	0,2	200	Topológica	Logaritmica
Radon	0,1	2000	Topológica	Logaritmica
Zernike	0,2	1000	Topológica	Sigmoidal

Passo (h) = Passo de aprendizagem

F. Ativ. = Função de ativação da camada escondida

A técnica utilizada para treinamento da RNA primária foi a de validação cruzada (fator 10), utilizando-se a cada 8 elementos de treinamento, 1 elemento para validação e 1 elemento para testes em cada metaclasse, como mostra a Figura 4.3. Os processos de extração de característica para modelagem das bases de dados de treinamento, validação e testes foram realizados de forma independente. Após treinada a rede neural, em cada teste de validação cruzada, realizou-se um teste para avaliação do desempenho para os 10 conjuntos de treinamento, teste e validação.

Com os testes de avaliação realizados, treina-se novamente a RNA primária com toda a base de dados (exceto com os símbolos “outliers”, que não pertencem à base de dados), para realizar-se uma nova avaliação. Os símbolos “outliers” utilizados representam uma base de dados criada artificialmente, devido a não existência de uma metodologia de segmentação dos símbolos para este caso. Aplicam-se agora os símbolos “outliers” para testes de confiabilidade do modelo, observando-se a capacidade seletiva do conjunto.

As RNAs secundárias operam como memória, e para os seus treinamentos aplica-se toda a base de dados, (sem os símbolos “outliers”). O funcionamento da RNA como memória caracteriza um detector de “menor distância euclidiana” do conjunto.

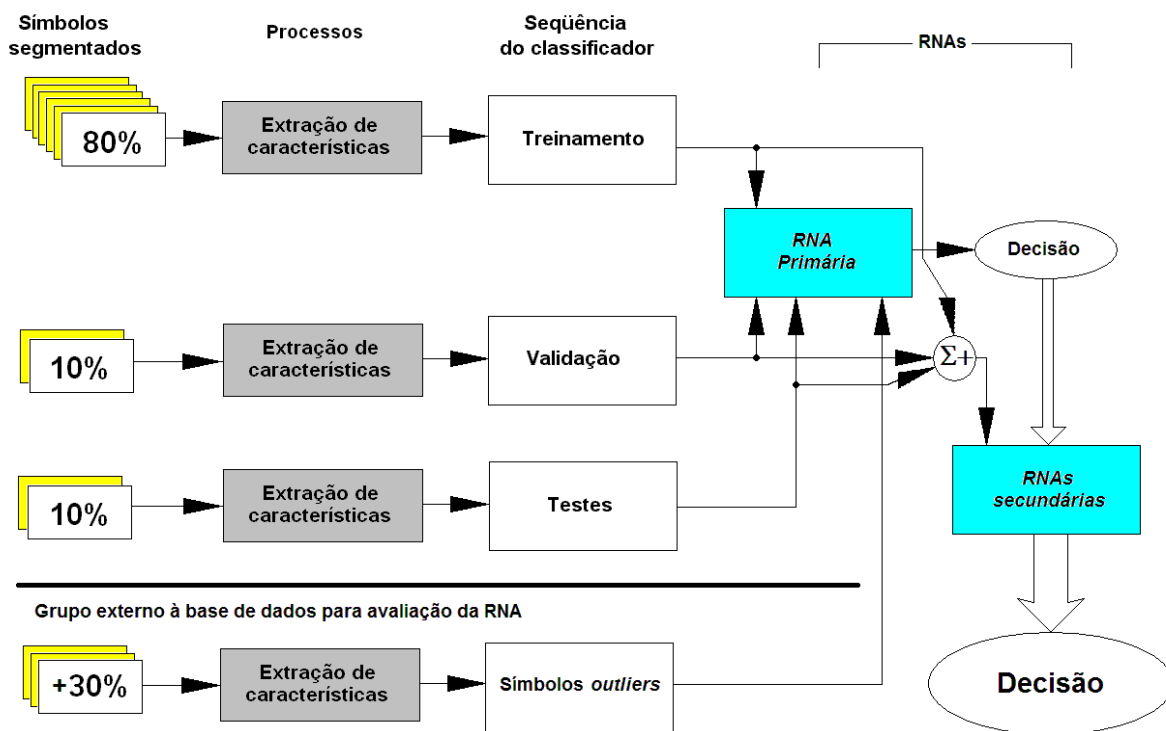


Figura 4.3: Demonstração de como realizaram-se os testes com as RNAs

4.2. Redes Neurais: Classificadores Secundários

Os classificadores secundários são formados por redes Hebbianas e MLP-BP, trabalhando como memória, determinando a distância Euclidiana entre os padrões e o centro dos *clusters* (agrupamentos). A escolha do funcionamento das RNAs como memória deve-se ao fato de que, por definição pela ABNT[ABNTa-d], há somente uma representação possível para alguns símbolos, conseqüentemente para algumas saídas das metaclasses II e III (uma vez que escalas, espessura de traçados e orientações dos desenhos já são resolvidos pelas técnicas de extração de características). Os treinamentos para as redes Hebbianas e MLP-BP são supervisionados, e consiste em aplicar-se toda a base de dados da metaclasses correspondente para realizar-se um futuro reconhecimento. Deve-se observar que o SSE (*Sum of Square Errors* – Soma Quadrática dos Erros) durante o treinamento não faz sentido para esta topologia. Para cada metaclasses, o número de saídas é definido pelo tipo de classe do símbolo no grupo:

- **1ª metaclasses:** Rede Hebbiana, definida com 2 saídas, identificando reatores e autotransformadores, com e sem comutadores de TAP (Terminal de Ajuste de Potencial), como mostra a Figura 4.4.



Figura 4.4: Classificador secundário para a 1ª metaclasses

- **2ª metaclasses:** Rede MLP-BP, definida com 11 saídas, identificando conectores de aterramento, conectores de terra digital, seccionadoras, um tipo de representação de disjuntores, módulos de funções (como relés de proteção, unidades de automatismo, etc...),

dispositivos eletrônicos (como diodos, raramente utilizados em diagramas unifilares) e entradas ou saídas de linhas, como mostra a Figura 4.5.

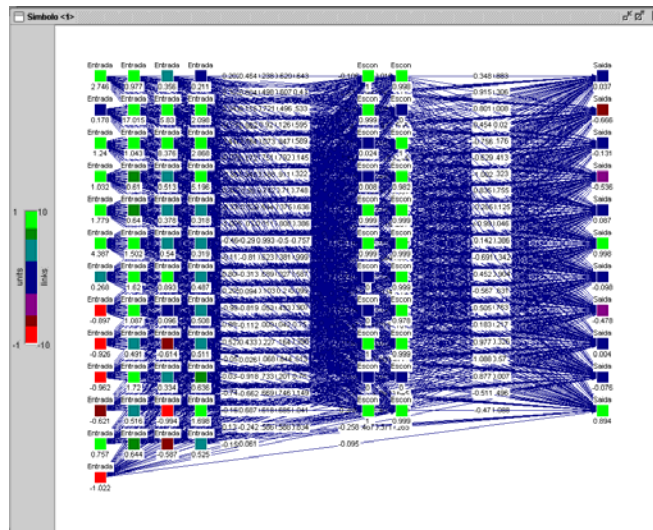


Figura 4.5: Classificador secundário para a 2ª metaclasses

- **3ª metaclasses:** Rede MLP-BP, definida com 6 saídas, identificando outras representações de disjuntores, dispositivos de painel (como p.ex: sefímetros, voltímetros, amérímetros, watímetros, etc...), como mostra a Figura 4.6.

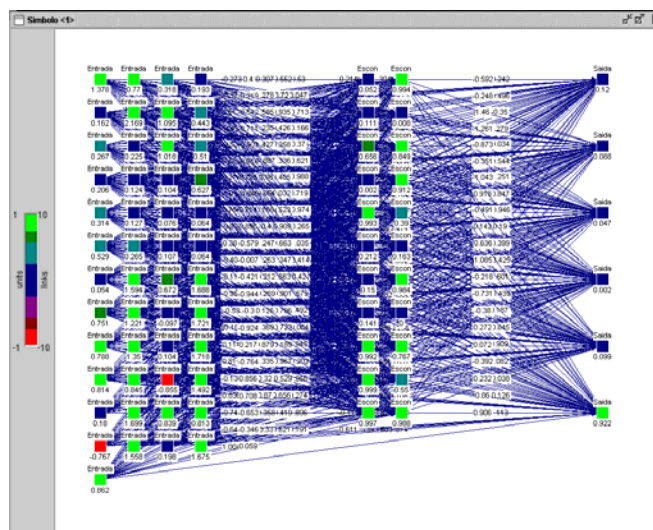


Figura 4.6: Classificador secundário para a 3ª metaclasses

• **4ª metaclassa:** Rede MLP-BP, definida com 3 saídas, identificando transformadores com 2 enrolamentos, transformadores de potência com e sem TAP (Terminal de Ajuste de Potencial), transformadores de potencial (TP de instrumentação), o qual é um caso particular do transformador de potência e a representação do Transformador de Corrente (TC), como mostra a Figura 4.7.

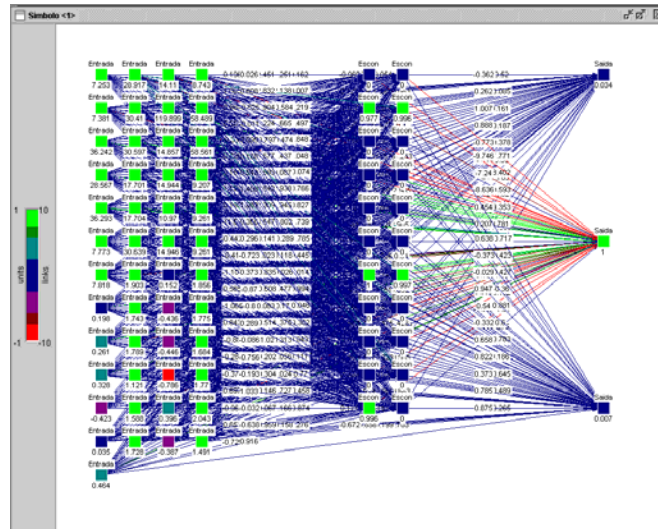


Figura 4.7: Classificador secundário para a 4ª metaclassa

• **5ª metaclassa:** Rede MLP-BP, definida com 2 saídas, identificando transformadores com 3 enrolamentos, com e sem TAP (Terminal de Ajuste de Potencial), como mostra a Figura 4.8.

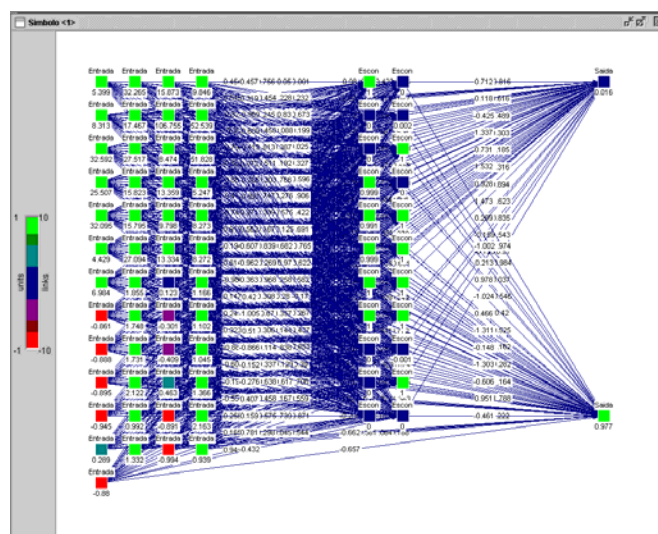


Figura 4.8: Classificador secundário para a 5ª metaclassa

4.3. Transformada de Chebyshev

Para o treinamento inicial da rede, no classificador primário, para o JavaNNS, utilizou-se uma distribuição de pesos aleatória na inicialização, variando entre -5 a $+5$ (*random weights*). A atualização com ordem serial (*serial order*) e aprendizado utilizando *Backpropagation Momentum* com $\eta=0,2$, $d_{\max}=0,01$, $\mu = 0,5$ e $c=0,1$, épocas definidas com 200. Os padrões de treinamento de saída foram definidos como *Threshold*, com valor inferior (*lower*) de $-1,0$, superior (*upper*) de $+1,0$, centrado (*inner*) em $0,0$ e saída (*outer*) predefinida em $1,0$.

A Transformada de Chebyshev apresentou um reconhecimento médio de $63,1 \pm 14,2\%$, e durante os testes de reconhecimento, obtiveram-se os resultados da Tabela 4.2 como matriz de confusão para o classificador primário, distribuídos em percentuais relativos à quantidade de símbolos por metaclassa. Para maiores detalhes, ver Apêndice B – Curvas de treinamento e testes de validação cruzada para a Transformada de Chebyshev. A Curva de treinamento global pode ser vista na Figura 4.9, e o registro do treinamento na Listagem 4.1.

A Tabela 4.2 mostra a identificação das metaclasses por grupo após o treinamento com toda a base de dados (sem os símbolos *outliers*) e com a re-aplicação de toda a base, para visualizar-se uma matriz de confusão global e preparar o classificador primário para os futuros testes com *outliers*. O reconhecimento médio apresentado refere-se à média obtida pelos testes de validação cruzada, apresentados no Apêndice B.

Tabela 4.2: Chebyshev: Matriz de confusão (%)

Meta-classe	Resultado da análise (%)						
	I	II	III	IV	V	N.R.	Rec. médio
I	76,7	1,7	6,0	8,6	0,0	7,0	69,3 ± 9,5
II	4,0	82,0	8,0	2,0	0,0	4,0	75,0 ± 16,1
III	5,0	21,7	60,0	11,7	0,0	1,6	51,5 ± 16,7
IV	0,0	2,0	37,3	58,8	0,0	1,9	49,2 ± 16,5
V	0,0	0,0	0,0	17,2	72,4	10,4	65,0 ± 20,9
N.R. = Não reconhecido							

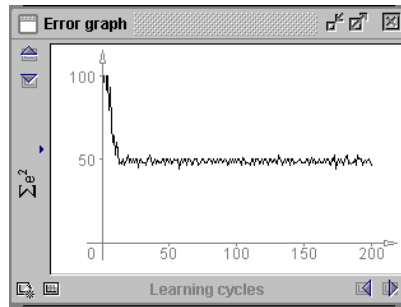


Figura 4.9: Chebyshev: Curva de aprendizagem apresentando o Erro \times Época

opened at: Mon Dec 15 11:19:30 BRST 2003

Step 20 SSE: 48.0
 Step 40 SSE: 47.0
 Step 60 SSE: 48.0
 Step 80 SSE: 50.0
 Step 100 SSE: 48.0
 Step 120 SSE: 47.0
 Step 140 SSE: 46.0
 Step 160 SSE: 48.0
 Step 180 SSE: 46.0
 Step 200 SSE: 46.0

Listagem 4.1: Chebyshev: Registro do treinamento fornecido pelo simulador JavaNNS

Após extraídas as características dos símbolos para testes de rejeição, aplicam-se estes dados na entrada do classificador primário, obtendo-se os resultados de *outliers* como listados na Tabela 4.3. Foram utilizados 102 símbolos para testes, simulando erros de segmentação. Os grupos da coluna vertical mostram a que classe o símbolo para rejeição foi identificado, e os grupos da linha horizontal mostram o grupo sob experimentação pelo classificador primário. Obteve-se, para a Transformada de Chebyshev, a taxa de rejeição de 24,5%, com o erro calculado em 75,5%.

Tabela 4.3: Chebyshev: Testes com rejeições

Grupo	I	II	III	IV	V	Nenhum
I	5	6	13	7	--	9
II	3	1	4	--	--	6
III	--	7	4	8	3	6
IV	--	2	5	1	4	3
V	--	--	1	4	--	1

4.2. Transformada Discreta do Coseno (DCT)

Para o treinamento inicial da rede, no classificador primário, para o JavaNNS, utilizou-se uma distribuição de pesos aleatória na inicialização, variando entre -5 a $+5$ (*random weights*). A atualização com ordem topológica (*topological order*) e aprendizado utilizando *Backpropagation Momentum* com $\eta=0,2$, $d_{\max}=0,01$, $\mu = 0,5$ e $c=0,1$, épocas definidas com 5000 ciclos. Os padrões de treinamento de saída foram definidos como *Threshold*, com valor inferior (*lower*) de $-1,0$, superior (*upper*) de $+1,0$, centrado (*inner*) em $0,0$ e saída (*outer*) predefinida em $1,0$.

A Transformada do Coseno apresentou um reconhecimento médio de $65,2 \pm 17,6\%$, e durante os testes de reconhecimento, obtiveram-se os resultados da Tabela 4.4 como matriz de confusão para o classificador primário, distribuídos em percentuais relativos a quantidade de símbolos por metaclasse. Para maiores detalhes, ver Apêndice C – Curvas de treinamento e testes de validação cruzada para a Transformada Discreta do Coseno. A Curva de treinamento global pode ser vista na Figura 4.10, e o registro do treinamento na Listagem 4.2.

Tabela 4.4: DCT: Matriz de confusão (%)

Meta-classe	Resultado da análise (%)						
	I	II	III	IV	V	N.R.	Rec. Médio
I	85,3	0,0	0,0	10,3	4,4	0,0	77,3 ± 12,5
II	0,0	49,0	33,3	3,9	0,0	13,8	46,3 ± 20,1
III	1,7	13,6	62,7	6,8	3,4	11,8	57,0 ± 16,3
IV	1,7	0,0	9,8	76,5	0,0	12,0	66,0 ± 21,2
V	6,7	0,0	3,4	10,3	72,4	7,2	63,3 ± 30,0
N.R. = Não reconhecido							

A Tabela 4.5 mostra os resultados dos testes de *outliers*. Conseguiu-se, para a Transformada do Coseno, a taxa de Rejeição de $23,6\%$, com o Erro calculado em $76,4\%$.

opened at: Mon Dec 29 18:12:11 BRST 2003

Backprop-Momentum selected

Step 500 SSE: 41.0

Step 1000 SSE: 42.0

Step 1500 SSE: 42.0

Step 2000 SSE: 45.0

Step 2500 SSE: 30.0

Step 3000 SSE: 31.0

Step 3500 SSE: 40.0

Step 4000 SSE: 33.0

Step 4500 SSE: 40.0

Step 5000 SSE: 27.0

Listagem 4.2: DCT: Registro do treinamento fornecido pelo simulador JavaNNS

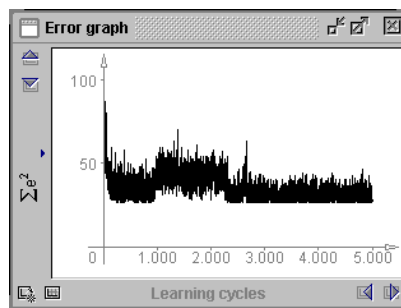


Figura 4.10: DCT: Curva de aprendizagem apresentando o Erro \times Época

Tabela 4.5: DCT: Testes com rejeições

Grupo	I	II	III	IV	V	Nenhum
I	7	5	11	5	--	11
II	1	2	4	2	--	5
III	2	7	4	7	4	4
IV	2	2	4	3	2	2
V	--	--	--	2	2	2

4.5. Transformada de Fourier

Para o treinamento inicial da rede, no classificador primário, para o JavaNNS, utilizou-se uma distribuição de pesos aleatória na inicialização, variando entre -5 a $+5$ (*random weights*). A atualização com ordem serial (*serial order*) e aprendizado utilizando *Backpropagation* com $\eta=0,2$ e $d=0,1$, épocas definidas com 200 ciclos. Os padrões de treinamento de saída foram definidos como *Threshold*, com valor inferior (*lower*) de $-1,0$, superior (*upper*) de $+1,0$, centrado (*inner*) em $0,0$ e saída (*outer*) predefinida em $1,0$.

A Transformada de Fourier apresentou um reconhecimento médio de $63,5 \pm 14,3\%$, e durante os testes de reconhecimento, obtiveram-se os resultados da Tabela 4.6 como matriz de confusão para o classificador primário, distribuídos em percentuais relativos a quantidade de símbolos por metaclasse. Para maiores detalhes, ver Apêndice D – Curvas de treinamento e testes de validação cruzada para a Transformada de Fourier. A Curva de treinamento global pode ser vista na Figura 4.11, e o registro do treinamento na Listagem 4.3.

Tabela 4.6: Fourier: Matriz de confusão (%)

Meta-classe	Resultado da análise (%)						
	I	II	III	IV	V	N.R.	Rec. Médio
I	99,1	0,0	0,0	0,0	0,0	0,9	89,2 ± 10,8
II	3,9	17,6	17,6	3,9	3,9	53,1	14,0 ± 18,0
III	6,9	5,2	39,6	5,2	1,7	41,4	33,0 ± 15,2
IV	0,0	0,0	2,0	96,1	0,0	1,9	86,3 ± 13,1
V	0,0	0,0	0,0	13,8	72,4	13,8	66,7 ± 22,2

N.R. = Não Reconhecido

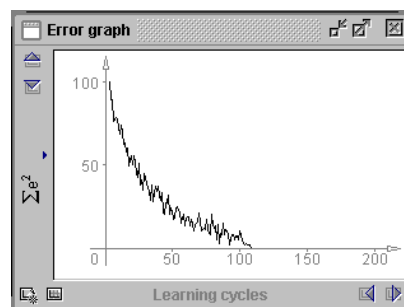


Figura 4.11: Fourier: Curva de aprendizagem apresentando o Erro x Época

A Tabela 4.7 mostra os resultados dos testes de rejeição. Conseguiu-se, para a Transformada de Fourier, a taxa de Rejeição de 36,3%, com o Erro calculado em 63,7%.


```

opened at: Mon Dec 15 12:24:13 BRST 2003
Step 20 SSE: 50.0
Step 40 SSE: 31.0
Step 60 SSE: 18.0
Step 80 SSE: 13.0
Step 100 SSE: 10.0
Step 120 SSE: 0.0
Step 140 SSE: 0.0
Step 160 SSE: 0.0
Step 180 SSE: 0.0
Step 200 SSE: 0.0

```

Listagem 4.3: Fourier: Registro do treinamento fornecido pelo simulador JavaNNS

Tabela 4.7: Fourier: Testes com rejeições

Grupo	I	II	III	IV	V	Nenhum
I	2	9	8	2	1	17
II	--	6	1	1	2	4
III	--	6	10	2	1	9
IV	--	2	2	6	--	5
V	1	1	1	--	1	2

4.6. Transformada de Radon

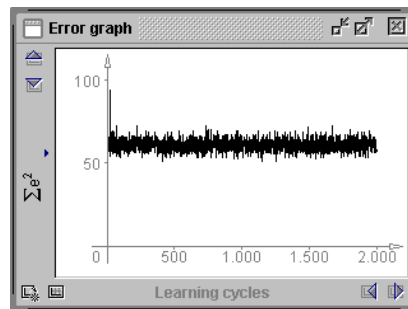
Para o treinamento inicial da rede, no classificador primário, para o JavaNNS, utilizou-se uma distribuição de pesos aleatória na inicialização, variando entre -5 a $+5$ (*random weights*). A atualização com ordem topológica (*topological order*) e aprendizado utilizando *Backpropagation* com $\eta=0,2$ e $d_{\max}=0,1$, épocas definidas com 2000 ciclos. Os padrões de treinamento de saída foram definidos como *Threshold*, com valor inferior (*lower*) de $-1,0$, superior (*upper*) de $+1,0$, centrado (*inner*) em $0,0$ e saída (*outer*) predefinida em $1,0$.

A Transformada de Radon apresentou um reconhecimento médio de $63,4 \pm 15,5\%$, e durante os testes de reconhecimento, obtiveram-se os resultados da Tabela 4.8 como matriz de confusão para o classificador primário, distribuídos em percentuais relativos a quantidade de símbolos por metaclassa. Para maiores detalhes, ver Apêndice E – Curvas de treinamento e testes de validação cruzada para a Transformada de Radon. A Curva de treinamento global pode ser vista na Figura 4.12, e o registro do treinamento na Listagem 4.4.

Tabela 4.8: Radon: Matriz de confusão (%)

Meta-classe	Resultado da análise (%)						
	I	II	III	IV	V	N.R.	Rec. médio
I	78,4	14,7	6,0	0,9	--	--	70,3 ± 10,5
II	8,0	68,0	14,0	--	--	10,0	56,7 ± 21,8
III	1,7	18,6	62,7	10,2	--	6,8	55,4 ± 14,5
IV	--	2,0	33,3	64,7	--	--	58,7 ± 19,4
V	--	--	10,3	10,3	79,4	--	71,2 ± 20,6

N.R. = Não reconhecido

Figura 4.12: Radon: Curva de aprendizagem apresentando o Erro \times Época

```

opened at: Mon Dec 15 12:39:37 BRST 2003
Step 200 SSE: 63.0
Step 400 SSE: 63.0
Step 600 SSE: 62.0
Step 800 SSE: 64.0
Step 1000 SSE: 59.0
Step 1200 SSE: 56.0
Step 1400 SSE: 59.0
Step 1600 SSE: 60.0
Step 1800 SSE: 59.0
Step 2000 SSE: 57.0

```

Listagem 4.4: Radon: Registro do treinamento fornecido pelo simulador JavaNNS

A Tabela 4.9 mostra os resultados dos testes de rejeição. Conseguiu-se, para a Transformada de Radon, a taxa de Rejeição de 20,6%, com o Erro calculado em 79,4%.

Tabela 4.9: Radon: Testes com rejeições

Grupo	I	II	III	IV	V	Nenhum
I	3	2	8	9	5	12
II	2	2	4	4	1	1
III	2	6	6	10	--	4
IV	2	2	4	3	2	2
V	--	--	--	2	2	2

4.7. Transformada de Zernike

Para o treinamento inicial da rede, no classificador primário, para o JavaNNS, utilizou-se uma distribuição de pesos aleatória na inicialização, variando entre -5 a $+5$ (*random weights*). A atualização com ordem topológica (*topological order*) e aprendizado utilizando *Backpropagation Momentum* com $\eta=0,2$, $d=0,1$, épocas definidas com 1000 ciclos. Os padrões de treinamento de saída foram definidos como *Threshold*, com valor inferior (*lower*) de $-1,0$, superior (*upper*) de $+1,0$, centrado (*inner*) em $0,0$ e saída (*outer*) predefinida em $1,0$.

A Transformada de Zernike apresentou um reconhecimento médio de $78,6 \pm 15,3\%$, e durante os testes de reconhecimento, obtiveram-se os resultados da Tabela 4.10 como matriz de confusão para o classificador primário, distribuídos em percentuais relativos a quantidade de símbolos por metaclasse. Para maiores detalhes, ver Apêndice F – Curvas de treinamento e testes de validação cruzada para a Transformada de Zernike. A Curva de treinamento global pode ser vista na Figura 4.13, e o registro do treinamento na Listagem 4.5.

Tabela 4.10: Zernike: Matriz de confusão (%)

Meta-classe	Resultado da análise (%)						Rec. Médio
	I	II	III	IV	V	N.R.	
I	94,8	--	--	1,7	3,5	--	84,9 ± 12,1
II	--	76,5	9,8	2,0	--	11,7	68,7 ± 17,3
III	1,7	1,7	89,7	--	1,7	5,2	81,0 ± 12,8
IV	3,9	--	2,0	78,4	15,7	--	68,3 ± 20,5
V	--	--	--	--	93,1	6,9	91,7 ± 6,0
N.R. = Não reconhecido							

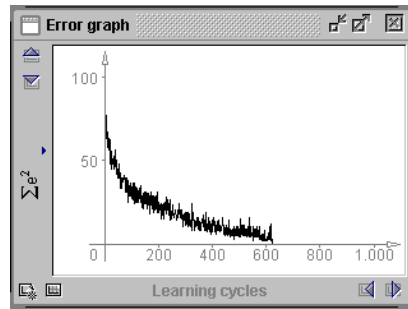


Figura 4.13: Zernike: Curva de aprendizagem apresentando o Erro x Época

opened at: Mon Dec 15 12:37:11 BRST 2003

Step 100 SSE: 31.0

Step 200 SSE: 23.0

Step 300 SSE: 16.0

Step 400 SSE: 8.0

Step 500 SSE: 4.0

Step 600 SSE: 2.0

Step 700 SSE: 0.0

Step 800 SSE: 0.0

Step 900 SSE: 0.0

Step 1000 SSE: 0.0

Listagem 4.5: Zernike: Registro do treinamento fornecido pelo simulador JavaNNS

A Tabela 4.11 mostra os resultados dos testes de rejeições. Conseguiu-se, para a Transformada de Zernike, a taxa de Rejeição de 40,2%, com o Erro calculado em 59,8%.

Tabela 4.11: Zernike: Testes com rejeições

Grupo	I	II	III	IV	V	Nenhum
I	2	9	8	2	1	17
II	--	6	1	1	2	4
III	--	6	10	2	1	9
IV	--	2	2	6	--	5
V	1	1	1	--	1	2

4.8. Transformadas de Zernike e Chebyshev

O agrupamento inicial (mesmo de forma visual) dos dados foi disposto para testes com as transformadas combinadas de Zernike e Chebyshev. As características, após extraídas, foram montadas para que fossem compostas com aproximadamente metade dos dados

compostos por cada transformada. Para que isso fosse possível, escolheu-se como a ordem de grandeza da transformada de Zernike a ordem 6, devolvendo 12 valores no domínio dos números complexos, que desmembrados geram um vetor de 24 valores reais. Para Chebyshev, por ser uma transformação no domínio dos números reais, gerou-se com ordem 5, devolvendo 25 elementos.

4.8.1. Classificador primário

Para o treinamento inicial da rede, no classificador primário, para o JavaNNS, utilizou-se uma distribuição de pesos aleatória na inicialização, variando entre -5 a $+5$ (*random weights*). A atualização com ordem topológica (*topological order*) e aprendizado utilizando *Backpropagation Momentum* com $\eta=0,2$, $d=0,1$, épocas definidas com 5000 ciclos. Os padrões de treinamento de saída foram definidos como *Threshold*, com valor inferior (*lower*) de $-1,0$, superior (*upper*) de $+1,0$, centrado (*inner*) em $0,0$ e saída (*outer*) predefinida em $1,0$.

As Transformadas associadas de Zernike e Chebyshev apresentaram um reconhecimento de 100%, e durante os testes de reconhecimento, obtiveram-se os resultados da Tabela 4.12 como matriz de confusão para o classificador primário, distribuídos em percentuais relativos a quantidade de símbolos por metaclasses. Para maiores detalhes, ver Apêndice G – Curvas de treinamento e testes de validação cruzada para as Transformadas de Zernike e Chebyshev. A Curva de treinamento global pode ser vista na Figura 4.14, e o registro do treinamento na Listagem 4.6.

Tabela 4.12: Zernike/Chebyshev: Matriz de confusão (%)

Meta-classe	Resultado da análise (%)						
	I	II	III	IV	V	N.R.	Rec. médio
I	100	--	--	--	--	--	100
II	--	100	--	--	--	--	100
III	--	--	100	--	--	--	100
IV	--	--	--	100	--	--	100
V	--	--	--	--	100	--	100
N.R. = Não reconhecido							

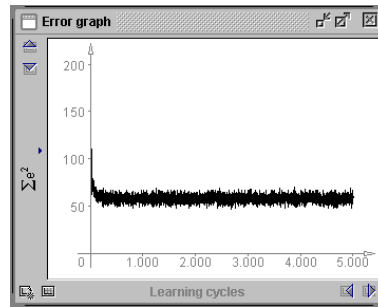


Figura 4.14: Zernike/Chebyshev: Curva de aprendizagem apresentando o Erro \times Época

```

opened at: Tue Sep 02 10:29:52 BRT 2003
Step 500 SSE: 72.0
Step 1000 SSE: 71.0
Step 1500 SSE: 70.0
Step 2000 SSE: 67.0
Step 2500 SSE: 70.0
Step 3000 SSE: 65.0
Step 3500 SSE: 63.0
Step 4000 SSE: 61.0
Step 4500 SSE: 65.0
Step 5000 SSE: 60.0

```

Listagem 4.5: Zernike/Chebyshev: Registro do treinamento fornecido pelo simulador JavaNNS

A Tabela 4.13 mostra os resultados dos testes de rejeição. Conseguiu-se, para as Transformadas de Zernike e Chebyshev, a taxa de rejeição de 65,7%, com o Erro calculado em 34,3% .

Tabela 4.13: Zernike/Chebyshev: Testes com rejeição

Grupo	I	II	III	IV	V	Nenhum
I	4	2	3	--	--	30
II	--	5	--	--	--	9
III	--	4	7	3	--	14
IV	--	3	--	3	--	9
V	--	1	--	--	--	5

4.8.2. Classificadores secundários

Para os classificadores secundários, baseados em redes Hebbianas e MLP-BP, efetuou-se o treinamento da seguinte forma: todos os dados pertencentes àquela rede foram introduzidos, para que a rede monte os seus espaços dimensionais. Após treinada a rede, a

mesma base de dados é utilizada nos testes. Observa-se que para esta estrutura de redes hebbianas, não existe a filosofia de erro de treinamento, mesmo que supervisionada. Montam-se então as tabelas de reconhecimentos dos símbolos pelos classificadores secundários, dispostos como mostra a Tabela 4.14.

Tabela 4.14: Zernike/Chebyshev: Relação de quantidade de símbolos por metaclasses

Meta-classe	I	II	III	IV	V
Quantidade de símbolos	116	50	60	46	28

Para os testes com os classificadores secundários, utilizou-se a seguinte distribuição dos dados em cada metaclasses, como mostra a Tabela 4.15.

Tabela 4.15: Zernike/Chebyshev: Distribuição dos elementos por metaclasses

Metaclasse	Saída do classificador secundário (em elementos)											Total
	1	2	3	4	5	6	7	8	9	10	11	
I	62	54										116
II	17	6	4	1	5	8	1	1	2	3	2	50
III	7	17	4	19	7	6						60
IV	12	18	16									46
V	17	11										28

A Tabela 4.16 mostra as saídas do classificador secundário (funcionando como memória) da metaclasses 1, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclasses foi de 95,7%, com perdas na identificação do primeiro grupo de símbolos (autotransformadores sem comutadores de TAP – Terminal de Ajuste de Potencial). Para a Rede Hebbiana, utilizou-se o treinamento com 5000 épocas, $\eta=0,0001$ e $w_{\max}=1,0$ (todos os pesos normalizados com o máximo em 1).

Tabela 4.16: Zernike/Chebyshev: Saída do classificador secundário 1, (%)

Saída	1	2	N.R.
1	91,9	--	8,1
2	--	100	--
N.R. = Não reconhecido			

A Tabela 4.17 mostra as saídas dos classificadores secundários, da metaclasses 2, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclasses foi de 84,0%, com perdas parciais na identificação do segundo grupo de símbolos (aterramentos) e décimo grupo (resistores), e com perdas totais no reconhecimento no sétimo

grupo (dispositivos especiais de painéis – Dispositivo20, no Apêndice A) e no nono grupo (Disjuntor08 e Disjuntor09, no Apêndice A). A curva de erro de treinamento é apresentada na Figura 4.15 para a rede neural MLP-BP utilizada com 5000 épocas e $\eta=0,0001$ e atualização topológica.

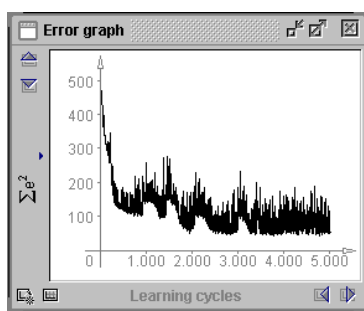


Figura 4.15 Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 2

Tabela 4.17: Zernike/Chebyshev: Saída do classificador secundário 2, (%)

Saída	1	2	3	4	5	6	7	8	9	10	11	N.R.
1	100	--	--	--	--	--	--	--	--	--	--	--
2	--	50,0	--	--	50,0	--	--	--	--	--	--	--
3	--	--	100	--	--	--	--	--	--	--	--	--
4	--	--	--	100	--	--	--	--	--	--	--	--
5	--	--	--	--	100	--	--	--	--	--	--	--
6	--	--	--	--	--	100	--	--	--	--	--	--
7	--	--	--	--	--	--	--	--	--	--	--	100
8	--	--	--	--	--	--	--	100	--	--	--	--
9	--	--	--	--	--	--	--	--	--	--	--	100
10	--	--	--	--	--	--	--	--	--	66,7	--	33,3
11	--	--	--	--	--	--	--	--	--	--	100	--

N.R. = Não reconhecido

A Tabela 4.18 mostra as saídas dos classificadores secundários, da metaclasse 3, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclasse foi de 90,0%, com perdas parciais na identificação do segundo grupo de símbolos (chaves seccionadoras), terceiro grupo (ilhas de conexões), e no quinto grupo (disjuntores, tipo arco – Disjuntor06 e 07, no Apêndice A). A curva de erro de treinamento é apresentada na Figura 4.16 para a rede neural MLP-BP utilizada com 5000 épocas e $\eta=0,002$ e atualização topológica.

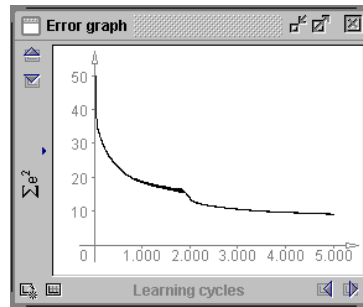


Figura 4.16: Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 3

Tabela 4.18: Zernike/Chebyshev: Saída do classificador secundário 3, (%)

Saída	1	2	3	4	5	6	N.R.
1	100	--	--	--	--	--	--
2	--	94,1	--	5,9	--	--	--
3	--	--	50,0	--	--	--	50,0
4	--	--	--	100	--	--	--
5	--	--	--	14,3	57,1	--	28,6
6	--	--	--	--	--	100	--
N.R. = Não reconhecido							

A Tabela 4.19 mostra as saídas dos classificadores secundários, da metaclasse 4, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclasse foi de 64,7%, com perdas parciais na identificação no primeiro grupo de símbolos (confundindo transformadores sem TAP com os que contém TAP), segundo grupo (reconhecendo transformadores com TAP que não contém comutadores) e no terceiro grupo (identificando TCs como transformadores com TAP). A curva de erro de treinamento é apresentada na Figura 4.17 para a rede neural MLP-BP *with momentum*, utilizada com 5000 épocas, $\eta=0,0001$, $\mu=0,5$, $c=0,1$ e $d_{\max}=01$ com atualização topológica.

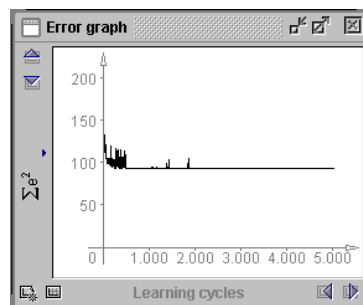


Figura 4.17: Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 4

Tabela 4.19: Zernike/Chebyshev: Saída do classificador secundário 4, (%)

Saída	1	2	3	NR
1	25,0	75,0	--	--
2	5,6	94,4	--	--
3	--	18,7	81,3	--
N.R. = Não reconhecido				

A Tabela 4.20 mostra as saídas dos classificadores secundários, da metaclasse 5, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclasse foi de 82,2%, com perdas parciais na identificação no primeiro grupo de símbolos (confundindo transformadores sem TAP com os que contém TAP). A curva de erro de treinamento é apresentada na Figura 4.18 para a rede neural MLP-BP *with momentum*, utilizada com 40000 épocas, $\eta=0,0001$, $\mu=0,5$, $c=0,1$ e $d_{\max}=01$ com atualização topológica.

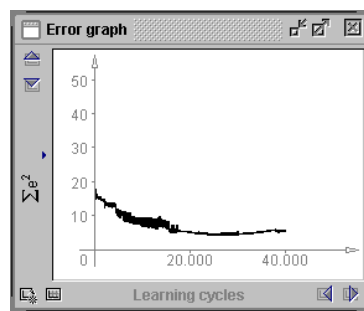


Figura 4.18: Zernike/Chebyshev – Curva de erro de treinamento para o classificador secundário 5

Tabela 4.20: Zernike/Chebyshev: Saída do classificador secundário 5, (%)

Saída	1	2	N.R.
1	70,6	29,4	--
2	--	100	--
N.R. = Não reconhecido			

4.9. Transformadas de Zernike e Fourier

Utilizou-se o mesmo agrupamento dos dados dos testes das transformadas combinadas de Zernike e Chebyshev. As características, após extraídas, foram montadas para que fossem compostas com aproximadamente metade dos dados compostos por cada transformada. Para que isso fosse possível, escolheu-se como a ordem de grandeza da transformada de Zernike a ordem 6, devolvendo 12 valores no domínio dos números complexos, que desmembrados geram um vetor de 24 valores reais. Para Fourier, embora seja uma transformada no domínio

dos complexos, utilizou-se somente o módulo da transformada, gerou-se com ordem 5, devolvendo 25 elementos reais.

4.9.1. Classificador primário

Para o treinamento inicial da rede, no classificador primário, para o JavaNNS, utilizou-se uma distribuição de pesos aleatória na inicialização, variando entre -5 a $+5$ (*random weights*). A atualização com ordem topológica (*topological order*) e aprendizado utilizando *Backpropagation Momentum* com $\eta=0,0002$, $d=0,1$, épocas definidas com 50000 ciclos. Os padrões de treinamento de saída foram definidos como lineares (*Identify*), com valor centrado inicialmente em 0, e bias ajustado em zero.

As Transformadas associadas de Zernike e Fourier apresentaram um reconhecimento médio ponderado de $96,8 \pm 1,94\%$, e durante os testes de reconhecimento, obtiveram-se os resultados da Tabela 4.21 como matriz de confusão para o classificador primário, distribuídos em percentuais relativos a quantidade de símbolos por metaclasse. Para maiores detalhes, ver Apêndice H – Curvas de treinamento e testes de validação cruzada para as Transformadas de Zernike e Fourier. A Curva de treinamento global pode ser vista na Figura 4.19, e o registro do treinamento na Listagem 4.6. O caráter de “alisamento” da curva de erro deve-se ao pequeno passo de aprendizagem (η) utilizado. Pelos testes realizados, estes valores baixos, além de manter mais estável a aprendizagem (evitando ruídos na curva de treinamento), aumenta a capacidade discriminatória da rede, sob pena de se impor altos valores de ciclos de aprendizagem (épocas).

Tabela 4.21: Zernike/Fourier: Matriz de confusão (%)

Meta-classe	Resultado da análise (%)						
	I	II	III	IV	V	N.R.	Rec. médio
I	100	--	--	--	--	--	100
II	2,0	88,2	7,8	--	--	2,0	94,9 ± 2,0
III	--	--	91,4	6,7	1,9	--	94,7 ± 3,9
IV	2,0	--	2,0	94,1	--	1,9	94,0 ± 3,2
V	--	--	--	3,4	96,6	--	96,7 ± 3,3
N.R. = Não reconhecido							

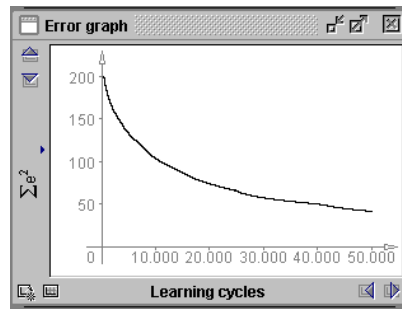


Figura 4.19: Zernike/Fourier: Curva de aprendizagem apresentando o Erro x Época

opened at: Mon Jan 19 15:06:06 BRST 2004

Step 5000 SSE: 132.52549743652344
 Step 10000 SSE: 103.19200897216797
 Step 15000 SSE: 86.33396911621094
 Step 20000 SSE: 73.4946060180664
 Step 25000 SSE: 65.19180297851562
 Step 30000 SSE: 56.994667053222656
 Step 35000 SSE: 52.900150299072266
 Step 40000 SSE: 49.696407318115234
 Step 45000 SSE: 44.42436218261719
 Step 50000 SSE: 41.171634674072266

Listagem 4.6: Zernike/Fourier: Registro do treinamento fornecido pelo simulador JavaNNS

A Tabela 4.22 mostra os resultados dos testes de *outliers*. Conseguiu-se, para as Transformadas de Zernike e Fourier, a taxa de rejeição de 61,8%, com o Erro calculado em 38,2%.

Tabela 4.22: Zernike/Fourier: Testes com *outliers*

Grupo	I	II	III	IV	V	Nenhum
I	1	4	2	1	2	29
II	1	4	1	--	--	8
III	3	2	7	--	3	13
IV	2	--	2	2	--	9
V	--	--	--	2	--	4

4.9.2. Classificadores secundários

Para os classificadores secundários, baseados em redes MLP-BP, efetuou-se o treinamento da seguinte forma: todos os dados pertencentes àquela rede foram introduzidos,

para que a rede monte os seus espaços dimensionais. Após treinada a rede, a mesma base de dados é utilizada teste. A rede Hebbiana não conseguiu apresentar durante os testes resultados satisfatórios, necessitando-se substituí-la pelo modelo mostrado na Figura 4.20 – Rede MLP-BP, com as saídas interligadas a entrada, além da camada escondida. A estrutura da rede consiste em 22 neurônios nas camadas escondidas⁸ e 2 neurônios de saída, ambos com função de ativação logarítmica e saídas lineares. Para treinamento, utilizou-se o algoritmo de treinamento *Backpropagation with momentum*, com 20000 épocas e passo de treinamento de 0,0002 para garantir a convergência do modelo em uma base de dados não separáveis linearmente. Aproveitam-se então as estruturas das tabelas de reconhecimentos dos símbolos pelos classificadores secundários, gerados para os testes com as transformadas de Zernike e Chebyshev (Tabela 4.14 para visualizar a quantidade de elementos por metaclassa e Tabela 4.15 para os testes com os classificadores secundários).

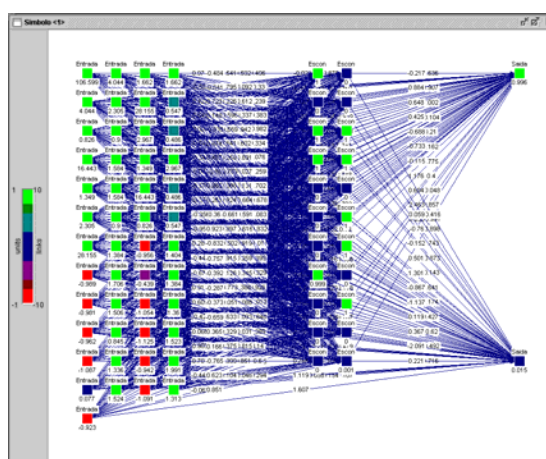


Figura 4.20 – Modelo substitutivo para o classificador secundário – metaclassa 1

A Tabela 4.23 mostra as saídas dos classificadores secundários, da metaclassa 1, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclassa foi de 100%, sem perdas no reconhecimento e nem confusão entre os símbolos sob análise. A curva de erro de treinamento é apresentada na Figura 4.21 para a rede neural MLP-BP *with momentum*, utilizada com 20000 épocas, $\eta=0,0002$, $\mu=0,5$, $c=0,1$ e $d_{\max}=01$ com atualização topológica.

⁸ Obtida através da relação custo/benefício, de onde se obteve o limiar de 95% da eficiência da rede, considerada como aceitável, a partir do mesmo número de neurônios da camada de entrada aplicados à camada intermediária (ou escondida).

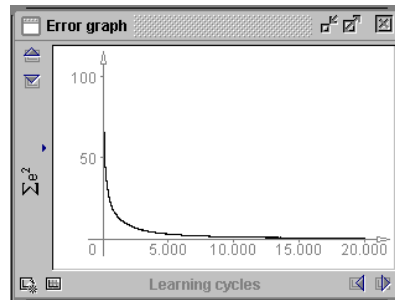


Figura 4.21: Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 1

Tabela 4.23: Zernike/Fourier: Saída do classificador secundário 1, (%)

Saída	1	2	N.R.
Rec.	100	100	--
Rec. = Reconhecido		N.R. = Não reconhecido	

A Tabela 4.24 mostra as saídas dos classificadores secundários, da metaclassa 2, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclassa foi de 100%, sem perdas no reconhecimento e nem confusão entre os símbolos sob análise. A curva de erro de treinamento é apresentada na Figura 4.22 para a rede neural MLP-BP *with momentum*, utilizada com 20000 épocas, $\eta=0,0002$, $\mu=0,5$, $c=0,1$ e $d_{\max}=01$ com atualização topológica.

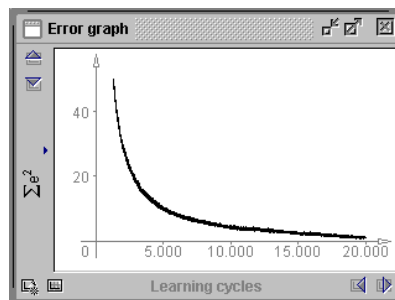


Figura 4.22: Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 2

Tabela 4.24: Zernike/Fourier: Saída do classificador secundário 2, (%)

Saída	1	2	3	4	5	6	7	8	9	10	11	N.R.
Rec.	100	100	100	100	100	100	100	100	100	100	100	--
Rec. = Reconhecido						N.R. = Não reconhecido						

A Tabela 4.25 mostra as saídas dos classificadores secundários, da metaclassa 3, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclassa foi de 100%, sem perdas no reconhecimento e nem confusão entre os símbolos sob análise. A curva de erro de treinamento é apresentada na Figura 4.23 para a rede neural MLP-

BP *with momentum*, utilizada com 20000 épocas, $\eta=0,0002$, $\mu=0,5$, $c=0,1$ e $d_{\max}=01$ com atualização topológica.

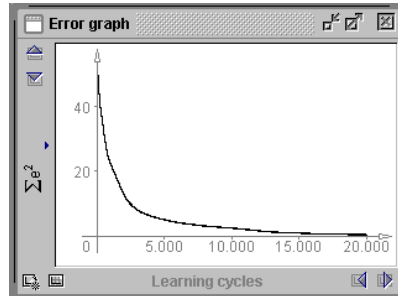


Figura 4.23: Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 3

Tabela 4.25: Zernike/Fourier: Saída do classificador secundário 3, (%)

Saída	1	2	3	4	5	6	N.R.
Rec.	100	100	100	100	100	100	--
Rec. = Reconhecido				N.R. = Não reconhecido			

A Tabela 4.26 mostra as saídas dos classificadores secundários, da metaclasse 4, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da metaclasse foi de 100%, sem perdas no reconhecimento e nem confusão entre os símbolos sob análise. A curva de erro de treinamento é apresentada na Figura 4.24 para a rede neural MLP-BP *with momentum*, utilizada com 20000 épocas, $\eta=0,0002$, $\mu=0,5$, $c=0,1$ e $d_{\max}=01$ com atualização topológica.

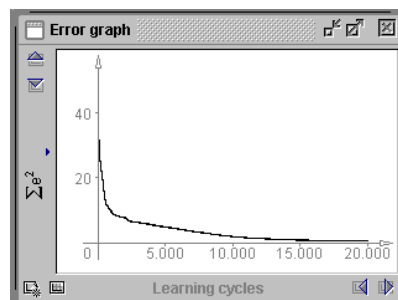


Figura 4.24: Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 4

Tabela 4.26: Zernike/Fourier: Saída do classificador secundário 4, (%)

Saída	1	2	3	N.R.
Rec.	100	100	100	--
Rec. = Reconhecido		N.R. = Não reconhecido		

A Tabela 4.27 mostra as saídas dos classificadores secundários, da metaclasse 5, indicando o grau de acerto (%) do símbolo por saída, para cada saída. O reconhecimento da

metaclassa foi de 100%, sem perdas no reconhecimento e nem confusão entre os símbolos sob análise. A curva de erro de treinamento é apresentada na Figura 4.25 para a rede neural MLP-BP *with momentum*, utilizada com 20000 épocas, $\eta=0,0002$, $\mu=0,5$, $c=0,1$ e $d_{\max}=01$ com atualização topológica.

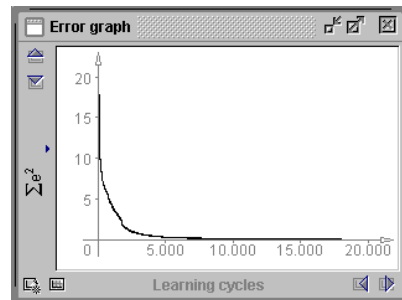


Figura 4.25: Zernike/Fourier – Curva de erro de treinamento para o classificador secundário 5

Tabela 4.27: Zernike/Fourier: Saída do classificador secundário 5, em percentual

Saída	1	2	NR
Rec.	100	100	--
Rec. = Reconhecido		N.R. = Não reconhecido	

4.10. Comentários finais

Para o classificador primário, a análise comparativa está listada com o erro obtido nos modelos implementados, como na Tabela 4.28, comparando-se os diversos resultados atingidos, observa-se o desempenho superior das transformadas combinadas, referente à classificação e discriminação seletiva dos dados. Conseqüentemente, conclui-se que as transformadas combinadas são os melhores candidatos a um classificador baseado em processos estatísticos, devido às suas combinações de momentos matemáticos distintos.

Tabela 4.28: Comparação com os resultados obtidos para rejeições, (%)

Transformada	Rejeição	Erro
Chebyshev	24,5	75,5
Discreta do Coseno	23,6	76,4
Fourier	36,3	63,7
Radon	20,6	79,4
Zernike	40,2	59,8
Zernike+Chebyshev	65,7	34,3
Zernike+Fourier	61,8	38,2

Como uma análise inicial, pode-se selecionar as transformadas combinadas de Zernike e Chebyshev, que apresentaram uma capacidade de rejeição de 65,7% ou $(102 * 0,657) = 67$ símbolos, e Zernike e Fourier, que apresentaram uma capacidade de rejeição de 61,8% ou $(102 * 0,618) = 63$ símbolos.

Mas para que se tenha uma avaliação mais precisa, deve-se considerar o resultado dos classificadores secundários, em relação aos símbolos reconhecidos. Para os classificadores secundários, a Tabela 4.29 apresenta um resultado comparativo entre os modelos combinados de transformadas.

Pelas restrições de normas de desenhos [ABNTa-d], para alguns símbolos só há uma representação gráfica, e pelo protocolo da base de dados utilizada, fixando os comprimentos dos terminais em 10% da sua altura (ou largura, dependendo da disposição), não se permitiu variações gráficas no desenho, assim utilizou-se as redes neurais secundárias como memória.

Tabela 4.29: Comparação com os resultados globais obtidos pelas transformadas combinadas

Meta-classe	I	II	III	IV	V	$\Sigma(\text{Rec})$
Quantidade de símbolos	116	50	60	50	28	304
Zernike/Chebyshev (Rec)	111	43	54	37	26	271
Zernike/Fourier (Rec)	116	44	55	47	27	289

Considerando então os resultados obtidos pela transformadas combinadas de Zernike/Chebyshev, obtemos uma perda de $(304 - 271) = 33$ símbolos não identificados, e pelas transformadas combinadas de Zernike/Fourier uma perda de $(304 - 289) = 15$ símbolos não classificados corretamente.

Pode-se considerar então, que para esta aplicação, a utilização combinada das transformadas de Zernike e Fourier obtiveram um desempenho superior às demais, mesmo com perdas no classificador primário.

Os resultados podem ser melhorados utilizando-se inicialmente um método de agrupamento não supervisionado para a definição das metaclasses (como por exemplo k-means, KNN, Bayes, etc...). Os testes efetuados serviram para demonstrar a eficiência das extrações de características baseadas em métodos estatísticos.

Os desenhos utilizados foram extraídos de desenhos técnicos em engenharia de plantas elétricas industriais, e convertidos do padrão de arquivo vetorial para o padrão de *bitmap*. Realizou-se a segmentação manual, e após feitas as segmentações, os símbolos foram classificados manualmente formando metaclasses em função das suas semelhanças visuais

(em primeira ordem), seguido da sua funcionalidade (em segunda ordem). Criou-se uma base de dados com as representações gráficas definidas pelas normas ABNT [ABNTa-d], e definiu-se uma métrica para garantir uma possível interface com um sistema de segmentação, adotando-se o comprimento dos terminais em 10% da altura (ou largura, conforme o caso). Para que seja possível a utilização das transformadas, necessitou-se de um procedimento que exige o pré-processamento da imagem, com reduções de cores, limiarização e conversão da imagem para o formato binário, invertendo-se os valores numéricos dos *pixels* ($0 \Leftrightarrow 1$). Esta conversão reduziu os valores numéricos operados pelas transformadas, conseqüentemente reduzindo-se os módulos dos valores obtidos pelas transformadas. Além destes cuidados prévios com as imagens, normalizou-se o resultado obtido pelas transformadas para atenuar os problemas com a escala e espessura do traçado, embora a matriz resultante da transformada mude com alguns valores, mas permanece a influência da PCA (*Principal Component Analysis* – Análise pela Componente Principal), assegurando o reconhecimento do símbolo. Extraídas as características, o vetor resultante é aplicado a uma rede neural hierárquica, para treinamento e/ou reconhecimento, identificando-se a que classe pertence. A camada superior da hierarquia formada pelas redes neurais, realiza um papel de chave seletora, identificando qual rede deverá ser ativada para o reconhecimento do símbolo (vetor característico, no caso). A rede selecionada (classificador secundário), faz o papel de identificar o símbolo (ou o vetor) na sua correta família, dentro da sua metaclasses.

Deve-se citar também, o cuidado a ser tomado com as normalizações dos dados, diminuindo as suas faixas de domínio (variâncias) ou então expandindo-as, conforme o caso. Este procedimento deve ser feito para evitar erros de processamento numérico (*overflow*) e diminuir a carga de processamento dos classificadores para os cálculos dos pesos durante o treinamento. Observou-se também que, para as transformadas no domínio dos números complexos, o processamento da normalização é mais simples que as transformadas no domínio dos números reais. Isto porque, torna-se fácil, em um domínio complexo, extrair a raiz quadrada ou o logaritmo de um número negativo, o que não ocorre no domínio dos reais.

Partindo do que até aqui foi exposto e analisado, este trabalho serve de embasamento para o próximo capítulo, onde é feita uma conclusão sobre a presente pesquisa, cumprindo com os objetivos propostos no início do trabalho.

Capítulo 5

Conclusão

Este trabalho representa um modelo de reconhecimento dos símbolos por métodos estatísticos usando transformadas numéricas, trabalhando isoladas e também trabalhando em paralelo (agrupadas). Observa-se que as transformadas associadas apresentaram um desempenho superior às isoladas, mesmo com as suas dimensões reduzidas⁹ para formarem um vetor concatenado com a mesma grandeza. Durante os experimentos realizados com as transformadas agrupadas (Zernike/Chebyshev e Zernike/Fourier), observou-se que para os classificadores secundários os dados tornaram-se não linearmente separáveis¹⁰ (com sobreposições de classes no mesmo hiper-espço), necessitando de uma rede discriminatória mais elaborada (complexa).

Os testes com as transformadas do Coseno e Radon apresentaram variações quanto a rotação nos seus respectivos vetores característicos, desmotivando trabalhos com associações envolvendo as demais transformadas. Para Chebyshev, esta dificuldade pôde ser amenizada pela forma de extração de características pela divisão da imagem por quadrantes simétricos. Fourier e Zernike apresentaram uma tolerância maior quanto à rotação. No referente à escala e espessura do traçado, durante a normalização, preservaram-se as principais características dos vetores (Componente pela Análise Principal), auxiliando na tarefa de reconhecimento.

Com as saídas das transformadas concatenadas, observou-se um desempenho melhor, como mostrado nos gráficos dos experimentos efetuados, na tentativa de se conseguir obter a

⁹ Por exemplo: Para a Transformada de Zernike, quando operando isoladamente, utilizou-se uma matriz de 8ª ordem para se obter um vetor 49 elementos, e quando associada, utilizou-se de 5ª ordem para se obter um vetor 24 elementos a ser concatenado (aproximadamente metade do vetor característico).

¹⁰ Para os classificadores secundários das metaclasses II, III, IV e V das transformadas associadas de Zernike e Chebyshev e todos os classificadores secundários utilizados pelas Transformadas de Zernike e Fourier.

melhor das características (maiores variâncias dos elementos em cada vetor) dos dados. Os índices atingidos foram de 100% no classificador primário para Zernike e Chebyshev e 89,1% no classificador primário para Zernike e Fourier. As transformadas combinadas de Zernike e Fourier, embora apresentem um resultado pior no classificador primário, apresentaram um desempenho melhor quanto a classificação global dos símbolos (classificador primário + classificadores secundários), com perdas de reconhecimento de 15 símbolos (ou $15 \div 304 = 0,049 \Rightarrow 4,9\%$) contra os 33 símbolos (ou $33 \div 304 = 0,109 \Rightarrow 10,9\%$) das transformadas combinadas de Zernike e Chebyshev. Considerando-se as aplicações de símbolos *outliers*, observa-se que as transformadas combinadas de Zernike e Fourier obtiveram um desempenho superior, apresentando uma rejeição de ($34,3\% \times 102 \text{ símbolos} =$) 35 símbolos, contra a transformada de Zernike e Chebyshev, com rejeição de ($38,2\% \times 102 \text{ símbolos} =$) 39 símbolos. Mesmo considerando uma menor sensibilidade aos símbolos “*outliers*”, as transformadas associadas de Zernike e Fourier obtiveram um melhor desempenho global.

Os fatores positivos da metodologia estão na utilização de abordagens hierárquicas do problema definindo-os em camadas, e pelo uso de algoritmos estatísticos de extração de características dos símbolos para redução do montante de informações à discriminar para o reconhecimento, otimizando assim a região de decisão.

Como ponto negativo da metodologia, deve-se observar a tolerância das transformadas aos símbolos “*outliers*”, conforme cita Perpiñán [PER97]. A Transformada de Zernike, pela sua definição de ser radial e invariante à rotação, não é capaz de identificar se o segmento de reta pertencente ao desenho está a 30° graus ou a 120° para o mesmo desenho. O mesmo acontece com os polinômios cartesianos de Chebyshev, aonde a transformada não é capaz de identificar a que quadrante pertence o segmento de reta, porquê na sua implementação, para que se torne invariante à rotação, a transformada final é a soma matricial dos elementos resultantes das transformações (soma das transformadas de cada quadrante). Para a Transformada de Fourier, deve-se dividir a técnica de normalização em duas partes, devida a componente fundamental. Esta componente altera o seu valor em função da escala e espessura do traçado, por isso este tratamento individualizado. Conseqüentemente, a transformada perderá a sua eficiência em desenhos com ordens de grandeza superiores a 100 vezes ao tamanho dos símbolos utilizados para testes. Para as Transformadas de Radon e DCT, deve-se restringir os tamanhos dos símbolos sob análise á uma faixa de escalas (como por exemplo de

0,1x até 10x o tamanho do símbolo), devido às próprias características intrínsecas dos modelos (devido ao passo cosenoidal da DCT e da rotação de histogramas de Radon).

Como conclusão final, os testes realizados demonstraram uma maior capacidade discriminatória das transformadas associadas de Zernike e Fourier, com perdas no classificador primário, mas com ganhos na classificação secundária (apresentou menos perdas de reconhecimento dos símbolos).

Pode-se citar como contribuições originais deste trabalho:

- a) Criação de uma metodologia para reconhecimento de símbolos em engenharia elétrica, como passo inicial para confecção de uma nova ferramenta para análises de circuitos, como topologias e determinação de fluxos de potências em diagramas elétricos (com informações auxiliares provenientes dos sistemas de automação existentes),
- b) Um padrão de referências utilizando-se de métodos estatísticos em diagramas unifilares (elétricos, hidro-pneumáticos, etc.), baseados nos comparativos de desempenho das transformadas numéricas,
- c) Interface de trabalhos com pesquisas em processos de segmentação e interpretação de desenhos técnicos, e
- d) Criação de uma base de dados de símbolos gráficos no padrão ABNT, para o contexto de Engenharia Elétrica.

Como trabalhos futuros, pode-se desenvolver novas metodologias de segmentação de imagem, para extrair os símbolos a partir de um desenho unifilar, e formatá-lo com um protocolo compatível com a base de dados especificada neste trabalho (imagem binária, com os comprimentos dos terminais definidos em 10% de sua dimensão transversal), para possibilitar uma nova metodologia que, integrada ao algoritmo de segmentação possa interpretar o desenho e auxiliar as equipes de manutenção preventiva em estabelecer agendas de serviços em plantas industriais, ou ainda auxiliar as equipes de manutenção corretiva na determinação de onde ocorreu o evento que necessite a intervenção (como detecção de fusíveis em bancos de capacitores ou mecanismos de comutação de TAP em transformadores).

Ainda como trabalhos futuros, no referente a identificação dos símbolos, pode-se criar um outro modelo classificatório baseado em Redes Neurais Artificiais com as transformadas associadas em paralelo (como no presente trabalho), mas com duas camadas intermediárias

isoladas, como por exemplo: a entrada da Transformada de Zernike interligada a uma camada intermediária (escondida) e a Transformada de Fourier interligada a outra camada intermediária (escondida), isoladas entre si, mas ambas interligadas com as saídas.

Referências¹¹

- [ABNTa] Norma ABNT, *Símbolos Gráficos para Produção e Conversão de Energia Elétrica*, NBR-12522, Abril de 1992.
- [ABNTb] Norma ABNT, *Símbolos Gráficos para Transmissão e Distribuição de Energia Elétrica*, NBR-12523, Abril de 1992.
- [ABNTc] Norma ABNT, *Relés de Medição com uma Grandeza de Alimentação de Entrada a Tempo Dependente Especificado*, NBR-7099, Dezembro de 1991.
- [ABNTd] Norma ABNT, *Desempenho dos Contatos dos Relés Elétricos*, NBR-7098, Dezembro de 1981.
- [BAR03] BARRETO, A. S., *Introdução às Transformadas de Radon*, Symposium on Spectral and Scattering Theory Recife, Brazil, August 11, 2003.
- [BAN03] BARANIDHARAN, R., IOERGER, T. R., *Enhancing Learning using Feature and Example selection*, Journal of Machine Learning Research, 2003
- [BOY00] BOYD, JOHN P., *Chebyshev and Fourier Spectral Methods - Second Edition*, 2000 - DOVER Publications, Inc.
- [CAS01] CASTLEMAN, K. R., *Digital Image Processing, - Pattern Recognition: Image Segmentation*, 2001/00 - Addison Wesley Publishing Company.
- [CHE93] CHENG, T., KHAN, J., Liu H., YUN, D. Y. Y., *A symbol Recognition System*, Electrical Engineering Department, University of Hawaii, 1993 – IEEE.
- [CHH00] CHHABRA, A. K., PHILLIPS, I. T., *Performance Evaluation of Line Drawing Recognition Systems - International Conference on Pattern Recognition (ICPR'00) - Volume 4*, pp 4864, September 03 - 08, 2000 - Barcelona, Spain
- [DAV93] DAVIS, M. H. A., *Markov Models and Optimization*, Monographs on Statistics and Applied Probability No. 49, Chapman & Hall, 1993
- [DIL96] DILLON, THARAN S., *Neural Networks Applications in Power Systems*, CRL Publishing Ltd, 1996, United Kingdom
- [DIM00] DIMACS – DISCRETE MATHEMATICS AND THEORETICAL COMPUTER SCIENCE, *REU 2000 Project: Wavelets for local tomographic medical imaging*, Center for Discrete Mathematics & Theoretical Computer Science, Rutgers and Princeton Universities
- [DIN02] DING, C., *Adaptive dimension reduction for clustering high dimensional data*, IEEE International Conference on Data Mining (ICDM'02), December 09, 2002 Maebashi City, Japan
- [FAC02] FACON, J., *Processamento de Imagens*, 2002, apostila de aula, <http://www.ppgia.pucpr.br/~facon/IndexPrincipalBrMaterialMestrado.htm>
- [FAC96] FACON, J., *Morfologia Matemática: Teoria e Exemplos*, 1996, Editora Champagnat, Curitiba
- [FAU94] FAUSETT, L., *Fundamentals of Neural Networks – Architecture, algorithms and applications*, 1994 – Prentice-Hall.

¹¹ Este modelo segue o estilo “alpha” do sistema LaTeX, que permite memorizar melhor as referências durante a leitura do texto e é bem mais compacto que o proposto pela ABNT.

- [FER86] FERRATÉ, G., PAVLIDIS, T., SANFELIU, A., BUNKE H., *Syntatic and Structural Pattern Recognition*, Springer-Verlag, 1986, NATO ASI Series F: Computer and Systems Sciences, Vol. 45.
- [FRE92] FREEMAN, J. A., *Neural Networks – Algorithms, Applications and Programming Techniques*, 1992 -Addison Wesley Co.
- [FUM00] FUMERA, G., ROLI, F., GIACINTO, G., *Reject option with multiple thesholds*, Pattern Recognition, issue 33, 2000.
- [GOS96] GOSE, E., JOHNSONBRUGH, R., JOST, S., *Pattern Recognition and Image Analysis*, Prentice-Hall, 1996
- [GRAN92] GRANVILLE, W. A.. *Elementos de cálculo integral e diferencial*, Ed. ÂMBITO CULTURAL – Rio de Janeiro, 1992.
- [HAL00] HALL, MARK A., *Correlation-based feature selection for discrete and numeric class machine learning*, Proceedings of the Seventeenth International Conference on Machine Learning, 2000, Stanford University, CA. Morgan Kaufmann Publishers.
- [HUS02] *Orthogonal Complex Zernike Moments*, acessada na internet em 19/04/2002, em http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/SHUTLER/node1.html, Edinburgh University/Artificial Intelligence.
- [JAE88] JAE, S. L., OPPENHEIM, A. V., *Advanced Topics in Signal Processing*, 1988 – Prentice Hall
- [JIS00] JISHENG, L., CHALANA, V., PHILLIPS, T., HARALICK, R. M., *A Methodology for Special Symbol Recognitions*, 4011-4014, International Conference on Pattern Recognition (ICPR'00), 2000, Barcelona, Spain, Volume IV. IEEE Computer Society, Session O4.1A. OCR and Document Analysis
- [KAR00] KARABOGA, D., PHAM, D. T., *Intelligent Optimisation Techniques*, Springer Verlag London, 2000
- [KAS90] KASTURI, R., TRIVERDI, M. M., *Image Analysis Applications*, 1990 – Marcel Dekker Inc.
- [KEI00] KEIJI G., *Optimization-based Image analysis dealing with symbolic constraints using hierarchical multi-agent system*, 2000, IEEE.
- [KIN03] KINGSTON, A., *Adaptive Discrete Radon Transforms for Greyscale Images*, 2003, Monash University, Australia
- [LAY02] LAY, DAVID C., *Linear Algebra and its Applications - Third Edition*, 2002 - Pearson Addison Wesley
- [LIA00] LIANG, J., PHILLIPS, I. T., CHALANA, V., HARALICK, R., *A Methodology for Special Symbol Recognition*, 2000, Proceedings of the 15th International Conference on Pattern Recognition (ICPR-00), Barcelona, Spain
- [LIK95] LIKFORMAN-SULEM, L., HANIMYAN A., FAURE, C., *A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents*, Third International Conference on Documento Analysis and Recognition (Volume 2), 1995, Montréal, Canada
- [LIU02] LIU, C. L., SAKO, H., FUJISAWA, H., *Performance evaluation of pattern classifiers for handwritten character recognition*, International Journal on Document Analysis and Recognition – IJDAR, Springer-Verlag, 2002
- [LIU03] LIU, H., MOTODA, H., *Feature Extraction, Selection, and Construction, The Handbook of Data Mining*, Lawrence Erlbaum Associates, Inc. Publishers, 2003.
- [LLA01] LLADOS, J., MARTI E., VILLANUEVA, J. J., *Symbol Recognition by Error Tolerant Subgraph Matching between Region Adjacency Graphs*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, VOL 23, No. 10

- [MAD01] MADHVANATH, S., GOVINDARAJU V., *The Role of Holistic Paradigms in Handwritten Word Recognition*, 2001, IEEE Transaction on Pattern Analysis and Machine Intelligence, VOL 23, No. 2
- [MCC01] McCULLOCH, C. E., SEARLE, S. R., *Generalised, Linear, and Mixed Models*, 2000, *Wiley Series on Probability and Statistics*, John Wiley & sons, inc – New York.
- [MOO94] MOODY, J., *Prediction risk and Architecture Selection for Neural Networks*, in *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, V. Cherkassky, J.H. Friedman, and H. Wechsler (eds), Springer-Verlag, 1994.
- [MUK01] MUKUNDAN, R., ONG, S. H., LEE, P. A., *Discrete vs. Continuous Orthogonal Moments for Image Analysis*, 2001, CISST01 – International Conference, <http://www.cosc.canterbury.ac.nz/people/mukundan/cisst01.pdf>
- [MUK02] MUKUNDAN, R., ONG, S. H., LEE, P. A., *Discrete Orthogonal Moment Features Using Chebyshev Polynomials*, acessada na internet em 17/11/2002, em <http://www.cosc.canterbury.ac.nz/people/mukundan/ivcnz00.pdf>, Faculty of Information Science and Technology, Multimedia University 75450 Malacca, Malaysia.
- [OH_02] OH, I. S., SUEN, C. Y., *A class-modular feedforward neural network for handwriting recognition*, *The Journal of Pattern Recognition Society*, pp 229-244, 2002
- [PER97] PERPIÑÁN, M. A. C., *A review of Dimension Reduction Techniques*, University of Sheffield, 1997 – Technical Report CS-96-09
- [POU00] POULARIKAS, A. D., *The Transforms and Applications Handbook - Second Edition*, CRC/IEEE PRESS, 1999
- [PRI99] PRINCIPE, J. C., EULIANO, N. R., LEFEBVRE, W. C., *Neural and Adaptive Systems*, Jhon Wisley & Sons Inc, 1999.
- [RAB93] RABINER, L., JUANG, B. H., *Fundamentals of Speech Recognition*, Prentice-HALL, 1993
- [SCH92] SCHALKOFF, R. J., *Pattern Recognition: Statistical, Structural and Neural Approaches*, pp 2-30, 1992 – Jhon Wisley & Sons Inc.
- [SEZ01] SEZGIN, M., SANKUR, B., *Selection of thresholding methods for non destructive testing applications*, International Conference on Image Processing, Proceedings. 2001
- [SIL92] SILVA, E. S., MUNTZ, R. R., *Métodos Computacionais de Solução de Cadeias de Markov:Aplicações a Sistemas de Computação e Comunicação*, VIII Escola de Computação, Gramado, 1992
- [SMI97] SMITH, S. W., *The Scientist and Engineer's Guide to Digital Signal Processing*, 1997 - California Technical Publishing.
- [TOM96] TOMBRE, K., *Quelques contributions à l'interpretation de documents techniques*, Centre de Recherche en Informatique de Nancy, 1996.
- [TRI95] TRIER, O. D., JAIN, A. K., TAXT, T., *Feature Extraction Methods for Character Recognition – A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 17, N. 1, January 1995
- [TON03a] TONON Jr, W., FREITAS, C. O. A., NIEVOLA, J. C., *Sistema de reconhecimento de símbolos*. Anais do XXIII Congresso Brasileiro da Sociedade Brasileira de Computação - IV Encontro Nacional de Inteligência Artificial (ENIA'2003), Campinas, SP, Brazil, agosto, 2003. Vol VII, 335-344p. ISBN: 85-88442-5X(v.7)

- [TON03b] TONON Jr, W., FREITAS, C. O. A., BORTOLOZZI, F., *A symbol recognition system*. 5th IAPR International Workshop on Graphics Recognition (GREC'2003), Barcelona, Spain, july, 2003. Vol I, 299-307p. ISBN: 84-932156-6-X
- [ZEL94] ZELL, A., MAMIER, G., VOGT, M., MACHE, N., *SNNS – Stuttgart Neural Network Simulator, User Manual, Version 4.2*, University of Stuttgart, 1994.

Apêndice A

Base de dados utilizada

A base de dados utilizadas está listada abaixo, por metaclassa e com os símbolos *outliers* compondo uma figura á parte. A base de dados é composta de 304 símbolos gráficos especificados dentro da norma ABNT (com graus de liberdade utilizada por desenhistas e técnicos para os dispositivos com enrolamentos, ex: indutores e transformadores).

Os símbolos estão dispostos em 5 metaclassas para facilitar a visualização, e os símbolos *outliers* são apresentados no ítem A.2.

A.1. Símbolos gráficos

Segue a apresentação da base de dados utilizada para treinamento, validação e testes.

A.1.1. Metaclassa I

Compõe-se de autotransformadores e reatores com comutação de TAP, como mostra a Figura A.1.

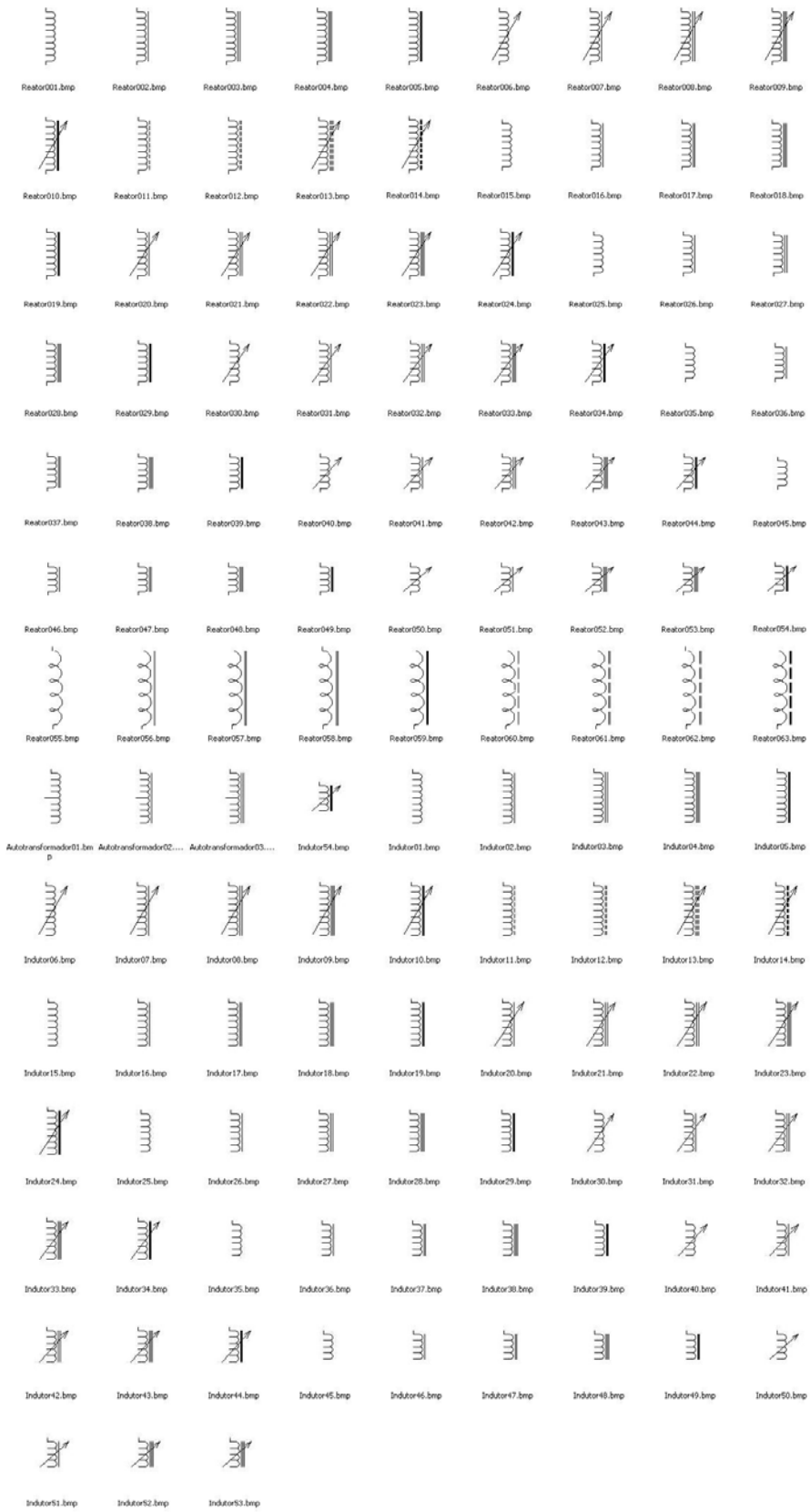


Figura A.1: Metaclassse I

A.1.2. Metaclasse II

Compõe-se de capacitores, aterramentos para equipamentos analógicos e sistemas de potência, diodos, ponte retificadora, instrumentos de painéis, disjuntores (norma ABNT-1), resistores, entradas e saídas de linha e muflas, como mostra a Figura A.2.

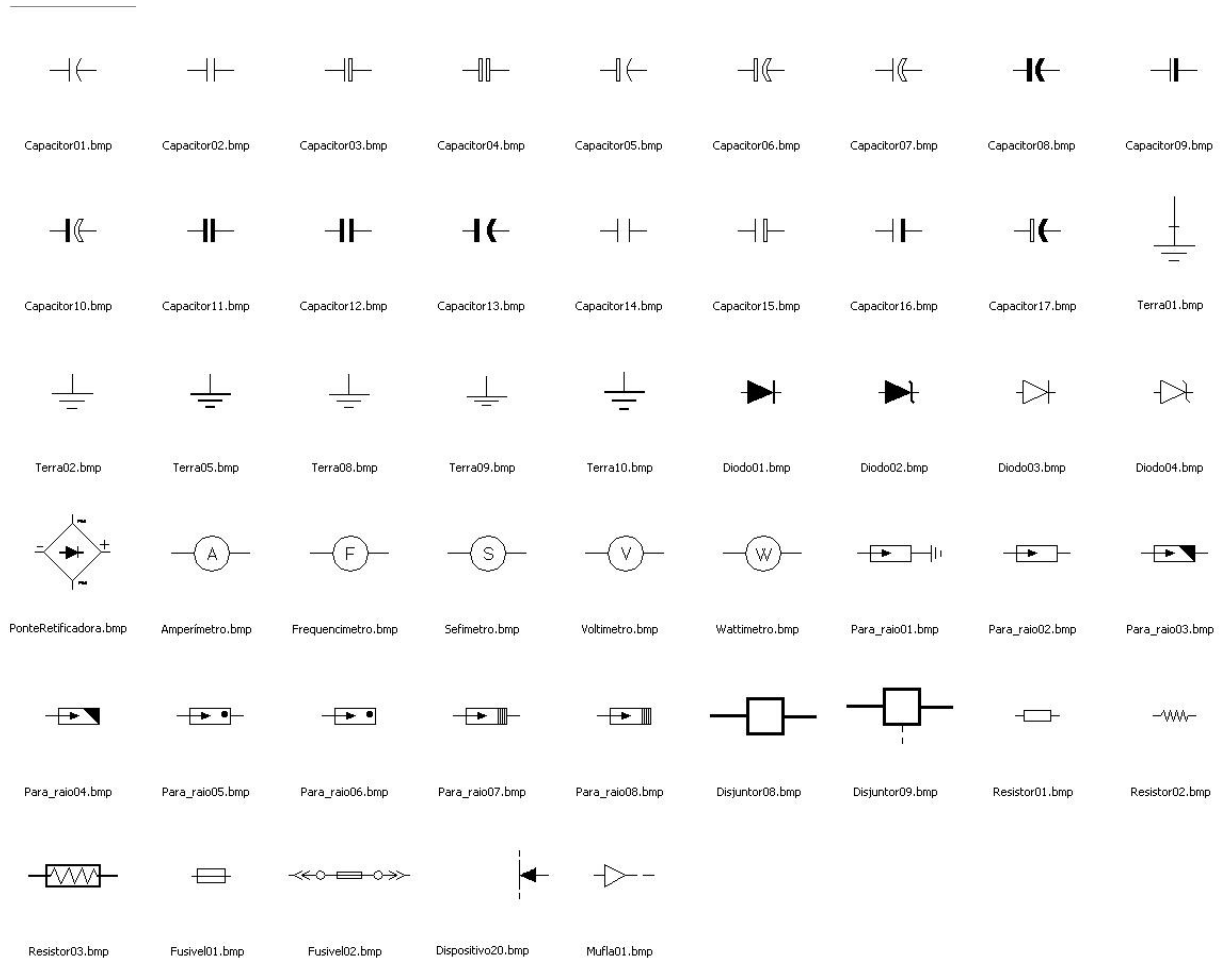


Figura A.2: Metaclasse II

A.1.3. Metaclasse III

Compõe-se de capacitores, aterramentos para equipamentos digitais, chaves seccionadoras, dispositivos de funções (sistemas de funções, como relés microprocessados, clps, etc..) e relés convencionais (inclusive biestáveis e temporizados), como mostra a Figura A.3.

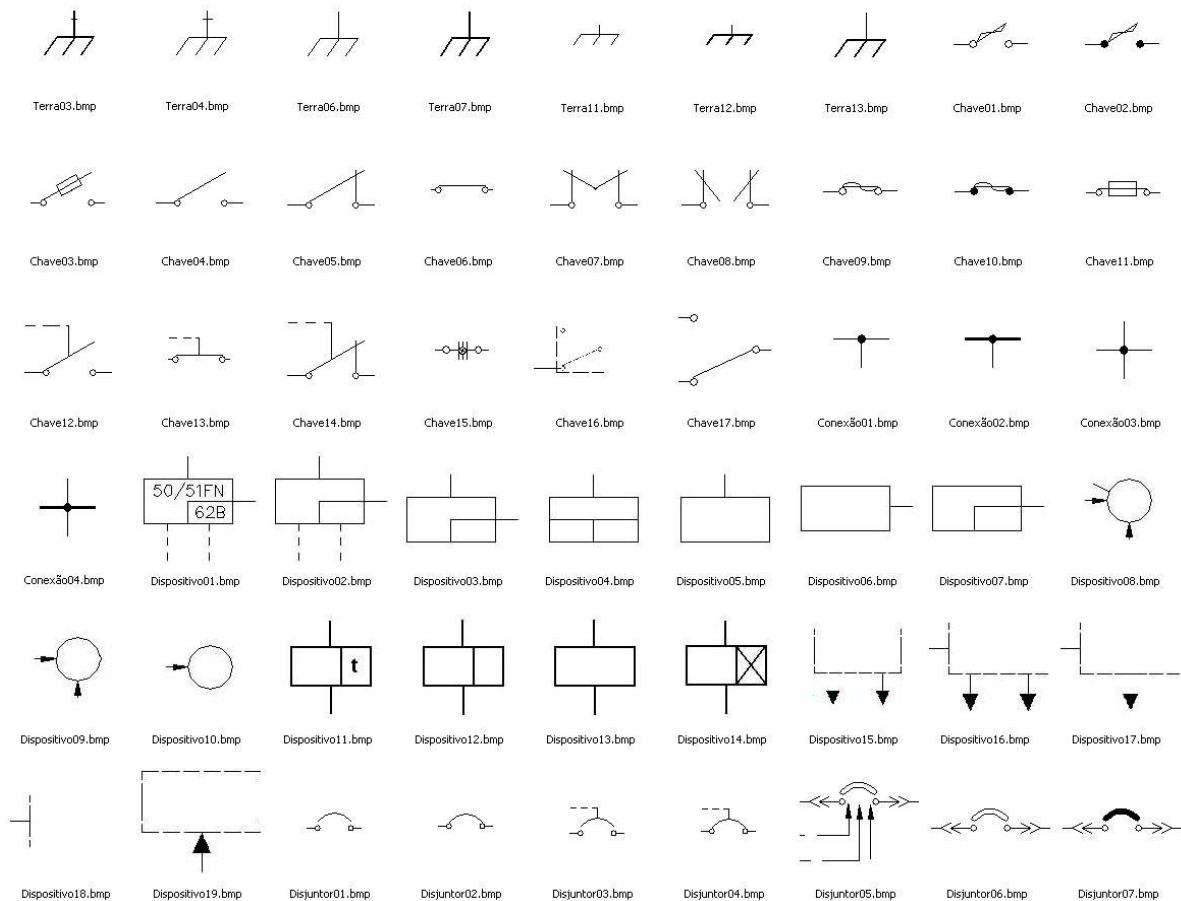


Figura A.3: Metaclasse III

A.1.4. Metaclasse IV

Compõe-se de transformadores com 2 enrolamentos, com e sem ajustes de TAP, e também contém a representação dos transformadores de corrente (TCs), como mostra a Figura A.4.

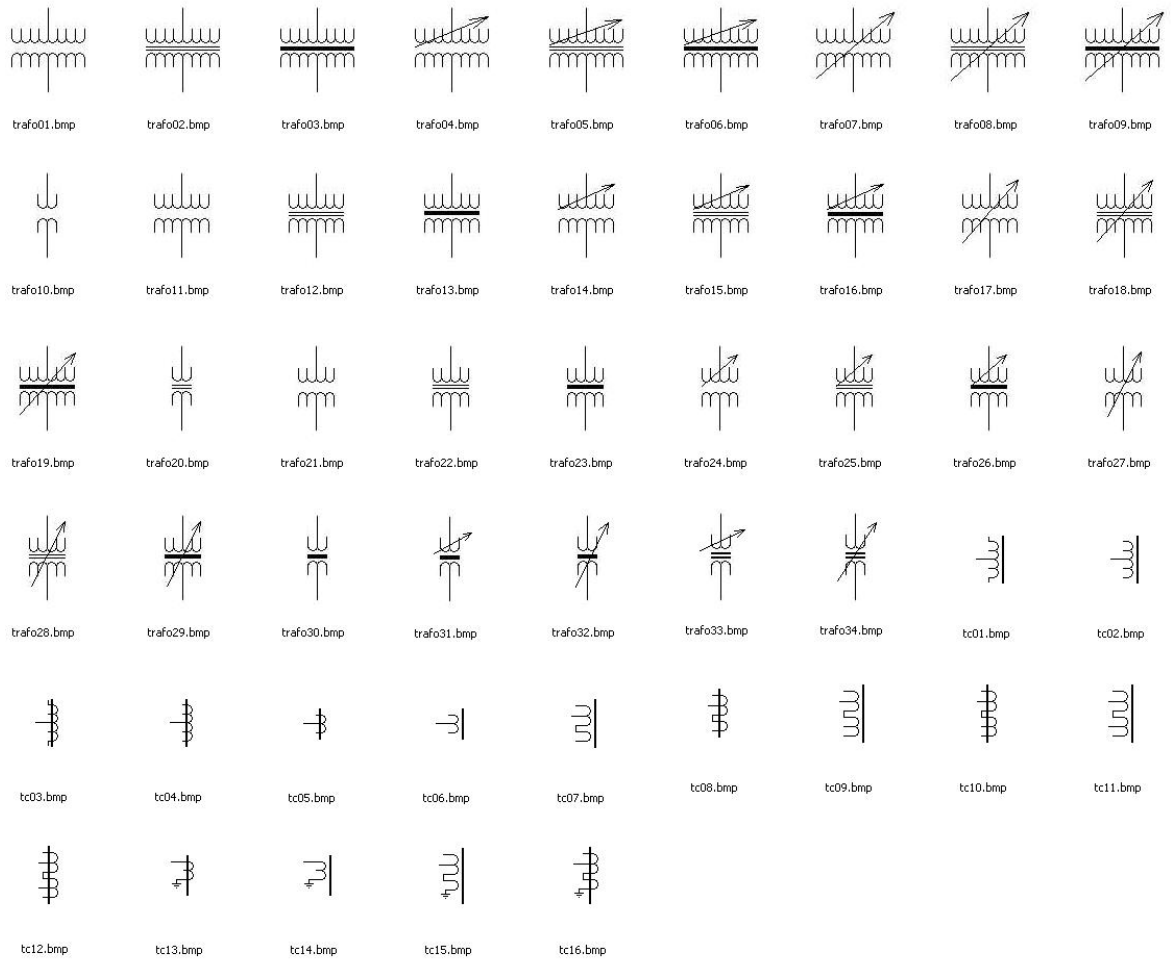


Figura A.4: Metaclasse IV

A.1.5. Metaclasse V

A mais simples de todas, porém com os desenhos mais complexos, e com os dispositivos mais onerosos em uma planta, compondo-se de transformadores com 3 enrolamentos, com e sem ajustes de TAP, como mostra a Figura A.5.

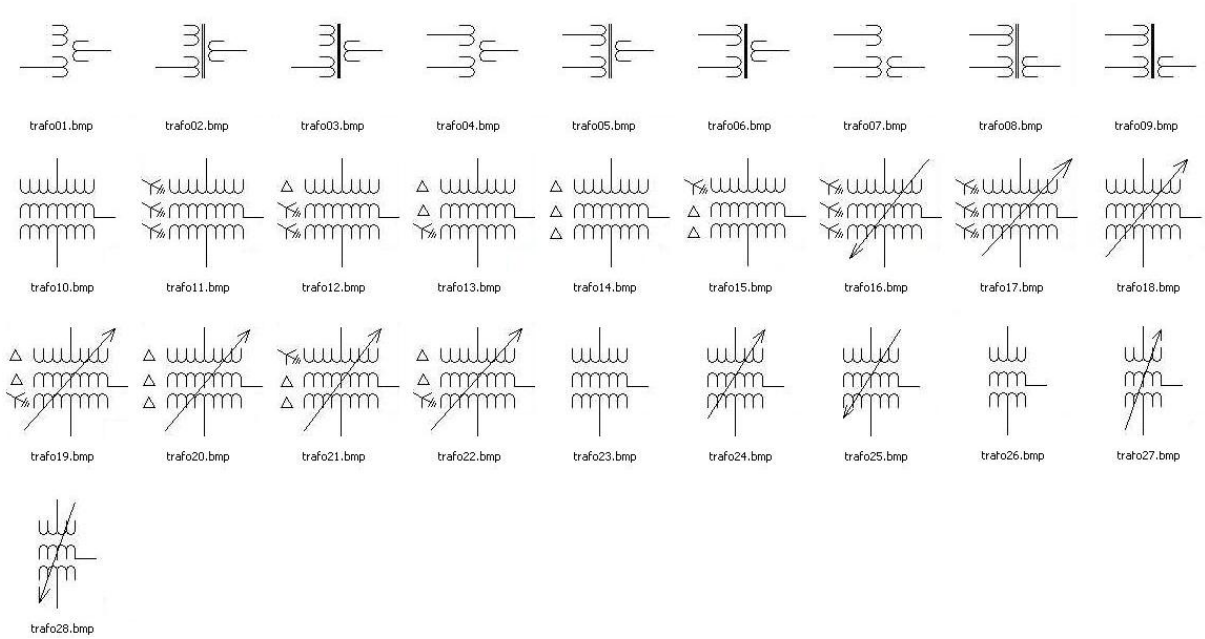


Figura A.5: Metaclasse V

A.2. Símbolos *outliers*

Os símbolos foram divididos por metaclasses, para uma melhor compreensão. Alguns símbolos foram gerados de forma aleatória, devido a dificuldade de se gerar desenhos simulando erros de segmentações.

A.2.1. Metaclasses I

A Figura A.6 mostra os símbolos *outliers* para a Metaclasses I.

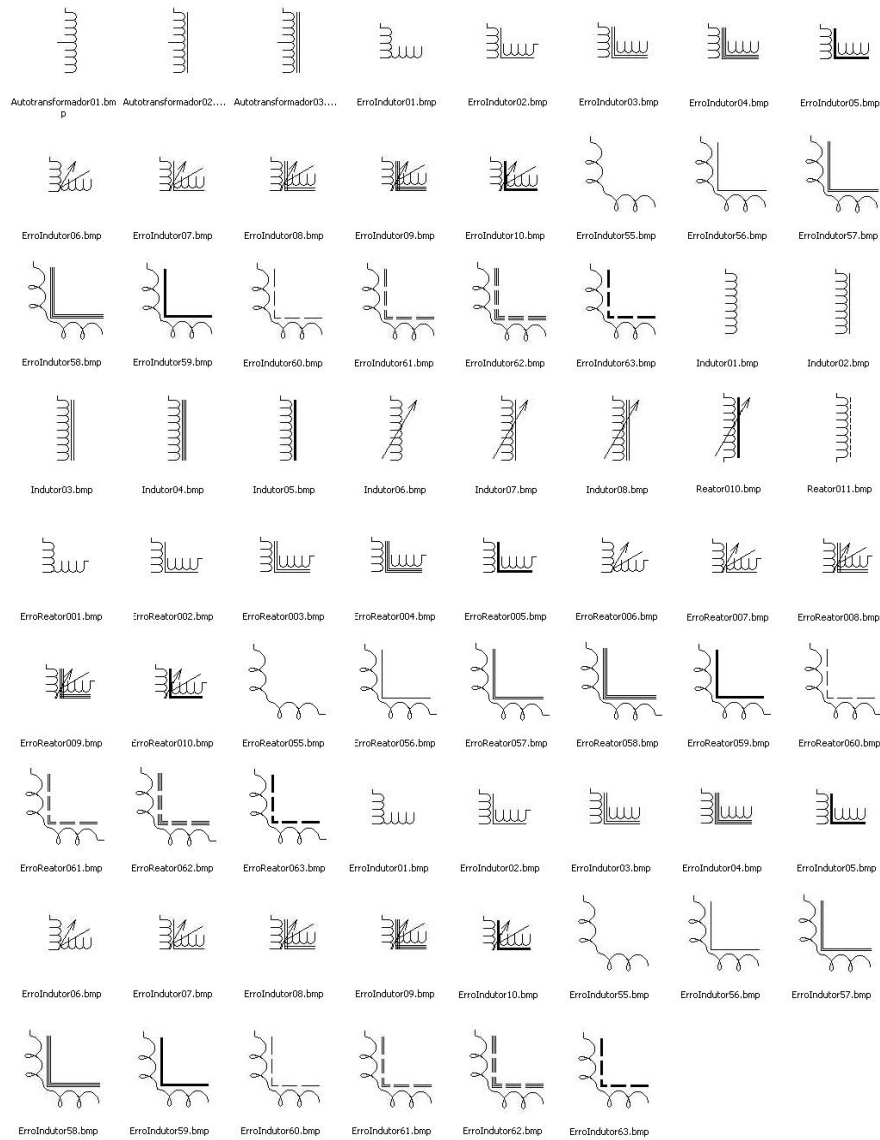


Figura A.6: Símbolos *outliers* para a Metaclasses I.

A.2.2. Metaclasse II

A Figura A.7 mostra os símbolos *outliers* para a Metaclasse II.

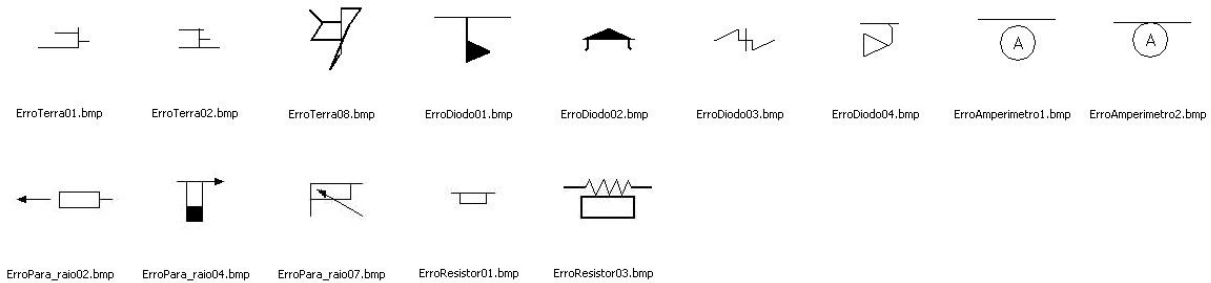


Figura A.7: Símbolos *outliers* para a Metaclasse II.

A.2.3. Metaclasse III

A Figura A.8 mostra os símbolos *outliers* para a Metaclasse III.

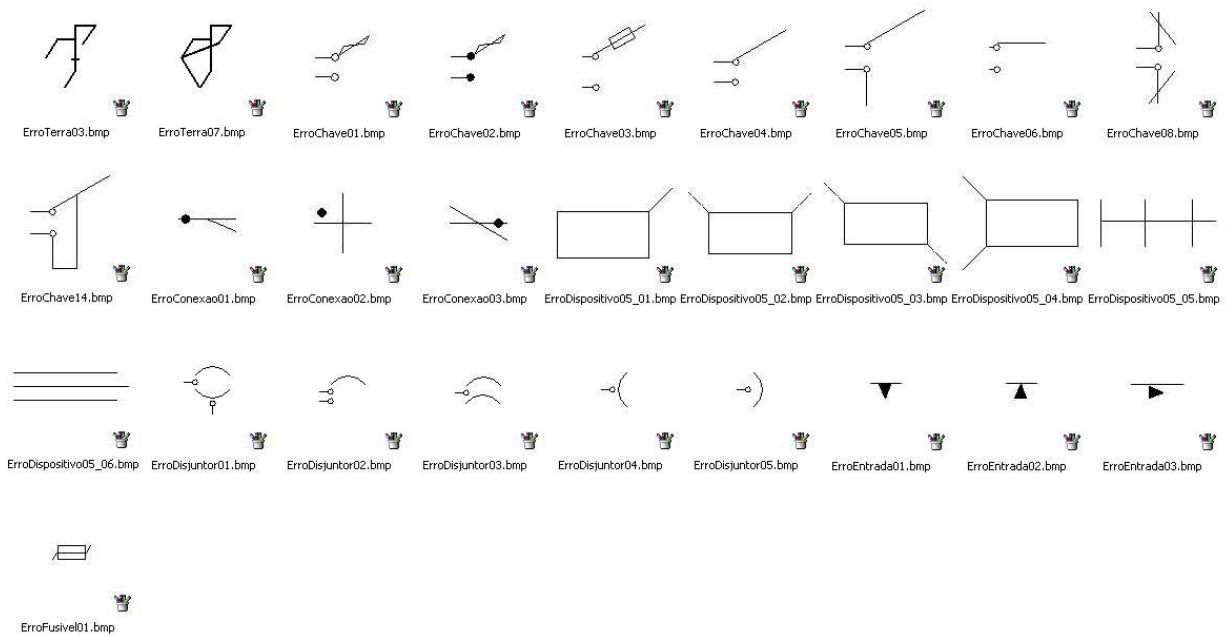


Figura A.8: Símbolos *outliers* para a Metaclasse III.

A.2.4. Metaclasse IV

A Figura A.9 mostra os símbolos *outliers* para a Metaclasse IV.

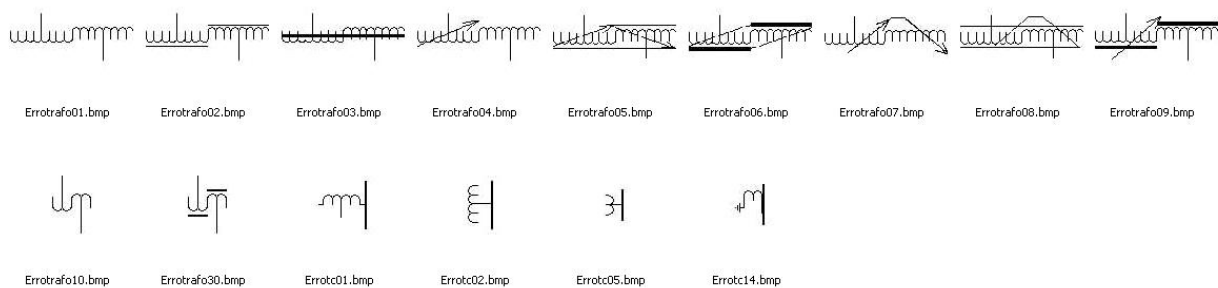


Figura A.9: Símbolos *outliers* para a Metaclasse IV.

A.2.5. Metaclasse V

A Figura A.10 mostra os símbolos *outliers* para a Metaclasse V.

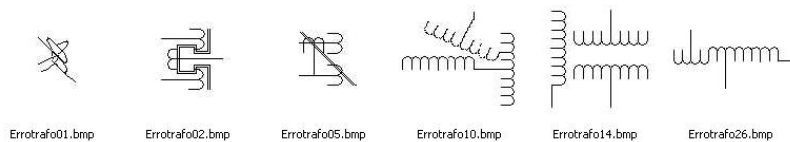


Figura A.10: Símbolos *outliers* para a Metaclasse V.

Apêndice B

Curvas de treinamento e testes de validação para a Transformada de Chebyshev

Seguem as curvas de treinamento, os arquivos de registro e os resultados parciais realizados durante os testes com a transformada. Os testes seguiram um padrão de validação cruzada a 10%, significando que há 10 testes parciais realizados por modelo de transformada.

Para o treinamento, utilizou-se o algoritmo *Backpropagation with momentum* (retropropagação com momento), definido com $\eta=0,2$, $\mu=0,5$, $c=0,1$ e $d_{\max}=0,1$.

Para treinamento do primeiro conjunto de testes, obteve-se a curva da Figura B.1, apresentando o registro de testes da listagem B.1 e com a tabela de reconhecimento B.1.

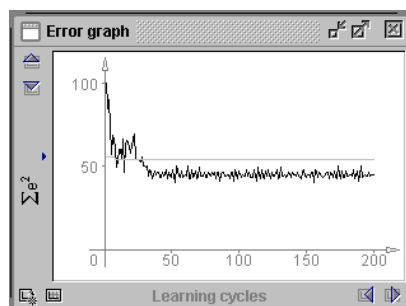


Figura B.1: Curva de erro para o 1º conjunto de testes

Listagem B.1: Registro dos testes do 1º grupo

opened at: Mon Dec 01 11:27:08 BRST 2003	
Step 20 SSE: 62.0	validation: 54.0
Step 40 SSE: 46.0	validation: 54.0
Step 60 SSE: 45.0	validation: 54.0
Step 80 SSE: 46.0	validation: 54.0
Step 100 SSE: 47.0	validation: 54.0

Step 120 SSE: 47.0	validation: 54.0
Step 140 SSE: 46.0	validation: 54.0
Step 160 SSE: 46.0	validation: 54.0
Step 180 SSE: 47.0	validation: 54.0
Step 200 SSE: 45.0	validation: 54.0

Tabela B.1: Reconhecimento do 1º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	1	1	--	--
2	--	4	--	1	--	--
3	1	1	5	--	--	--
4	--	1	1	2	--	1
5	--	--	--	--	1	2
N.R. = Não reconhecido						

Para treinamento do segundo conjunto de testes, obteve-se a curva da Figura B.2, apresentando o registro de testes da listagem B.2 e com a tabela de reconhecimento B.2.

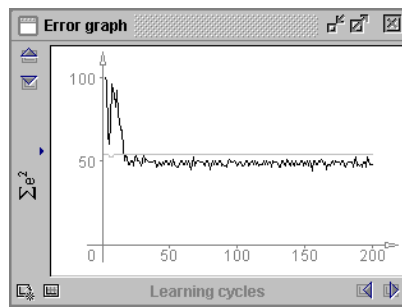


Figura B.2: Curva de erro para o 2º conjunto de testes

Listagem B.2: Registro dos testes do 2º grupo

opened at: Thu Dec 04 09:44:22 BRST 2003		
Step 20 SSE: 91.0	validation:	54.0
Step 40 SSE: 50.0	validation:	54.0
Step 60 SSE: 50.0	validation:	54.0
Step 80 SSE: 50.0	validation:	54.0
Step 100 SSE: 48.0	validation:	54.0
Step 120 SSE: 46.0	validation:	54.0
Step 140 SSE: 48.0	validation:	54.0
Step 160 SSE: 48.0	validation:	54.0
Step 180 SSE: 46.0	validation:	54.0
Step 200 SSE: 46.0	validation:	54.0

Tabela B.2: Reconhecimento do 2º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	8	--	1	1	--	--
2	--	5	1	--	--	--
3	--	1	4	1	--	--
4	--	--	2	3	--	--
5	--	--	--	--	1	1
N.R. = Não reconhecido						

Para treinamento do terceiro conjunto de testes, obteve-se a curva da Figura B.3, apresentando o registro de testes da listagem B.3 e com a tabela de reconhecimento B.3.

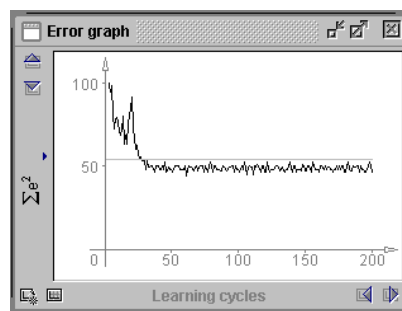


Figura B.3: Curva de erro para o 3º conjunto de testes

Listagem B.3: Registro dos testes do 3º grupo

opened at: Thu Dec 04 09:44:22 BRST 2003			
Step 20	SSE: 91.0	validation:	54.0
Step 40	SSE: 50.0	validation:	54.0
Step 60	SSE: 50.0	validation:	54.0
Step 80	SSE: 50.0	validation:	54.0
Step 100	SSE: 48.0	validation:	54.0
Step 120	SSE: 46.0	validation:	54.0
Step 140	SSE: 48.0	validation:	54.0
Step 160	SSE: 48.0	validation:	54.0
Step 180	SSE: 46.0	validation:	54.0
Step 200	SSE: 46.0	validation:	54.0

Tabela B.3: Reconhecimento do 3º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	1	--	--	1
2	--	4	--	--	--	--
3	1	1	3	1	--	--
4	--	--	2	3	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do quarto conjunto de testes, obteve-se a curva da Figura B.4, apresentando o registro de testes da listagem B.4 e com a tabela de reconhecimento B.4.

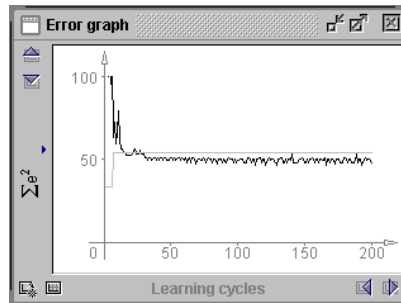


Figura B.4: Curva de erro para o 4º conjunto de testes

Listagem B.4: Registro dos testes do 4º grupo

opened at: Thu Dec 04 09:49:58 BRST 2003		
Step 20	SSE: 52.0	validation: 54.0
Step 40	SSE: 51.0	validation: 54.0
Step 60	SSE: 49.0	validation: 54.0
Step 80	SSE: 50.0	validation: 54.0
Step 100	SSE: 49.0	validation: 54.0
Step 120	SSE: 50.0	validation: 54.0
Step 140	SSE: 49.0	validation: 54.0
Step 160	SSE: 50.0	validation: 54.0
Step 180	SSE: 51.0	validation: 54.0
Step 200	SSE: 48.0	validation: 54.0

Tabela B.4: Reconhecimento do 4º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	8	--	1	1	--	1
2	--	5	--	--	--	--
3	--	1	2	2	--	1
4	--	--	2	3	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do quinto conjunto de testes, obteve-se a curva da Figura B.5, apresentando o registro de testes da listagem B.5 e com a tabela de reconhecimento B.5.

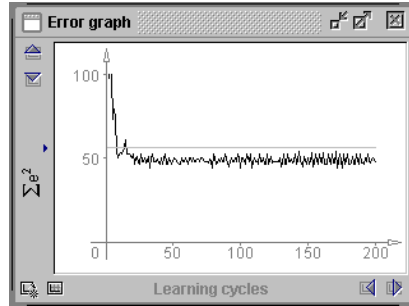


Figura B.5: Curva de erro para o 5º conjunto de testes

Listagem B.5: Registro dos testes do 5º grupo

opened at: Thu Dec 04 09:54:14 BRST 2003

Step 20 SSE:	52.0	validation:	56.0
Step 40 SSE:	46.0	validation:	56.0
Step 60 SSE:	48.0	validation:	56.0
Step 80 SSE:	52.0	validation:	56.0
Step 100 SSE:	44.0	validation:	56.0
Step 120 SSE:	52.0	validation:	56.0
Step 140 SSE:	48.0	validation:	56.0
Step 160 SSE:	48.0	validation:	56.0
Step 180 SSE:	52.0	validation:	56.0
Step 200 SSE:	48.0	validation:	56.0

Tabela B.5: Reconhecimento do 5º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	--	1	--	1
2	--	4	--	--	--	1
3	--	2	3	1	--	--
4	--	--	2	3	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do sexto conjunto de testes, obteve-se a curva da Figura B.6, apresentando o registro de testes da listagem B.6 e com a tabela de reconhecimento B.6.

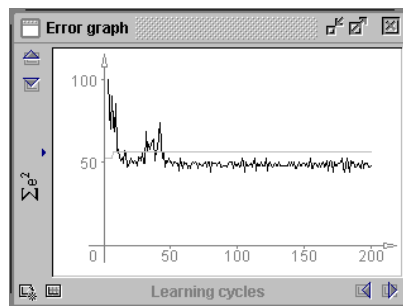


Figura B.6: Curva de erro para o 6º conjunto de testes

Listagem B.6: Registro dos testes do 6º grupo

opened at: Thu Dec 04 09:59:00 BRST 2003		
Step 20 SSE:	52.0	validation: 56.0
Step 40 SSE:	58.0	validation: 56.0
Step 60 SSE:	50.0	validation: 56.0
Step 80 SSE:	44.0	validation: 56.0
Step 100 SSE:	48.0	validation: 56.0
Step 120 SSE:	48.0	validation: 56.0
Step 140 SSE:	48.0	validation: 56.0
Step 160 SSE:	48.0	validation: 56.0
Step 180 SSE:	46.0	validation: 56.0
Step 200 SSE:	48.0	validation: 56.0

Tabela B.6: Reconhecimento do 6º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	8	1	1	1	--	1
2	--	5	--	--	--	--
3	--	3	3	--	--	--
4	--	--	2	3	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sétimo conjunto de testes, obteve-se a curva da Figura B.7, apresentando o registro de testes da listagem B.7 e com a tabela de reconhecimento B.7.

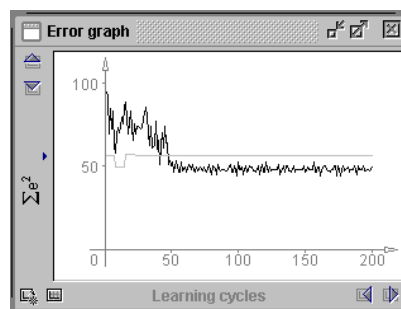


Figura B.7: Curva de erro para o 7º conjunto de testes

Listagem B.7: Registro dos testes do 7º grupo

opened at: Thu Dec 04 10:03:36 BRST 2003		
Step 20 SSE:	72.0	validation: 57.0
Step 40 SSE:	65.0	validation: 56.0
Step 60 SSE:	49.0	validation: 56.0
Step 80 SSE:	45.0	validation: 56.0
Step 100 SSE:	52.0	validation: 56.0
Step 120 SSE:	46.0	validation: 56.0

Step 140 SSE: 46.0	validation: 56.0
Step 160 SSE: 49.0	validation: 56.0
Step 180 SSE: 47.0	validation: 56.0
Step 200 SSE: 49.0	validation: 56.0

Tabela B.7: Reconhecimento do 7º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	--	2	--	1
2	1	4	--	--	--	--
3	1	1	4	--	--	--
4	--	--	3	2	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do oitavo conjunto de testes, obteve-se a curva da Figura B.8, apresentando o registro de testes da listagem B.8 e com a tabela de reconhecimento B.8.

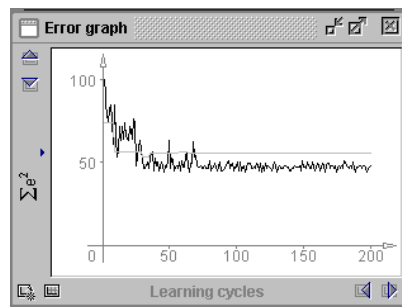


Figura B.8: Curva de erro para o 8º conjunto de testes

Listagem B.8: Registro dos testes do 8º grupo

opened at: Thu Dec 04 10:09:41 BRST 2003		
Step 20 SSE: 71.0	validation: 56.0	
Step 40 SSE: 48.0	validation: 55.0	
Step 60 SSE: 47.0	validation: 56.0	
Step 80 SSE: 51.0	validation: 55.0	
Step 100 SSE: 47.0	validation: 55.0	
Step 120 SSE: 48.0	validation: 55.0	
Step 140 SSE: 45.0	validation: 55.0	
Step 160 SSE: 47.0	validation: 55.0	
Step 180 SSE: 46.0	validation: 55.0	
Step 200 SSE: 48.0	validation: 55.0	

Tabela B.8: Reconhecimento do 8º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	1	1	--	1
2	1	4	--	--	--	--
3	--	1	4	1	--	--
4	--	--	2	3	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do nono conjunto de testes, obteve-se a curva da Figura B.9, apresentando o registro de testes da listagem B.9 e com a tabela de reconhecimento B.9.

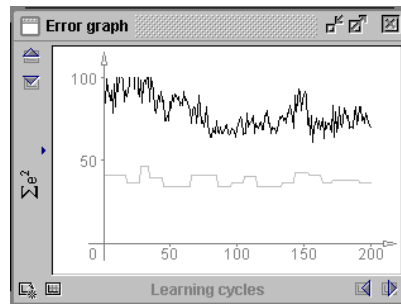


Figura B.9: Curva de erro para o 9º conjunto de testes

Listagem B.9: Registro dos testes do 9º grupo

opened at: Thu Dec 04 11:01:06 BRST 2003		
Step 20	SSE: 106.0	validation: 36.0
Step 40	SSE: 83.0	validation: 39.0
Step 60	SSE: 91.0	validation: 34.0
Step 80	SSE: 64.0	validation: 41.0
Step 100	SSE: 68.0	validation: 36.0
Step 120	SSE: 68.0	validation: 34.0
Step 140	SSE: 83.0	validation: 36.0
Step 160	SSE: 75.0	validation: 41.0
Step 180	SSE: 74.0	validation: 38.0
Step 200	SSE: 70.0	validation: 36.0

Tabela B.9: Reconhecimento do 9º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	--	2	--	1
2	--	3	2	--	--	--
3	--	1	5	--	--	--
4	--	--	2	3	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do décimo conjunto de testes, obteve-se a curva da Figura B.10, apresentando o registro de testes da listagem B.10 e com a tabela de reconhecimento B.10.

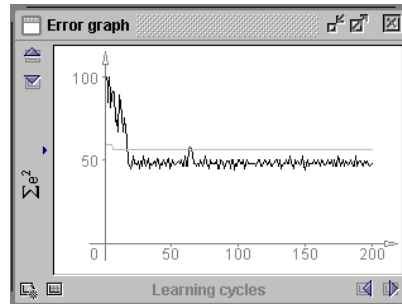


Figura B.10: Curva de erro para o 10º conjunto de testes

Listagem B.10: Registro dos testes do 10º grupo

opened at: Thu Dec 04 10:55:36 BRST 2003		
Step 20 SSE:	46.0	validation: 56.0
Step 40 SSE:	49.0	validation: 56.0
Step 60 SSE:	52.0	validation: 56.0
Step 80 SSE:	48.0	validation: 56.0
Step 100 SSE:	50.0	validation: 56.0
Step 120 SSE:	50.0	validation: 56.0
Step 140 SSE:	48.0	validation: 56.0
Step 160 SSE:	50.0	validation: 56.0
Step 180 SSE:	46.0	validation: 56.0
Step 200 SSE:	48.0	validation: 56.0

Tabela B.10: Reconhecimento do 10º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	--	2	--	1
2	--	3	1	--	--	1
3	--	1	3	1	--	--
4	--	--	1	5	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Apêndice C

Curvas de treinamento e testes de validação para a Transformada Discreta do Coseno (DCT)

Seguem as curvas de treinamento, os arquivos de registro e os resultados parciais realizados durante os testes com a transformada. Os testes seguiram um padrão de validação cruzada a 10%, significando que há 10 testes parciais realizados por modelo de transformada.

Para o treinamento, utilizou-se o algoritmo *Backpropagation with momentum* (retropropagação com momento), definido com $\eta=0,2$, $\mu=0,5$, $c=0,1$ e $d_{\max}=0,01$.

Para treinamento do primeiro conjunto de testes, obteve-se a curva da Figura C.1, apresentando o registro de testes da listagem C.1 e com a tabela de reconhecimento C.1.

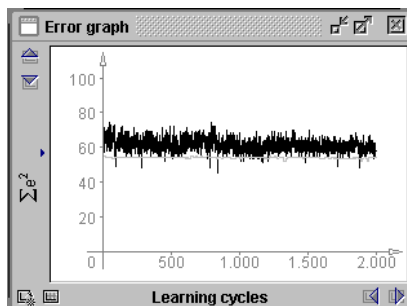


Figura C.1: Curva de erro para o 1º conjunto de testes

Listagem C.1: Registro dos testes do 1º grupo

opened at: Sat Nov 29 14:34:44 GMT-03:00 2003	
Step 200 SSE: 56.0	validation: 54.0
Step 400 SSE: 58.0	validation: 54.0
Step 600 SSE: 58.0	validation: 54.0
Step 800 SSE: 74.0	validation: 54.0
Step 1000 SSE: 56.0	validation: 54.0

Step 1200 SSE: 58.0	validation: 54.0
Step 1400 SSE: 67.0	validation: 54.0
Step 1600 SSE: 57.0	validation: 53.0
Step 1800 SSE: 63.0	validation: 53.0
Step 2000 SSE: 58.0	validation: 54.0

Tabela C.1: Reconhecimento do 1º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	--	1	--	2
2	--	3	1	--	--	1
3	--	--	4	--	--	2
4	--	--	--	4	--	1
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do segundo conjunto de testes, obteve-se a curva da Figura C.2, apresentando o registro de testes da listagem C.2 e com a tabela de reconhecimento C.2.

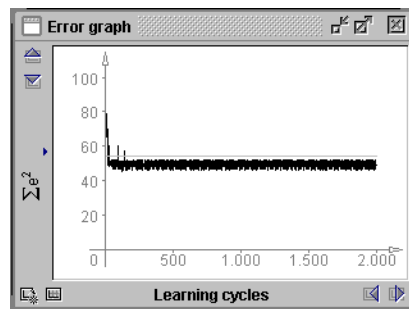


Figura C.2: Curva de erro para o 2º conjunto de testes

Listagem C.2: Registro dos testes do 2º grupo

opened at: Sat Nov 29 13:56:33 GMT-03:00 2003	
Step 200 SSE: 50.0	validation: 54.0
Step 400 SSE: 51.0	validation: 54.0
Step 600 SSE: 50.0	validation: 54.0
Step 800 SSE: 52.0	validation: 54.0
Step 1000 SSE: 51.0	validation: 54.0
Step 1200 SSE: 50.0	validation: 54.0
Step 1400 SSE: 51.0	validation: 54.0
Step 1600 SSE: 51.0	validation: 54.0
Step 1800 SSE: 50.0	validation: 54.0
Step 2000 SSE: 49.0	validation: 54.0

Tabela C.2: Reconhecimento do 2º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	3	1	--	--	1
3	--	--	5	--	--	1
4	1	--	--	3	--	1
5	1	--	--	--	--	1
N.R. = Não reconhecido						

Para treinamento do terceiro conjunto de testes, obteve-se a curva da Figura C.3, apresentando o registro de testes da listagem C.3 e com a tabela de reconhecimento C.3.

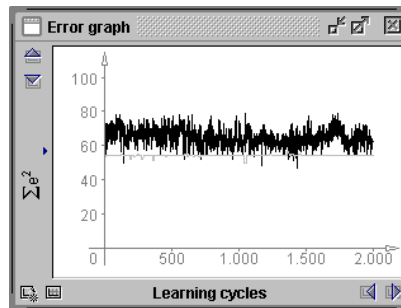


Figura C.3: Curva de erro para o 3º conjunto de testes

Listagem C.3: Registro dos testes do 3º grupo

opened at: Sat Nov 29 14:53:55 GMT-03:00 2003	
Step 200 SSE: 71.0	validation: 52.0
Step 400 SSE: 59.0	validation: 54.0
Step 600 SSE: 67.0	validation: 55.0
Step 800 SSE: 67.0	validation: 54.0
Step 1000 SSE: 57.0	validation: 54.0
Step 1200 SSE: 69.0	validation: 54.0
Step 1400 SSE: 67.0	validation: 55.0
Step 1600 SSE: 63.0	validation: 54.0
Step 1800 SSE: 61.0	validation: 54.0
Step 2000 SSE: 62.0	validation: 54.0

Tabela C.3: Reconhecimento do 3º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	--	1	--	--
2	--	2	3	--	--	--
3	--	2	4	--	--	--
4	--	--	2	3	--	--
5	1	--	--	--	2	--
N.R. = Não reconhecido						

Para treinamento do quarto conjunto de testes, obteve-se a curva da Figura C.4, apresentando o registro de testes da listagem C.4 e com a tabela de reconhecimento C.4.

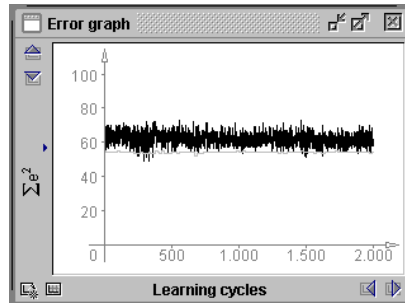


Figura C.4: Curva de erro para o 4º conjunto de testes

Listagem C.4: Registro dos testes do 4º grupo

```

opened at: Sat Nov 29 15:08:27 GMT-03:00 2003
Step 200 SSE: 61.0  validation: 54.0
Step 400 SSE: 67.0  validation: 54.0
Step 600 SSE: 63.0  validation: 54.0
Step 800 SSE: 60.0  validation: 54.0
Step 1000 SSE: 60.0 validation: 55.0
Step 1200 SSE: 58.0 validation: 54.0
Step 1400 SSE: 64.0 validation: 54.0
Step 1600 SSE: 59.0 validation: 54.0
Step 1800 SSE: 65.0 validation: 54.0
Step 2000 SSE: 58.0 validation: 54.0
  
```

Tabela C.4: Reconhecimento do 4º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	--	1	--	1
2	--	5	1	--	--	--
3	--	--	3	1	--	1
4	--	--	--	3	--	2
5	--	--	--	--	2	1
N.R. = Não reconhecido						

Para treinamento do quinto conjunto de testes, obteve-se a curva da Figura C.5, apresentando o registro de testes da listagem C.5 e com a tabela de reconhecimento C.5.

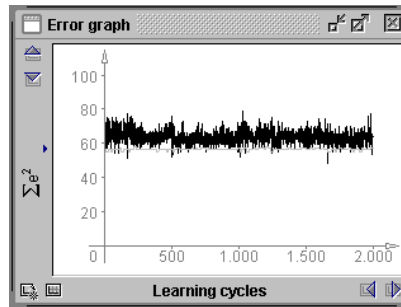


Figura C.5: Curva de erro para o 5º conjunto de testes

Listagem C.5: Registro dos testes do 5º grupo

opened at: Sat Nov 29 15:33:08 GMT-03:00 2003	
Step 200 SSE: 76.0	validation: 56.0
Step 400 SSE: 64.0	validation: 55.0
Step 600 SSE: 64.0	validation: 56.0
Step 800 SSE: 63.0	validation: 56.0
Step 1000 SSE: 62.0	validation: 57.0
Step 1200 SSE: 64.0	validation: 56.0
Step 1400 SSE: 65.0	validation: 56.0
Step 1600 SSE: 68.0	validation: 57.0
Step 1800 SSE: 62.0	validation: 57.0
Step 2000 SSE: 64.0	validation: 57.0

Tabela C.5: Reconhecimento do 5º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	5	1	--	--	--
3	--	--	3	1	--	1
4	--	--	--	3	--	2
5	--	--	--	--	2	1
N.R. = Não reconhecido						

Para treinamento do sexto conjunto de testes, obteve-se a curva da Figura C.6, apresentando o registro de testes da listagem C.6 e com a tabela de reconhecimento C.6.

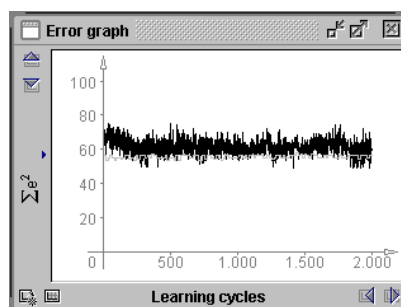


Figura C.6: Curva de erro para o 6º conjunto de testes

Listagem C.6: Registro dos testes do 6º grupo

opened at: Sat Nov 29 15:47:15 GMT-03:00 2003	
Step 200 SSE: 61.0	validation: 54.0
Step 400 SSE: 57.0	validation: 55.0
Step 600 SSE: 62.0	validation: 57.0
Step 800 SSE: 66.0	validation: 56.0
Step 1000 SSE: 65.0	validation: 55.0
Step 1200 SSE: 59.0	validation: 55.0
Step 1400 SSE: 59.0	validation: 56.0
Step 1600 SSE: 60.0	validation: 56.0
Step 1800 SSE: 59.0	validation: 55.0
Step 2000 SSE: 59.0	validation: 56.0

Tabela C.6: Reconhecimento do 6º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	--	1	--	1
2	--	3	--	--	--	2
3	--	1	2	--	1	2
4	--	--	1	4	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do sétimo conjunto de testes, obteve-se a curva da Figura C.7, apresentando o registro de testes da listagem C.7 e com a tabela de reconhecimento C.7.

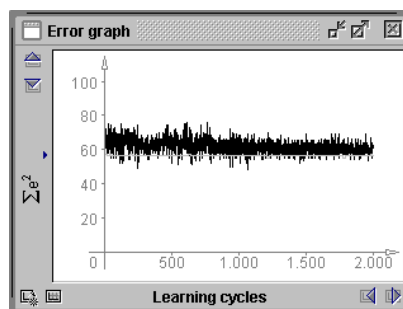


Figura C.7: Curva de erro para o 7º conjunto de testes

Listagem C.7: Registro dos testes do 7º grupo

opened at: Sat Nov 29 15:53:29 GMT-03:00 2003	
Step 200 SSE: 69.0	validation: 57.0
Step 400 SSE: 63.0	validation: 56.0
Step 600 SSE: 72.0	validation: 56.0
Step 800 SSE: 68.0	validation: 56.0
Step 1000 SSE: 57.0	validation: 56.0
Step 1200 SSE: 57.0	validation: 56.0

Step 1400 SSE: 65.0	validation: 56.0
Step 1600 SSE: 58.0	validation: 56.0
Step 1800 SSE: 60.0	validation: 56.0
Step 2000 SSE: 61.0	validation: 56.0

Tabela C.7: Reconhecimento do 7º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	--	1	--	1
2	--	2	3	--	--	--
3	--	1	4	1	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do oitavo conjunto de testes, obteve-se a curva da Figura C.8, apresentando o registro de testes da listagem C.8 e com a tabela de reconhecimento C.8.

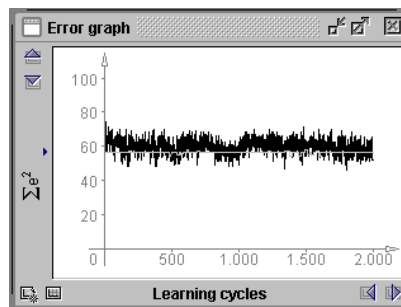


Figura C.8: Curva de erro para o 8º conjunto de testes

Listagem C.8: Registro dos testes do 8º grupo

opened at: Thu Dec 04 10:09:41 BRST 2003	
Step 200 SSE: 60.0	validation: 56.0
Step 400 SSE: 63.0	validation: 56.0
Step 600 SSE: 63.0	validation: 56.0
Step 800 SSE: 62.0	validation: 56.0
Step 1000 SSE: 49.0	validation: 56.0
Step 1200 SSE: 64.0	validation: 56.0
Step 1400 SSE: 62.0	validation: 56.0
Step 1600 SSE: 60.0	validation: 54.0
Step 1800 SSE: 59.0	validation: 56.0
Step 2000 SSE: 52.0	validation: 56.0

Tabela C.8: Reconhecimento do 8º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	--	2	--	--
2	--	1	3	1	--	--
3	--	--	5	--	--	1
4	--	--	1	3	--	1
5	--	--	1	--	2	--
N.R. = Não reconhecido						

Para treinamento do nono conjunto de testes, obteve-se a curva da Figura C.9, apresentando o registro de testes da listagem C.9 e com a tabela de reconhecimento C.9.

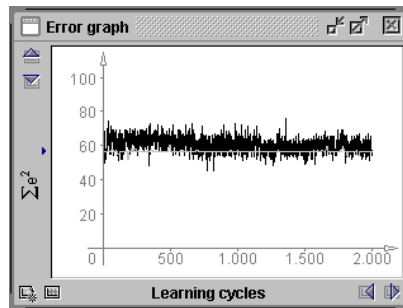


Figura C.9: Curva de erro para o 9º conjunto de testes

Listagem C.9: Registro dos testes do 9º grupo

opened at: Thu Dec 04 11:01:06 BRST 2003	
Step 200 SSE: 64.0	validation: 56.0
Step 400 SSE: 61.0	validation: 56.0
Step 600 SSE: 67.0	validation: 57.0
Step 800 SSE: 60.0	validation: 56.0
Step 1000 SSE: 62.0	validation: 56.0
Step 1200 SSE: 54.0	validation: 56.0
Step 1400 SSE: 60.0	validation: 56.0
Step 1600 SSE: 59.0	validation: 56.0
Step 1800 SSE: 65.0	validation: 56.0
Step 2000 SSE: 54.0	validation: 56.0

Tabela C.9: Reconhecimento do 9º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	--	2	--	--
2	--	1	3	--	--	1
3	--	1	4	1	--	--
4	--	--	--	5	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do décimo conjunto de testes, obteve-se a curva da Figura C.10, apresentando o registro de testes da listagem C.10 e com a tabela de reconhecimento C.10.

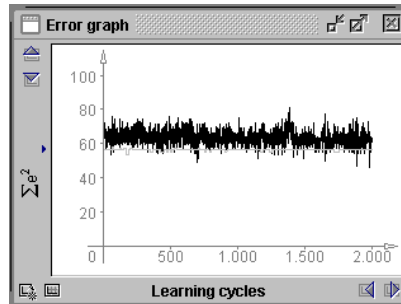


Figura C.10: Curva de erro para o 10^o conjunto de testes

Listagem C.10: Registro dos testes do 10^o grupo

opened at: Sat Nov 29 16:41:36 GMT-03:00 2003

Step 200 SSE: 64.0 validation: 56.0
 Step 400 SSE: 63.0 validation: 56.0
 Step 600 SSE: 68.0 validation: 56.0
 Step 800 SSE: 56.0 validation: 55.0
 Step 1000 SSE: 58.0 validation: 56.0
 Step 1200 SSE: 71.0 validation: 56.0
 Step 1400 SSE: 67.0 validation: 56.0
 Step 1600 SSE: 58.0 validation: 56.0
 Step 1800 SSE: 59.0 validation: 56.0
 Step 2000 SSE: 63.0 validation: 56.0

Tabela C.10: Reconhecimento do 10^o conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	--	--	3	--	--
2	--	3	1	--	--	1
3	--	1	3	--	1	--
4	--	--	--	6	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Apêndice D

Curvas de treinamento e testes de validação para a Transformada de Fourier

Seguem as curvas de treinamento, os arquivos de registro e os resultados parciais realizados durante os testes com a transformada. Os testes seguiram um padrão de validação cruzada a 10%, significando que há 10 testes parciais realizados por modelo de transformada.

Para o treinamento, utilizou-se o algoritmo *Backpropagation* (retro-propagação), definido com $\eta=0,2$ e $d_{\max}=0,1$.

Para treinamento do primeiro conjunto de testes, obteve-se a curva da Figura D.1, apresentando o registro de testes da listagem D.1 e com a tabela de reconhecimento D.1.

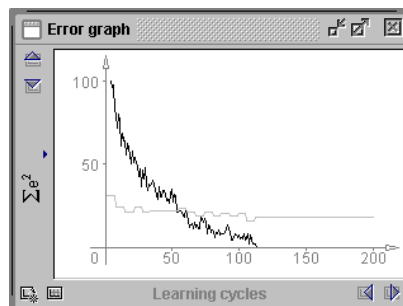


Figura D.1: Curva de erro para o 1º conjunto de testes

Listagem D.1: Registro dos testes do 1º grupo

opened at: Sun Dec 07 14:38:05 BRST 2003	
Step 20 SSE: 58.0	validation: 21.0
Step 40 SSE: 36.0	validation: 22.0
Step 60 SSE: 23.0	validation: 23.0
Step 80 SSE: 10.0	validation: 21.0
Step 100 SSE: 7.0	validation: 20.0

Step 120 SSE: 0.0	validation: 18.0
Step 140 SSE: 0.0	validation: 18.0
Step 160 SSE: 0.0	validation: 18.0
Step 180 SSE: 0.0	validation: 18.0
Step 200 SSE: 0.0	validation: 18.0

Tabela D.1: Reconhecimento do 1º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	--	1	--	1	3
3	--	--	3	1	--	2
4	--	--	--	4	--	1
5	--	--	--	--	2	1
N.R. = Não reconhecido						

Para treinamento do segundo conjunto de testes, obteve-se a curva da Figura D.2, apresentando o registro de testes da listagem D.2 e com a tabela de reconhecimento D.2.

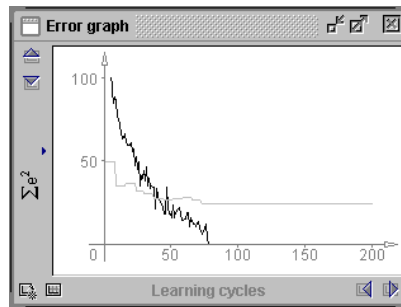


Figura D.2: Curva de erro para o 2º conjunto de testes

Listagem D.2: Registro dos testes do 2º grupo

opened at: Sun Dec 07 14:53:17 BRST 2003	
Step 20 SSE: 61.0	validation: 36.0
Step 40 SSE: 29.0	validation: 27.0
Step 60 SSE: 15.0	validation: 28.0
Step 80 SSE: 0.0	validation: 24.0
Step 100 SSE: 0.0	validation: 24.0
Step 120 SSE: 0.0	validation: 24.0
Step 140 SSE: 0.0	validation: 24.0
Step 160 SSE: 0.0	validation: 24.0
Step 180 SSE: 0.0	validation: 24.0
Step 200 SSE: 0.0	validation: 24.0

Tabela D.2: Reconhecimento do 2º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	1	1	--	1	2
3	1	--	3	--	--	2
4	--	--	--	5	--	--
5	--	--	--	--	2	--
N.R. = Não reconhecido						

Para treinamento do terceiro conjunto de testes, obteve-se a curva da Figura D.3, apresentando o registro de testes da listagem D.3 e com a tabela de reconhecimento D.3.

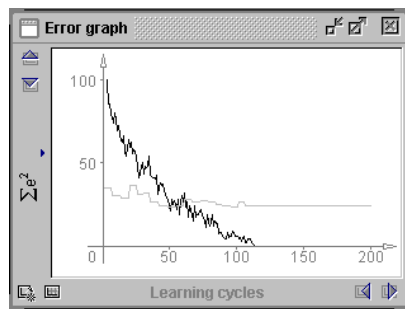


Figura D.3: Curva de erro para o 3º conjunto de testes

Listagem D.3: Registro dos testes do 3º grupo

opened at: Sun Dec 07 15:03:14 BRST 2003	
Step 20 SSE: 59.0	validation: 36.0
Step 40 SSE: 42.0	validation: 26.0
Step 60 SSE: 26.0	validation: 24.0
Step 80 SSE: 12.0	validation: 26.0
Step 100 SSE: 6.0	validation: 23.0
Step 120 SSE: 0.0	validation: 24.0
Step 140 SSE: 0.0	validation: 24.0
Step 160 SSE: 0.0	validation: 24.0
Step 180 SSE: 0.0	validation: 24.0
Step 200 SSE: 0.0	validation: 24.0

Tabela D.3: Reconhecimento do 3º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	--	2	1	--	3
3	1	--	2	1	--	1
4	--	--	--	5	--	--
5	--	--	--	--	2	1
N.R. = Não reconhecido						

Para treinamento do quarto conjunto de testes, obteve-se a curva da Figura D.4, apresentando o registro de testes da listagem D.4 e com a tabela de reconhecimento D.4.

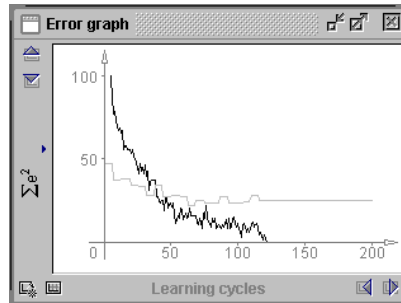


Figura D.4: Curva de erro para o 4º conjunto de testes

Listagem D.4: Registro dos testes do 4º grupo

opened at: Sun Dec 07 15:17:39 BRST 2003	
Step 20 SSE: 53.0	validation: 34.0
Step 40 SSE: 27.0	validation: 34.0
Step 60 SSE: 15.0	validation: 27.0
Step 80 SSE: 9.0	validation: 23.0
Step 100 SSE: 11.0	validation: 23.0
Step 120 SSE: 4.0	validation: 25.0
Step 140 SSE: 0.0	validation: 25.0
Step 160 SSE: 0.0	validation: 25.0
Step 180 SSE: 0.0	validation: 25.0
Step 200 SSE: 0.0	validation: 25.0

Tabela D.4: Reconhecimento do 4º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	1	2	--	--	--	2
3	--	--	1	--	--	6
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do quinto conjunto de testes, obteve-se a curva da Figura D.5, apresentando o registro de testes da listagem D.5 e com a tabela de reconhecimento D.5.

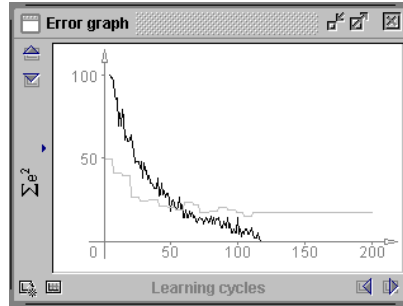


Figura D.5: Curva de erro para o 5º conjunto de testes

Listagem D.5: Registro dos testes do 5º grupo

opened at: Sun Dec 07 15:28:58 BRST 2003	
Step 20 SSE: 64.0	validation: 26.0
Step 40 SSE: 38.0	validation: 25.0
Step 60 SSE: 14.0	validation: 23.0
Step 80 SSE: 15.0	validation: 18.0
Step 100 SSE: 14.0	validation: 17.0
Step 120 SSE: 0.0	validation: 17.0
Step 140 SSE: 0.0	validation: 17.0
Step 160 SSE: 0.0	validation: 17.0
Step 180 SSE: 0.0	validation: 17.0
Step 200 SSE: 0.0	validation: 17.0

Tabela D.5: Reconhecimento do 5º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	--	1	--	--	4
3	--	--	2	--	--	4
4	--	--	--	5	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do sexto conjunto de testes, obteve-se a curva da Figura D.6, apresentando o registro de testes da listagem D.6 e com a tabela de reconhecimento D.6.

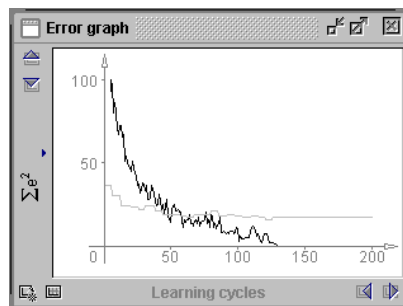


Figura D.6: Curva de erro para o 6º conjunto de testes

Listagem D.6: Registro dos testes do 6º grupo

opened at: Sun Dec 07 15:39:01 BRST 2003		
Step 20 SSE:	45.0	validation: 23.0
Step 40 SSE:	25.0	validation: 21.0
Step 60 SSE:	16.0	validation: 18.0
Step 80 SSE:	19.0	validation: 21.0
Step 100 SSE:	10.0	validation: 18.0
Step 120 SSE:	7.0	validation: 17.0
Step 140 SSE:	0.0	validation: 17.0
Step 160 SSE:	0.0	validation: 17.0
Step 180 SSE:	0.0	validation: 17.0
Step 200 SSE:	0.0	validation: 17.0

Tabela D.6: Reconhecimento do 6º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	--	2	--	--	3
3	--	1	2	--	--	3
4	--	--	--	5	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do sétimo conjunto de testes, obteve-se a curva da Figura D.7, apresentando o registro de testes da listagem D.7 e com a tabela de reconhecimento D.7.

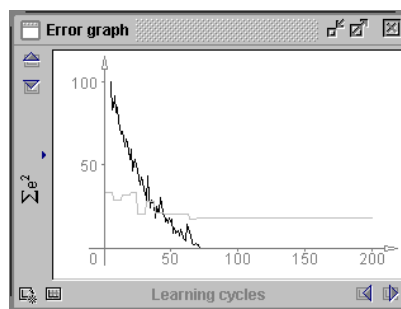


Figura D.7: Curva de erro para o 7º conjunto de testes

Listagem D.7: Registro dos testes do 7º grupo

opened at: Sun Dec 07 15:50:13 BRST 2003		
Step 20 SSE:	59.0	validation: 33.0
Step 40 SSE:	22.0	validation: 21.0
Step 60 SSE:	5.0	validation: 20.0
Step 80 SSE:	0.0	validation: 18.0
Step 100 SSE:	0.0	validation: 18.0
Step 120 SSE:	0.0	validation: 18.0

Step 140 SSE: 0.0	validation: 18.0
Step 160 SSE: 0.0	validation: 18.0
Step 180 SSE: 0.0	validation: 18.0
Step 200 SSE: 0.0	validation: 18.0

Tabela D.7: Reconhecimento do 7º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	1	1	--	1	--	2
3	--	1	2	--	--	3
4	--	--	--	5	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do oitavo conjunto de testes, obteve-se a curva da Figura D.8, apresentando o registro de testes da listagem D.8 e com a tabela de reconhecimento D.8.

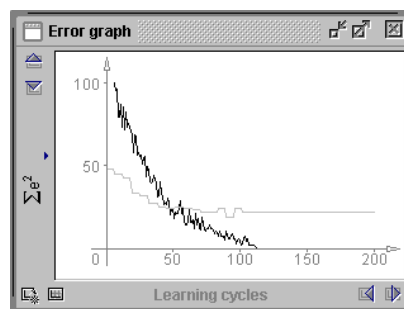


Figura D.8: Curva de erro para o 8º conjunto de testes

Listagem D.8: Registro dos testes do 8º grupo

opened at: Sun Dec 07 16:00:25 BRST 2003	
Step 20 SSE: 58.0	validation: 33.0
Step 40 SSE: 40.0	validation: 25.0
Step 60 SSE: 14.0	validation: 24.0
Step 80 SSE: 10.0	validation: 22.0
Step 100 SSE: 4.0	validation: 24.0
Step 120 SSE: 0.0	validation: 22.0
Step 140 SSE: 0.0	validation: 22.0
Step 160 SSE: 0.0	validation: 22.0
Step 180 SSE: 0.0	validation: 22.0
Step 200 SSE: 0.0	validation: 22.0

Tabela D.8: Reconhecimento do 8º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	2	1	--	--	2
3	--	--	4	--	--	2
4	--	--	--	5	--	--
5	--	--	--	1	1	1
N.R. = Não reconhecido						

Para treinamento do nono conjunto de testes, obteve-se a curva da Figura D.9, apresentando o registro de testes da listagem D.9 e com a tabela de reconhecimento D.9.

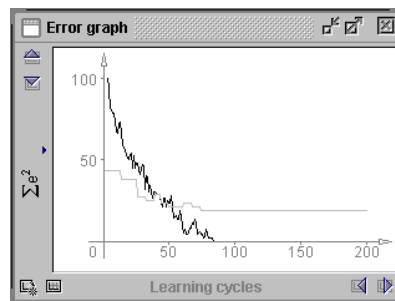


Figura D.9: Curva de erro para o 9º conjunto de testes

Listagem D.9: Registro dos testes do 9º grupo

opened at: Sun Dec 07 16:12:20 BRST 2003	
Step 20 SSE: 53.0	validation: 38.0
Step 40 SSE: 29.0	validation: 29.0
Step 60 SSE: 9.0	validation: 21.0
Step 80 SSE: 2.0	validation: 19.0
Step 100 SSE: 0.0	validation: 19.0
Step 120 SSE: 0.0	validation: 19.0
Step 140 SSE: 0.0	validation: 19.0
Step 160 SSE: 0.0	validation: 19.0
Step 180 SSE: 0.0	validation: 19.0
Step 200 SSE: 0.0	validation: 19.0

Tabela D.9: Reconhecimento do 9º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	2	1	--	--	2
3	--	--	4	--	--	2
4	--	--	--	5	--	--
5	--	--	--	--	2	1
N.R. = Não reconhecido						

Para treinamento do décimo conjunto de testes, obteve-se a curva da Figura D.10, apresentando o registro de testes da listagem D.10 e com a tabela de reconhecimento D.10.

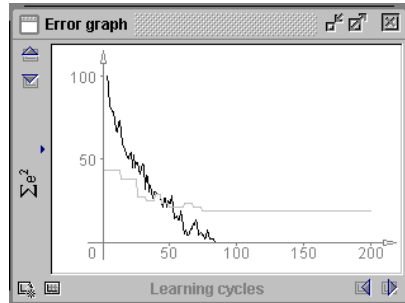


Figura D.10: Curva de erro para o 10^o conjunto de testes

Listagem D.10: Registro dos testes do 10^o grupo

opened at: Sun Dec 07 16:21:49 BRST 2003	
Step 20 SSE: 49.0	validation: 37.0
Step 40 SSE: 23.0	validation: 33.0
Step 60 SSE: 11.0	validation: 33.0
Step 80 SSE: 0.0	validation: 34.0
Step 100 SSE: 0.0	validation: 34.0
Step 120 SSE: 0.0	validation: 34.0
Step 140 SSE: 0.0	validation: 34.0
Step 160 SSE: 0.0	validation: 34.0
Step 180 SSE: 0.0	validation: 34.0
Step 200 SSE: 0.0	validation: 34.0

Tabela D.10: Reconhecimento do 10^o conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	1	--	--	--	4
3	1	--	2	1	1	--
4	--	--	1	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Apêndice E

Curvas de treinamento e testes de validação para a Transformada de Radon

Seguem as curvas de treinamento, os arquivos de registro e os resultados parciais realizados durante os testes com a transformada. Os testes seguiram um padrão de validação cruzada a 10%, significando que há 10 testes parciais realizados por modelo de transformada.

Para o treinamento, utilizou-se o algoritmo *Backpropagation* (retro-propagação), definido com $\eta=0,2$ e $d_{\max}=0,1$.

Para treinamento do primeiro conjunto de testes, obteve-se a curva da Figura E.1, apresentando o registro de testes da listagem E.1 e com a tabela de reconhecimento E.1.

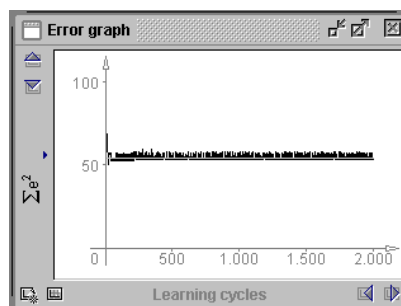


Figura E.1: Curva de erro para o 1º conjunto de testes

Listagem E.1: Registro dos testes do 1º grupo

opened at: Sat Nov 29 18:25:27 BRST 2003	
Step 200 SSE: 52.0	validation: 54.0
Step 400 SSE: 53.0	validation: 54.0
Step 600 SSE: 56.0	validation: 54.0
Step 800 SSE: 56.0	validation: 54.0
Step 1000 SSE: 55.0	validation: 54.0

Step 1200 SSE: 56.0	validation: 54.0
Step 1400 SSE: 53.0	validation: 54.0
Step 1600 SSE: 56.0	validation: 54.0
Step 1800 SSE: 54.0	validation: 54.0
Step 2000 SSE: 53.0	validation: 54.0

Tabela E.1: Reconhecimento do 1º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	8	1	2	--	--	1
2	--	5	--	--	--	--
3	--	2	4	--	--	--
4	--	--	1	4	--	--
5	--	--	1	--	2	--
N.R. = Não reconhecido						

Para treinamento do segundo conjunto de testes, obteve-se a curva da Figura E.2, apresentando o registro de testes da listagem E.2 e com a tabela de reconhecimento E.2.

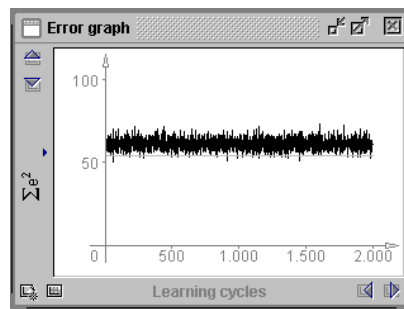


Figura E.2: Curva de erro para o 2º conjunto de testes

Listagem E.2: Registro dos testes do 2º grupo

Topological order selected		
Backpropagation selected		
Step 200 SSE: 61.0	validation:	54.0
Step 400 SSE: 59.0	validation:	54.0
Step 600 SSE: 58.0	validation:	54.0
Step 800 SSE: 58.0	validation:	54.0
Step 1000 SSE:	60.0	validation: 54.0
Step 1200 SSE:	59.0	validation: 54.0
Step 1400 SSE:	64.0	validation: 54.0
Step 1600 SSE:	64.0	validation: 54.0
Step 1800 SSE:	60.0	validation: 54.0
Step 2000 SSE:	61.0	validation: 54.0

Tabela E.2: Reconhecimento do 2º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	8	3	--	--	--	--
2	--	4	1	--	--	--
3	--	--	4	1	--	1
4	--	--	2	3	--	--
5	--	--	1	--	1	--
N.R. = Não reconhecido						

Para treinamento do terceiro conjunto de testes, obteve-se a curva da Figura E.3, apresentando o registro de testes da listagem E.3 e com a tabela de reconhecimento E.3.

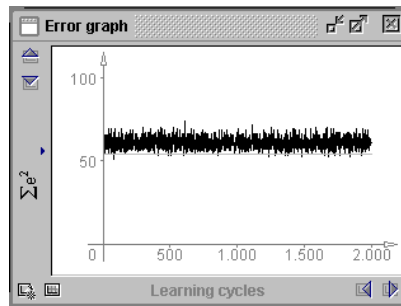


Figura E.3: Curva de erro para o 3º conjunto de testes

Tabela E.3: Reconhecimento do 3º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	1	1	--	--	--
2	1	2	1	--	--	1
3	--	--	4	1	--	1
4	--	--	1	4	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Listagem E.3: Registro dos testes do 3º grupo

```

opened at: Sun Nov 30 14:54:43 BRST 2003
Step 200 SSE: 61.0 validation: 54.0
Step 400 SSE: 64.0 validation: 54.0
Step 600 SSE: 56.0 validation: 54.0
Step 800 SSE: 57.0 validation: 54.0
Step 1000 SSE: 67.0 validation: 54.0
Step 1200 SSE: 63.0 validation: 54.0
Step 1400 SSE: 63.0 validation: 54.0
Step 1600 SSE: 63.0 validation: 54.0

```

Step 1800 SSE: 62.0 validation: 54.0
Step 2000 SSE: 61.0 validation: 54.0

Para treinamento do quarto conjunto de testes, obteve-se a curva da Figura E.4, apresentando o registro de testes da listagem E.4 e com a tabela de reconhecimento E.4.

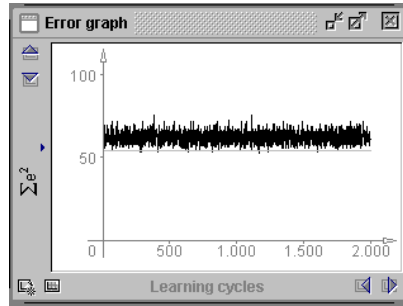


Figura E.4: Curva de erro para o 4º conjunto de testes

Listagem E.4: Registro dos testes do 4º grupo

opened at: Sun Nov 30 15:21:49 BRST 2003
Step 200 SSE: 65.0 validation: 54.0
Step 400 SSE: 66.0 validation: 54.0
Step 600 SSE: 55.0 validation: 54.0
Step 800 SSE: 67.0 validation: 54.0
Step 1000 SSE: 60.0 validation: 54.0
Step 1200 SSE: 65.0 validation: 54.0
Step 1400 SSE: 61.0 validation: 54.0
Step 1600 SSE: 59.0 validation: 54.0
Step 1800 SSE: 65.0 validation: 54.0
Step 2000 SSE: 61.0 validation: 54.0

Tabela E.4: Reconhecimento do 4º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	2	--	--	--	--
2	--	4	--	--	--	1
3	--	2	4	--	--	--
4	--	--	2	3	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do quinto conjunto de testes, obteve-se a curva da Figura E.5, apresentando o registro de testes da listagem E.5 e com a tabela de reconhecimento E.5.

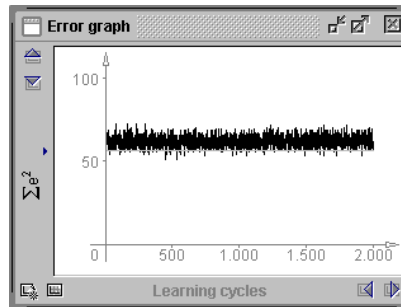


Figura E.5: Curva de erro para o 5º conjunto de testes

Listagem E.5: Registro dos testes do 5º grupo

```

opened at: Sun Nov 30 15:35:00 BRST 2003
Step 200 SSE: 65.0 validation: 56.0
Step 400 SSE: 56.0 validation: 56.0
Step 600 SSE: 66.0 validation: 56.0
Step 800 SSE: 58.0 validation: 56.0
Step 1000 SSE: 65.0 validation: 56.0
Step 1200 SSE: 67.0 validation: 56.0
Step 1400 SSE: 63.0 validation: 56.0
Step 1600 SSE: 62.0 validation: 56.0
Step 1800 SSE: 62.0 validation: 56.0
Step 2000 SSE: 57.0 validation: 56.0

```

Tabela E.5: Reconhecimento do 5º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	8	2	1	--	--	--
2	1	3	--	--	--	1
3	--	1	4	1	--	--
4	--	--	1	4	--	--
5	--	--	1	--	2	--
N.R. = Não reconhecido						

Para treinamento do sexto conjunto de testes, obteve-se a curva da Figura E.6, apresentando o registro de testes da listagem E.6 e com a tabela de reconhecimento E.6.

Listagem E.6: Registro dos testes do 6º grupo

```

opened at: Sun Nov 30 15:44:38 BRST 2003
Step 200 SSE: 56.0 validation: 56.0
Step 400 SSE: 55.0 validation: 56.0
Step 600 SSE: 66.0 validation: 56.0
Step 800 SSE: 65.0 validation: 56.0
Step 1000 SSE: 59.0 validation: 56.0

```

Step 1200 SSE: 61.0	validation: 56.0
Step 1400 SSE: 59.0	validation: 56.0
Step 1600 SSE: 61.0	validation: 56.0
Step 1800 SSE: 64.0	validation: 56.0
Step 2000 SSE: 63.0	validation: 56.0

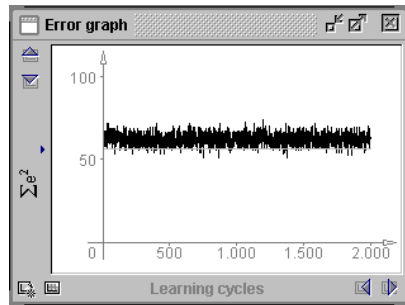


Figura E.6: Curva de erro para o 6º conjunto de testes

Tabela E.6: Reconhecimento do 6º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	1	1	--	--
2	1	3	--	--	--	1
3	--	2	2	2	--	--
4	--	--	2	3	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sétimo conjunto de testes, obteve-se a curva da Figura E.7, apresentando o registro de testes da listagem E.7 e com a tabela de reconhecimento E.7.

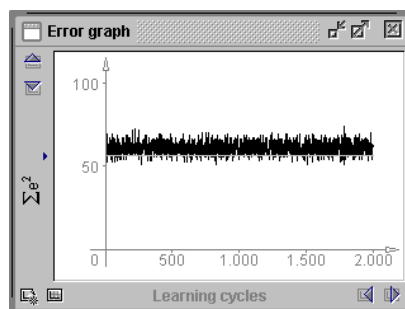


Figura E.7: Curva de erro para o 7º conjunto de testes

Listagem E.7: Registro dos testes do 7º grupo

opened at: Sun Nov 30 16:07:28 BRST 2003	
Step 200 SSE: 58.0	validation: 56.0
Step 400 SSE: 59.0	validation: 56.0

Step 600 SSE: 63.0	validation: 56.0
Step 800 SSE: 58.0	validation: 56.0
Step 1000 SSE: 59.0	validation: 56.0
Step 1200 SSE: 60.0	validation: 56.0
Step 1400 SSE: 59.0	validation: 56.0
Step 1600 SSE: 53.0	validation: 56.0
Step 1800 SSE: 56.0	validation: 56.0
Step 2000 SSE: 62.0	validation: 56.0

Tabela E.7: Reconhecimento do 7º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	1	1	--	--	--
2	1	2	1	--	--	1
3	1	1	4	--	--	--
4	--	--	1	4	--	--
5	--	--	--	--	3	--

N.R. = Não reconhecido

Para treinamento do oitavo conjunto de testes, obteve-se a curva da Figura E.8, apresentando o registro de testes da listagem E.8 e com a tabela de reconhecimento E.8.

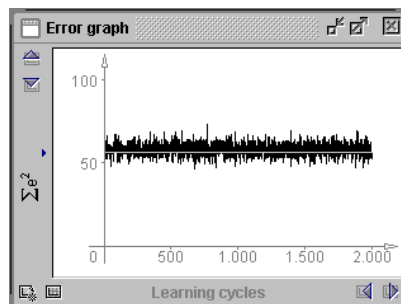


Figura E.8: Curva de erro para o 8º conjunto de testes

Listagem E.8: Registro dos testes do 8º grupo

opened at: Sun Nov 30 16:19:26 BRST 2003	
Step 200 SSE: 54.0	validation: 56.0
Step 400 SSE: 60.0	validation: 56.0
Step 600 SSE: 53.0	validation: 56.0
Step 800 SSE: 57.0	validation: 56.0
Step 1000 SSE: 59.0	validation: 56.0
Step 1200 SSE: 63.0	validation: 56.0
Step 1400 SSE: 61.0	validation: 56.0
Step 1600 SSE: 60.0	validation: 56.0
Step 1800 SSE: 54.0	validation: 56.0
Step 2000 SSE: 52.0	validation: 56.0

Tabela E.8: Reconhecimento do 8º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	3	--	--	--	--
2	--	4	1	--	--	--
3	--	1	4	1	--	--
4	--	1	1	3	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do nono conjunto de testes, obteve-se a curva da Figura E.9, apresentando o registro de testes da listagem E.9 e com a tabela de reconhecimento E.9.

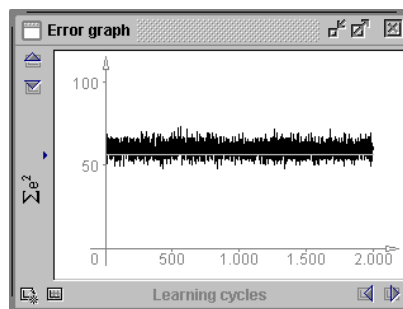


Figura E.9: Curva de erro para o 9º conjunto de testes

Tabela E.9: Reconhecimento do 9º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	2	--	--	--	--
2	--	4	1	--	--	--
3	--	2	3	--	--	1
4	--	--	4	1	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Listagem E.9: Registro dos testes do 9º grupo

```

opened at: Sun Nov 30 15:40:18 BRST 2003
Step 200 SSE: 55.0 validation: 56.0
Step 400 SSE: 57.0 validation: 56.0
Step 600 SSE: 58.0 validation: 56.0
Step 800 SSE: 57.0 validation: 56.0
Step 1000 SSE: 66.0 validation: 56.0
Step 1200 SSE: 50.0 validation: 56.0
Step 1400 SSE: 61.0 validation: 56.0

```

Step 1600 SSE: 54.0	validation: 56.0
Step 1800 SSE: 58.0	validation: 56.0
Step 2000 SSE: 60.0	validation: 56.0

Para treinamento do décimo conjunto de testes, obteve-se a curva da Figura E.10, apresentando o registro de testes da listagem E.10 e com a tabela de reconhecimento E.10.

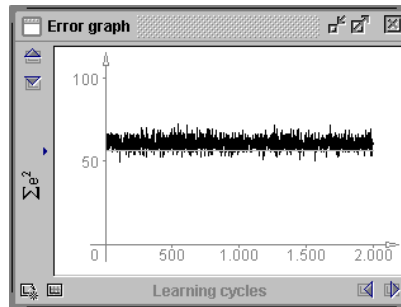


Figura E.10: Curva de erro para o 10^o conjunto de testes

Listagem E.10: Registro dos testes do 10^o grupo

opened at: Sun Nov 30 14:58:01 BRST 2003	
Step 200 SSE: 57.0	validation: 56.0
Step 400 SSE: 66.0	validation: 56.0
Step 600 SSE: 63.0	validation: 56.0
Step 800 SSE: 57.0	validation: 56.0
Step 1000 SSE: 57.0	validation: 56.0
Step 1200 SSE: 63.0	validation: 56.0
Step 1400 SSE: 61.0	validation: 56.0
Step 1600 SSE: 59.0	validation: 56.0
Step 1800 SSE: 60.0	validation: 56.0
Step 2000 SSE: 61.0	validation: 56.0

Tabela E.10: Reconhecimento do 10^o conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	9	2	1	--	--	--
2	--	3	2	--	--	--
3	--	--	4	--	--	1
4	--	--	2	4	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Apêndice F

Curvas de treinamento e testes de validação para a Transformada de Zernike

Seguem as curvas de treinamento, os arquivos de registro e os resultados parciais realizados durante os testes com a transformada. Os testes seguiram um padrão de validação cruzada a 10%, significando que há 10 testes parciais realizados por modelo de transformada.

Para o treinamento, utilizou-se o algoritmo *Backpropagation* (retro-propagação), definido com $\eta=0,2$ e $d_{\max}=0,1$.

Para treinamento do primeiro conjunto de testes, obteve-se a curva da Figura F.1, apresentando o registro de testes da listagem F.1 e com a tabela de reconhecimento F.1.

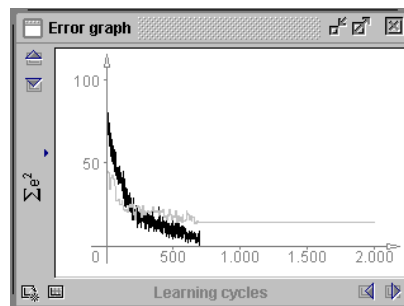


Figura F.1: Curva de erro para o 1º conjunto de testes

Listagem F.1: Registro dos testes do 1º grupo

opened at: Sat Nov 29 17:39:26 BRST 2003	
Step 200 SSE: 27.0	validation: 25.0
Step 400 SSE: 14.0	validation: 19.0
Step 600 SSE: 8.0	validation: 21.0
Step 800 SSE: 0.0	validation: 14.0
Step 1000 SSE: 0.0	validation: 14.0

Step 1200 SSE: 0.0	validation: 14.0
Step 1400 SSE: 0.0	validation: 14.0
Step 1600 SSE: 0.0	validation: 14.0
Step 1800 SSE: 0.0	validation: 14.0
Step 2000 SSE: 0.0	validation: 14.0

Tabela F.1: Reconhecimento do 1º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	--	--	--	2
2	--	4	1	--	--	--
3	--	--	5	--	--	1
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do segundo conjunto de testes, obteve-se a curva da Figura F.2, apresentando o registro de testes da listagem F.2 e com a tabela de reconhecimento F.2.

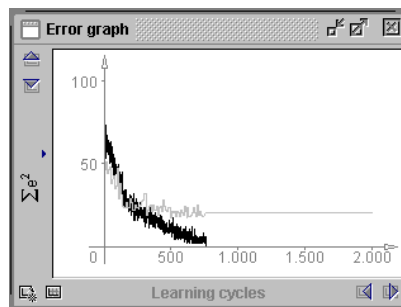


Figura F.2: Curva de erro para o 2º conjunto de testes

Listagem F.2: Registro dos testes do 2º grupo

opened at: Sat Nov 29 17:55:47 BRST 2003	
Step 200 SSE: 22.0	validation: 29.0
Step 400 SSE: 17.0	validation: 24.0
Step 600 SSE: 5.0	validation: 19.0
Step 800 SSE: 0.0	validation: 20.0
Step 1000 SSE: 0.0	validation: 20.0
Step 1200 SSE: 0.0	validation: 20.0
Step 1400 SSE: 0.0	validation: 20.0
Step 1600 SSE: 0.0	validation: 20.0
Step 1800 SSE: 0.0	validation: 20.0
Step 2000 SSE: 0.0	validation: 20.0

Tabela F.2: Reconhecimento do 2º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	1	--	--	4	--	--
5	--	--	--	--	1	1
N.R. = Não reconhecido						

Para treinamento do terceiro conjunto de testes, obteve-se a curva da Figura F.3, apresentando o registro de testes da listagem F.3 e com a tabela de reconhecimento F.3.

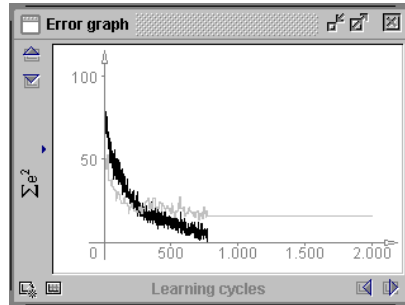


Figura F.3: Curva de erro para o 3º conjunto de testes

Listagem F.3: Registro dos testes do 3º grupo

opened at: Sat Nov 29 18:11:39 BRST 2003	
Step 200 SSE: 25.0	validation: 21.0
Step 400 SSE: 16.0	validation: 21.0
Step 600 SSE: 10.0	validation: 21.0
Step 800 SSE: 0.0	validation: 16.0
Step 1000 SSE: 0.0	validation: 16.0
Step 1200 SSE: 0.0	validation: 16.0
Step 1400 SSE: 0.0	validation: 16.0
Step 1600 SSE: 0.0	validation: 16.0
Step 1800 SSE: 0.0	validation: 16.0
Step 2000 SSE: 0.0	validation: 16.0

Tabela F.3: Reconhecimento do 3º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	4	--	--	--	6
3	--	--	4	--	--	1
4	--	--	--	4	--	1
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do quarto conjunto de testes, obteve-se a curva da Figura F.4, apresentando o registro de testes da listagem F.4 e com a tabela de reconhecimento F.4.

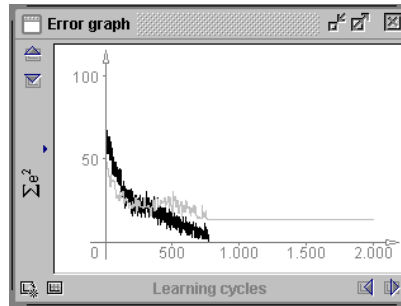


Figura F.4: Curva de erro para o 4º conjunto de testes

Listagem F.4: Registro dos testes do 4º grupo

opened at: Sat Nov 29 18:30:06 BRST 2003	
Step 200 SSE: 27.0	validation: 20.0
Step 400 SSE: 20.0	validation: 21.0
Step 600 SSE: 6.0	validation: 25.0
Step 800 SSE: 0.0	validation: 13.0
Step 1000 SSE: 0.0	validation: 13.0
Step 1200 SSE: 0.0	validation: 13.0
Step 1400 SSE: 0.0	validation: 13.0
Step 1600 SSE: 0.0	validation: 13.0
Step 1800 SSE: 0.0	validation: 13.0
Step 2000 SSE: 0.0	validation: 13.0

Tabela F.4: Reconhecimento do 4º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	10	--	--	1	--	--
2	--	4	1	--	--	--
3	1	--	5	--	--	--
4	--	--	--	4	--	1
5	--	--	--	--	2	1
N.R. = Não reconhecido						

Para treinamento do quinto conjunto de testes, obteve-se a curva da Figura F.5, apresentando o registro de testes da listagem F.5 e com a tabela de reconhecimento F.5.

Listagem F.5: Registro dos testes do 5º grupo

opened at: Sat Nov 29 18:44:07 BRST 2003	
Step 200 SSE: 23.0	validation: 29.0
Step 400 SSE: 12.0	validation: 18.0
Step 600 SSE: 5.0	validation: 16.0

Step 800 SSE: 0.0	validation: 17.0
Step 1000 SSE: 0.0	validation: 17.0
Step 1200 SSE: 0.0	validation: 17.0
Step 1400 SSE: 0.0	validation: 17.0
Step 1600 SSE: 0.0	validation: 17.0
Step 1800 SSE: 0.0	validation: 17.0
Step 2000 SSE: 0.0	validation: 17.0

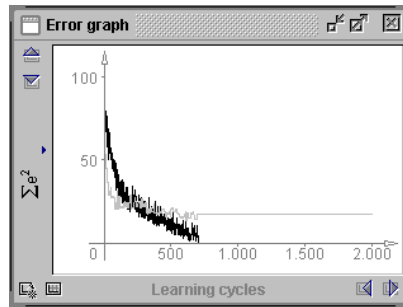


Figura F.5: Curva de erro para o 5º conjunto de testes

Tabela F.5: Reconhecimento do 5º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	3	1	--	--	1
3	--	--	5	--	--	1
4	1	--	--	3	--	1
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sexto conjunto de testes, obteve-se a curva da Figura F.6, apresentando o registro de testes da listagem F.6 e com a tabela de reconhecimento F.6.

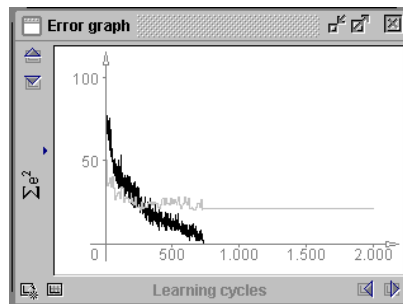


Figura F.6: Curva de erro para o 6º conjunto de testes

Listagem F.6: Registro dos testes do 6º grupo

opened at: Sat Nov 29 18:58:38 BRST 2003	
Step 200 SSE: 25.0	validation: 24.0
Step 400 SSE: 15.0	validation: 22.0
Step 600 SSE: 6.0	validation: 25.0
Step 800 SSE: 0.0	validation: 21.0
Step 1000 SSE: 0.0	validation: 21.0
Step 1200 SSE: 0.0	validation: 21.0
Step 1400 SSE: 0.0	validation: 21.0
Step 1600 SSE: 0.0	validation: 21.0
Step 1800 SSE: 0.0	validation: 21.0
Step 2000 SSE: 0.0	validation: 21.0

Tabela F.6: Reconhecimento do 6º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	1
2	--	3	1	--	--	1
3	--	--	6	--	--	--
4	--	--	--	4	--	1
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sétimo conjunto de testes, obteve-se a curva da Figura F.7, apresentando o registro de testes da listagem F.7 e com a tabela de reconhecimento F.7.

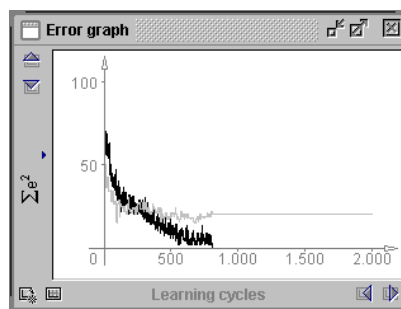


Figura F.7: Curva de erro para o 7º conjunto de testes

Listagem F.7: Registro dos testes do 7º grupo

opened at: Sat Nov 29 19:37:47 BRST 2003	
Step 200 SSE: 23.0	validation: 22.0
Step 400 SSE: 18.0	validation: 22.0
Step 600 SSE: 5.0	validation: 17.0
Step 800 SSE: 2.0	validation: 19.0
Step 1000 SSE: 0.0	validation: 20.0
Step 1200 SSE: 0.0	validation: 20.0

Step 1400 SSE: 0.0	validation: 20.0
Step 1600 SSE: 0.0	validation: 20.0
Step 1800 SSE: 0.0	validation: 20.0
Step 2000 SSE: 0.0	validation: 20.0

Tabela F.7: Reconhecimento do 7º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	1
2	--	3	1	--	--	1
3	--	--	6	--	--	--
4	--	--	--	4	--	1
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do oitavo conjunto de testes, obteve-se a curva da Figura F.8, apresentando o registro de testes da listagem F.8 e com a tabela de reconhecimento F.8.

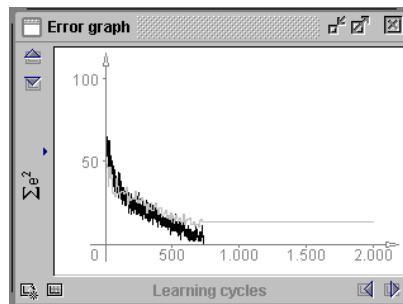


Figura F.8: Curva de erro para o 8º conjunto de testes

Listagem F.8: Registro dos testes do 8º grupo

opened at: Sat Nov 29 21:09:04 BRST 2003	
Step 200 SSE: 21.0	validation: 27.0
Step 400 SSE: 16.0	validation: 18.0
Step 600 SSE: 7.0	validation: 14.0
Step 800 SSE: 0.0	validation: 13.0
Step 1000 SSE: 0.0	validation: 13.0
Step 1200 SSE: 0.0	validation: 13.0
Step 1400 SSE: 0.0	validation: 13.0
Step 1600 SSE: 0.0	validation: 13.0
Step 1800 SSE: 0.0	validation: 13.0
Step 2000 SSE: 0.0	validation: 13.0

Tabela F.8: Reconhecimento do 8º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	4	--	1	--	--
3	--	--	6	--	--	--
4	--	--	--	2	--	3
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do nono conjunto de testes, obteve-se a curva da Figura F.9, apresentando o registro de testes da listagem F.9 e com a tabela de reconhecimento F.9.

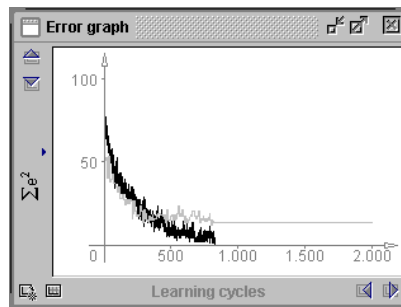


Figura F.9: Curva de erro para o 9º conjunto de testes

Tabela F.9: Reconhecimento do 9º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	4	--	--	--	1
3	--	1	5	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Listagem F.9: Registro dos testes do 9º grupo

opened at: Sat Nov 29 21:22:15 BRST 2003	
Step 200 SSE: 35.0	validation: 27.0
Step 400 SSE: 14.0	validation: 17.0
Step 600 SSE: 6.0	validation: 17.0
Step 800 SSE: 6.0	validation: 15.0
Step 1000 SSE: 0.0	validation: 13.0
Step 1200 SSE: 0.0	validation: 13.0

Step 1400 SSE: 0.0	validation: 13.0
Step 1600 SSE: 0.0	validation: 13.0
Step 1800 SSE: 0.0	validation: 13.0
Step 2000 SSE: 0.0	validation: 13.0

Para treinamento do décimo conjunto de testes, obteve-se a curva da Figura F.10, apresentando o registro de testes da listagem F.10 e com a tabela de reconhecimento F.10.

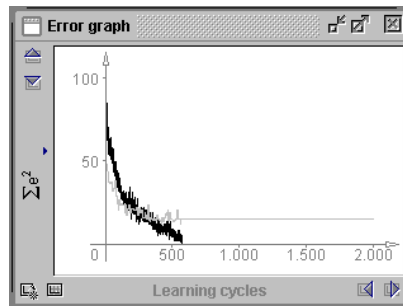


Figura F.10: Curva de erro para o 10^o conjunto de testes

Listagem F.10: Registro dos testes do 10^o grupo

opened at: Sat Nov 29 21:34:52 BRST 2003	
Step 200 SSE: 20.0	validation: 21.0
Step 400 SSE: 8.0	validation: 15.0
Step 600 SSE: 0.0	validation: 15.0
Step 800 SSE: 0.0	validation: 15.0
Step 1000 SSE: 0.0	validation: 15.0
Step 1200 SSE: 0.0	validation: 15.0
Step 1400 SSE: 0.0	validation: 15.0
Step 1600 SSE: 0.0	validation: 15.0
Step 1800 SSE: 0.0	validation: 15.0
Step 2000 SSE: 0.0	validation: 15.0

Tabela F.10: Reconhecimento do 10^o conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	1	--	--
2	--	5	--	--	--	--
3	--	--	4	--	1	--
4	--	--	1	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Apêndice G

Curvas de treinamento e testes de validação para as Transformadas de Zernike e Chebyshev

Seguem as curvas de treinamento, os arquivos de registro e os resultados parciais realizados durante os testes com a transformada. Os testes seguiram um padrão de validação cruzada a 10%, significando que há 10 testes parciais realizados por modelo de transformada.

Para o treinamento, utilizou-se o algoritmo *Backpropagation with momentum* (retropropagação com momento), definido com $\eta=0,2$, $\mu=0,5$, $c=0,1$ e $d_{\max}=0,1$.

Para treinamento do primeiro conjunto de testes, obteve-se a curva da Figura G.1, apresentando o registro de testes da listagem G.1 e com a tabela de reconhecimento G.1.

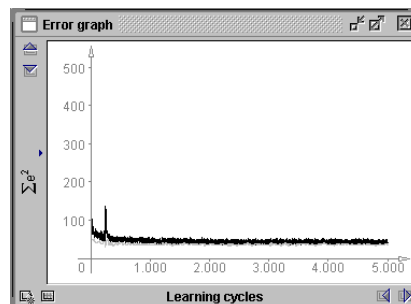


Figura G.1: Curva de erro para o 1º conjunto de testes

Listagem G.1: Registro dos testes do 1º grupo

opened at: Sat Aug 30 15:27:02 GMT-03:00 2003

Step 500 SSE: 49.0 validation: 39.0

Step 1000 SSE: 43.0 validation: 34.0

Step 1500 SSE: 43.0 validation: 37.0

Step 2000 SSE: 46.0 validation: 34.0

Step 2500 SSE: 45.0 validation: 35.0

Step 3000 SSE: 41.0	validation: 34.0
Step 3500 SSE: 50.0	validation: 36.0
Step 4000 SSE: 43.0	validation: 35.0
Step 4500 SSE: 39.0	validation: 37.0
Step 5000 SSE: 42.0	validation: 36.0

Tabela G.1: Reconhecimento do 1º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	7	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do segundo conjunto de testes, obteve-se a curva da Figura G.2, apresentando o registro de testes da listagem G.2 e com a tabela de reconhecimento G.2.

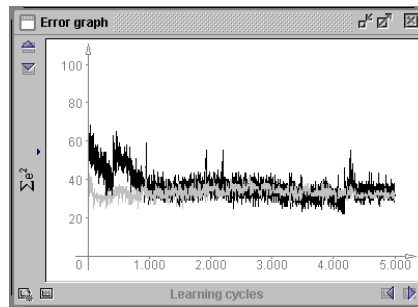


Figura G.2: Curva de erro para o 2º conjunto de testes

Listagem G.2: Registro dos testes do 2º grupo

opened at: Sat Aug 30 14:43:10 BRT 2003	
Step 500 SSE: 52.0	validation: 33.0
Step 1000 SSE: 34.0	validation: 32.0
Step 1500 SSE: 39.0	validation: 37.0
Step 2000 SSE: 36.0	validation: 34.0
Step 2500 SSE: 31.0	validation: 35.0
Step 3000 SSE: 35.0	validation: 33.0
Step 3500 SSE: 32.0	validation: 34.0
Step 4000 SSE: 30.0	validation: 33.0
Step 4500 SSE: 37.0	validation: 32.0
Step 5000 SSE: 31.0	validation: 33.0

Tabela G.2: Reconhecimento do 2º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	6	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do terceiro conjunto de testes, obteve-se a curva da Figura G.3, apresentando o registro de testes da listagem G.3 e com a tabela de reconhecimento G.3.

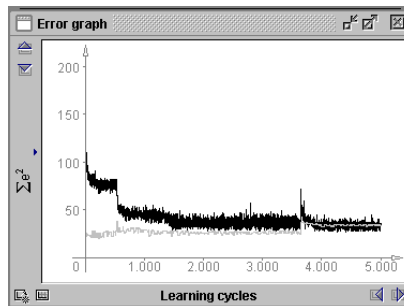


Figura G.3: Curva de erro para o 3º conjunto de testes

Listagem G.3: Registro dos testes do 3º grupo

opened at: Sat Aug 30 16:45:24 GMT-03:00 2003	
Step 500 SSE: 77.0	validation: 26.0
Step 1000 SSE: 41.0	validation: 29.0
Step 1500 SSE: 33.0	validation: 28.0
Step 2000 SSE: 35.0	validation: 25.0
Step 2500 SSE: 36.0	validation: 26.0
Step 3000 SSE: 33.0	validation: 26.0
Step 3500 SSE: 34.0	validation: 24.0
Step 4000 SSE: 32.0	validation: 33.0
Step 4500 SSE: 34.0	validation: 34.0
Step 5000 SSE: 35.0	validation: 33.0

Tabela G.3: Reconhecimento do 3º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	4	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do quarto conjunto de testes, obteve-se a curva da Figura G.4, apresentando o registro de testes da listagem G.4 e com a tabela de reconhecimento G.4.

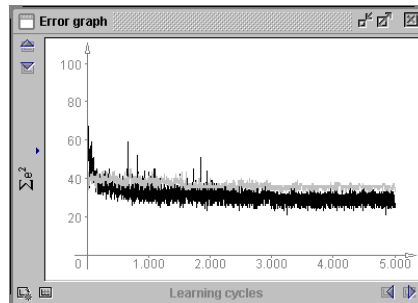


Figura G.4: Curva de erro para o 4º conjunto de testes

Listagem G.4: Registro dos testes do 4º grupo

```

opened at: Sat Aug 30 16:32:49 BRT 2003
Step 500 SSE: 33.0  validation: 38.0
Step 1000 SSE: 37.0  validation: 36.0
Step 1500 SSE: 31.0  validation: 38.0
Step 2000 SSE: 32.0  validation: 35.0
Step 2500 SSE: 31.0  validation: 38.0
Step 3000 SSE: 31.0  validation: 36.0
Step 3500 SSE: 30.0  validation: 34.0
Step 4000 SSE: 30.0  validation: 36.0
Step 4500 SSE: 32.0  validation: 36.0
Step 5000 SSE: 25.0  validation: 33.0
  
```

Tabela G.4: Reconhecimento do 4º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do quinto conjunto de testes, obteve-se a curva da Figura G.5, apresentando o registro de testes da listagem .5 e com a tabela de reconhecimento G.5.

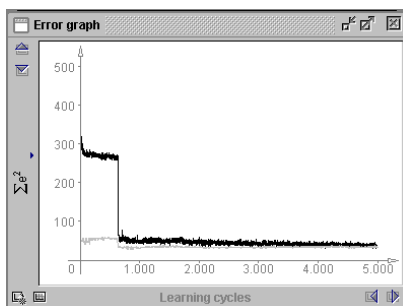


Figura G.5: Curva de erro para o 5º conjunto de testes

Listagem G.5: Registro dos testes do 5º grupo

opened at: Sun Aug 31 00:51:46 BRT 2003

Step 500 SSE: 267.0 validation: 56.0
 Step 1000 SSE: 51.0 validation: 31.0
 Step 1500 SSE: 50.0 validation: 36.0
 Step 2000 SSE: 46.0 validation: 34.0
 Step 2500 SSE: 49.0 validation: 34.0
 Step 3000 SSE: 39.0 validation: 33.0
 Step 3500 SSE: 48.0 validation: 32.0
 Step 4000 SSE: 43.0 validation: 32.0
 Step 4500 SSE: 40.0 validation: 35.0
 Step 5000 SSE: 32.0 validation: 33.0

Tabela G.5: Reconhecimento do 5º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sexto conjunto de testes, obteve-se a curva da Figura G.6, apresentando o registro de testes da listagem G.6 e com a tabela de reconhecimento G.6.

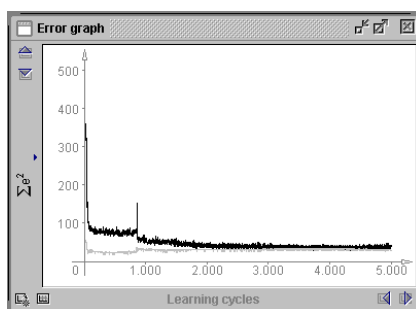


Figura G.6: Curva de erro para o 6º conjunto de testes

Listagem G.6: Registro dos testes do 6º grupo

opened at: Sun Aug 31 15:00:40 BRT 2003	
Step 500 SSE: 71.0	validation: 24.0
Step 1000 SSE: 54.0	validation: 31.0
Step 1500 SSE: 46.0	validation: 28.0
Step 2000 SSE: 46.0	validation: 30.0
Step 2500 SSE: 38.0	validation: 31.0
Step 3000 SSE: 45.0	validation: 32.0
Step 3500 SSE: 42.0	validation: 33.0
Step 4000 SSE: 36.0	validation: 33.0
Step 4500 SSE: 33.0	validation: 32.0
Step 5000 SSE: 42.0	validation: 27.0

Tabela G.6: Reconhecimento do 6º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sétimo conjunto de testes, obteve-se a curva da Figura G.7, apresentando o registro de testes da listagem G.7 e com a tabela de reconhecimento G.7.

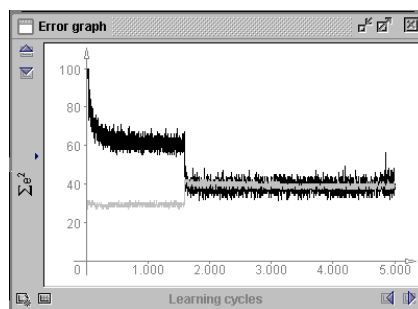


Figura G.7: Curva de erro para o 7º conjunto de testes

Listagem G.7: Registro dos testes do 7º grupo

opened at: Sat Aug 30 18:42:33 BRT 2003	
Step 500 SSE: 56.0	validation: 30.0
Step 1000 SSE: 58.0	validation: 30.0
Step 1500 SSE: 59.0	validation: 29.0
Step 2000 SSE: 36.0	validation: 40.0
Step 2500 SSE: 41.0	validation: 41.0
Step 3000 SSE: 42.0	validation: 39.0

Step 3500 SSE: 35.0	validation: 39.0
Step 4000 SSE: 40.0	validation: 40.0
Step 4500 SSE: 36.0	validation: 37.0
Step 5000 SSE: 37.0	validation: 39.0

Tabela G.7: Reconhecimento do 7º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do oitavo conjunto de testes, obteve-se a curva da Figura G.8, apresentando o registro de testes da listagem G.8 e com a tabela de reconhecimento G.8.

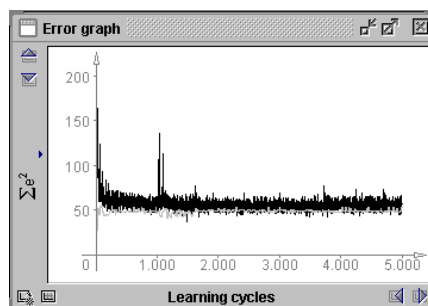


Figura G.8: Curva de erro para o 8º conjunto de testes

Listagem G.8: Registro dos testes do 8º grupo

opened at: Sat Aug 30 20:37:51 GMT-03:00 2003	
Step 500 SSE: 54.0	validation: 49.0
Step 1000 SSE: 52.0	validation: 46.0
Step 1500 SSE: 55.0	validation: 48.0
Step 2000 SSE: 53.0	validation: 49.0
Step 2500 SSE: 56.0	validation: 50.0
Step 3000 SSE: 55.0	validation: 50.0
Step 3500 SSE: 53.0	validation: 49.0
Step 4000 SSE: 56.0	validation: 48.0
Step 4500 SSE: 57.0	validation: 48.0
Step 5000 SSE: 57.0	validation: 53.0

Tabela G.8: Reconhecimento do 8º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do nono conjunto de testes, obteve-se a curva da Figura G.9, apresentando o registro de testes da listagem G.9 e com a tabela de reconhecimento G.9.

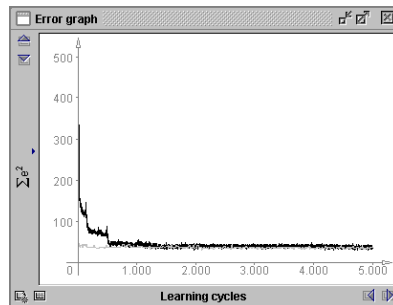


Figura G.9: Curva de erro para o 9º conjunto de testes

Tabela G.9: Reconhecimento do 9º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Listagem G.9: Registro dos testes do 9º grupo

```

opened at: Sun Aug 31 02:26:43 GMT-03:00 2003
Step 500 SSE: 63.0 validation: 37.0
Step 1000 SSE: 40.0 validation: 37.0
Step 1500 SSE: 36.0 validation: 38.0
Step 2000 SSE: 42.0 validation: 36.0
Step 2500 SSE: 34.0 validation: 38.0

```


Step 3000 SSE: 37.0	validation: 36.0
Step 3500 SSE: 41.0	validation: 36.0
Step 4000 SSE: 39.0	validation: 34.0
Step 4500 SSE: 34.0	validation: 34.0
Step 5000 SSE: 33.0	validation: 37.0

Para treinamento do décimo conjunto de testes, obteve-se a curva da Figura G.10, apresentando o registro de testes da listagem G.10 e com a tabela de reconhecimento G.10.

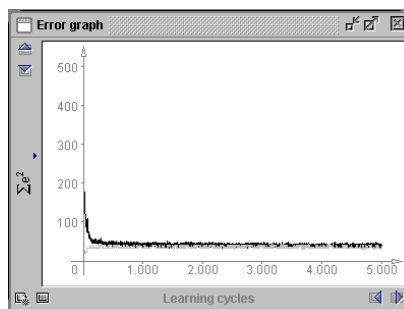


Figura G.10: Curva de erro para o 10^o conjunto de testes

Listagem G.10: Registro dos testes do 10^o grupo

opened at: Thu Dec 04 10:55:36 BRST 2003	
Step 500 SSE: 46.0	validation: 40.0
Step 1000 SSE: 41.0	validation: 34.0
Step 1500 SSE: 39.0	validation: 40.0
Step 2000 SSE: 38.0	validation: 35.0
Step 2500 SSE: 44.0	validation: 34.0
Step 3000 SSE: 41.0	validation: 40.0
Step 3500 SSE: 36.0	validation: 35.0
Step 4000 SSE: 38.0	validation: 35.0
Step 4500 SSE: 40.0	validation: 35.0
Step 5000 SSE: 41.0	validation: 34.0

Tabela G.10: Reconhecimento do 10^o conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	5	--	--	--
4	--	--	--	6	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Apêndice H

Curvas de treinamento e testes de validação para as Transformadas de Zernike e Fourier

Seguem as curvas de treinamento, os arquivos de registro e os resultados parciais realizados durante os testes com a transformada. Os testes seguiram um padrão de validação cruzada a 10%, significando que há 10 testes parciais realizados por modelo de transformada.

Para o treinamento, utilizou-se o algoritmo *Backpropagation with momentum* (retropropagação com momento), definido com $\eta=0,0002$, $\mu=0,5$, $c=0,1$ e $d_{\max}=0,1$.

Para treinamento do primeiro conjunto de testes, obteve-se a curva da Figura H.1, apresentando o registro de testes da listagem H.1 e com a tabela de reconhecimento H.1.

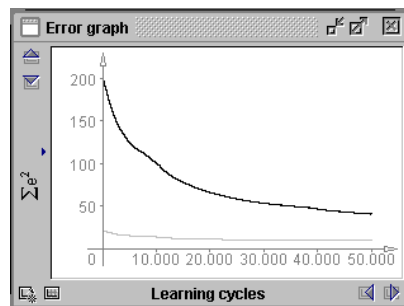


Figura H.1: Curva de erro para o 1º conjunto de testes

Listagem H.1: Registro dos testes do 1º grupo

opened at: Mon Jan 19 11:08:56 BRST 2004	
Step 5000 SSE: 123.96759796142578	validation: 15.017308235168457
Step 10000 SSE: 99.99260711669922	validation: 13.405961036682129
Step 15000 SSE: 77.65050506591797	validation: 11.435474395751953
Step 20000 SSE: 65.71058654785156	validation: 10.771419525146484
Step 25000 SSE: 58.05405044555664	validation: 10.32621955871582

Step 30000 SSE: 53.07328796386719	validation: 9.916178703308105
Step 35000 SSE: 49.65794372558594	validation: 9.546957015991211
Step 40000 SSE: 46.328338623046875	validation: 9.311796188354492
Step 45000 SSE: 42.7908821105957	validation: 9.20659065246582
Step 50000 SSE: 40.30757141113281	validation: 9.192544937133789

Tabela H.1: Reconhecimento do 1º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	1	4	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do segundo conjunto de testes, obteve-se a curva da Figura H.2, apresentando o registro de testes da listagem H.2 e com a tabela de reconhecimento H.2.

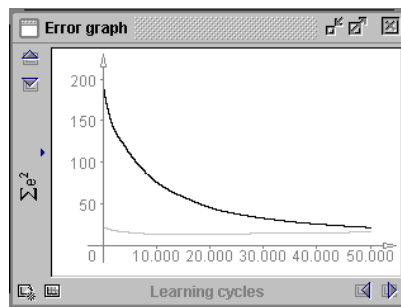


Figura H.2: Curva de erro para o 2º conjunto de testes

Listagem H.2: Registro dos testes do 2º grupo

opened at: Mon Jan 19 12:30:40 BRST 2004	
Step 5000 SSE: 110.63128662109375	validation: 15.195387840270996
Step 10000 SSE: 75.98703002929688	validation: 12.8894681930542
Step 15000 SSE: 58.15780258178711	validation: 13.049829483032227
Step 20000 SSE: 45.19396209716797	validation: 13.447929382324219
Step 25000 SSE: 37.47862243652344	validation: 13.577385902404785
Step 30000 SSE: 32.29035949707031	validation: 13.926490783691406
Step 35000 SSE: 28.286930084228516	validation: 14.242891311645508
Step 40000 SSE: 25.28924560546875	validation: 14.561919212341309
Step 45000 SSE: 22.934907913208008	validation: 15.18902587890625
Step 50000 SSE: 21.086565017700195	validation: 15.613879203796387

Tabela H.2: Reconhecimento do 2º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	1	4	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	2	--
N.R. = Não reconhecido						

Para treinamento do terceiro conjunto de testes, obteve-se a curva da Figura H.3, apresentando o registro de testes da listagem H.3 e com a tabela de reconhecimento H.3.

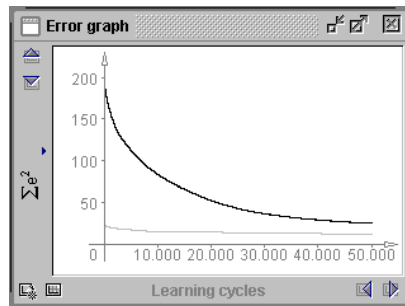


Figura H.3: Curva de erro para o 3º conjunto de testes

Listagem H.3: Registro dos testes do 3º grupo

opened at: Sun Jan 18 22:50:09 BRST 2004	
Step 5000 SSE: 112.46414184570312	validation: 16.88605308532715
Step 10000 SSE: 83.67121124267578	validation: 14.961990356445312
Step 15000 SSE: 65.47774505615234	validation: 14.095856666564941
Step 20000 SSE: 51.92000198364258	validation: 13.90936279296875
Step 25000 SSE: 42.44295120239258	validation: 13.456046104431152
Step 30000 SSE: 36.18263626098633	validation: 13.112337112426758
Step 35000 SSE: 31.84613609313965	validation: 12.678356170654297
Step 40000 SSE: 28.742231369018555	validation: 12.247517585754395
Step 45000 SSE: 26.352636337280273	validation: 11.955524444580078
Step 50000 SSE: 24.519113540649414	validation: 11.629825592041016

Tabela H.3: Reconhecimento do 3º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	6	--	--	--	--
3	--	--	4	--	1	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do quarto conjunto de testes, obteve-se a curva da Figura H.4, apresentando o registro de testes da listagem H.4 e com a tabela de reconhecimento H.4.

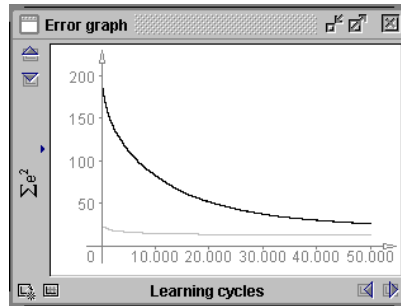


Figura H.4: Curva de erro para o 4º conjunto de testes

Listagem H.4: Registro dos testes do 4º grupo

opened at: Sun Jan 18 18:05:41 BRST 2004	
Step 5000 SSE: 111.1146011352539	validation: 16.03199005126953
Step 10000 SSE: 82.64796447753906	validation: 14.663970947265625
Step 15000 SSE: 63.280269622802734	validation: 13.831846237182617
Step 20000 SSE: 51.47857666015625	validation: 13.218230247497559
Step 25000 SSE: 42.96327209472656	validation: 12.888580322265625
Step 30000 SSE: 37.10708236694336	validation: 12.53139591217041
Step 35000 SSE: 32.81752395629883	validation: 12.336837768554688
Step 40000 SSE: 29.89203643798828	validation: 12.470572471618652
Step 45000 SSE: 27.549968719482422	validation: 12.667787551879883
Step 50000 SSE: 25.69331169128418	validation: 12.729928970336914

Tabela H.4: Reconhecimento do 4º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	3	2	--	--	--
3	--	--	4	1	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do quinto conjunto de testes, obteve-se a curva da Figura H.5, apresentando o registro de testes da listagem H.5 e com a tabela de reconhecimento H.5.

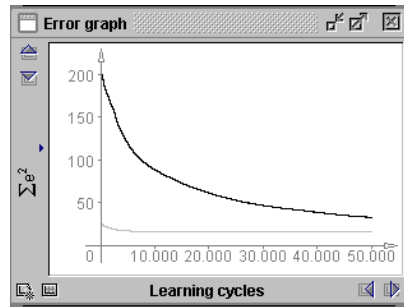


Figura H.5: Curva de erro para o 5º conjunto de testes

Listagem H.5: Registro dos testes do 5º grupo

opened at: Sun Jan 18 18:02:41 BRST 2004	
Step 5000 SSE: 119.30965423583984	validation: 16.746231079101562
Step 10000 SSE: 88.1236572265625	validation: 15.402545928955078
Step 15000 SSE: 72.4482421875	validation: 15.859447479248047
Step 20000 SSE: 61.01617431640625	validation: 15.78019905090332
Step 25000 SSE: 52.35237121582031	validation: 15.380062103271484
Step 30000 SSE: 46.37800979614258	validation: 15.110827445983887
Step 35000 SSE: 42.067752838134766	validation: 15.169549942016602
Step 40000 SSE: 38.30263137817383	validation: 15.373802185058594
Step 45000 SSE: 34.88198471069336	validation: 15.507752418518066
Step 50000 SSE: 32.39691925048828	validation: 15.634113311767578

Tabela H.5: Reconhecimento do 5º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	11	--	--	--	--	--
2	--	5	--	--	--	--
3	--	2	4	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sexto conjunto de testes, obteve-se a curva da Figura H.6, apresentando o registro de testes da listagem H.6 e com a tabela de reconhecimento H.6.

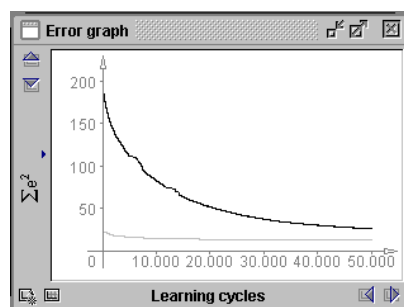


Figura H.6: Curva de erro para o 6º conjunto de testes

Listagem H.6: Registro dos testes do 6º grupo

opened at: Mon Jan 19 00:49:13 BRST 2004	
jStep 5000 SSE: 124.78665924072266	validation: 15.751924514770508
Step 10000 SSE: 99.97513580322266	validation: 14.169809341430664
Step 15000 SSE: 82.00343322753906	validation: 13.961902618408203
Step 20000 SSE: 67.68363952636719	validation: 13.507122993469238
Step 25000 SSE: 56.862152099609375	validation: 13.81599235534668
Step 30000 SSE: 49.65619659423828	validation: 13.935522079467773
Step 35000 SSE: 44.54906463623047	validation: 13.511116027832031
Step 40000 SSE: 41.275657653808594	validation: 13.56494140625
Step 45000 SSE: 38.78004455566406	validation: 13.667192459106445
Step 50000 SSE: 36.68597412109375	validation: 13.872335433959961

Tabela H.6: Reconhecimento do 6º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	4	--	1
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do sétimo conjunto de testes, obteve-se a curva da Figura H.7, apresentando o registro de testes da listagem H.7 e com a tabela de reconhecimento H.7.

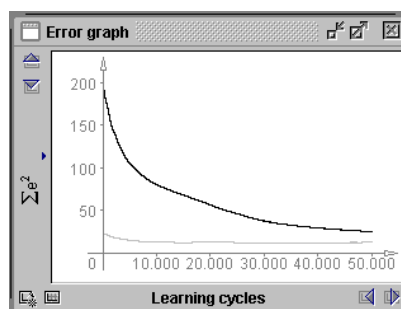


Figura H.7: Curva de erro para o 7º conjunto de testes

Listagem H.7: Registro dos testes do 7º grupo

opened at: Tue Jan 20 02:49:38 BRST 2004	
Step 5000 SSE: 105.03565979003906	validation: 14.602828979492188
Step 10000 SSE: 79.85977172851562	validation: 12.526758193969727
Step 15000 SSE: 67.643310546875	validation: 11.79844856262207
Step 20000 SSE: 55.96036148071289	validation: 12.439284324645996
Step 25000 SSE: 45.8327522277832	validation: 11.86569595336914

Step 30000 SSE: 37.25325393676758	validation: 11.470197677612305
Step 35000 SSE: 32.21105194091797	validation: 11.525318145751953
Step 40000 SSE: 29.012847900390625	validation: 11.69292163848877
Step 45000 SSE: 26.553281784057617	validation: 11.882139205932617
Step 50000 SSE: 24.679039001464844	validation: 11.955754280090332

Tabela H.7: Reconhecimento do 7º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do oitavo conjunto de testes, obteve-se a curva da Figura H.8, apresentando o registro de testes da listagem H.8 e com a tabela de reconhecimento H.8.

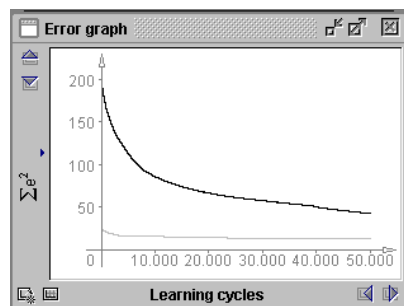


Figura H.8: Curva de erro para o 8º conjunto de testes

Listagem H.8: Registro dos testes do 8º grupo

opened at: Wed Jan 21 22:18:59 BRST 2004	
Step 5000 SSE: 112.74357604980469	validation: 15.39281940460205
Step 10000 SSE: 85.4887466430664	validation: 15.152975082397461
Step 15000 SSE: 73.60733795166016	validation: 14.541053771972656
Step 20000 SSE: 66.0712890625	validation: 13.7140531539917
Step 25000 SSE: 60.971771240234375	validation: 13.317052841186523
Step 30000 SSE: 57.22629165649414	validation: 13.060089111328125
Step 35000 SSE: 53.73779296875	validation: 12.849196434020996
Step 40000 SSE: 49.965553283691406	validation: 12.824210166931152
Step 45000 SSE: 45.57184982299805	validation: 12.815736770629883
Step 50000 SSE: 42.19322204589844	validation: 12.812286376953125

Tabela H.8: Reconhecimento do 8º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	3	1	--	--	1
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	--	3	--
N.R. = Não reconhecido						

Para treinamento do nono conjunto de testes, obteve-se a curva da Figura H.9, apresentando o registro de testes da listagem H.9 e com a tabela de reconhecimento H.9.

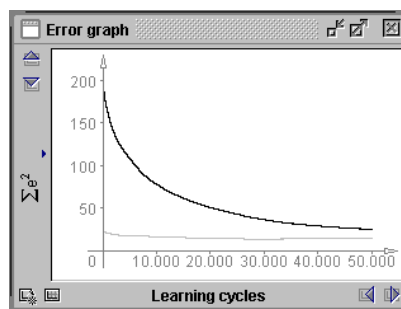


Figura H.9: Curva de erro para o 9º conjunto de testes

Listagem H.9: Registro dos testes do 9º grupo

opened at: Wed Jan 21 10:54:32 BRST 2004	
Step 5000 SSE: 107.99140167236328	validation: 16.99859046936035
Step 10000 SSE: 77.6335220336914	validation: 16.192039489746094
Step 15000 SSE: 61.28025817871094	validation: 14.978880882263184
Step 20000 SSE: 50.18557357788086	validation: 14.228326797485352
Step 25000 SSE: 42.63035583496094	validation: 13.909867286682129
Step 30000 SSE: 36.050498962402344	validation: 13.340888977050781
Step 35000 SSE: 31.625205993652344	validation: 13.445019721984863
Step 40000 SSE: 28.64473533630371	validation: 13.626410484313965
Step 45000 SSE: 26.42254066467285	validation: 13.717503547668457
Step 50000 SSE: 24.662519454956055	validation: 13.826786041259766

Tabela H.9: Reconhecimento do 9º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	5	--	--	--	--
3	--	--	6	--	--	--
4	--	--	--	5	--	--
5	--	--	--	1	2	--
N.R. = Não reconhecido						

Para treinamento do décimo conjunto de testes, obteve-se a curva da Figura H.10, apresentando o registro de testes da listagem H.10 e com a tabela de reconhecimento H.10.

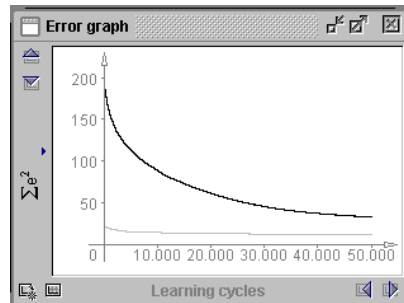


Figura H.10: Curva de erro para o 10º conjunto de testes

Listagem H.10: Registro dos testes do 10º grupo

opened at: Sun Jan 18 15:15:53 BRST 2004	
Step 5000 SSE: 113.26277160644531	validation: 14.853893280029297
Step 10000 SSE: 88.43305969238281	validation: 13.909408569335938
Step 15000 SSE: 72.72820281982422	validation: 13.106990814208984
Step 20000 SSE: 61.073280334472656	validation: 13.038887977600098
Step 25000 SSE: 51.88932800292969	validation: 12.483888626098633
Step 30000 SSE: 45.05032730102539	validation: 12.028910636901855
Step 35000 SSE: 40.20952606201172	validation: 11.575902938842773
Step 40000 SSE: 36.8609504699707	validation: 11.454875946044922
Step 45000 SSE: 34.372920989990234	validation: 11.473245620727539
Step 50000 SSE: 32.335166931152344	validation: 11.789361000061035

Tabela H.10: Reconhecimento do 10º conjunto de teste/validação

Grupo	1	2	3	4	5	N.R.
1	12	--	--	--	--	--
2	--	4	1	--	--	--
3	--	--	5	--	--	--
4	--	--	--	6	--	--
5	--	--	--		3	--
N.R. = Não reconhecido						