

**COMPARACIÓN ENTRE ONDITAS Y CEPSTRO PARA RECONOCIMIENTO
AUTOMÁTICO DE FONEMAS**

**CARLOS ANDRÉS MONCAYO RAMÍREZ
CARLOS EUGENIO THOMPSON PINZÓN**

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ, D.C.**

2000

**COMPARACIÓN ENTRE ONDITAS Y CEPSTRO PARA RECONOCIMIENTO
AUTOMÁTICO DE FONEMAS**

**CARLOS ANDRÉS MONCAYO RAMÍREZ
CARLOS EUGENIO THOMPSON PINZÓN**

**Informe Final de Trabajo de Grado
para optar al título de Ingeniero Electrónico**

Director

PEDRO RAÚL VIZCAYA GUARÍN, Ph.D.

Ingeniero Electrónico

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ, D.C.**

2000

NOTA DE ACEPTACIÓN

Presidente del jurado

Jurado

Jurado

Ciudad y fecha (día, mes, año)

A mis padres.

— Carlos Andrés Moncayo Ramírez

A mi novia por todo el apoyo que nos ha prestado.

A mis padres y a mi familia.

— Carlos Eugenio Thompson Pinzón

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

Pedro Raúl Vizcaya Guarín, Ph.D. Ingeniero Electrónico y Director del Trabajo de Grado, por sus valiosas orientaciones.

CONTENIDO

	pág.
Introducción	24
1. Marco teórico	25
1.1 Cepstro	25
1.2 Onditas	26
1.3 Redes Neurales	26
2. Especificaciones	28
2.1 Adquisición	28
2.1.1 Base de datos	28
2.1.2 Rotulación	29
2.1.3 Segmentación	30
2.2 Extracción de características	31
2.3 Clasificación	32
2.3.1 Criterios de diseño de las redes	32
2.4 Presentación	34
3. Desarrollo	35
3.1 Adquisición	35
3.1.1 Base de datos	35
3.1.2 Rotulación	36
3.1.3 Segmentación	36
3.2 Extracción de características	39
3.3 Clasificación	39
3.4 Presentación	40

3.4.1	Presentación de resultados.....	40
3.4.2	Programas utilizados	41
4.	Análisis de resultados	42
4.1	Análisis de pruebas fallidas.....	42
4.1.1	Segmentación uniforme.....	42
4.1.2	Reconocimiento de los 18 fonemas.....	43
4.1.3	Número muy grande de coeficientes.....	43
4.1.4	Rotulación imprecisa	43
4.1.5	Bases de datos de entrada muy grandes e imprecisas	44
4.2	Análisis de pruebas finales.....	44
4.2.1	Análisis de los entrenamientos.....	44
4.2.2	Análisis de pruebas	46
4.2.3	Resultados consolidados en consonantes.....	46
4.2.4	Resultados consolidados en vocales.	50
4.3	Notas varias de conclusión.....	Error! Bookmark not defined.
5.	Conclusiones.....	52
	Bibliografía.....	54
	Índice.....	56

LISTA DE TABLAS

	pág.
Tabla 1. Tablas de convergencia.....	45
Tabla 2. Onditas /p/, /t/, /k/; ajuste aleatorio, selección aleatoria, con exclusión	47
Tabla 3. Cepstro /p/, /t/, /k/; ajuste aleatorio, selección aleatoria, con exclusión.....	47
Tabla 4. Onditas /p/, /t/, /k/; ajuste aleatorio, selección aleatoria, sin exclusión	48
Tabla 5. Onditas /p/, /t/, /k/; ajuste binario, selección uniforme, sin exclusión	48
Tabla 6. Onditas /p/, /t/, /k/; ajuste binario, selección uniforme, con exclusión	49
Tabla 7. Consolidados totales, /a/ y /e/ con onditas.....	50
Tabla 8. Consolidados totales, /a/ y /e/ con cepstro.....	50
Tabla 14. Entrenamiento y pruebas del experimento 1	65
Tabla 15. Entrenamiento y pruebas del experimento 2.....	66
Tabla 16. Entrenamiento y pruebas del experimento 3.....	67
Tabla 17. Entrenamiento y pruebas del experimento 4.....	67
Tabla 18. Entrenamiento y pruebas del experimento 5.....	68
Tabla 19. Entrenamiento y pruebas del experimento 6.....	68
Tabla 20. Entrenamiento y pruebas del experimento 7	69
Tabla 21. Entrenamiento y pruebas del experimento 8	70
Tabla 22. Entrenamiento y pruebas del experimento 9	70
Tabla 23. Entrenamiento y pruebas del experimento 10	71
Tabla 24. Entrenamiento y pruebas del experimento 11	72
Tabla 25. Entrenamiento y pruebas del experimento 12.....	72
Tabla 26. Entrenamiento del experimento 1.....	73

Tabla 27. Pruebas del experimento 1, instancia 1	74
Tabla 28. Pruebas del experimento 1, instancia 2	74
Tabla 29. Pruebas del experimento 1, instancia 3	75
Tabla 30. Pruebas del experimento 1, instancia 4	75
Tabla 31. Pruebas del experimento 1, instancia 5	76
Tabla 32. Entrenamiento del experimento 2	77
Tabla 33. Pruebas del experimento 2, instancia 1.....	77
Tabla 34. Pruebas del experimento 2, instancia 2.....	77
Tabla 35. Pruebas del experimento 2, instancia 3	78
Tabla 36. Pruebas del experimento 2, instancia 4.....	78
Tabla 37. Pruebas del experimento 2, instancia 5	79
Tabla 38. Entrenamiento del experimento 3	80
Tabla 39. Pruebas del experimento 3, instancia 1	80
Tabla 40. Pruebas del experimento 3, instancia 1	80
Tabla 41. Pruebas del experimento 3, instancia 3	81
Tabla 42. Pruebas del experimento 3, instancia 4.....	81
Tabla 43. Pruebas del experimento 3, instancia 1	82
Tabla 44. Entrenamiento del experimento 4	83
Tabla 45. Pruebas del experimento 4, instancia 1.....	83
Tabla 46. Pruebas del experimento 4, instancia 2.....	83
Tabla 47. Pruebas del experimento 4, instancia 3	84
Tabla 48. Pruebas del experimento 4, instancia 4.....	84
Tabla 49. Pruebas del experimento 4, instancia 5.....	85
Tabla 50. Entrenamiento del experimento 5	86
Tabla 51. Pruebas del experimento 5, instancia 1, computador 1	86
Tabla 52. Pruebas del experimento 5, instancia 2, computador 1	86
Tabla 53. Pruebas del experimento 5, instancia 3, computador 1	87
Tabla 54. Pruebas del experimento 5, instancia 4, computador 1	87
Tabla 55. Pruebas del experimento 5, instancia 5, computador 1.....	88

Tabla 56. Pruebas del experimento 5, instancia 1, computador 2	88
Tabla 57. Pruebas del experimento 5, instancia 2, computador 2	89
Tabla 58. Pruebas del experimento 5, instancia 3, computador 2.....	89
Tabla 59. Pruebas del experimento 5, instancia 4, computador 2.....	90
Tabla 60. Pruebas del experimento 5, instancia 5, computador 2.....	90
Tabla 61. Entrenamiento del experimento 6.....	91
Tabla 62. Pruebas del experimento 6, instancia 1, computador 1	91
Tabla 63. Pruebas del experimento 6, instancia 2, computador 1	92
Tabla 64. Pruebas del experimento 6, instancia 3, computador 1	92
Tabla 65. Pruebas del experimento 6, instancia 4, computador 1	93
Tabla 66. Pruebas del experimento 6, instancia 5, computador 1	93
Tabla 67. Pruebas del experimento 6, instancia 1, computador 2	94
Tabla 68. Pruebas del experimento 6, instancia 2, computador 2	94
Tabla 69. Pruebas del experimento 6, instancia 3, computador 2.....	95
Tabla 70. Pruebas del experimento 6, instancia 4, computador 2.....	95
Tabla 71. Pruebas del experimento 6, instancia 5, computador 2	96
Tabla 72. Entrenamiento del experimento 7.....	96
Tabla 73. Pruebas del experimento 7, instancia 1, computador 1	97
Tabla 74. Pruebas del experimento 7, instancia 2, computador 1	97
Tabla 75. Pruebas del experimento 7, instancia 3, computador 1.....	98
Tabla 76. Pruebas del experimento 7, instancia 4, computador 1	98
Tabla 77. Pruebas del experimento 7, instancia 5, computador 1.....	99
Tabla 78. Pruebas del experimento 7, instancia 1, computador 2	99
Tabla 79. Pruebas del experimento 7, instancia 2, computador 2	100
Tabla 80. Pruebas del experimento 7, instancia 3, computador 2	100
Tabla 81. Pruebas del experimento 7, instancia 4, computador 2	101
Tabla 82. Pruebas del experimento 7, instancia 5, computador 2.....	101
Tabla 83. Entrenamiento del experimento 8	102
Tabla 84. Pruebas del experimento 8, instancia 1, computador 1	102

Tabla 85. Pruebas del experimento 8, instancia 2, computador 1.....	103
Tabla 86. Pruebas del experimento 8, instancia 3, computador 1.....	103
Tabla 87. Pruebas del experimento 8, instancia 4, computador 1.....	104
Tabla 88. Pruebas del experimento 8, instancia 5, computador 1.....	104
Tabla 89. Pruebas del experimento 8, instancia 1, computador 2.....	105
Tabla 90. Pruebas del experimento 8, instancia 2, computador 2.....	105
Tabla 91. Pruebas del experimento 8, instancia 3, computador 2.....	106
Tabla 92. Pruebas del experimento 8, instancia 4, computador 2.....	106
Tabla 93. Pruebas del experimento 8, instancia 5, computador 2.....	107
Tabla 94. Entrenamiento del experimento 9.....	107
Tabla 95. Pruebas del experimento 9, instancia 1, computador 1.....	108
Tabla 96. Pruebas del experimento 9, instancia 2, computador 1.....	108
Tabla 97. Pruebas del experimento 9, instancia 3, computador 1.....	108
Tabla 98. Pruebas del experimento 9, instancia 4, computador 1.....	109
Tabla 99. Pruebas del experimento 9, instancia 5, computador 1.....	109
Tabla 100. Pruebas del experimento 9, instancia 1, computador 2.....	109
Tabla 101. Pruebas del experimento 9, instancia 2, computador 2.....	110
Tabla 102. Pruebas del experimento 9, instancia 3, computador 2.....	110
Tabla 103. Pruebas del experimento 9, instancia 4, computador 2.....	110
Tabla 104. Pruebas del experimento 9, instancia 5, computador 2.....	111
Tabla 105. Entrenamiento del experimento 10.....	112
Tabla 106. Pruebas del experimento 10, instancia 1, computador 1.....	112
Tabla 107. Pruebas del experimento 10, instancia 2, computador 1.....	112
Tabla 108. Pruebas del experimento 10, instancia 3, computador 1.....	112
Tabla 109. Pruebas del experimento 10, instancia 4, computador 1.....	113
Tabla 110. Pruebas del experimento 10, instancia 5, computador 1.....	113
Tabla 111. Pruebas del experimento 10, instancia 1, computador 2.....	114
Tabla 112. Pruebas del experimento 10, instancia 2, computador 2.....	114
Tabla 113. Pruebas del experimento 10, instancia 3, computador 2.....	114

Tabla 114. Pruebas del experimento 10, instancia 4, computador 2.....	115
Tabla 115. Pruebas del experimento 10, instancia 5, computador 2.....	115
Tabla 116. Entrenamiento del experimento 10	116
Tabla 117. Pruebas del experimento 11, instancia 1, computador 1	116
Tabla 118. Pruebas del experimento 11, instancia 2, computador 1.....	116
Tabla 119. Pruebas del experimento 11, instancia 3, computador 1.....	117
Tabla 120. Pruebas del experimento 11, instancia 4, computador 1	117
Tabla 121. Pruebas del experimento 11, instancia 5, computador 1.....	117
Tabla 122. Pruebas del experimento 11, instancia 1, computador 2	118
Tabla 123. Pruebas del experimento 11, instancia 2, computador 2.....	118
Tabla 124. Pruebas del experimento 11, instancia 3, computador 2.....	118
Tabla 125. Pruebas del experimento 11, instancia 4, computador 2.....	119
Tabla 126. Pruebas del experimento 11, instancia 5, computador 2.....	119
Tabla 127. Entrenamiento del experimento 10.....	120
Tabla 128. Pruebas del experimento 12, instancia 1, computador 1	120
Tabla 129. Pruebas del experimento 12, instancia 2, computador 1.....	120
Tabla 130. Pruebas del experimento 12, instancia 3, computador 1.....	121
Tabla 131. Pruebas del experimento 12, instancia 4, computador 1	121
Tabla 132. Pruebas del experimento 12, instancia 5, computador 1.....	121
Tabla 133. Pruebas del experimento 12, instancia 1, computador 2.....	122
Tabla 134. Pruebas del experimento 12, instancia 2, computador 2	122
Tabla 135. Pruebas del experimento 12, instancia 3, computador 2	122
Tabla 136. Pruebas del experimento 12, instancia 4, computador 2	123
Tabla 137. Pruebas del experimento 12, instancia 5, computador 2.....	123
Tabla 138. Entrenamiento del experimento 13.....	124
Tabla 139. Pruebas del experimento 13, instancia 1, computador 1	124
Tabla 140. Pruebas del experimento 13, instancia 2, computador 1.....	125
Tabla 141. Pruebas del experimento 13, instancia 3, computador 1	125
Tabla 142. Pruebas del experimento 13, instancia 4, computador 1.....	125

Tabla 143. Pruebas del experimento 13, instancia 5, computador 1	126
Tabla 144. Pruebas del experimento 13, instancia 1, computador 2	126
Tabla 145. Pruebas del experimento 13, instancia 2, computador 2	126
Tabla 146. Pruebas del experimento 13, instancia 3, computador 2	127
Tabla 147. Pruebas del experimento 13, instancia 4, computador 2.....	127
Tabla 148. Pruebas del experimento 13, instancia 5, computador 2	127
Tabla 149. Entrenamiento del experimento 14.....	128
Tabla 150. Pruebas del experimento 14, instancia 1, computador 1	128
Tabla 151. Pruebas del experimento 14, instancia 2, computador 1	129
Tabla 152. Pruebas del experimento 14, instancia 3, computador 1	129
Tabla 153. Pruebas del experimento 14, instancia 4, computador 1.....	129
Tabla 154. Pruebas del experimento 14, instancia 5, computador 1	130
Tabla 155. Pruebas del experimento 14, instancia 1, computador 2	130
Tabla 156. Pruebas del experimento 14, instancia 2, computador 2	130
Tabla 157. Pruebas del experimento 14, instancia 3, computador 2.....	131
Tabla 158. Pruebas del experimento 14, instancia 4, computador 2	131
Tabla 159. Pruebas del experimento 14, instancia 5, computador 2.....	131
Tabla 160. Entrenamiento del experimento 15.....	132
Tabla 161. Pruebas del experimento 15, instancia 1, computador 1.....	132
Tabla 162. Pruebas del experimento 15, instancia 2, computador 1.....	133
Tabla 163. Pruebas del experimento 15, instancia 3, computador 1	133
Tabla 164. Pruebas del experimento 15, instancia 4, computador 1.....	133
Tabla 165. Pruebas del experimento 5, instancia 5, computador 1.....	134
Tabla 166. Pruebas del experimento 15, instancia 1, computador 2.....	134
Tabla 167. Pruebas del experimento 15, instancia 2, computador 2.....	134
Tabla 168. Pruebas del experimento 15, instancia 3, computador 2	135
Tabla 169. Pruebas del experimento 15, instancia 4, computador 2	135
Tabla 170. Pruebas del experimento 15, instancia 5, computador 2.....	136
Tabla 171. Entrenamiento del experimento 10	136

Tabla 172. Pruebas del experimento 16, instancia 1, computador 1	137
Tabla 173. Pruebas del experimento 16, instancia 2, computador 1	137
Tabla 174. Pruebas del experimento 16, instancia 3, computador 1	137
Tabla 175. Pruebas del experimento 16, instancia 4, computador 1	138
Tabla 176. Pruebas del experimento 16, instancia 5, computador 1	138
Tabla 177. Pruebas del experimento 16, instancia 1, computador 2	138
Tabla 178. Pruebas del experimento 16, instancia 2, computador 2	139
Tabla 179. Pruebas del experimento 16, instancia 3, computador 2.....	139
Tabla 180. Pruebas del experimento 16, instancia 4, computador 2.....	139
Tabla 181. Pruebas del experimento 16, instancia 5, computador 2.....	140

LISTA DE CUADROS

Cuadro 1. Experimento 1, descripción del experimento.....	65
Cuadro 2. Experimento 1, selección y ajuste de ventanas.....	65
Cuadro 3. Experimento 1, características de la red	65
Cuadro 4. Experimento 2, descripción del experimento.....	66
Cuadro 5. Experimento 2, selección y ajuste de ventanas.....	66
Cuadro 6. Experimento 2, características de la red	66
Cuadro 7. Experimento 3, descripción del experimento	66
Cuadro 8. Experimento 3, selección y ajuste de ventanas	66
Cuadro 9. Experimento 3, características de la red	66
Cuadro 10. Experimento 4, descripción del experimento	67
Cuadro 11. Experimento 4, selección y ajuste de ventanas	67
Cuadro 12. Experimento 4, características de la red	67
Cuadro 13. Experimento 5, descripción del experimento	67
Cuadro 14. Experimento 5, selección y ajuste de ventanas.....	67
Cuadro 15. Experimento 5, características de la red.....	68
Cuadro 16. Experimento 6, descripción del experimento	68
Cuadro 17. Experimento 6, selección y ajuste de ventanas	68
Cuadro 18. Experimento 6, características de la red	68
Cuadro 19. Experimento 7, descripción del experimento	69
Cuadro 20. Experimento 7, selección y ajuste de ventanas	69
Cuadro 21. Experimento 7, características de la red.....	69
Cuadro 22. Experimento 8, descripción del experimento.....	69
Cuadro 23. Experimento 8, selección y ajuste de ventanas	69

Cuadro 24. Experimento 8, características de la red.....	69
Cuadro 25. Experimento 9, descripción del experimento.....	70
Cuadro 26. Experimento 9, selección y ajuste de ventanas	70
Cuadro 27. Experimento 9, características de la red	70
Cuadro 28. Experimento 10, descripción del experimento.....	70
Cuadro 29. Experimento 10, selección y ajuste de ventanas.....	71
Cuadro 30. Experimento 10, características de la red.....	71
Cuadro 31. Experimento 11, descripción del experimento	71
Cuadro 32. Experimento 11, selección y ajuste de ventanas	71
Cuadro 33. Experimento 11, características de la red.....	71
Cuadro 34. Experimento 12, descripción del experimento	72
Cuadro 35. Experimento 12, selección y ajuste de ventanas.....	72
Cuadro 36. Experimento 12, características de la red	72
Cuadro 37. Experimento 1, descripción del experimento	73
Cuadro 38. Experimento 1, selección y ajuste de ventanas.....	73
Cuadro 39. Experimento 1, características de la red	73
Cuadro 40. Experimento 2, descripción del experimento	76
Cuadro 41. Experimento 2, selección y ajuste de ventanas.....	76
Cuadro 42. Experimento 2, características de la red.....	76
Cuadro 43. Experimento 3, descripción del experimento.....	79
Cuadro 44. Experimento 3, selección y ajuste de ventanas	79
Cuadro 45. Experimento 3, características de la red.....	79
Cuadro 46. Experimento 4, descripción del experimento.....	82
Cuadro 47. Experimento 4, selección y ajuste de ventanas.....	82
Cuadro 48. Experimento 4, características de la red.....	82
Cuadro 49. Experimento 5, descripción del experimento.....	85
Cuadro 50. Experimento 5, selección y ajuste de ventanas	85
Cuadro 51. Experimento 5, características de la red.....	85
Cuadro 52. Experimento 6, descripción del experimento.....	91

Cuadro 53. Experimento 6, selección y ajuste de ventanas.....	91
Cuadro 54. Experimento 6, características de la red	91
Cuadro 55. Experimento 7, descripción del experimento	96
Cuadro 56. Experimento 7, selección y ajuste de ventanas.....	96
Cuadro 57. Experimento 7, características de la red	96
Cuadro 58. Experimento 8, descripción del experimento.....	102
Cuadro 59. Experimento 8, selección y ajuste de ventanas	102
Cuadro 60. Experimento 8, características de la red	102
Cuadro 61. Experimento 9, descripción del experimento	107
Cuadro 62. Experimento 9, selección y ajuste de ventanas	107
Cuadro 63. Experimento 9, características de la red.....	107
Cuadro 64. Experimento 10, descripción del experimento.....	111
Cuadro 65. Experimento 10, selección y ajuste de ventanas.....	111
Cuadro 66. Experimento 10, características de la red.....	111
Cuadro 67. Experimento 11, descripción del experimento.....	115
Cuadro 68. Experimento 11, selección y ajuste de ventanas	115
Cuadro 69. Experimento 11, características de la red.....	116
Cuadro 70. Experimento 12, descripción del experimento	119
Cuadro 71. Experimento 12, selección y ajuste de ventanas	120
Cuadro 72. Experimento 12, características de la red	120
Cuadro 73. Experimento 13, descripción del experimento	124
Cuadro 74. Experimento 13, selección y ajuste de ventanas.....	124
Cuadro 75. Experimento 13, características de la red.....	124
Cuadro 76. Experimento 14, descripción del experimento	128
Cuadro 77. Experimento 14, selección y ajuste de ventanas	128
Cuadro 78. Experimento 14, características de la red	128
Cuadro 79. Experimento 15, descripción del experimento	132
Cuadro 80. Experimento 5, selección y ajuste de ventanas	132
Cuadro 81. Experimento 5, características de la red	132

Cuadro 82. Experimento 13, descripción del experimento.....	136
Cuadro 83. Experimento 13, selección y ajuste de ventanas.....	136
Cuadro 84. Experimento 13, características de la red.....	136

LISTA DE FIGURAS

Figura 1. Diagrama en bloques.	28
Figura 2. Diagrama en bloques extracción de características y clasificación.....	31

LISTA DE ANEXOS

Anexo A. Base de datos de pruebas.....	58
Anexo B. Resultados detallados, experimentos preliminares	65
Anexo C. Resultados detallados, experimentos finales	73
Anexo D. Programas para extracción de la base de datos	141
Anexo E. Programas para manipulación de archivos . wav	152
Anexo F. Programa de extracción aleatoria de fonemas	167
Anexo G. Programas de MATLAB para manipulación de archivos	182
Anexo H. Programas de MATLAB para el proceso de entrenamiento y pruebas definitivos.....	188

GLOSARIO

ALÓFONO: corresponde a las variaciones de los sonidos (fonos) que pueden coexistir en un determinado idioma, como variantes de un mismo fonema.

APARATO FONADOR: conjunto de órganos del cuerpo humano que producen los sonidos del habla.

ATAQUE: sobre la curva de envolvente de un sonido es el periodo en el cual se cambia de silencio a un sonido de voz.

CEPSTRO: transformada matemática no lineal correspondiente a la transformada inversa de fourier del logaritmo de la transformada de fourier.

DIFONEMA: conjunto de dos fonemas oconsecutivos y la transición entre los mismos.

FONEMA: cada uno de los sonidos de un idioma que son distinguibles por un hablante nativo como sonidos diferentes.

FONO: cada uno de los posibles sonidos que puede contener un idioma.

FOURIER: transformada matemática lineal que describe una señal en términos de sus componentes de frecuencia.

OCCLUSIÓN: cierre del flujo de aire durante la fonación. Durante la oclusión no existe sonido.

ONDITAS: conjunto de transformadas matemáticas lineales de resolución múltiple en tiempo y frecuencia. Cada una de las señales base de una transformación de onditas.

ROTULACIÓN: acción o efecto de rotular.

ROTULAR: asignar una etiqueta a una muestra. En este caso en particular, asignar las etiquetas de los fonemas a las muestras de voz.

SEGMENTACIÓN: acción o efecto de segmentar.

SEGMENTAR: extraer ventanas de una muestra de voz para procesarlas individualmente.

WAVELETS: nombre internacional para las onditas.

RESUMEN

El trabajo realizado compara la transformada onditas y cepstro para reconocimiento de fonemas explosivos y vocálicos, utilizando coeficientes de energía por nivel en el caso de onditas y los 14 coeficientes después del primero en el caso de cepstro. Como clasificador se escogió una red neural con las mismas especificaciones para onditas y para cepstro. Los fonemas que se utilizaron fueron /p/, /t/, /k/, /a/ y /e/, extraídos en igual número de ventanas por fonema de una base de datos de 93 frases pronunciadas por un solo hablante. Onditas y cepstro se comportan de manera similar en reconocimiento de fonemas explosivos, y en fonemas vocálicos cepstro definitivamente es mejor, sin embargo onditas arroja malos resultados .

INTRODUCCIÓN

El reconocimiento de habla permite que una máquina entienda los comandos y se comunique con un ser humano de una forma natural. Un sistema de reconocimiento de habla puede incluir aplicaciones tales como reconocimiento de comandos de voz, transcripción de habla, traducción automática y ayudas a personas inválidas, tales como máquinas comandadas por voz para personas con disfunción en las manos o en la comunicación de hablantes con sordos.

Los trabajos que se han hecho hasta este momento en la Universidad Javeriana han evaluado distintas tecnologías de forma individual. Sus objetivos, métodos y condiciones de evaluación han sido muy diversos, lo cual hace imposible una comparación. Onditas tiene una característica multirresolución que se puede explorar y comparar con el análisis cespstral que es el método que ha dado mejores resultados y que ha sido el más utilizado.

El trabajo realizado compara la transformada onditas y el análisis cespstral en reconocimiento de fonemas explosivos y vocálicos, utilizando coeficientes de energía por nivel en el caso de onditas y los 14 coeficientes después del primero en el caso de cepstro. Como clasificador se escogió una red neural con las mismas especificaciones para onditas y para cepstro, quedando así en igualdad de condiciones para poder comparar.

1. MARCO TEÓRICO

Una vez la señal de voz es dividida, generalmente en segmentos entre 10 ms y 40 ms, se extraen sus características o componentes esenciales con el fin de obtener una representación de la señal en otro espacio donde resulte más fácil su procesamiento. Particularmente se busca que los patrones parecidos se concentren y los patrones diferentes se distancien.

1.1 CEPSTRO

Una forma de caracterizar las señales de voz consiste en calcular el cepstro, el cepstro se define como:

$$C_{X(n)} = F^{-1} \{ \log | F \{ X(n) \} | \}$$

Donde F es la transformada de Fourier y $X(n)$ la señal en tiempo. El análisis cepstral está inspirado en el hecho de que la señal de voz se forma por la convolución entre una señal semiperiódica (secuencia de excitación) y una envolvente (la respuesta impulso del sistema o tracto vocal). Así se tiene que:

$$x(n) = e(n) * h(n)$$

Al aplicar transformada de Fourier a ambos lados de la ecuación se obtiene el producto de las señales en el dominio de la frecuencia, al aplicar logaritmo se obtiene la suma de las señales en frecuencia, las cuales se pueden separar fácilmente en el caso de la señal de voz donde la señal de excitación es rápida (componentes de alta frecuencia) mientras la envolvente es lenta (componentes de baja frecuencia); sobre esta última es la que se trabaja posteriormente en etapas de reconocimiento.

1.2 ONDITAS

Las onditas se derivan a partir de corrimientos en escala y tiempo de una función principal llamada “Ondita Madre”; hay diferentes tipos de ondita madre y se utilizan según la aplicación. Onditas se puede ver como un banco de filtros más selectivos en bajas frecuencias y menos selectivos en altas; la respuesta impulso de los filtros son versiones escalizadas de un mismo prototipo:

$$h_a(t) = \frac{1}{|a|^{1/2}} h\left(\frac{t}{a}\right)$$

Donde a es el factor de escala y el factor $1/|a|^{1/2}$ es utilizado para la normalización en energía

La transformada onditas se define como:

$$CWT_x(t, a) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(\tau) h\left(\frac{t-\tau}{a}\right) d\tau$$

En onditas la resolución no es fija, depende de la frecuencia, $\Delta f/f = c$, es decir, el ancho de banda de los filtros depende de la frecuencia central del filtro; al cumplirse la desigualdad de Heisenberg para cada filtro, la resolución en tiempo es mayor en altas frecuencias y menor para las bajas, pero la resolución en frecuencia es mayor para frecuencias bajas y menor para las altas. Esta característica se denomina multirresolución y es óptima para señales como la voz donde las vocales son de baja frecuencia y alta duración, mientras que las consonantes son de alta frecuencia y corta duración.

Onditas actúa de manera similar a la cóclea y en general al mecanismo de percepción auditivo, pues nuestro oído es capaz de distinguir mejor las frecuencias bajas que las altas, podemos distinguir cambios de frecuencia entre los 400 Hz y 600 Hz pero se dificulta hacerlo conservando la misma diferencia de 200 Hz entre los 10 400 Hz y los 10 600 Hz.

1.3 REDES NEURALES

Una red neural es un procesador no lineal, distribuido, masivamente paralelo que tiene la capacidad de almacenar conocimiento experimental (aprende mediante ejemplos). La red aprende por medio

de un algoritmo de aprendizaje que se encarga de modificar los pesos sinápticos de la red; en estos pesos la red almacena el conocimiento. Se han utilizado las redes neurales con éxito en reconocimiento de patrones y como clasificadores son muy eficientes.

2. ESPECIFICACIONES

El trabajo consiste en comparar el comportamiento de la transformada onditas con respecto a cepstro para reconocimiento automático de fonemas. El objetivo principal es determinar que ventajas y/o desventajas presenta la transformada onditas (*wavelets*) frente a la representación cepstral en el contexto de reconocimiento de fonemas vocálicos y consonánticos. Para lograrlo se desarrolló un sistema de reconocimiento automático cuyo diagrama en bloques se muestra en la siguiente figura:

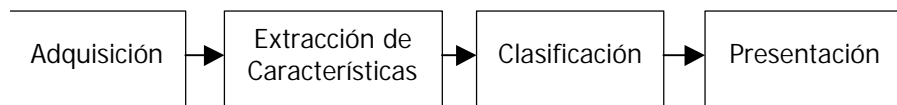


Figura 1. Diagrama en bloques.

A continuación se explicará en que consistió cada uno de estos bloques con sus respectivas especificaciones técnicas.

2.1 ADQUISICIÓN

Esta etapa incluye el diseño y creación de la base de datos y la adquisición, rotulación y segmentación de las frases utilizadas en las pruebas.

2.1.1 Base de datos

Se diseñó una base de datos de frases de tal forma que cumpliera estadísticamente con la distribución de los fonemas en el español, para esto primero se tomaron textos del español de gran variedad de medios; como periódicos, revistas, trabajos y libros; y de diversos contenidos: educación, noticias, política, literatura, arte, académicos, diversión, deportes, actualidad, etc. Luego

se realizó la transcripción fonética, se calcularon las distribuciones de los fonemas y difonemas y se tabularon los resultados. De este mismo conjunto se generó automáticamente un conjunto de frases de cuatro palabras, se revisaron y se dejaron únicamente las que tenían sentido. Este conjunto fue de 40150 frases, de las cuales se extrajeron al azar ocho grupos de 100 frases cada uno; dos grupos por cada hablante. Se comprobó que cada uno de los grupos cumpliera con las distribuciones estadísticas medidas anteriormente. Se determinó el número de hablantes en cuatro: dos hombres y dos mujeres. Se seleccionaron de acuerdo a su buena dicción, condición necesaria para obtener una base de datos de buena calidad.

La adquisición de las frases leídas por los hablantes se llevó a cabo en un estudio de grabación, de forma digital. Se utilizó un micrófono de condensador, el de mejor en calidad y rechazo a ruido. El estudio garantiza mínimo ruido externo y una excelente calidad en la señal de voz. Al ser grabación digital se elimina el ruido ocasionado por partes mecánicas o cintas.

Por razones de complejidad y capacidad de computo expuestas más adelante, los experimentos se realizaron con un solo hablante y con 93 frases, también representativas del español. El número de frases del conjunto de entrenamiento fue de 58 y el de frases de pruebas fue de 35. Cada frase fue almacenada como un archivo .wav independiente, con una tasa de 22050 muestras por segundo.

2.1.2 Rotulación

La rotulación consiste en establecer los límites de comienzo y finalización de los diferentes fonemas, se utilizó un programa *freeware* diseñado exclusivamente para rotulación de archivos de voz, desarrollado por SIL International, el cual modifica los archivos .wav añadiéndole la rotulación al final del de los datos.

El criterio que se utilizó para rotular fue el de eliminar al máximo la coarticulación, sólo se rotuló en el caso en que se estaba completamente seguro del fonema y donde se tenía menor variabilidad. En los fonemas explosivos se tomó solamente el ataque, excluyendo así silencios, comienzos de otros

fonemas y apagados rápidos de fonemas anteriores; que si bien cuentan en el proceso de reconocimiento del habla, hacen que la red se confunda y no converja.

2.1.3 Segmentación

En esta etapa se realizaron varios experimentos para llegar a la mejor forma de segmentar. Según las pruebas realizadas, en los entrenamientos por frases la segmentación se hacía tomando ventanas en forma secuencial a lo largo de la frase. Esta forma de segmentar no separa los fonemas de forma precisa y como consecuencia las redes no convergen. Se optó por tomar ventanas de ancho fijo dentro de los fonemas rotulados. El tamaño de las ventanas se tomó de 512 muestras que equivalen a 23.2 ms, se hicieron pruebas con ventanas de 1024 y 256 pero las que mejores resultados arrojaron fueron las ventanas de 512.

Los fonemas se tomaron aleatoria o uniformemente distribuidos dentro de un conjunto de archivos de frases rotuladas. El ajuste de la ventana dentro del fonema puede ser con tendencia centrada o aleatoria.

A continuación se explican los diferentes métodos de segmentación utilizados:

2.1.3.1 Formas de extracción de fonemas

Aleatoria: entre todos fonemas válidos dentro del conjunto de frases rotuladas, se toma aleatoriamente cada fonema, con reposición. Aún cuando la función aleatoria usada tiene una distribución homogénea, no hay garantía de que todas las instancias de un sonido queden representadas.

Uniforme: entre el número de fonemas válidos, se toma con una función pseudo-aleatoria que asegura una distribución homogénea con pocas muestras. Esta función garantiza que si hay N fonemas y $k \times N$ muestras, cada fonema estará representado entre $k-1$ y $k+1$ veces.

2.1.3.2 Tipos de ajuste de las ventanas dentro de los fonemas

Aleatoria: dentro de los límites del fonema seleccionado se ajusta la ventana mediante una función aleatoria homogénea cumpliendo las reglas.

Binaria: se ajusta la ventana mediante una función aleatoria con distribución binaria de orden 6, donde la probabilidad de que el centro de la ventana esté cerca al centro es mayor a la probabilidad de que esté cerca de los bordes.

Centrada: el punto medio de la ventana coincide con el punto medio de del fonema rotulado.

En todos los casos si el fonema es más grande que la ventana, la ventana debe quedar contenida dentro del fonema. Si el fonema es más pequeño que la ventana, el fonema debe quedar contenido dentro de la ventana.

2.2 EXTRACCIÓN DE CARACTERÍSTICAS

Esta etapa consta de tres partes como se puede observar en la figura 2.

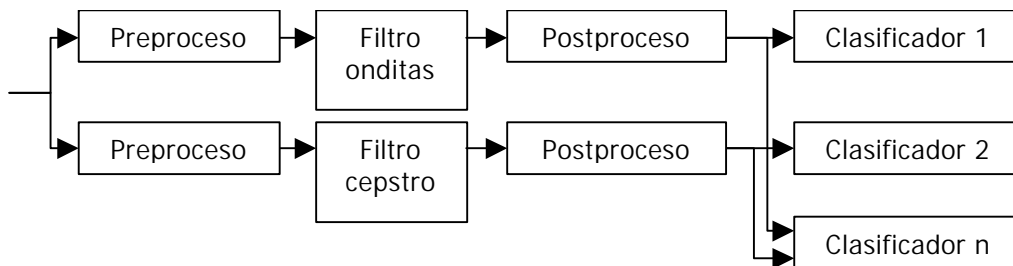


Figura 2. Diagrama en bloques extracción de características y clasificación.

La primera etapa consiste en un preproceso que se hace a las ventanas y en el caso de este proyecto consiste en una normalización en amplitud. Este preproceso es necesario debido a que los fonemas son expresados con diferentes niveles de energía. Adicionalmente se aplica un filtro Hamming que destaca las características sobre la mitad de la ventana con baja distorsión en amplitud.

Una vez la señal (ventana) es normalizada, pasa por el respectivo filtro, onditas o cepstro.

Onditas: para onditas se tomó como ondita madre *Haar* con 6 niveles de descomposición, más uno de aproximación. Después del filtro viene una etapa de extracción de coeficientes. No todos los coeficientes tienen igual importancia en el momento de reconocer. En onditas planteó la posibilidad de utilizar todos los coeficientes y descubrir cuales de ellos eran los que más contribuyen en el proceso de reconocimiento, pero la complejidad y la cantidad de cálculos que se tenían que realizar eran muy grandes ya que los clasificadores eran redes neurales, donde la complejidad de los algoritmos depende en gran medida del número de coeficientes de entrada. En los experimentos se probaron varios métodos de reducción de coeficientes pero con el método de energía por nivel, fue en el que se obtuvieron mejores resultados y menor número de coeficientes.

Cepstro: en el caso de cepstro se tomaron los 14 coeficientes después del primero que son los que concentran la información de las formantes, el primer coeficiente de cepstro se descarta ya que posee información de la potencia de la señal.

2.3 CLASIFICACIÓN

La etapa de clasificación consiste en una red neural en configuración de perceptrón multinivel, *feedforward* de 3 niveles. Se escogió este tipo de red porque lo que se pretendía era tener un buen clasificador que aprendiera y que generalizara. Esto nos permitiría ver si los coeficientes que estabamos utilizando eran los apropiados o no, también nos permitiría comparar, bajo las mismas circunstancias, a cepstro y a onditas.

2.3.1 Criterios de diseño de las redes

Número de neuronas en la capa de salida: se escogen según el número de clases que se quieran separar o discriminar; por ejemplo, en el caso de /p/, /t/ y /k/, las neuronas en la etapa de salida son tres. Para los experimento de los fonemas vocálicos se tomaron dos neuronas en la capa de salida.

Número de neuronas en la capa de entrada: no existe una regla para calcular este número con exactitud, sino que se toma de acuerdo a ciertos criterios como número de dimensiones del conjunto de entrada, el número de clases que se quiere tener a la salida o que tan dispersos o aglomerados estén los datos que se quieren separar.

Número de neuronas en la capa de intermedia: para el número de neuronas de este nivel no hay ninguna regla en particular, se llega también mediante ensayo y error o mediante el sistema de poda, en el trabajo no se utilizó el sistema de poda por la cantidad de tiempo de proceso que se requiere, en consecuencia se utilizó el método de ensayo y error pero controlado, se empezó con pocas neuronas (5) y se fueron midiendo los resultados, luego se fueron aumentando hasta que la generalización mejoraba, aunque con 10 neuronas se comportaba bien, se llegó a (15) para asegurar una mejor generalización.

Funciones de transferencia: se utilizaron *tansig* (arcotangente), *tansig* y *logsig* (logarítmica) respectivamente en cada capa de la red, la razón por la cual se tomó en la capa de salida la función de transferencia *logsig* es porque permite tener una salida entre 0 y 1. Las otras funciones se tomaron *tansig* porque esta es una función impar cuyo rango está entre 1 y -1 lo que acelera el proceso de cálculos internos.

Criterio de error alcanzado: es un parámetro que manualmente se puede modificar para que la red termine su proceso de entrenamiento cuando el error cuadrado medio sea menor que el límite prefijado. El límite de error que se utilizó en los experimentos fue cero, que aunque se sabía que no se alcanzaría, permitiría observar los comportamientos de las redes: hasta donde llegaban y como respondían a los conjuntos de pruebas en cuanto a generalización.

Mínimo gradiente alcanzado: es un parámetro que indica qué tanto puede la red mejorar en la siguiente época (minimizar el error cuadrado medio). Un gradiente muy pequeño significa que la red tiene un potencial bajo de seguir mejorando y esto puede deberse a que el sistema se encuentra

en un pozo local o a que el sistema no puede converger. El criterio que se utilizó en los experimentos fue de 1×10^{-12} .

2.4 PRESENTACIÓN

Esta etapa estaba prevista para extraer los resultados de las pruebas en formato de texto plano para posteriormente ser tabulados, consolidados y analizados.

Antes de generar estos archivos de resultados para ser analizados tienen que pasar por un proceso de decisión, en el cual se toman las salidas de la red (valores continuos entre cero y uno), y se pasan a través de una función de decisión que clasifica estos valores en una de tres posibilidades dependiendo de un parámetro e que establece que si el número se encuentra entre 0 y e , se asocia con “rechazo”, si el número se encuentra entre $1-e$ y 1, se asocia como “aceptación”, y si se encuentra entre e y $1-e$ se asocia con “indecisión”.

Los resultados de mayor interés para el trabajo de investigación son las matrices de aceptación, rechazo e indecisión de los fonemas a reconocer. Estas matrices se calculan a partir de las simulaciones obtenidas de la red y de los resultados deseados o *targets* y son éstas las que van a soportar la comparación y análisis los resultados. Las operaciones realizadas con dichas matrices arrojarán datos que se pueden ver como aspectos diferentes o puntos de vista diferentes sobre los mismos resultados obtenidos. Estos tipos de datos serán muy útiles al momento de los análisis de los resultados y de la comparación entre los métodos utilizados.

3. DESARROLLO

La metodología utilizada en la investigación se basó en el desarrollo de las etapas, análisis de resultados preliminares y depuración, dentro de estos resultados hay muchos datos interesantes que van a hacer que la investigación tome otro rumbo o no.

Se empieza con pruebas sencillas las cuales se van volviendo más complejas y si hay que revisar y modificar, incluso hasta la primera etapa, se revisa y se modifica, y se vuelven a repetir todas las pruebas.

En este capítulo se mencionará el desarrollo tanto de las pruebas finales como de algunas de las pruebas preliminares que resultaron infelices, las que resultaron de mayor interés.

3.1 ADQUISICIÓN

3.1.1 Base de datos

Se utilizó un programa realizado en C para la transcripción fonética de los textos, el programa se llama phons.exe (ver D.1 en la página 141)

Para los conteos de fonemas y difonemas de un texto transcrito se utilizaron los programas phons.exe y diphon.exe (ver D.3 en la página 146) respectivamente. Para extraer las frases se desarrolló otro programa frases.exe (ver D.5 en la página 149) que buscaba cadenas de cuatro palabras no separadas por puntuación. Finalmente se procesaron todos los datos con ayuda de Microsoft Access y Microsoft Excel.

3.1.2 Rotulación

La rotulación inicialmente se había pensado de tal forma que abarcara la mayor cantidad de fonemas, sin importar la coarticulación existente, se pensaba que entre más coarticulación tuviese el fonema rotulado, mejor, porque los ejemplos iban a ser más diversos y la red podría aprender a generalizar al tener contextos diferentes. En realidad no ocurrió así, la coarticulación hacía que las redes se confundieran o no convergieran

Se desarrolló un programa en Microsoft Visual Basic, para que fuera más amigable y cómodo de manejar, para que lo utilizaran terceras personas con el fin de que ayudaran a rotular, pero en la práctica se vio que era más fácil y menos costoso en cuanto a tiempo que una sola persona rotule y que sea uno de los autores del proyecto ya que conoce qué hacer y cómo.

Más adelante se consiguió por Internet una herramienta excelente desarrollada exclusivamente para este propósito, el software se llama Speech Analyzer, versión 1.5 test versión 10.6 desarrollado por SIL International, un instituto que se dedica a la investigación de lenguas y apoya trabajos de lingüística de campo.

Esta parte de rotulación fue clave para que las redes empezaran a converger, con un error que exista en el conjunto de entrenamiento ya la red no converge y ese fue uno de los principales problemas que se tuvieron. Hasta que no se revisó la rotulación más de tres veces, frase por frase, fonema por fonema, no empezaron a converger las redes.

3.1.3 Segmentación

En un principio se entrenó las redes mediante frases, es decir, tomando una frase y dividiéndola en ventanas, y esta frase era la que se le enseñaba a la red.

Se tuvieron muchos problemas, tales como que la red se cargaba hacia los fonemas más frecuentes o más largos del español, se aprendía de memoria la frase y no generalizaba, esto ocurría por las

dimensiones que se estaban manejando para lograr que generalice con este tipo de segmentación se hubiera requerido, muchos más ejemplos y muchas horas más de entrenamiento.

Analizando las fallas de los experimentos anteriores, el trabajo se reorientó, en primer lugar hacia la reducción de la complejidad, y en segundo lugar, a encontrar una forma más equitativa de enseñar los fonemas a la red sin cargar los resultados hacia los fonemas más repetidos o más largos del español. En consecuencia se replanteó la forma en la que se entrenaría la red, ya no sería por frases sino por fonemas, pudiendo escoger el número de estos dentro de un conjunto de frases rotuladas, esto nos permitiría alcanzar los dos propósitos.

Se optó por la forma de entrenar por fonemas ya que al entrenar por frases, la misma distribución natural de los fonemas del español, hacía que la red se cargara hacia los fonemas que mas se repetían o a los más largos, es decir que se aprendía mejor los fonemas más frecuentes del español. Los fonemas explosivos son menos frecuentes, por lo tanto, se requería un numero mayor de frases para que hubiera una cantidad suficiente y representativa de estos. Aumentar el numero de frases significaba un gran tiempo de proceso, y una gran demanda de recursos (El equipo después de 8 o 10 horas de proceso abortaba por memoria, tanto en RAM como en espacio libre en disco para paginación), Aumentar los recursos era algo que no se justificaba, dado que al aumentar el número de frases la distribución de fonemas explosivos se iba a mantener y la por lo tanto la red iba a cargarse hacia los más frecuentes, y no había garantías de que se pudiera tener buenos resultados, únicamente que se lograría ser más robusto en los recursos.

Con el método de entrenamiento por fonemas, por el contrario, se podía asegurar que, en el conjunto de entrenamiento habría una buena cantidad de los fonemas no frecuentes del español, incluso configurable en número, y por lo tanto reducir considerablemente el número de ejemplos enseñados y bajar la demanda de recursos y tiempo de proceso.

Se utilizaron los programas realizados en c para la extracción de ventanas de fonemas: fonextr al cual le entra un archivo de texto con la ubicación de los archivos .wav rotulados, en el mismo texto

se especifica el número de fonemas a extraer, el nombre de los archivos de salida . **wnd** (archivo con las ventanas de fonemas extraídos en un formato creado para ser leído desde MATLAB como una matriz) y . **tgt** (archivo de *targets*, donde se indica a que clase corresponde cada ventana este archivo y cuyo formato está diseñado para ser leído fácilmente desde MATLAB y manipularlos directamente como matrices)

Del conjunto de entrenamiento se excluyeron los silencios y los fonemas distintos a /p/, /t/, /k/, /a/ y /e/, ya que según las pruebas realizadas, con grupos grandes de fonemas, v.g. sordos y sonoros, las redes se demoraban horas y nunca convergían, se confundían con tantos ejemplos diferentes. Quizás, para lograr que la red encuentre estos patrones comunes, hubiera sido necesario una red mucho más compleja y con una cantidad de ejemplos mucho más grande, pero intensiva en tiempo de procesamiento y recursos de máquina.

En cuanto a la reducción de coeficientes Onditas se pensó en la menor cantidad posible: energía por nivel. De esta forma se logró que los entrenamientos fueran menos exigentes en tiempo de proceso y en recursos, a la vez que fueran representativos del conjunto de coeficientes que se obtienen al utilizar esta transformada.

Al reducir el tamaño de la red, reducir el número de coeficientes y reducir el conjunto de entrenamiento, la complejidad se redujo considerablemente, la redes empezaron a converger en menor número de épocas. Con menos datos de entrada había que hacer menos cálculos por época, por consiguiente, se obtuvo un menor tiempo por época. En general se obtuvieron tiempos manejables por experimento del orden de aproximadamente tres horas de computo en el peor de los casos.

3.2 EXTRACCIÓN DE CARACTERÍSTICAS

En cuanto a la extracción de características se utilizó la función `wavedec` de MATLAB, la cual a la señal de entrada le calcula la transformada ondita discreta hasta varios niveles, y se puede especificar el tipo de ondita madre en este caso *Haar*, y el número de niveles de descomposición. Para la reducción de los coeficientes se utilizó el valor de energía por nivel, el cual se obtenía del programa `matwl et2wcoef. m`

En el caso de cepstro se utilizó la función interna que trae MATLAB para la extracción del cepstro real: `rceps`, al cual le entra como parámetro de entrada la señal, en este caso las ventanas de fonemas extraídas de los diferentes archivos `. wav`.

3.3 CLASIFICACIÓN

Se sabía de antemano que la transformada cepstro funcionaba bastante bien con fonemas vocálicos, sin embargo, para fonemas explosivos no representaba una ventaja significativa, precisamente esta ventaja, sobre los fonemas explosivos, era la que se esperaba obtener de la transformada Onditas. La idea era demostrarlo, los esfuerzos se concentraron, entonces, en reducir las pruebas tanto como fuera posible, orientando el trabajo hacia este objetivo. Se pensó en reducir el tamaño de la red, el conjunto de entrenamiento y el número de coeficientes, donde prácticamente se evaluaría comparativamente las transformadas sobre los fonemas explosivos, y luego se compararían estos resultados contra fonemas como los vocálicos.

En cuanto a la red, uno de los problemas era el tiempo de entrenamiento, por esto se planteó una mínima configuración de red, es decir, que la red aprendiera a reconocer solo dos fonemas. Se utilizaron redes para el caso de los fonemas explosivos `/p/`, `/t/` y `/k/`, en combinaciones de dos: una red para que discrimine entre `/p/` y `/t/` otra para el caso `/p/` y `/k/` y para `/t/` y `/k/`. Como estas pruebas resultaron exitosas se pasó a redes que discriminaran entre los tres fonemas al tiempo. En el caso de los vocálicos los fonemas escogidos fueron `/a/` y `/e/` debido a que su

frecuencia dentro del español es la más alta. Se probó también con redes que discriminaran entre los 5 fonemas pero eran más lentas y se prefirió manejarlas independientes, para concentrarse en las características propias y poder analizar mejor los resultados separados, antes que mezclar diferentes tipos de fonemas.

Todas estas redes se crearon y se entrenaron en ambiente de MATLAB, fueron guardadas en disco como archivos `.mat` para más adelante con los programas de presentación de resultados, calcular o leer los datos que se tenían almacenados en estos archivos. Los programas que se desarrollaron para entrenar estas redes fueron `entptk.m` y `entae.m`, para entrenar por bloques de redes se utilizaron los programas: `entptk4.m` que incluye cuatro redes una de cada una de los cuatro tipos de ventanas, `entptk5.m` que incluye las cinco redes de cada experimento con uno de los cuatro tipos de ventanas, `entptk5x4.m` entrena las 20 redes de la transformada especificada como parámetro de entrada. (Ver Anexo H)

3.4 PRESENTACIÓN

3.4.1 Presentación de resultados

Por cada experimento se realizan tres matrices de $n \times n$, donde n es el número de fonemas a reconocer. La primera matriz es la matriz de aceptación y coloca en la fila i , columna j , el número de veces que dado el i -ésimo fonema, aceptó como el j -ésimo fonema. La diagonal contiene las aceptaciones correctas y el resto de la matriz contiene las falsas aceptaciones.

La segunda matriz es la matriz de rechazos. En esta matriz la fila i , columna j , contiene el número de veces que dado el i -ésimo fonema, rechazó como el j -ésimo fonema. La diagonal contiene las falsas aceptaciones y el resto muestra los rechazos correctos.

La tercera matriz es la matriz de indecisiones y se arma de forma análoga a las anteriores. El número total de indecisiones consiste en la suma de los elementos de esta tercera matriz. El

número total de aciertos es la suma de la diagonal de la primera matriz con los elementos fuera de la diagonal de la segunda y el número total de desaciertos es la suma de la diagonal de la segunda matriz con los elementos no diagonales de la primera.

Cuando se agrupan experimentos o cuando la representación de fonemas es diferente dentro de un experimento, las matrices de totales no son fáciles de analizar al carecer de una referencia y se pueden preferir las matrices de porcentaje. Las matrices de porcentaje se arman igual a las matrices de totales pero finalmente cada valor en la i -ésima fila es dividido por el número total de ocurrencias del i -ésimo fonema en el experimento o los experimentos agregados.

3.4.2 Programas utilizados

Los programas utilizados en esta parte de presentación de resultados fueron: `simptk.m` que con un cambio en uno de los parámetros de entrada funcionaba para `/a/` y `/e/`, funcionaba simulando las redes con los diferentes conjuntos de entrada y calculando sus respectivas matrices de aceptación rechazo e indecisión por medio de otra función, la función que permitía establecer los límites de delta y que además calcular las matrices de aceptación rechazo e indecisión fue: `acierto_error.m` la función que calculaba el error cuadrado medio de la simulación de las redes con otros conjuntos de entrenamiento fue: `perfsimptk.m` que calcula el error cuadrado medio de una red, con `perfsimptk4.m`, `perfsimptk5.m`, `perfsimptk5x5.m`, `perfsimptk5x5x4.m` se logran obtener de una vez varios errores de diferentes redes de la misma manera que en el punto de clasificación con `entptk`. (Ver Anexo H)

4. ANÁLISIS DE RESULTADOS

4.1 ANÁLISIS DE PRUEBAS FALLIDAS

En un trabajo de investigación son muchas las variables con las que no se cuentan que aún cuando se planea rigurosamente, se va a incurrir errores, una de las partes más valiosas del trabajo es un análisis de los errores que se tuvieron, ya que de ellos se aprende y da la posibilidad a otras personas de que no los repitan.

Gran parte de las primeras pruebas realizadas durante el trabajo fueron infructuosas debido a que no se utilizó una metodología adecuada, las dimensiones de los datos de entrada eran muy grandes y aún así no se garantizaba una representación de los fonemas, también había fallas graves en la rotulación que no se sabía hasta que se replanteó el concepto de rotulación.

El entrenamiento por frases no fue el adecuado porque producía, como se mencionó anteriormente, carga de la red hacia los fonemas más frecuentes y más largos del español.

4.1.1 Segmentación uniforme

Se usa para el entrenamiento y pruebas de frases completas. Entre los inconvenientes presentados para el entrenamiento se encuentra que las ventanas no corresponden a los fonemas rotulados, la representación de los fonemas depende de su representación en tiempo en las frases y no de su número de ocurrencias. No todos los fonemas están representados. Si se aumenta el número de frases, no se garantiza una presencia más equitativa de los fonemas pero sí aumenta la base de entrenamiento con el respectivo aumento de las necesidades de proceso.

4.1.2 Reconocimiento de los 18 fonemas

Intentar reconocer los 18 fonemas al tiempo requiere una complejidad muy grande en la red neural la cual se refleja en la demanda de recursos y el tiempo invertido en el proceso.

4.1.3 Número muy grande de coeficientes

A falta de criterios claros para la reducción de los coeficientes para la representación en ondas, se intentó introducir la totalidad de los coeficientes que arroja la transformada, se probaron con los más grandes, con los más grandes por nivel y no funcionó. Se buscaba con esto descubrir los más representativos, pero en la práctica el conjunto resultaba muy grande. Se tiene que tener un concepto claro de reducción para aplicar a una transformada de este tipo. Lo que puede servir en compresión no necesariamente es bueno para reconocimiento, ni tampoco malo, lo demuestran los resultados.

4.1.4 Rotulación imprecisa

El programa usado para la rotulación de la mayor parte de las frases, que fue realizado por los autores del trabajo, no tenía mayores comprobaciones de error, por lo que la rotulación era susceptible de errores de anotación. Adicionalmente no había criterios claros para distinguir unos fonemas de otros, se permitía el solapamiento cuando se consideraba que una muestra podía representar a dos sonidos consecutivos y las personas que ayudaron en la rotulación tenían criterios diferentes para aceptar o rechazar los sonidos y la calidad variaba entre sujetos, la experiencia adquirida durante frases previas y el cansancio.

Es un error tratar de rotular todo y es un error delegar la rotulación, se deben conseguir herramientas adecuadas para el manejo de la rotulación se tiene que estar entrenado y saber qué es lo que se está rotulando, una sola persona rotulando permite tener el mismo criterio de rotulación en todos los casos .

4.1.5 Bases de datos de entrada muy grandes e imprecisas

En las pruebas en las cuales el número de coeficientes de entrada era muy grande, el tiempo de proceso y la demanda de memoria son muy grandes. En muchas de las pruebas realizadas el proceso abortaba después de varias horas de trabajo reportando memoria insuficiente (incluyendo RAM + memoria virtual en disco duro).

Las imprecisiones en la rotulación acentuadas por el solapamiento propio de la segmentación uniforme hacen que en los casos en los que se logra la convergencia durante el entrenamiento, muestren resultados erráticos. Sin un cálculo detallado de error salvo la comparación visual entre los resultados deseados, la muestra de voz y los resultados obtenidos, se puede asegurar que el sistema no reconocía. Esto hace pensar que la red neural no generalizaba sino que se aprendía de memoria las frases.

4.2 ANÁLISIS DE PRUEBAS FINALES

4.2.1 Análisis de los entrenamientos

El tipo de ajuste de la ventana dentro del fonema y la forma de selección del fonema dentro del conjunto de frases, el tipo de transformada y los fonemas utilizados son las variables que influyen en el siguiente análisis comparativo.

Cada combinación de variables se entrenó diez veces, esto puede mostrar algunas tendencias, mas no resultados concluyentes, sobre la convergencia que puede tener cada conjunto de variables.

Según la definición de error definida en el capítulo de especificaciones tenemos que:

Las confrontaciones de los entrenamientos que arrojaron valores de error finales entre $e^2/(N \cdot n)$ y e^2 , en este trabajo entre: $6,67 \times 10^{-5}$ y 9×10^{-2} con $e = 0.3$. mostraron al menos un desacierto en todos los casos, lo que significa que la red no quedó completamente entrenada.

En la siguiente tabla de combinaciones se muestran la cantidad de redes entrenadas y las no entrenadas para su comparación:

Tabla 1. Tablas de convergencia

Transformada	Fonemas	Ajuste	Selección	Entrenadas	No entrenadas
onditas	p t k	aleatorio	aleatorio	3	2
onditas	p t k	aleatorio	uniforme	3	2
onditas	p t k	normal	aleatorio	4	1
onditas	p t k	normal	uniforme	4	1
cepstro	p t k	aleatorio	aleatorio	6	4
cepstro	p t k	aleatorio	uniforme	7	3
cepstro	p t k	normal	aleatorio	8	2
cepstro	p t k	normal	uniforme	5	5
onditas	a e	aleatorio	aleatorio	2	8
onditas	a e	aleatorio	uniforme	3	7
onditas	a e	normal	aleatorio	0	10
onditas	a e	normal	uniforme	2	8
cepstro	a e	aleatorio	aleatorio	8	2
cepstro	a e	aleatorio	uniforme	10	0
cepstro	a e	normal	aleatorio	6	4
cepstro	a e	normal	uniforme	7	3

Salvo en la combinación: vocales/normal/aleatorio con onditas, cada combinación de variables arrojó al menos una red entrenada. Se puede interpretar que los coeficientes si son suficientes para lograr una convergencia siempre y cuando las condiciones iniciales de punto de partida sean las correctas. Esto sin embargo, no es garantía de que los coeficientes sean los correctos para generalizar.

Se destaca el caso de las vocales, que entrenan sin mayores dificultades con cepstro pero tienen problemas con onditas, esto indica que los coeficientes para onditas no son tan claros o no están en un espacio lo suficientemente separado para que la red converja rápidamente.

Las consonantes tienen un comportamiento más homogéneo, al menos con los datos obtenidos esto hace pensar que onditas aunque es excelente para compresión no es descabellado utilizarla en reconocimiento, o por lo menos en cierto tipo de fonemas como los explosivos.

Se tiene claro que el hecho de que, aunque una red converja no es garantía de que sea capaz de generalizar.

El error cuadrado medio (MSE), es una medida de error continua que compara la simulación de un conjunto de entrada a través de la red con un conjunto de objetivos. El MSE es una medida global y el valor que arroja puede no ser muy representativo sobre el comportamiento de la red como clasificador.

El número de desaciertos es una medición de error que es más importante para evaluar el desempeño de un clasificador. Esta medición está ligada a la forma como se considere un acierto o un desacierto. En particular, si la evaluación consiste en decidir una entre las posibles salidas o si decidir por cada salida si se trata de una aceptación o un rechazo.

4.2.2 Análisis de pruebas

En un experimento de clasificación donde el objetivo es (0; 1; 0) se interpreta que la muestra corresponde al segundo de tres posibles valores. Si el resultado de la simulación es (0,06; 0,98; 0,04) se puede aplicar un criterio de tomar el valor más alto, en este caso el segundo valor 0,98 y se tiene un acierto. En este ejemplo el MSE es de 0,0019. Si se aplica el mismo criterio de tomar el valor más alto, en otra simulación que arroja valores como (0,09; 0,10; 0,02), implicaría un acierto, pero ahora el MSE es de 0,2728 y la diferencia entre 0,09 y 0,10 puede interpretarse como poco relevante para tomar una decisión, al tiempo que 0,10 es un valor más cercano a cero que a uno. Un buen criterio debe establecer, por lo tanto, un límite para la decisión y dejar un campo adicional para la indecisión.

4.2.3 Resultados consolidados en consonantes

Las siguientes tablas de resultados consolidados muestran el porcentaje de fonemas x (en la fila) que fue aceptado, rechazado o indeciso al ser reconocido como y (en la columna respectiva). Para la creación de estas tablas se tomaron sólo los resultados de las redes que convergieron completamente.

El caso ideal debe mostrar una matriz identidad en “Aceptación” y su complemento a uno (100%) en “Rechazo”.

Tabla 2. Onditas /p/, /t/, /k/; ajuste aleatorio, selección aleatoria, con exclusión

Fonema	Aceptación			Rechazo			Indecisión		
	p	t	k	$\neg p$	$\neg t$	$\neg k$	$\zeta p?$	$\zeta t?$	$\zeta k?$
/p/	94,2%	2,2%	6,1%	5,7%	97,8%	93,9%	0,1%	0,0%	0,0%
/t/	1,3%	94,1%	3,5%	98,4%	5,2%	96,4%	0,3%	0,6%	0,1%
/k/	2,8%	8,6%	88,0%	97,1%	91,1%	11,4%	0,2%	0,3%	0,6%

Tabla 3. Cepstro /p/, /t/, /k/; ajuste aleatorio, selección aleatoria, con exclusión

Fonema	Aceptación			Rechazo			Indecisión		
	p	t	k	$\neg p$	$\neg t$	$\neg k$	$\zeta p?$	$\zeta t?$	$\zeta k?$
/p/	85,5%	5,1%	7,5%	13,9%	94,5%	92,2%	0,6%	0,4%	0,3%
/t/	7,7%	86,5%	5,1%	91,9%	12,4%	94,4%	0,4%	1,1%	0,5%
/k/	4,8%	7,9%	87,7%	94,8%	91,9%	11,4%	0,4%	0,2%	0,9%

En estos casos, la diagonal de la matriz de aceptación muestra el porcentaje de aceptaciones correctas (reconocimiento positivo correcto). El complemento de esta matriz muestra el porcentaje de falsas aceptaciones.

En la matriz de rechazos la diagonal muestra el porcentaje de falsos rechazos, mientras que el complemento muestra el porcentaje de rechazo correcto.

Tener índices bajos de indecisión, muy por abajo de los índices de falso reconocimiento y falso rechazo, es un síntoma de que la red fue entrenada antes de cumplir los parámetros de salida y que la mejora del MSE se debe a que el entrenamiento afinó los niveles de tolerancia sobre la decisión tomada. Esto se traduce en las pruebas como una alta probabilidad de que la red se decida acertada o desacertadamente.

Sin embargo los índices de falsa decisión son bajos en comparación a los de decisión correcta, por lo que puede concluirse que las redes sí generalizaron.

En la comparación entre onditas y cepstro, hay una tendencia a favor de onditas. En particular en el caso de las pruebas hechas con selección aleatoria de fonemas y ajuste aleatorio de ventana, se observa:

Mejor reconocimiento afirmativo de la /p/ en onditas: 94,2% contra 85,5%.

Mejor reconocimiento afirmativo de la /t/ en onditas: 94,1% contra 86,5%.

Mejor reconocimiento afirmativo de la /k/ en onditas: 88,0% contra 87,7%.

Mejor rechazo de la /k/ como /t/ en cepstro: 91.9% contra 91.1%.

Comparando la globalidad de los experimentos, hay una tendencia a mejores resultados para onditas en contraposición con cepstro, pero tal tendencia no es contundente. En varios casos los resultados para cepstro son individualmente mejores que los de onditas.

La misma matriz consolidada de selección aleatoria y ajuste aleatorio en onditas, pero sin excluir los resultados de redes no entrenadas, muestra cómo las matrices no entrenadas afectan.

Tabla 4. Onditas /p/, /t/, /k/; ajuste aleatorio, selección aleatoria, sin exclusión

Fonema	Aceptación			Rechazo			Indecisión		
	p	t	k	¬p	¬t	¬k	¿p?	¿t?	¿k?
/p/	87,7%	1,5%	6,1%	12,0%	98,3%	93,8%	0,3%	0,1%	0,1%
/t/	4,6%	75,5%	2,8%	95,2%	24,1%	97,1%	0,2%	0,5%	0,1%
/k/	2,5%	6,9%	70,6%	97,3%	92,8%	28,9%	0,2%	0,3%	0,5%

No necesariamente los resultados mejoran al excluir experimentos con matrices no entrenadas, salvo que las matrices no tengan tendencia al entrenar.

En este caso en particular, el resultado de la segunda instancia afecta enormemente los resultados (ver C.1.2, página 74).

Sin embargo en otro experimento, se puede observar:

Tabla 5. Onditas /p/, /t/, /k/; ajuste binario, selección uniforme, sin exclusión

Fonema	Aceptación	Rechazo	Indecisión
--------	------------	---------	------------

	p	t	k	$\neg p$	$\neg t$	$\neg k$	$\xi p?$	$\xi t?$	$\xi k?$
/p/	88,7%	0,4%	9,7%	11,0%	99,3%	90,1%	0,3%	0,3%	0,2%
/t/	0,2%	96,7%	2,4%	99,7%	2,8%	97,3%	0,1%	0,5%	0,3%
/k/	2,9%	5,3%	91,3%	97,1%	94,3%	8,0%	0,1%	0,4%	0,7%

Tabla 6. Onditas /p/, /t/, /k/; ajuste binario, selección uniforme, con exclusión

Fonema	Aceptación			Rechazo			Indecisión		
	p	t	k	$\neg p$	$\neg t$	$\neg k$	$\xi p?$	$\xi t?$	$\xi k?$
/p/	89,7%	0,2%	9,4%	10,0%	99,7%	90,4%	0,3%	0,1%	0,2%
/t/	0,3%	96,1%	2,8%	99,6%	3,3%	96,9%	0,1%	0,6%	0,3%
/k/	3,5%	5,5%	90,3%	96,5%	94,1%	8,7%	0,1%	0,4%	0,9%

Donde los resultados para entradas /t/ y /k/ son mejores al tomar los resultados de las redes que con convergieron completamente.

Una de las redes excluidas en la Tabla 2 fue el resultado de un experimento en el cual la matriz final rechazaba todos los fonemas como /t/ y como /k/ y tenía un MSE de $2,59 \times 10^{-1}$ con el propio conjunto de entrenamiento. Excluir esta red se justifica como un entrenamiento fallido. Los otros entrenamientos no perfectos pueden arrojar tasas de reconocimiento superiores a los entrenamientos perfectos y la decisión de excluirlos o no de los consolidados finales es menos clara. Como no hay un criterio claro de “que tan imperfecto” un experimento puede ser, se decidió que se incluiría cero imperfección en el entrenamiento.

Los experimentos con entrenamientos no perfectos que pueden dar mejores resultados que las matrices completamente entrenadas pueden interpretarse, sin embargo, como un buen síntoma, porque indica que la red es capaz de generalizar, aún cuando en el conjunto de entrada existan algunas pocas muestras que no se ciñen a los parámetros aprendidos. Por el contrario, un entrenamiento perfecto, que implique cero desaciertos y cero indecisiones, pero que logra pobres tasas de reconocimiento, puede ser síntoma de que la red no generalizó sino que se aprendió de memoria las muestras de entrada. Sin embargo, este no parece ser el caso de ninguno de los experimentos realizados.

4.2.4 Resultados consolidados en vocales.

La siguiente tabla muestra los consolidados de todas las pruebas con redes que entrenaron, discriminadas por la forma de seleccionar los fonemas y ajustar las ventanas, para el caso de distinguir las vocales /a/ y /e/ con onditas.

Tabla 7. Consolidados totales, /a/ y /e/ con onditas.

consolidados	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
aleatorio/ aleatorio	a	70,5%	28,5%	28,5%	70,5%	0,9%	0,9%
	e	26,1%	72,8%	72,5%	26,1%	1,3%	1,1%
uniforme/ aleatorio	a	73,7%	25,2%	25,2%	73,7%	1,1%	1,1%
	e	28,9%	70,3%	70,3%	28,9%	0,8%	0,8%
aleatorio/ binario	a	#####	#####	#####	#####	#####	#####
	e	#####	#####	#####	#####	#####	#####
uniforme/ binario	a	83,5%	16,3%	16,3%	83,5%	0,3%	0,3%
	e	34,8%	63,9%	63,9%	34,8%	1,3%	1,3%

La siguiente tabla muestra los consolidados de todas las pruebas con redes que entrenaron, discriminadas por la forma de seleccionar los fonemas y ajustar las ventanas, para el caso de distinguir las vocales /a/ y /e/ con cepstro.

Tabla 8. Consolidados totales, /a/ y /e/ con cepstro.

consolidados	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
aleatorio/ aleatorio	a	97,0%	2,7%	2,8%	97,1%	0,2%	0,2%
	e	5,6%	94,1%	94,1%	5,5%	0,3%	0,4%
uniforme/ aleatorio	a	98,0%	1,8%	1,8%	98,0%	0,2%	0,2%
	e	3,5%	96,2%	96,2%	3,6%	0,3%	0,2%
aleatorio/ binario	a	94,9%	4,8%	4,9%	95,1%	0,2%	0,1%
	e	4,8%	94,9%	95,0%	4,8%	0,2%	0,3%
uniforme/ binario	a	96,5%	3,5%	3,3%	96,2%	0,2%	0,3%
	e	4,6%	95,1%	95,2%	4,7%	0,2%	0,2%

Claramente se observa una ventaja a favor de cepstro, la cual es consecuente con los resultados de entrenamiento de las redes.

Sin embargo los resultados consolidados de onditas distan de ser aleatorios, y dado que sólo se exploró con siete coeficientes de energía, contra 14 coeficientes en cepstro, es posible que al incluir otros coeficientes se puedan obtener mejores resultados para distinguir estos dos fonemas.

5. CONCLUSIONES

Dentro de las conclusiones del trabajo se incluyen dos tipos de conclusiones las de los experimentos en si y las obtenidas como experiencia al realizar el trabajo.

5.1 COMPARACIÓN ENTRE ONDITAS Y CEPSTRO

De los resultados de las pruebas se puede concluir que bajo las condiciones dadas del experimento, cepstro es definitivamente mejor que onditas en la tarea de extracción de características para reconocimiento de fonemas vocálicos, sin importar el ajuste de ventanas, o la escogencia de fonemas dentro de los archivos.

Respecto a los fonemas explosivos no se puede concluir de manera determinante que onditas es mejor. Según los datos, onditas es ligeramente mejor, pero no se puede establecer con absoluta seguridad, dado que los márgenes de error son estrechos.

Aunque onditas es excelente en compresión, no quiere decir que no sea buena en extracción de características para reconocimiento de fonemas, se puede asegurar que con los coeficientes de energía por nivel, se obtienen buenos resultados para los fonemas explosivos y no tan malos para los fonemas vocálicos lo cual indica que se debe explorar más, la forma de extraer coeficientes de onditas, las diferentes onditas madre, y los diferentes niveles de filtrado.

En forma general se puede afirmar que cepstro es una herramienta excelente para comparar con cualquier otro método, da una idea clara de qué tan bueno o malo es el otro método y ayuda a discernir cuando algo va por mal camino si no funciona con cepstro.

5.2 DEASARROLLO DEL TRABAJO

La rotulación es un trabajo muy grande, arduo y es crítico para el resultado final. Fue la parte que más retrasó los experimentos, no se sabía rotular hasta que por ensayo y error se lograron buenos resultados y se observó que se había comenzado a rotular mal desde el principio, no hubo criterios claros para distinguir los fonemas se delegó rotulación y es algo que se tiene que hacer de forma consistente y por una persona que utilice los mismos criterios de rotulación en todos los archivos y los revise fonema por fonema más de una vez entre prueba y prueba. Hay que contar con una buena herramienta de rotulación.

Los sonidos explosivos tomados desde la oclusión hasta el ataque contienen espacios de silencio que pueden ser demasiado grandes y con poca o nula información en el silencio. Se obtienen mejores resultados cuando se toma sólo el ataque.

Las dimensiones del trabajo se deben reducir al máximo posible para luego ir subiendo la complejidad, al trabajar y hacer pruebas con miles de datos se pierde mucho tiempo, sobretodo cuando se trata de pruebas preliminares donde lo que se quiere es evaluar tendencias.

Todo el desarrollo del software debe ser modular de tal forma que si se desea cambiar cualquier parámetro de alguna etapa lo pueda hacer y el cambio sea reversible.

BIBLIOGRAFÍA

ANTOGNETTI, Paolo and MILUTINOVIC, Veljko. Neural Networks : Concepts, Applications and Implementations. *s.l.* : Prentice Hall, 1991. v. 1.

CLARKSON, Peter and STARK, Henry. Signal Processing Methods for Audio Images and Telecommunication. *s.l.* : Academic Press, 1995.

COLESTOCK, M. A. Wavelets a New Tool for Signal Processing Analysts. AIAA/IEEE, Digital Avionics System Conference, 12th DASC, 1994. p. 54-59

DAUBECHIES, Ingrid. Where Do Wavelets Come From? : A Personal Point of View. En Proceedings Of the IEEE. Vol. 84, No. 4 (abr. 1996); p. 510-513

DAVENPORT, M. R. and GARUDADRI, H. A Neural Net Acoustic Phonetic Feature Extractor Based on Wavelets. En : IEEE Pacific Rim Conference of Communication Computers and Signal Processing (may. 1991) p. 449-453.

DELLER, John R., HANSEN, John H. L. and PROAKIS, John G. Discrete-Time Processing of Speech Signals. *s.l.* : Prentice Hall, 1993.

D'MESSANDRO, C. and Richard G. Random Wavelet Representation of Unvoiced Speech. *s.l.* : IEEE, 1992. p. 41-44.

FAVERO, R. F. Compound: Wavelets for Speech Recognition. En : Proceedings of the IEEE-SP International Symposium Time-Frequency and Time Scale Analysis. 1994. p. 600-604.

FAVERO, R. F. and KING, R. W. Wavelet Parameterization for Speech Recognition : Variation in Translation and Scale Parameters. En : IEEE International Symposium on Speech, Image Processing and Neural Networks (abr. 1994 : Hong Kong). p. 694-698.

FORERO, Andrés; NAVARRO, Miguel y OTERO, Andrés. Introducción a la Transformada Onditas. Bogotá, 1996. Trabajo de grado (Ingeniero Electrónico). Pontificia Universidad Javeriana. Facultad de Ingeniería. Departamento de Electrónica.

FREEMAN, J. A. y SKAPURA, David M. Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones. *s.l.* : Adison Wesley, 1993.

FUENTES, M. y ZULUAGA, D. Estudio de los Modelos Ocultos de Markov para Reconocimiento de Voz. Bogotá, 1994. Trabajo de Grado (Ingeniero Electrónico). Pontificia Universidad Javeriana. Facultad de Ingeniería. Departamento de Electrónica.

GALLI, A. W.; HEYDT, G. T. and RIBEIRO, P. F. Exploring the Power of Wavelets Analysis. *s.l.* : IEEE, oct. 1996. p. 37-41.

GHITZA, O. Auditory Models and Human Performance in Task Related to Speech Coding and Speech Recognition. En : IEEE Transaction on Speech and Audio Processing. Vol. 2, No. 1 (ene. 1994); p. 115-132.

GIDAS, B. and MURUA, A. Classification and Clustering of Stop Consonants Via Non Parametric Transformations and Wavelets. *s.l.* : IEEE, 1995. p. 872-876.

GONZÁLEZ, E. y MERCHÁN J. Cuantificación Vectorial Para el Reconocimiento de Voz. Bogotá, 1993. Trabajo de Grado (Ingeniero Electrónico). Pontificia Universidad Javeriana. Facultad de Ingeniería. Departamento de Electrónica.

GRANADOS, Johner y MEDINA, Vladimir. Reconocimiento de Habla Discreta por medio de la Transformada Wavelet. Bogotá, 1997. Trabajo de Grado (Ingeniero Electrónico). Pontificia Universidad Javeriana. Facultad de Ingeniería. Departamento de Electrónica.

GRAPS, A. Introduction to Wavelets. *s.l.* : IEEE Computational Science & Engineering, 1995. p. 58-61.

GREENWOOD, D. A Cochlear Frequency-Position Function for Several Species -29 Years Later. En : Journal of Acoustic Society of America. Vol. 87, No. 6 (1990); p. 2592-2605.

GUYTON, Arthur C. Anatomy and Physiology. *s.l.* : Saunders College, 1984.

HABIT, A. Introduction to Wavelets. *s.l.* : IEEE, 1995. p. 879-886.

HAYKIN, Simon S. Neural Networks : A Comprehensive Foundations. *s.l.* : Macmillan, 1994.

LEE, T.; CHING, P. and CHAN, L. W. Recurrent Neural Networks for Speech Modeling and Speech Recognition. *s.l.* : IEEE, 1995. p. 3319-3323.

LI, W. et al. Recurrent Sub-Neural Networks Applied to Speech Recognition. En : International Symposium on Speech, Image and Neural Networks (abr. 1994 : Hong Kong). p. 233-237.

LIU, W.; GOLDSTEIN, M. H. and ANDREU, A. G. Multiresolution Speech Analysis With the Analog Cochlear Model. En : Proceedings of the IEEE-SP International Symposium Time-Frequency and Time Scale Analysis, 1992. p. 433-436.

MALBOS, F.; BAUDRY, M. and MONTRESOR, S. Detection of Stop Consonants With the Wavelets Transform. *s.l.* : IEEE, 1994. p. 612-616.

MORA, B.; GUALDRON, O. y TORRES Y. Reconocimiento de Fonemas mediante el uso de Redes Neuronales. En : Primer Congreso Nacional de Neuro Computación, 1995. p. 110-124.

PATERNINA, Walter y OROZCO, Javier M. Desarrollo y evaluación de Técnicas de Reconocimiento de Voz en Español. Bogotá, 1995. Trabajo de Grado (Ingeniero Electrónico). Pontificia Universidad Javeriana. Facultad de Ingeniería. Departamento de Electrónica.

RIOUL, O. and VETTERLI, M. Wavelets and Signal Processing. En : IEEE Signal Processing Magazine. Vol. 8, No. 4 (oct. 1991).

RIS, C.; FONTAINE, V. and LEICH, H. Speech Analysis Based on Malvar Wavelets Transform. *s.l.* : IEEE, 1995. p. 389-393.

WILSON, R. Wavelets?. *s.l.* : IEEE, 1994.

ÍNDICE

alófono

definición, xxii

cepstro, 25

definición, xxii

cóclea, 26

fonema

definición, xxii

fonemas, xxii, 170, 179

fono

definición, xxii

fourier

definición, xxii

transformada, xxii

transformada inversa, xxii

Fourier

transformada, 25

Markov

modelos ocultos, 53

onditas, xxii, 26

definición, xxii

ondita madre, 26

red neural, 26

wavelets. Véase onditas

NEXO A. BASE DE DATOS DE PRUEBAS

A.1 Frases, listado 1.

logística y mercadeo tienen
 embargo le dan algún
 del servicio por parte
 a veces es mejor
 evaluación es el conjunto
 entonces tiene algunos buenos
 de varios locales comerciales
 de hacer servir algo
 en empresa de servicio
 secciones de la tesis
 sufrimiento de los niños
 a domicilio y horario
 cultural de la nacionalidad
 básicos para el cálculo
 a conocer su cultura
 escasas horas de definir
 hace que los cálculos
 una empresa puede dividir
 que Montoya no será
 a pesar de estar
 el Coliseo el Campín
 de amor más especial
 la oportunidad de aplicar
 comparación desde el punto
 esta razón las aprobaciones
 el mediano el piñón
 en las que inicialmente
 de imponer los grupos
 recelo que inspiran todas
 los eructos son síntomas
 mismo episodio Jorge Posada
 se subdividen en seis
 el promedio de consumo
 del volumen de sólidos

García especial para viajar
 se abstuvo de señalar
 cambios en los procesos
 puede aliviarse al modificar
 fosfato que es menos
 en función de objetivos
 conocimiento de la organización
 de nivel dos dónde
 ello se pueden utilizar
 no sin una dosis
 número de compañías internacionales
 usan una medida propia
 país mal hecho cuya
 que había una nueva
 integral y el resto
 error que se comete
 las instalaciones de bienes
 de las casas disuergas
 la presidencia de dicha
 y con el otro
 costo de los cultivos
 abrirlo es necesario romper
 la utilización del caucho
 la institución con relación
 que tiene un compromiso
 del libro vectores carácter
 y hacia donde irían
 esperar que se tope
 los fluidos y control
 el satanismo se tome
 por componentes o créditos
 gran acogida el programa
 así llegaron las cosas
 se le ha retirado

logró conformar un cartel
 vez que grandes masas
 recibe por el alquiler
 estos dos últimos sistemas
 presentan momentos de valor
 de la segunda guerra
 también por el escritor
 alienta y es útil
 colores y los vestidos
 anterior se hará teniendo
 de consecución y transporte
 en los Estados Unidos
 no es solo problema
 azúcar señala al sacchara
 y quienes se sientan
 generalmente con un líder
 la tecnología más adecuada
 apesta como un cadáver
 para lograr un proceso
 a los supermercados ubicado
 salen a la superficie
 el mundo se encuentran
 que tenemos que salir
 el hablante más cercano
 ganancias crecían a ritmos
 mucho en la parte
 donde se ha hecho
 el concepto de servicio
 ubicación de las mazas
 o causas que inciden
 obrero introduce los otros
 colaboró en la redacción

A.2 Frases, listado 2.

con dos grandes cambios
 las directivas del hospital
 por medio de tres
 de minas se ratificó
 el lanzamiento de productos
 la respuesta del modelo
 una de las alumnas
 la década pasada confinadas
 dolor en el pecho
 su asesor puede ayudar
 hicieron que la creciente
 de que revelara secretos
 motivación para estos sería
 experiencia con un pequeño
 muy altas de balso
 y cargas ha situado
 extraditaría y a otros
 actividad era la minería
 al estado del lado
 nuevo sistema puede bajar
 teniendo en cuenta todos
 y sin un conocimiento
 exige que se haga
 de indagar y crear
 buscaron por las farmacias
 las tasas de retorno
 le pidió al alcalde
 tal como lo solicitó
 que es muy común
 cuales no hay apelación
 es importante no solo
 química en la escuela
 unidos por el medio
 la televisión por suscripción

explica como se hace
 frente a los métodos
 y con un peso
 el comportamiento que tiene
 de insuficiencia de ingreso
 la mezcla no sólo
 el presidente de nueva
 ve en la figura
 el reto de incrementar
 en México se comercializaba
 a veces se incluye
 también a las acciones
 ni puede decir nada
 el dormir poco contribuye
 de que el profesor
 crema que rápidamente mostré
 que conforma el proceso
 de otra persona porque
 análisis de los datos
 cultivo por otra parte
 un labrado que imita
 su empresa le ayude
 cultivada en el país
 e inspeccionar la seguridad
 que promueve la industria
 y propietario de productos
 impacto de las balas
 mientras en la calle
 quemas temas que atañen
 aplicada en muchos casos
 en áreas de actividad
 la mayoría se incorpora
 grandes bases de datos
 que se deben incluir

que no bombardearan buenos
 partir de categorías previa
 dentro de estos documentos
 con el mismo número
 para dar al lector
 se orientó a atender
 las naciones unidas reglamentan
 seguridad para los pueblos
 al país el primer
 ver con la actitud
 por todas las acciones
 cada vez que habla
 se gana por encima
 ha encontrado el orden
 la abundancia que adornan
 y de los propósitos
 apoyará en una secretaría
 le llegó la hora
 objetivo en los aspectos
 o conjunto de tejidos
 claro y más fácil
 a construir redes urbanas
 la cual ha obtenido
 fue por la declaración
 exactitud lo que debe
 utilizaremos en este caso
 que este proceso evaluativo
 de galones de petróleo
 los más asustados eran
 con la leve ayuda
 con firmeza y metía
 obtener y procesar dicha

A.3 Frases, listado 3.

de gas para Bogotá
 padres de familia discuten

los canales y entes
 un periodista no tiene

debido los malos manejos
 la prensa ha aprendido

que el gobierno garantice del proceso de obtención abordar ayer el tema de gas natural llegó que ya se habían mano de obra indirecta manos vacías y seguir un escrito científico debe la autorización y visto cumplido la fecha límite un estudiante de hacer de personal se describe su concepción fueron adapta la obtención del melote de los jueces implicaron se indica la energía la campaña arranca justo como un verdadero retroceso e implementar un proceso los productos al mercado los padres se refleja mi mamá me puso efectos de tres elementos una estrategia basada solo de la empresa renovación accionaria que se indica esta forma se puedan base de datos consta disminución en el monto de selección que describe cuales son las principales la prospección y trámites

imaginarse lo que significa le dijo que mientras cuando juró hacerlo hace manera breve y concisa estancia a la hacienda colombiano y con otros los puntos de dicha flujo de personas dentro espontánea de usos preserva a la fase antes la entonación y además pesar de las prevención humildad es bonita porque acuerdo sobre los derechos qué horas se precipitó sur tienden a ubicarse presidencia de la república compañías formarán un porta sólo el ser absoluto de salsas se hace misma ciencia se expresara responder a tan importante una oficina de pasajes encuentran en el nivel me daba unos siete docentes que no quiere mercado el mercado total la ciencia y debe que no da lugar de consumo que patrocinaban gracias a su vertiginoso de un consejo distrital

paciencia para no desespera con la mirada puesta el producto más avanzado despachos y la producción valoren más lo interior al cual se dirigirá en la fundación hilos lima que el defensor el poder y mejoró con capacidad para tres de sus ingresos provienen el cual se desarrollaron un factor de ganancia ya que ésta enriquece ésta pueda ser comprendida hacerse un solo incremento cada vez más adultos en marcha la extradición procesos en curso cuya lácteos entre las cuales cargo se les realizan situados en el sector calculó que el promedio cuenta que la empresa que implica una labor sea por una reacción al estado como garante índice de logro educativo podrían tener un trabajo cabaña de trabajo patricio

A.4 Frases, listado 4.

el jugador colombiano puede simultánea con las tasas la intensidad del sonido diferencias entre un caso de Bogotá es atendido procesos de sus diferentes en todos los casos daños a las tuberías generan un efecto negativo de patrones un sistema

mercadeo o con directivos paila se forman hilos allí ni los publicistas encontrar quién se interesa Ospina Rodríguez y José a la empresa transportadora económica de la finca líder en el mercado la contratación y oferta caldera a vapor intervienen

de red que facilitan llantas elaboradas a partir logro de una evaluación de expansión del área y puerto rico superan pedía que no bombardearan Villa Carmenza Soledad Norte con un aumento importante se evita el deterioro juicios sobre el avance

el centro de Bogotá
 los medios para satisfacer
 de que la corte
 de terminar su gira
 que las compañías comienzan
 tratarán con mayores limita
 no significa que deben
 elaborado incluye una guía
 una vez se toma
 presidencia dijo a través
 cadena al quedar paralizado
 sufrimiento de los niños
 eficiencia del método tres
 de dar y recibir
 hasta tanto no haber
 utilizadas en la industria
 no se dedicaban exclusive
 nombre de la empresa
 estratégicos de la ciudad
 la coincidencia de colores
 el atentado del jueves
 de personal se describe
 en la crisis mundial
 concretaron citas con altos

uno de los voceros
 otras propuestas se obtiene
 su boca salen todos
 se reduce el esfuerzo
 ganadores de la pasada
 su evolución en tres
 notas de afuera crítica
 Usme y la calle
 la empresa habían sido
 la señal del pulsos
 y protagonistas del proceso
 toda la gente juegue
 noche y la madrugada
 que sucedió y además
 sobre todo sus decisiones
 la secta les daría
 dormitorios y esto pasarían
 del alquiler es aportado
 al aula de clases
 tiene por objetivo mejorar
 diseño y de ingeniería
 afuera en la batalla
 no puede ser independiente
 faltado liderazgo por parte

que hacer el bien
 caracterizó por el libertina
 la entrada de exceso
 de policías y otras
 conceptos y teorías desafío
 de rapidez y proceso
 tener estas sectas satánica
 cuando las ideas vayan
 bebé por lo tanto
 las pails de cobre
 relacionados con el entorno
 y es una fuente
 el hombro y angustia
 o consultados por niños
 de datos y conexión
 sobre la estructura cayó
 y no será fácil
 buena alumna como puede
 esto no ha ahorrado
 diseño de la cartera
 efectuar en dos períodos
 ver con la actitud

A.5 Frases, listado 5.

realizaba con una rueda
 que la tesis puede
 pero el niño empezó
 éstos le dijeron durante
 práctica personal y social
 que fueron las compañías
 del concepto y pruebas
 y el insertar tres
 pasan por un clasificador
 manuales de funciones anexo
 las reglas y normas
 Villa y Pablo José
 capital de la república
 que se ha evaluado
 ocurrir es que esas
 musicales acaso lo mejor
 nos rodean un proceso
 las deudas con ninguno

más lo que sentó
 que la empresa ataca
 porque no le interesa
 la paz necesita mucho
 él no tan solo
 con ellas se observa
 parte no ha sido
 de ideas y posteriormente
 un déficit de controladores
 revela en su verdadera
 los dos maderos verticales
 y la información resultante
 rango medio de sistemas
 une con el bloque
 repiten en las aulas
 para obtener buenas notas
 obtenida en las condiciones
 a la persona continuar

todo caso no benefician
 procesos que ellos están
 de liberación de precios
 de sus diferentes áreas
 el profesor dirija todas
 esto incluye hacer respetar
 que ponen a disposición
 no son lo bastante
 la que se escoja
 regional completó la obra
 a procesos de erosión
 la mañana del domingo
 establecidos se le coloca
 la compañía deben pasar
 a orientar la discusión
 para la cual fueron
 a una presión máxima
 de hoy horóscopo hace

las FARC han alcanzado
lo cual es cierto
cuenta que en ninguna
interviene a la hora
Colombia sólo se produce
el manejo de datos
uno con una actualización
las sentencias por parte
las expuestas en clase
el ideal que impulsa
de vista lo correcto
ideado bajo el gobierno
solo en el sentido
la parálisis del crédito
la información y luego
mirar a nuestro alrededor

desvió por pocos centímetro
de mercados para analizar
inspirada en un cuento
encuentra en un punto
del museo de fotografía
una fuente te dice
ser elaborado un cronograma
caña para panela debe
ya que muchas personas
especial para el tiempo
aplicables al caso pero
sus usuarios estén dispuesta
en un curso tradicional
donde cambian de implemento
valor de la inversión
que según Manuel Alzate

el pasado domingo mató
áreas y su distribución
de las tres mostradas
gobierno que la liberación
la hornilla mejorada tabla
sistema ha generado proceso
una hoja los objetivos
castro lee los caballos
un blindaje legal mayor
de la canasta normativa
la facultad de comunicación
para el período estatutario
el hombre de negro
permitir una molestia extra

A.6 Frases, listado 6.

la población un servicio
disfrute de una vida
perciben rentas y controlan
moral a la protección
el sistema de percepción
que la industria panelera
con ellos y hágales
que el sector educativo
redacción de el tiempo
de la deuda externa
exactamente lo que trata
y en un tema
es la parte afectiva
de Antioquia están dispuesta
ve con mayor frecuencia
por encima del incremento
ser absoluto puede colmar
normas brindan un entorno
alarma por el posible
el alce son operaciones
surgió todo un éxodo
física de los estudiantes
a desarrollar la capacidad
frente a este tipo
ocupado un puesto destacado
los jugos pues permite

preparación a los participa
la propuesta final debe
al ministerio de minas
un consumo de bagazo
recorridos en la planta
mano de obra comprende
los estragos que produciría
ala labor de ventas
inició con una pequeña
porque solo debe describir
interesada en el tema
por estrato y teniendo
se mueve la industria
de gas natural llegó
del entorno la historia
algún empleado que labore
en unciendo para realizar
en el extremo opuesto
dado que el vapor
pero siguiendo los pasos
hechos tomó a todos
el sabor del producto
ejército se den cuenta
obtener una mayor participa
la caldera que utilice
un ejemplo de esto

primera en la industria
a los modelos ocultos
cuanto a la tecnología
de trabajo busca ponerle
se desea obtener panela
inició en las primeras
se planteará la solución
resaltar que de acuerdo
que favorezca el adecuado
medición y los errores
parece ser la política
porque conformamos un dócil
avanzan en el mismo
miembros de la facultad
colegio en previsión tomo
en el que ocurrieron
el jefe de publicidad
la recuperación se material
quiera que sea capaz
a los recursos necesarios
del primero de enero
del mundo e induciendo
se tienen las órdenes
demostrado que quien escucha
la matriz de coeficientes
para una mejor claridad

final de los años
situación actual se aprecia
la facultad de ingeniería
unos genios que entendieron
amasa con una espátula
entre las cuales destaca
las razones no pudiera
consideran una empresa ove

el criterio de logro
o mejoran la movilidad
de uno de estos
de televisión y hablan
se desea la felicidad
que la mujer podía
en espacio un edificio
a la secretaría general

esto se puede matizar
municipios con el propósito
conocer al jefe inmediato
en la buena intención
no han sido ratificados
presa de unas campañas

A.7 Frases, listado 7.

selección de personal estas
con sus nuevos aviones
primer ministro y diputados
de describir el ámbito
con respecto al cambio
es el número tres
pueblo y que sepan
un edificio hace seis
esto se hará hincapié
casa de estratos medios
un descenso del porcentaje
de esquemas de economía
de la pasada versión
que permitan la consulta
de septiembre se convoca
qué ofrecen el costo
se establecen los primeros
dependiendo de las escalas
con cobertura para toda
la empresa a principios
y cargas ha situado
del ministerio de minas
correctos de hace veinte
se encuentra el desarrollo
como los catálogos computar
textura y un color
del sistema con otras
la ayuda a Colombia
radical en el sistema
siguientes son las zonas
se presten a doble
el mapa de conexiones
proyecto con los profesores
Ospina Rodríguez y José

desempeño de sus alumnos
que acabe con toda
ponga fin al sufrimiento
domingo y fueron sembradas
concepto de lucha supone
las ventas han sobrepasado
distrital para la atención
el plan de masificación
permite hallar el libro
para confrontar la realidad
bares fueron su espacio
máximo en el fiscal
es la primera etapa
por un calentamiento lento
es a la escuela
experimentos en su finca
jugos en los diferentes
que desde su punto
sol y la luna
gobierno de julio cesar
con otras personas cercanas
los casos se utilizó
sacarosa y en azúcares
sus peticiones sobre seguir
de comprobarse la lentitud
de las niñas frente
el miedo se convierte
expansión de las redes
hombres por las expulsiones
una vía sin pavimento
ya que la nación
control en las fronteras
la creación de comités
dar alternativas que sean

dejó claro el publicista
control de la temperatura
caracteriza por haber quita
puntos de una manera
veces es necesario ajustar
aprender con la oportunidad
nuestro país el sistema
se usa para realizar
es imposible de separar
dejó en marcha programas
se expresa la opinión
las normas del derecho
fisco y la fuente
de las empresas cazadoras
mejor es el aire
para que se cree
se da que muchos
que se presenten mezclas
claro que la extradición
reto para la seguridad
porque su reputación quedar
oído sin tener nada
desdibujan ante la cruda
operación de la empresa
sido trasladado a centros
espía experta en labores
apoyándose en imágenes toma
tecnología es un mercado
no será lo suficientemente
municipal y el primer
a ser más productivos
se atenderá la industria

A.8 Frases, listado 8.

es normal que algunas
 generador de empleo después
 no requiere de agua
 el mercado no estaban
 de una actitud constructiva
 donde una elevada cifra
 aprecia que está formada
 causó un gran problema
 ventas que se realiza
 de formar al niño
 lo que había sido
 artículos que los sustentan
 a reconocer las bondades
 más comunes que cometen
 libro en una clase
 y del análisis crítico
 que a la final
 público con la razón
 evacuar a los afectados
 control en las fronteras
 con esa misma óptica
 el predio nube blanca
 uno de los cuatro
 los precios son bajos
 que es donde principalmente
 en el extremo opuesto
 eran los que cotizaban
 un teatro se convierten
 está creando al gobierno
 fiscalía general por parte
 que la nación estuvo
 los países de Latinoamérica
 no permite una buena
 en la crisis mundial

su taller había seis
 realizada en los inicios
 lado de una pareja
 no cuenta con presencia
 y seis para televisión
 contra de la ayuda
 opcionados para el cargo
 el próximo jueves vence
 bajos de la población
 descripción de las principia
 el manejo hace referencia
 parte de los docentes
 ya que éstas permiten
 de Juan Carlos Ramírez
 fundamental de los programa
 del bienestar de nuestros
 específicos en cada tema
 testigo de la historia
 de definir el título
 la población de mayores
 identificar en que momento
 del hombre al cargo
 flota de camiones propios
 volantes hay un escalón
 diciembre Víctor Mora vence
 el profesor deberá tener
 a la junta directiva
 su cargo el diseño
 que acabe con toda
 a los altos volúmenes
 esas cosas son personas
 en los jóvenes amor
 de lunes a sábado
 dice que la hornilla

entre ellos logran acuerdo
 los productos es repartido
 incluirse en el texto
 y otra que comenzó
 de un grupo reducido
 de personal se seguirán
 textos que al parecer
 contrato por unos años
 tomo como el cuarto
 de la jornada entregó
 la tasa de oportunidad
 aumentando la clase obrera
 y refinería de fabricante
 que no ha hecho
 regional completó la obra
 corriente de gases caliente
 principales calles del país
 las extensas canchas deporta
 que ayer fuer liberada
 ha sido la producción
 las dos terceras partes
 queso y los demás
 como el país menos
 las expulsiones de Leonardo
 se compare con todos
 las drogas y poniendo
 de la cual jamás
 la puerta sino impreso
 haber realizado un progreso
 miel del último tanque
 mercado natural y donde
 pasiva el maestro transmití

NEXO B. RESULTADOS DETALLADOS, EXPERIMENTOS PRELIMINARES

Todos los experimentos fueron ejecutados en el intérprete de comandos de MATLAB, corriendo sobre computadores Pentium II con Windows 98.

B.1 Experimento 1: /a/ y /e/ wavelets

Cuadro 1. Experimento 1, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	25	12

Cuadro 2. Experimento 1, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	53	150	25	75	Aleatorio	Centrado
/e/	66	150	30	75	Aleatorio	Centrado

Cuadro 3. Experimento 1, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 9. Entrenamiento y pruebas del experimento 1

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1		$5,66 \times 10^{-16}$	48	32	52	44	44	41	48		
2	Gradiente mínimo (1×10^{-10})	$4,91 \times 10^{-15}$	51	32	49	42	48	42	42		
3		$6,48 \times 10^{-16}$	42	25	38	39	35	40	30	28	39
4		$4,60 \times 10^{-15}$	56	38	51	45	34	48	39	47	38
5		$1,40 \times 10^{-16}$	36	22	33	46	45	40	42	46	42

B.2 Experimento 2: /a/ y /e/ cepstro

Cuadro 4. Experimento 2, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	del 2 al 15	25	12

Cuadro 5. Experimento 2, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	53	150	25	75	Aleatorio	Centrado
/e/	66	150	30	75	Aleatorio	Centrado

Cuadro 6. Experimento 2, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 10. Entrenamiento y pruebas del experimento 2

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1	Gradiente mínimo (1×10^{-11})	$8,66 \times 10^{-16}$	21	3	1	0	3	2	3	1	2
2		$4,96 \times 10^{-15}$	18	4	3	3	3	2	1	3	3
3		$6,36 \times 10^{-15}$	18	3	3	3	2	1	2	2	0
4		$6,79 \times 10^{-15}$	19	4	1	3	3	5	5	2	3

B.3 Experimento 3: /a/ y /e/ cepstro

Cuadro 7. Experimento 3, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	del 2 al 15	25	12

Cuadro 8. Experimento 3, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	53	180	25	75	Aleatorio	Centrado
/e/	66	180	30	75	Aleatorio	Centrado

Cuadro 9. Experimento 3, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 11. Entrenamiento y pruebas del experimento 3

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1	Gradiente mínimo (1×10^{-11})	$6,33 \times 10^{-16}$	24	8	6	7	10	13	7	8	6

B.4 Experimento 4: /t/ y /k/ wavelets**Cuadro 10. Experimento 4, descripción del experimento**

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/t/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	25	12

Cuadro 11. Experimento 4, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/t/	19	50	6	25	Aleatorio	Centrado
/k/	22	50	8	25	Aleatorio	Centrado

Cuadro 12. Experimento 4, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 12. Entrenamiento y pruebas del experimento 4

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas							
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7	8
1		$1,23 \times 10^{-16}$	27	10	4	5	3	9	2	4	2	4
2	Gradiente mínimo (1×10^{-12})	$5,77 \times 10^{-16}$	26	10	3	2	6	2	1	1	0	1
3		$6,99 \times 10^{-17}$	63	45	0	0	0	0	0	0	0	0
4		$2,60 \times 10^{-16}$	93	78	5	4	6	4	4	8	11	2
5		$2,34 \times 10^{-16}$	22	6	8	8	9	12	8	10	5	11

B.5 Experimento 5: /t/ y /k/ cepstro**Cuadro 13. Experimento 5, descripción del experimento**

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/t/ y /k/	512 muestras	del 2 al 15	25	12

Cuadro 14. Experimento 5, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		

/t/	19	50	6	25	Aleatorio	Centrado
/k/	22	50	8	25	Aleatorio	Centrado

Cuadro 15. Experimento 5, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 13. Entrenamiento y pruebas del experimento 5

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1		$1,40 \times 10^{-15}$	20	5	10	4	7	6	7	9	9
2	Gradiente	$2,71 \times 10^{-16}$	19	3	6	5	5	6	7	6	4
3	mínimo	$1,77 \times 10^{-16}$	21	4	5	4	6	4	8	5	4
4	(1×10^{-12})	$4,13 \times 10^{-16}$	25	9	6	8	6	10	7	6	5
5		$1,13 \times 10^{-15}$	22	6	7	9	9	6	4	2	9

B.6 Experimento 6: /p/ y /k/ wavelets

Cuadro 16. Experimento 6, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/p/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	25	12

Cuadro 17. Experimento 6, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	15	50	6	25	Aleatorio	Centrado
/k/	22	50	8	25	Aleatorio	Centrado

Cuadro 18. Experimento 6, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 14. Entrenamiento y pruebas del experimento 6

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1		$9,77 \times 10^{-17}$	23	6	0	0	0	0	0	0	0
2	Gradiente	$1,50 \times 10^{-16}$	21	4	5	4	2	4	1	2	4
3	mínimo	$2,56 \times 10^{-16}$	19	4	0	0	0	0	0	0	0
4	(1×10^{-12})	$9,30 \times 10^{-17}$	25	9	2	2	1	2	4	2	4
5		$1,32 \times 10^{-15}$	22	7	1	3	3	2	2	2	3

B.7 Experimento 7: /p/ y /k/ cepstro

Cuadro 19. Experimento 7, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro		Número de frases	
		Coefficientes		entrenamiento	pruebas
/p/ y /k/	512 muestras	del 2 al 15		25	12

Cuadro 20. Experimento 7, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	15	50	6	25	Aleatorio	Centrado
/k/	22	50	8	25	Aleatorio	Centrado

Cuadro 21. Experimento 7, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 15. Entrenamiento y pruebas del experimento 7

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1		$2,11 \times 10^{-16}$	21	5	0	0	0	0	0	0	0
2	Gradiente mínimo (1×10^{-12})	$1,33 \times 10^{-16}$	25	9	0	0	0	0	0	0	0
3		$1,98 \times 10^{-15}$	26	10	6	7	9	5	5	11	8
4		$2,16 \times 10^{-16}$	18	2	1	2	2	2	2	2	1
5		$6,17 \times 10^{-16}$	21	7	2	1	2	2	2	1	1

B.8 Experimento 8: /p/ y /t/ wavelets

Cuadro 22. Experimento 8, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/p/ y /t/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	25	12

Cuadro 23. Experimento 8, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	15	50	6	25	Aleatorio	Centrado
/t/	19	50	6	25	Aleatorio	Centrado

Cuadro 24. Experimento 8, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

<i>feedforward</i>							
--------------------	--	--	--	--	--	--	--

Tabla 16. Entrenamiento y pruebas del experimento 8

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1		$2,14 \times 10^{-16}$	21	7	0	0	0	0	0	0	0
2	Gradiente	$4,61 \times 10^{-17}$	25	10	2	3	4	5	4	2	3
3	mínimo	$1,06 \times 10^{-15}$	20	5	0	0	0	0	0	0	0
4	(1×10^{-12})	$3,00 \times 10^{-16}$	18	8	2	3	3	1	4	2	5
5		$2,80 \times 10^{-16}$	20	4	0	0	0	0	0	0	0

B.9 Experimento 9: /p/ y /t/ cepstro

Cuadro 25. Experimento 9, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/p/ y /t/	512 muestras	del 2 al 15	25	12

Cuadro 26. Experimento 9, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	15	50	6	25	Aleatorio	Centrado
/t/	19	50	6	25	Aleatorio	Centrado

Cuadro 27. Experimento 9, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 17. Entrenamiento y pruebas del experimento 9

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1		$1,19 \times 10^{-16}$	22	5	3	1	3	1	2	2	2
2	Gradiente	$2,50 \times 10^{-10}$	22								
3	mínimo	$2,97 \times 10^{-16}$	18	3	0	0	0	0	0	0	0
4	(1×10^{-12})	$1,09 \times 10^{-16}$	30	14	4	6	3	5	3	3	5
5		$4,26 \times 10^{-16}$	29	11	3	2	4	3	4	0	5

B.10 Experimento 10: /p/, /t/ y /k/ wavelets

Cuadro 28. Experimento 10, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	25	12

Cuadro 29. Experimento 10, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	15	50	6	25	Aleatorio	Centrado
/t/	19	50	6	25	Aleatorio	Centrado
/k/	22	50	8	25	Aleatorio	Centrado

Cuadro 30. Experimento 10, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 18. Entrenamiento y pruebas del experimento 10

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1	Motivo de culminación	$1,08 \times 10^{-16}$	24	9	5	7	3	0	2	3	2
2					1	3	4	3	0	2	3
2	Gradiente mínimo (1×10^{-12})	$7,04 \times 10^{-17}$	68	NE	1	3	2	4	6	6	4
3					1	6	4	8	13	13	7
3	Gradiente mínimo (1×10^{-12})	$1,51 \times 10^{-17}$	24	5	4	5	7	1	1	5	4
4					7	8	10	2	3	7	7
4	Gradiente mínimo (1×10^{-12})	$1,40 \times 10^{-16}$	40	23	3	3	3	1	2	2	3
5					8	5	4	4	2	9	4
5	Gradiente mínimo (1×10^{-12})	$4,88 \times 10^{-17}$	30	16	9	6	9	10	5	9	7
6					11	9	9	8	5	8	8
6	Gradiente mínimo (1×10^{-12})	$4,88 \times 10^{-17}$	30	16	1	7	1	3	5	5	2
7					6	9	9	14	14	14	6
7	Gradiente mínimo (1×10^{-12})	$4,88 \times 10^{-17}$	30	16	5	8	8	11	9	9	4
8					5	8	8	11	9	9	4

B.11 Experimento 11: /p/, /t/ y /k/ cepstro**Cuadro 31. Experimento 11, descripción del experimento**

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	del 2 al 15	25	12

Cuadro 32. Experimento 11, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	15	50	6	25	Aleatorio	Centrado
/t/	19	50	6	25	Aleatorio	Centrado
/k/	22	50	8	25	Aleatorio	Centrado

Cuadro 33. Experimento 11, características de la red

Tipo de red	Número de neuronas	Función de adaptación	Algoritmo de
-------------	--------------------	-----------------------	--------------

	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	entrenamiento
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 19. Entrenamiento y pruebas del experimento 11

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas						
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6	7
1	Gradiente mínimo (1×10^{-12})	$1,97 \times 10^{-16}$	141	128	18	23	20	19	25	18	18
					3	11	8	10	3	7	6
					12	14	15	20	18	17	12
2	Gradiente mínimo (1×10^{-12})	$2,60 \times 10^{-16}$	23	8	5	4	4	2	4	4	3
					9	10	7	6	8	8	12
					12	10	8	6	4	8	7
3	Gradiente mínimo (1×10^{-12})	$1,60 \times 10^{-16}$	31	15	3	1	6	2	2	1	4
					9	8	10	7	11	5	15
					5	9	4	6	6	7	9

B.12 Experimento 12: /t/ y /k/ wavelets

Cuadro 34. Experimento 12, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/t/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	25	12

Cuadro 35. Experimento 12, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/t/	19	80	6	40	Aleatorio	Centrado
/k/	22	80	8	40	Aleatorio	Centrado

Cuadro 36. Experimento 12, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	2	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 20. Entrenamiento y pruebas del experimento 12

Instancia	Parámetros de entrenamiento				Desaciertos en pruebas					
	Motivo de culminación	MSE final	Épocas alcanzadas	Época de quiebre	1	2	3	4	5	6
1	Gradiente mínimo (1×10^{-12})	$7,65 \times 10^{-17}$	57	45	7	11	11	6	9	
2					9	6	6	12	7	
3					12	8	11	16	13	
4					3	3	8	8	6	5
4		$1,92 \times 10^{-17}$	25	8	8	9	9	12	9	12

NEXO C. RESULTADOS DETALLADOS, EXPERIMENTOS FINALES

C.1 Experimento final 1: /p/, /t/ y /k/ wavelets

Cuadro 37. Experimento 1, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 38. Experimento 1, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	aleatorio	aleatorio
/t/	44	150	23	75	aleatorio	aleatorio
/k/	49	150	25	75	aleatorio	aleatorio

Cuadro 39. Experimento 1, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 21. Entrenamiento del experimento 1

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	41	$7,41 \times 10^{-04}$	1071,7 s	1	41	$7,41 \times 10^{-04}$	645,8 s s
2	88	$2,59 \times 10^{-01}$	2184,3 s	2	88	$2,59 \times 10^{-01}$	1343,5 s
3	44	$9,61 \times 10^{-19}$	1134,3 s	3	44	$9,61 \times 10^{-19}$	700,7 s s
4	69	$3,13 \times 10^{-19}$	1811,8 s	4	69	$3,13 \times 10^{-19}$	1083,8 s
5	46	$5,24 \times 10^{-17}$	1205,0 s	5	46	$5,24 \times 10^{-17}$	746,9 s s

C.1.1 Experimento final 1, instancia 1

Tabla 22. Pruebas del experimento 1, instancia 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	65	0	7	9	74	68	1	1	0
	t	0	72	3	75	3	72	0	0	0
	k	2	4	68	73	70	6	0	1	1
2	p	66	1	6	7	74	68	2	0	1
	t	0	72	3	75	2	72	0	1	0
	k	2	6	67	73	69	7	0	0	1
3	p	60	0	13	14	74	61	1	1	1
	t	0	73	1	75	2	73	0	0	1
	k	1	10	65	73	65	10	1	0	0
4	p	64	1	8	10	74	67	1	0	0
	t	1	70	3	74	4	72	0	1	0
	k	3	8	64	71	66	10	1	1	1
5	p	63	2	11	12	73	64	0	0	0
	t	1	69	4	74	6	71	0	0	0
	k	2	5	69	73	70	6	0	0	0

C.1.2 Experimento final 1, instancia 2

Tabla 23. Pruebas del experimento 1, instancia 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	57	0	0	18	75	75	0	0	0
	t	13	0	0	62	75	75	0	0	0
	k	1	0	0	74	75	75	0	0	0
2	p	58	0	0	17	75	75	0	0	0
	t	16	0	0	59	75	75	0	0	0
	k	1	0	0	74	75	75	0	0	0
3	p	47	0	0	28	75	75	0	0	0
	t	12	0	0	63	75	75	0	0	0
	k	0	0	0	75	75	75	0	0	0
4	p	51	0	0	24	75	75	0	0	0
	t	13	0	0	62	75	75	0	0	0
	k	3	0	0	72	75	75	0	0	0
5	p	53	0	0	22	75	75	0	0	0
	t	16	0	0	59	75	75	0	0	0
	k	0	0	0	75	75	75	0	0	0

C.1.3 Experimento final 1, instancia 3

Tabla 24. Pruebas del experimento 1, instancia 3

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	74	0	1	1	75	74	0	0	0
	t	3	70	3	72	5	72	0	0	0
	k	1	6	67	74	68	7	0	1	1
2	p	74	0	0	1	75	75	0	0	0
	t	1	72	2	74	3	73	0	0	0
	k	1	7	66	73	68	8	1	0	1
3	p	71	0	4	4	75	71	0	0	0
	t	1	67	5	73	6	70	1	2	0
	k	1	8	66	74	67	9	0	0	0
4	p	74	0	1	1	75	74	0	0	0
	t	3	69	3	72	6	72	0	0	0
	k	4	8	63	71	67	12	0	0	0
5	p	73	0	0	2	75	75	0	0	0
	t	3	70	0	70	3	75	2	2	0
	k	0	5	69	75	70	5	0	0	1

C.1.4 Experimento final 1, instancia 4

Tabla 25. Pruebas del experimento 1, instancia 4

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	67	5	5	8	70	70	0	0	0
	t	0	72	3	75	3	72	0	0	0
	k	1	4	69	74	70	6	0	1	0
2	p	70	5	4	5	70	71	0	0	0
	t	0	74	1	75	1	74	0	0	0
	k	2	7	64	73	68	11	0	0	0
3	p	66	6	11	9	69	64	0	0	0
	t	0	73	2	75	2	73	0	0	0
	k	2	8	66	73	67	9	0	0	0
4	p	68	5	6	7	70	69	0	0	0
	t	2	71	2	73	4	73	0	0	0
	k	5	10	60	70	64	14	0	1	1
5	p	67	3	5	8	72	70	0	0	0
	t	0	72	3	75	3	72	0	0	0
	k	2	8	66	73	67	9	0	0	0

C.1.5 Experimento final 1, instancia 5

Tabla 26. Pruebas del experimento 1, instancia 5

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	\hat{p} ?	\hat{t} ?	\hat{k} ?
1	p	73	0	5	2	75	70	0	0	0
	t	0	69	4	75	6	71	0	0	0
	k	0	3	72	75	72	3	0	0	0
2	p	73	0	4	2	75	71	0	0	0
	t	0	70	3	75	3	72	0	2	0
	k	5	5	64	69	70	11	1	0	0
3	p	69	0	10	5	75	65	1	0	0
	t	0	69	3	75	5	72	0	1	0
	k	1	6	68	74	69	7	0	0	0
4	p	72	0	7	3	75	68	0	0	0
	t	2	68	5	73	7	70	0	0	0
	k	4	8	63	71	67	9	0	0	3
5	p	69	1	6	6	74	69	0	0	0
	t	0	73	0	75	2	74	0	0	1
	k	2	4	67	73	71	8	0	0	0

C.2 Experimento final 2: /p/, /t/ y /k/ wavelets

Cuadro 40. Experimento 2, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 41. Experimento 2, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	aleatorio	uniforme
/t/	44	150	23	75	aleatorio	uniforme
/k/	49	150	25	75	aleatorio	uniforme

Cuadro 42. Experimento 2, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 27. Entrenamiento del experimento 2

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	56	$1,48 \times 10^{-03}$	1453,1 s	1	56	$1,48 \times 10^{-03}$	891,6 s
2	36	$8,37 \times 10^{-19}$	926,6 s	2	36	$8,37 \times 10^{-19}$	592,1 s
3	58	$3,39 \times 10^{-19}$	1549,8 s	3	58	$3,39 \times 10^{-19}$	932,0 s
4	33	$3,21 \times 10^{-18}$	864,0 s	4	33	$3,21 \times 10^{-18}$	540,9 s
5	100	$3,18 \times 10^{-02}$	2619,1 s	5	100	$3,18 \times 10^{-02}$	1646,5 s

C.2.1 Experimento final 2, instancia 1**Tabla 28. Pruebas del experimento 2, instancia 1**

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	65	6	6	10	68	68	0	1	1
	t	3	68	2	72	7	71	0	0	2
	k	5	9	62	70	64	13	0	2	0
2	p	64	4	6	11	71	69	0	0	0
	t	2	69	4	73	6	71	0	0	0
	k	2	12	64	73	63	11	0	0	0
3	p	62	6	11	13	68	63	0	1	1
	t	3	70	1	72	3	73	0	2	1
	k	3	10	63	72	65	12	0	0	0
4	p	63	3	11	11	72	64	1	0	0
	t	1	67	8	74	8	67	0	0	0
	k	1	10	62	74	65	13	0	0	0
5	p	66	2	9	9	73	66	0	0	0
	t	0	66	7	75	9	66	0	0	2
	k	3	10	67	72	65	7	0	0	1

C.2.2 Experimento final 2, instancia 2**Tabla 29. Pruebas del experimento 2, instancia 2**

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	69	0	7	6	75	68	0	0	0
	t	2	69	3	73	6	71	0	0	1
	k	0	9	63	75	63	11	0	3	1
2	p	71	1	5	3	74	70	1	0	0
	t	2	69	3	73	6	71	0	0	1
	k	0	10	63	75	64	11	0	1	1
3	p	70	0	7	5	75	68	0	0	0
	t	2	71	2	73	4	73	0	0	0
	k	1	9	67	74	65	8	0	1	0

4	p	70	1	5	5	74	70	0	0	0
	t	1	67	5	72	8	70	2	0	0
	k	0	11	61	75	64	13	0	0	1
5	p	67	0	8	7	75	67	1	0	0
	t	2	70	1	73	3	73	0	2	1
	k	1	8	66	73	67	9	1	0	0

C.2.3 Experimento final 2, instancia 3

Tabla 30. Pruebas del experimento 2, instancia 3

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	72	0	4	3	75	71	0	0	0
	t	2	69	3	73	5	71	0	1	1
	k	2	8	65	73	67	10	0	0	0
2	p	73	0	1	2	75	74	0	0	0
	t	1	62	10	74	12	63	0	1	2
	k	0	11	64	74	64	11	1	0	0
3	p	71	0	4	4	75	71	0	0	0
	t	0	61	13	75	13	60	0	1	2
	k	0	11	64	75	64	11	0	0	0
4	p	72	0	3	3	75	72	0	0	0
	t	1	64	7	74	10	66	0	1	2
	k	1	8	66	74	67	9	0	0	0
5	p	71	0	3	4	75	71	0	0	1
	t	0	68	6	75	6	68	0	1	1
	k	2	3	71	73	72	4	0	0	0

C.2.4 Experimento final 2, instancia 4

Tabla 31. Pruebas del experimento 2, instancia 4

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	67	0	7	8	75	68	0	0	0
	t	5	62	8	69	12	67	1	1	0
	k	1	6	65	73	69	8	1	0	2
2	p	67	0	8	8	75	66	0	0	1
	t	4	60	10	71	15	65	0	0	0
	k	0	9	64	74	65	10	1	1	1
3	p	64	0	10	10	75	65	1	0	0
	t	4	56	13	71	19	61	0	0	1
	k	0	7	68	73	68	7	2	0	0
4	p	70	0	5	5	75	70	0	0	0
	t	3	57	12	72	18	62	0	0	1
	k	0	8	66	75	67	9	0	0	0

	p	64	0	8	11	75	67	0	0	0
5	t	5	59	7	70	16	66	0	0	2
	k	0	7	66	75	68	9	0	0	0

C.2.5 Experimento final 2, instancia 5

Tabla 32. Pruebas del experimento 2, instancia 5

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	69	0	2	6	75	73	0	0	0
	t	3	66	11	71	8	64	1	1	0
	k	4	9	43	71	65	31	0	1	1
2	p	64	0	6	9	75	69	2	0	0
	t	4	66	12	70	8	63	1	1	0
	k	2	12	51	73	63	23	0	0	1
3	p	64	0	5	10	75	70	1	0	0
	t	2	69	13	72	4	62	1	2	0
	k	0	8	51	75	67	24	0	0	0
4	p	67	0	3	7	75	72	1	0	0
	t	1	69	13	74	6	62	0	0	0
	k	2	6	55	73	67	18	0	2	2
5	p	65	0	5	10	75	70	0	0	0
	t	1	68	16	74	7	59	0	0	0
	k	3	5	49	72	70	26	0	0	0

C.3 Experimento final 3: /p/, /t/ y /k/ wavelets

Cuadro 43. Experimento 3, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 44. Experimento 3, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	binario	aleatorio
/t/	44	150	23	75	binario	aleatorio
/k/	49	150	25	75	binario	aleatorio

Cuadro 45. Experimento 3, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 33. Entrenamiento del experimento 3

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	29	$7,06 \times 10^{-18}$	754,0 s s	1	29	$7,06 \times 10^{-18}$	475,5 s s
2	27	$7,35 \times 10^{-18}$	706,3 s s	2	27	$7,35 \times 10^{-18}$	432,6 s s
3	32	$1,70 \times 10^{-18}$	846,3 s s	3	32	$1,70 \times 10^{-18}$	508,8 s s
4	65	$5,78 \times 10^{-18}$	1710,1 s s	4	65	$5,78 \times 10^{-18}$	1053,3 s s
5	100	$2,33 \times 10^{-02}$	2567,9 s s	5	100	$2,33 \times 10^{-02}$	1550,6 s s

C.3.1 Experimento final 3, instancia 1**Tabla 34. Pruebas del experimento 3, instancia 1**

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp	ζt	ζk
1	p	71	0	3	4	75	72	0	0	0
	t	0	69	3	73	5	72	2	1	0
	k	0	3	70	75	72	4	0	0	1
2	p	68	0	6	7	75	69	0	0	0
	t	2	62	8	72	12	67	1	1	0
	k	0	4	69	75	71	5	0	0	1
3	p	64	0	11	11	75	64	0	0	0
	t	0	69	5	75	5	70	0	1	0
	k	2	5	69	73	70	6	0	0	0
4	p	62	0	15	13	75	60	0	0	0
	t	1	65	7	74	8	67	0	2	1
	k	0	4	71	75	71	4	0	0	0
5	p	62	0	12	13	75	62	0	0	1
	t	0	69	3	75	3	68	0	3	4
	k	0	5	70	75	70	5	0	0	0

C.3.2 Experimento final 3, instancia 2**Tabla 35. Pruebas del experimento 3, instancia 1**

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp	ζt	ζk
1	p	70	0	6	5	75	68	0	0	1
	t	0	73	1	75	2	73	0	0	1
	k	1	7	66	74	67	9	0	1	0
2	p	64	0	12	11	75	63	0	0	0
	t	0	73	2	75	2	73	0	0	0
	k	1	6	68	74	69	7	0	0	0
3	p	62	0	13	13	75	61	0	0	1
	t	0	75	0	75	0	75	0	0	0
	k	1	10	64	74	64	11	0	1	0

4	p	65	0	10	10	75	64	0	0	1
	t	0	74	1	75	1	74	0	0	0
	k	0	7	68	75	68	7	0	0	0
5	p	62	0	11	12	75	61	1	0	3
	t	0	73	2	75	2	73	0	0	0
	k	0	8	67	75	67	8	0	0	0

C.3.3 Experimento final 3, instancia 3

Tabla 36. Pruebas del experimento 3, instancia 3

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	73	0	3	2	75	70	0	0	2
	t	0	72	3	75	3	72	0	0	0
	k	1	4	70	74	71	5	0	0	0
2	p	69	0	6	6	75	67	0	0	2
	t	0	72	2	75	3	72	0	0	1
	k	1	7	67	74	68	8	0	0	0
3	p	66	0	10	9	75	64	0	0	1
	t	0	73	2	75	2	73	0	0	0
	k	1	4	70	74	70	5	0	1	0
4	p	68	0	10	7	75	65	0	0	0
	t	0	72	2	75	3	72	0	0	1
	k	0	4	71	75	71	4	0	0	0
5	p	67	0	11	8	75	63	0	0	1
	t	0	73	2	75	2	73	0	0	0
	k	0	4	71	75	70	4	0	1	0

C.3.4 Experimento final 3, instancia 4

Tabla 37. Pruebas del experimento 3, instancia 4

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	68	0	7	6	75	67	1	0	1
	t	0	71	4	75	4	70	0	0	1
	k	1	4	69	74	71	6	0	0	0
2	p	65	2	10	9	73	65	1	0	0
	t	0	68	6	75	7	68	0	0	1
	k	1	6	65	73	69	9	1	0	1
3	p	62	1	13	13	74	62	0	0	0
	t	0	71	4	75	3	71	0	1	0
	k	5	5	67	70	70	8	0	0	0
4	p	64	2	12	9	73	62	2	0	1
	t	0	75	0	75	0	75	0	0	0
	k	1	4	68	74	71	6	0	0	1

	p	64	0	10	10	75	65	1	0	0
5	t	0	72	2	75	3	73	0	0	0
	k	1	9	66	74	66	9	0	0	0

C.3.5 Experimento final 3, instancia 5

Tabla 38. Pruebas del experimento 3, instancia 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	63	4	6	12	71	69	0	0	0
	t	3	64	10	72	9	64	0	2	1
	k	0	15	58	73	59	12	2	1	5
2	p	59	7	9	16	68	66	0	0	0
	t	4	61	12	70	10	62	1	4	1
	k	2	10	65	71	65	8	2	0	2
3	p	59	3	11	15	72	63	1	0	1
	t	4	64	2	71	10	72	0	1	1
	k	2	14	59	72	61	13	1	0	3
4	p	54	3	17	20	72	58	1	0	0
	t	5	60	8	68	13	67	2	2	0
	k	0	16	63	74	59	10	1	0	2
5	p	56	3	12	19	72	59	0	0	4
	t	3	63	6	72	10	68	0	2	1
	k	0	14	61	74	61	11	1	0	3

C.4 Experimento final 4: /p/, /t/ y /k/ wavelets

Cuadro 46. Experimento 4, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 47. Experimento 4, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	binario	uniforme
/t/	44	150	23	75	binario	uniforme
/k/	49	150	25	75	binario	uniforme

Cuadro 48. Experimento 4, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 39. Entrenamiento del experimento 4

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	34	$3,01 \times 10^{-17}$	884,7 s	1	34	$3,01 \times 10^{-17}$	548,7 s
2	45	$7,45 \times 10^{-19}$	1193,4 s	2	45	$7,45 \times 10^{-19}$	708,5 s
3	54	$6,44 \times 10^{-19}$	1404,1 s	3	54	$6,44 \times 10^{-19}$	858,5 s
4	39	$4,73 \times 10^{-18}$	1002,7 s	4	39	$4,73 \times 10^{-18}$	627,9 s
5	100	$1,48 \times 10^{-03}$	2551,8 s	5	100	$1,48 \times 10^{-03}$	1514,0 s

C.4.1 Experimento final 4, instancia 1**Tabla 40. Pruebas del experimento 4, instancia 1**

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	69	0	5	5	75	70	1	0	0
	t	0	75	0	75	0	75	0	0	0
	k	11	5	61	64	70	11	0	0	3
2	p	65	0	7	10	75	68	0	0	0
	t	0	70	4	74	5	71	1	0	0
	k	5	3	66	70	72	8	0	0	1
3	p	66	0	8	8	75	67	1	0	0
	t	1	73	1	74	1	74	0	1	0
	k	4	3	69	71	72	5	0	0	1
4	p	69	0	6	6	75	69	0	0	0
	t	2	73	0	73	2	75	0	0	0
	k	7	7	65	68	68	10	0	0	0
5	p	66	0	8	9	74	67	0	1	0
	t	1	73	2	74	2	73	0	0	0
	k	4	2	70	71	71	5	0	2	0

C.4.2 Experimento final 4, instancia 2**Tabla 41. Pruebas del experimento 4, instancia 2**

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	64	0	13	11	75	61	0	0	1
	t	0	73	2	75	2	72	0	0	1
	k	4	2	69	71	73	6	0	0	0
2	p	62	0	13	13	75	62	0	0	0
	t	0	69	6	75	6	69	0	0	0
	k	3	2	70	72	73	5	0	0	0
3	p	65	0	10	10	75	65	0	0	0
	t	0	74	1	75	1	74	0	0	0

	k	2	3	69	73	72	6	0	0	0
4	p	65	0	9	10	75	65	0	0	1
	t	0	72	3	74	2	72	1	1	0
	k	1	4	68	74	70	6	0	1	1
5	p	59	0	16	16	75	59	0	0	0
	t	0	71	4	75	4	71	0	0	0
	k	0	3	73	75	72	2	0	0	0

C.4.3 Experimento final 4, instancia 3

Tabla 42. Pruebas del experimento 4, instancia 3

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	70	0	5	5	75	70	0	0	0
	t	0	69	5	75	4	69	0	2	1
	k	1	2	71	74	73	4	0	0	0
2	p	66	1	7	9	74	68	0	0	0
	t	0	68	4	75	6	70	0	1	1
	k	1	2	71	73	73	4	1	0	0
3	p	68	0	6	6	75	68	1	0	1
	t	0	73	1	75	1	73	0	1	1
	k	1	4	71	74	71	4	0	0	0
4	p	70	0	3	4	75	72	1	0	0
	t	0	73	2	75	2	73	0	0	0
	k	1	5	66	74	70	9	0	0	0
5	p	64	1	8	10	73	67	1	1	0
	t	0	73	1	75	2	74	0	0	0
	k	0	5	68	75	70	5	0	0	2

C.4.4 Experimento final 4, instancia 4

Tabla 43. Pruebas del experimento 4, instancia 4

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	70	0	5	5	75	70	0	0	0
	t	0	74	0	75	1	74	0	0	1
	k	3	6	64	72	67	11	0	2	0
2	p	70	0	5	5	75	70	0	0	0
	t	0	69	4	75	5	71	0	1	0
	k	2	4	68	73	71	6	0	0	1
3	p	72	1	2	3	74	73	0	0	0
	t	0	72	1	75	1	74	0	2	0
	k	0	7	66	75	67	8	0	1	1
4	p	73	0	2	2	75	73	0	0	0
	t	0	75	0	75	0	75	0	0	0

	k	1	9	61	74	66	10	0	0	4
5	p	72	0	3	3	75	72	0	0	0
	t	0	73	1	75	2	74	0	0	0
	k	1	5	69	74	70	6	0	0	0

C.4.5 Experimento final 4, instancia 5

Tabla 44. Pruebas del experimento 4, instancia 5

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	68	0	5	7	75	70	0	0	0
	t	0	75	0	75	0	75	0	0	0
	k	1	4	71	74	71	4	0	0	0
2	p	64	1	8	11	73	67	0	1	0
	t	0	73	2	75	2	73	0	0	0
	k	1	3	71	74	72	4	0	0	0
3	p	62	1	9	13	74	66	0	0	0
	t	0	75	0	75	0	75	0	0	0
	k	0	3	72	75	72	3	0	0	0
4	p	65	2	8	10	73	67	0	0	0
	t	0	75	0	75	0	75	0	0	0
	k	0	3	71	75	71	4	0	1	0
5	p	60	1	11	15	72	64	0	2	0
	t	0	74	1	75	1	74	0	0	0
	k	0	4	71	75	71	4	0	0	0

C.5 Experimento final 5: /p/, /t/ y /k/ cepstro

Cuadro 49. Experimento 5, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	del 2 al 15	58	35

Cuadro 50. Experimento 5, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	aleatorio	aleatorio
/t/	44	150	23	75	aleatorio	aleatorio
/k/	49	150	25	75	aleatorio	aleatorio

Cuadro 51. Experimento 5, características de la red

Tipo de red	Número de neuronas	Función de adaptación	Algoritmo de
-------------	--------------------	-----------------------	--------------

	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	entrenamiento
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 45. Entrenamiento del experimento 5

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	100	$4,44 \times 10^{-03}$	4180,1 s	1	30	$2,92 \times 10^{-18}$	741,1 s
2	4	$4,13 \times 10^{-01}$	188,4 s	2	58	$3,70 \times 10^{-03}$	1454,0 s
3	25	$4,97 \times 10^{-17}$	1012,8 s	3	28	$3,65 \times 10^{-17}$	691,6 s
4	30	$4,99 \times 10^{-18}$	1234,8 s	4	40	$7,30 \times 10^{-18}$	958,8 s
5	26	$1,56 \times 10^{-17}$	1036,5 s	5	100	$1,21 \times 10^{-01}$	2562,2 s

C.5.1 Experimento final 5, computador 1, instancia 1

Tabla 46. Pruebas del experimento 5, instancia 1, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp	ζt	ζk
1	p	71	7	2	4	68	73	0	0	0
	t	11	67	2	64	8	73	0	0	0
	k	0	6	70	75	69	5	0	0	0
2	p	72	5	2	3	70	73	0	0	0
	t	7	68	4	68	7	71	0	0	0
	k	0	8	67	75	67	8	0	0	0
3	p	68	10	5	7	65	69	0	0	1
	t	5	67	7	69	8	68	1	0	0
	k	0	12	65	75	63	10	0	0	0
4	p	74	8	3	1	67	72	0	0	0
	t	16	60	2	59	15	73	0	0	0
	k	0	7	69	75	68	6	0	0	0
5	p	69	9	5	6	66	70	0	0	0
	t	9	69	9	66	6	66	0	0	0
	k	0	5	70	75	70	5	0	0	0

C.5.2 Experimento final 5, computador 1, instancia 2

Tabla 47. Pruebas del experimento 5, instancia 2, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp	ζt	ζk
1	p	0	0	75	75	75	0	0	0	0
	t	3	47	72	72	28	3	0	0	0
	k	0	16	75	75	59	0	0	0	0

2	p	0	0	75	75	75	0	0	0	0
	t	1	47	75	74	28	0	0	0	0
	k	0	16	75	75	59	0	0	0	0
3	p	0	0	75	75	75	0	0	0	0
	t	0	47	75	75	28	0	0	0	0
	k	0	16	75	75	59	0	0	0	0
4	p	0	0	75	75	75	0	0	0	0
	t	1	34	74	74	41	1	0	0	0
	k	0	17	75	75	58	0	0	0	0
5	p	0	0	75	75	75	0	0	0	0
	t	3	42	72	72	33	3	0	0	0
	k	0	11	75	75	64	0	0	0	0

C.5.3 Experimento final 5, computador 1, instancia 3

Tabla 48. Pruebas del experimento 5, instancia 3, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp	ξt	ξk
								?	?	?
1	p	66	5	0	8	70	75	1	0	0
	t	3	66	3	71	8	71	1	1	1
	k	0	5	69	75	70	6	0	0	0
2	p	68	3	1	6	71	74	1	1	0
	t	5	64	2	69	11	71	1	0	2
	k	4	9	61	71	66	13	0	0	1
3	p	69	1	3	5	74	72	1	0	0
	t	2	68	2	73	5	72	0	2	1
	k	0	12	61	75	63	13	0	0	1
4	p	65	2	1	6	72	74	4	1	0
	t	6	63	3	68	8	72	1	4	0
	k	2	8	65	73	67	8	0	0	2
5	p	69	3	0	5	71	75	1	1	0
	t	6	61	5	68	11	70	1	3	0
	k	2	5	67	73	70	8	0	0	0

C.5.4 Experimento final 5, computador 1, instancia 4

Tabla 49. Pruebas del experimento 5, instancia 4, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp	ξt	ξk
								?	?	?
1	p	64	2	5	10	73	70	1	0	0
	t	2	68	2	73	6	73	0	1	0
	k	7	4	65	67	71	10	1	0	0
2	p	65	0	5	10	74	70	0	1	0
	t	1	68	2	74	5	73	0	2	0
	k	10	4	61	62	71	14	3	0	0

3	p	66	1	4	8	73	71	1	1	0
	t	1	68	3	74	5	72	0	2	0
	k	10	4	64	64	71	11	1	0	0
4	p	70	2	3	5	72	71	0	1	1
	t	4	60	5	71	15	69	0	0	1
	k	10	4	66	65	71	8	0	0	1
5	p	67	1	2	8	74	73	0	0	0
	t	1	68	1	74	7	73	0	0	1
	k	10	1	67	65	73	8	0	1	0

C.5.5 Experimento final 5, computador 1, instancia 5

Tabla 50. Pruebas del experimento 5, instancia 5, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	58	8	9	17	67	66	0	0	0
	t	5	66	3	70	9	72	0	0	0
	k	7	4	63	68	71	12	0	0	0
2	p	62	8	4	12	67	71	1	0	0
	t	8	66	2	67	8	73	0	1	0
	k	12	6	59	63	69	15	0	0	1
3	p	64	6	4	11	69	70	0	0	1
	t	7	68	0	68	7	75	0	0	0
	k	6	11	62	69	63	13	0	1	0
4	p	67	7	1	8	68	74	0	0	0
	t	9	63	2	65	11	73	1	1	0
	k	5	8	62	70	67	12	0	0	1
5	p	63	7	4	12	68	70	0	0	1
	t	12	59	3	62	16	72	1	0	0
	k	5	2	66	70	73	9	0	0	0

C.5.6 Experimento final 5, computador 2, instancia 1

Tabla 51. Pruebas del experimento 5, instancia 1, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	61	2	19	14	73	56	0	0	0
	t	3	62	12	72	13	62	0	0	1
	k	1	6	68	74	69	7	0	0	0
2	p	64	1	12	10	73	62	1	1	1
	t	4	65	8	71	10	64	0	0	3
	k	0	4	70	74	71	5	1	0	0
3	p	61	2	19	13	73	56	1	0	0
	t	1	71	11	74	4	64	0	0	0
	k	1	9	66	74	66	8	0	0	1

4	p	60	1	17	15	73	57	0	1	1
	t	6	64	9	69	11	66	0	0	0
	k	0	6	70	75	69	4	0	0	1
5	p	64	1	15	11	74	60	0	0	0
	t	4	68	15	71	6	60	0	1	0
	k	1	3	71	74	72	4	0	0	0

C.5.7 Experimento final 5, computador 2, instancia 2

Tabla 52. Pruebas del experimento 5, instancia 2, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	55	2	9	20	73	65	0	0	1
	t	7	64	4	68	11	71	0	0	0
	k	0	5	66	75	69	8	0	1	1
2	p	51	0	15	24	75	59	0	0	1
	t	4	67	1	71	7	74	0	1	0
	k	4	4	68	70	71	7	1	0	0
3	p	56	0	8	18	75	67	1	0	0
	t	1	66	2	73	9	73	1	0	0
	k	3	9	61	71	66	14	1	0	0
4	p	60	0	9	14	75	66	1	0	0
	t	8	63	5	67	10	69	0	2	1
	k	3	7	66	72	68	9	0	0	0
5	p	56	1	11	19	74	64	0	0	0
	t	5	63	1	70	11	74	0	1	0
	k	1	5	70	74	70	5	0	0	0

C.5.8 Experimento final 5, computador 2, instancia 3

Tabla 53. Pruebas del experimento 5, instancia 3, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	68	6	1	7	69	74	0	0	0
	t	4	67	1	70	6	74	1	2	0
	k	3	4	68	72	71	7	0	0	0
2	p	71	2	3	4	73	72	0	0	0
	t	4	71	0	71	3	75	0	1	0
	k	3	5	64	71	70	10	1	0	1
3	p	64	9	3	11	66	71	0	0	1
	t	5	70	0	70	5	75	0	0	0
	k	2	7	65	72	68	9	1	0	1
4	p	66	8	0	9	67	75	0	0	0
	t	10	61	1	64	13	74	1	1	0
	k	5	4	66	70	70	9	0	1	0

	p	67	6	0	7	69	74	1	0	1
5	t	4	70	0	71	5	75	0	0	0
	k	4	3	68	71	72	6	0	0	1

C.5.9 Experimento final 5, computador 2, instancia 4

Tabla 54. Pruebas del experimento 5, instancia 4, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	57	10	0	18	64	75	0	1	0
	t	4	65	8	71	10	67	0	0	0
	k	8	8	6	66	67	69	1	0	0
2	p	68	4	0	7	71	75	0	0	0
	t	5	63	6	70	12	69	0	0	0
	k	10	12	5	65	63	70	0	0	0
3	p	64	7	0	10	68	75	1	0	0
	t	3	65	4	72	10	71	0	0	0
	k	4	12	5	70	63	70	1	0	0
4	p	64	9	0	11	66	75	0	0	0
	t	10	55	4	65	19	69	0	1	2
	k	12	15	4	63	59	71	0	1	0
5	p	64	7	0	10	67	75	1	1	0
	t	5	62	6	70	13	69	0	0	0
	k	6	10	5	67	65	69	2	0	1

C.5.10 Experimento final 5, computador 2, instancia 5

Tabla 55. Pruebas del experimento 5, instancia 5, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	64	10	3	11	65	72	0	0	0
	t	12	62	5	63	12	69	0	1	1
	k	0	6	69	75	69	5	0	0	1
2	p	71	4	1	4	71	74	0	0	0
	t	10	61	4	64	13	71	1	1	0
	k	1	6	69	73	69	5	1	0	1
3	p	68	8	2	7	67	73	0	0	0
	t	13	59	7	62	15	68	0	1	0
	k	3	13	64	72	62	6	0	0	5
4	p	66	10	2	9	65	73	0	0	0
	t	18	57	5	57	18	70	0	0	0
	k	0	11	67	75	64	8	0	0	0
5	p	68	6	2	7	67	73	0	2	0
	t	8	63	6	67	11	69	0	1	0
	k	1	4	68	74	70	4	0	1	3

C.6 Experimento final 6: /p/, /t/ y /k/ cepstro

Cuadro 52. Experimento 6, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	del 2 al 15	58	35

Cuadro 53. Experimento 6, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	uniforme	aleatorio
/t/	44	150	23	75	uniforme	aleatorio
/k/	49	150	25	75	uniforme	aleatorio

Cuadro 54. Experimento 6, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 56. Entrenamiento del experimento 6

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	33	$9,14 \times 10^{-18}$	1349,1 s	1	28	$2,91 \times 10^{-17}$	689,5 s
2	100	$1,41 \times 10^{-02}$	4163,5 s	2	100	$3,70 \times 10^{-03}$	2516,8 s
3	22	$4,27 \times 10^{-17}$	894,4 s	3	31	$1,04 \times 10^{-17}$	767,8 s
4	25	$6,91 \times 10^{-17}$	1023,3 s	4	30	$2,31 \times 10^{-17}$	739,6 s
5	26	$2,20 \times 10^{-17}$	1028,7 s	5	100	$2,66 \times 10^{-02}$	2517,0 s

C.6.1 Experimento final 6, computador 1, instancia 1

Tabla 57. Pruebas del experimento 6, instancia 1, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	δp ?	δt ?	δk ?
1	p	66	4	5	9	71	70	0	0	0
	t	6	61	5	69	12	68	0	2	2
	k	0	3	72	75	72	3	0	0	0
2	p	66	5	5	9	70	70	0	0	0
	t	5	63	7	70	12	68	0	0	0
	k	2	7	65	72	68	9	1	0	1
3	p	66	4	4	9	71	71	0	0	0
	t	6	64	4	69	11	71	0	0	0
	k	1	6	67	74	69	8	0	0	0
4	p	68	4	4	7	71	70	0	0	1
	t	4	65	5	71	8	68	0	2	2
	k	1	7	67	74	68	8	0	0	0

5	p	63	5	7	12	70	67	0	0	1
	t	7	65	3	68	9	72	0	1	0
	k	2	4	70	73	70	4	0	1	1

C.6.2 Experimento final 6, computador 1, instancia 2

Tabla 58. Pruebas del experimento 6, instancia 2, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	73	5	0	2	70	75	0	0	0
	t	11	56	6	63	19	69	1	0	0
	k	9	5	58	66	70	17	0	0	0
2	p	72	5	0	3	70	75	0	0	0
	t	13	65	2	62	9	73	0	1	0
	k	6	7	61	69	68	13	0	0	1
3	p	72	5	0	3	70	75	0	0	0
	t	12	67	2	63	8	73	0	0	0
	k	2	9	57	73	66	18	0	0	0
4	p	73	5	0	2	70	75	0	0	0
	t	18	64	0	57	9	75	0	2	0
	k	3	7	63	72	68	12	0	0	0
5	p	73	6	0	2	69	75	0	0	0
	t	10	67	2	65	7	73	0	1	0
	k	7	3	62	68	72	13	0	0	0

C.6.3 Experimento final 6, computador 1, instancia 3

Tabla 59. Pruebas del experimento 6, instancia 3, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	72	3	0	3	69	75	0	3	0
	t	7	61	7	66	13	68	2	1	0
	k	0	6	72	75	68	2	0	1	1
2	p	68	5	0	5	69	75	2	1	0
	t	4	62	7	71	10	68	0	3	0
	k	0	7	69	75	67	5	0	1	1
3	p	69	5	0	5	70	74	1	0	1
	t	8	60	5	66	14	70	1	1	0
	k	0	8	71	75	65	4	0	2	0
4	p	67	4	0	5	70	74	3	1	1
	t	6	60	6	69	14	69	0	1	0
	k	1	8	66	74	66	9	0	1	0
5	p	72	1	2	3	74	73	0	0	0
	t	4	65	4	71	10	71	0	0	0
	k	1	8	70	74	66	4	0	1	1

C.6.4 Experimento final 6, computador 1, instancia 4

Tabla 60. Pruebas del experimento 6, instancia 4, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	71	3	0	4	72	75	0	0	0
	t	16	52	6	59	22	69	0	1	0
	k	2	2	72	73	72	2	0	1	1
2	p	72	3	0	3	72	75	0	0	0
	t	6	62	7	69	13	68	0	0	0
	k	5	2	69	69	72	5	1	1	1
3	p	72	3	0	3	72	75	0	0	0
	t	8	60	6	66	15	69	1	0	0
	k	1	2	72	74	72	3	0	1	0
4	p	71	3	0	3	72	75	1	0	0
	t	6	60	8	68	14	67	1	1	0
	k	4	8	64	71	66	11	0	1	0
5	p	72	3	0	3	72	75	0	0	0
	t	8	60	7	67	15	68	0	0	0
	k	3	2	71	72	73	3	0	0	1

C.6.5 Experimento final 6, computador 1, instancia 5

Tabla 61. Pruebas del experimento 6, instancia 5, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	70	5	0	5	70	75	0	0	0
	t	16	52	5	57	23	69	2	0	1
	k	1	2	72	74	71	3	0	2	0
2	p	72	2	0	1	72	75	2	1	0
	t	13	58	4	62	17	71	0	0	0
	k	1	3	71	74	72	4	0	0	0
3	p	69	6	0	5	69	75	1	0	0
	t	16	57	2	59	18	73	0	0	0
	k	0	8	67	75	67	8	0	0	0
4	p	70	5	0	4	70	75	1	0	0
	t	19	49	4	54	24	70	2	2	1
	k	1	12	62	74	63	13	0	0	0
5	p	71	4	0	4	71	75	0	0	0
	t	12	58	3	62	17	71	1	0	1
	k	1	5	68	74	69	6	0	1	1

C.6.6 Experimento final 6, computador 2, instancia 1

Tabla 62. Pruebas del experimento 6, instancia 1, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	66	6	4	9	68	71	0	1	0
	t	3	55	16	72	19	59	0	1	0
	k	2	3	70	73	72	5	0	0	0
2	p	66	7	4	9	68	71	0	0	0
	t	5	57	11	70	17	64	0	1	0
	k	3	3	68	72	72	7	0	0	0
3	p	67	6	4	7	69	71	1	0	0
	t	6	56	13	69	19	62	0	0	0
	k	5	6	66	70	69	9	0	0	0
4	p	66	6	8	7	68	67	2	1	0
	t	8	47	18	67	25	57	0	3	0
	k	3	5	67	72	70	8	0	0	0
5	p	65	9	5	9	66	70	1	0	0
	t	7	55	11	68	20	64	0	0	0
	k	1	3	70	74	71	5	0	1	0

C.6.7 Experimento final 6, computador 2, instancia 2

Tabla 63. Pruebas del experimento 6, instancia 2, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	72	0	4	3	72	71	0	3	0
	t	11	53	1	63	19	74	1	3	0
	k	4	14	57	71	61	18	0	0	0
2	p	72	0	4	3	73	71	0	2	0
	t	10	59	3	65	11	71	0	5	1
	k	2	13	60	73	62	15	0	0	0
3	p	75	1	4	0	74	71	0	0	0
	t	12	54	3	63	13	69	0	8	3
	k	5	14	58	70	61	17	0	0	0
4	p	73	0	3	2	74	72	0	1	0
	t	15	55	1	60	13	72	0	7	2
	k	2	16	54	73	58	21	0	1	0
5	p	72	0	6	3	74	69	0	1	0
	t	10	57	3	65	11	71	0	7	1
	k	7	8	59	66	67	16	2	0	0

C.6.8 Experimento final 6, computador 2, instancia 3

Tabla 64. Pruebas del experimento 6, instancia 3, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	60	8	2	14	67	71	1	0	2
	t	9	54	10	66	21	65	0	0	0
	k	3	4	65	71	71	10	1	0	0
2	p	62	8	0	12	67	71	1	0	4
	t	13	49	12	62	26	63	0	0	0
	k	3	6	62	72	69	13	0	0	0
3	p	67	6	1	8	69	74	0	0	0
	t	16	51	8	58	24	66	1	0	1
	k	1	8	65	74	66	10	0	1	0
4	p	64	6	2	11	69	71	0	0	2
	t	18	44	13	56	31	62	1	0	0
	k	3	11	59	72	63	16	0	1	0
5	p	61	6	2	14	68	72	0	1	1
	t	13	50	10	62	24	65	0	1	0
	k	4	11	57	71	64	17	0	0	1

C.6.9 Experimento final 6, computador 2, instancia 4

Tabla 65. Pruebas del experimento 6, instancia 4, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	67	5	2	8	69	73	0	1	0
	t	8	57	10	66	18	65	1	0	0
	k	1	3	71	74	72	4	0	0	0
2	p	63	6	5	7	68	70	5	1	0
	t	7	64	3	67	11	72	1	0	0
	k	2	3	72	73	72	3	0	0	0
3	p	69	4	3	5	71	72	1	0	0
	t	5	65	5	69	9	69	1	1	1
	k	0	7	68	75	68	7	0	0	0
4	p	69	5	3	6	70	72	0	0	0
	t	8	62	3	67	12	72	0	1	0
	k	1	5	69	74	70	5	0	0	1
5	p	63	5	8	10	70	66	2	0	1
	t	5	65	3	69	10	72	1	0	0
	k	0	5	71	75	70	4	0	0	0

C.6.10 Experimento final 6, computador 2, instancia 5

Tabla 66. Pruebas del experimento 6, instancia 5, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	\hat{p} ?	\hat{t} ?	\hat{k} ?
1	p	48	1	11	27	74	64	0	0	0
	t	0	63	0	75	11	75	0	1	0
	k	1	3	64	74	71	11	0	1	0
2	p	50	1	10	25	74	65	0	0	0
	t	0	56	3	75	16	71	0	3	1
	k	0	3	66	75	72	8	0	0	1
3	p	53	3	4	22	72	70	0	0	1
	t	0	58	3	75	16	71	0	1	1
	k	1	7	60	74	68	15	0	0	0
4	p	46	0	4	29	75	70	0	0	1
	t	1	56	3	74	19	72	0	0	0
	k	0	6	67	75	69	8	0	0	0
5	p	52	2	8	23	73	67	0	0	0
	t	0	59	2	75	13	72	0	3	1
	k	3	4	68	72	70	6	0	1	1

C.7 Experimento final 7: /p/, /t/ y /k/ cepstro

Cuadro 55. Experimento 7, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	del 2 al 15	58	35

Cuadro 56. Experimento 7, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	aleatorio	binario
/t/	44	150	23	75	aleatorio	binario
/k/	49	150	25	75	aleatorio	binario

Cuadro 57. Experimento 7, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 67. Entrenamiento del experimento 7

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	36	$2,06 \times 10^{-16}$	1469,0 s	1	22	$1,26 \times 10^{-16}$	545,1 s

2	100	$7,77 \times 10^{-02}$	4216,0 s	2	24	$5,98 \times 10^{-17}$	596,8 s
3	23	$7,39 \times 10^{-17}$	946,3 s	3	35	$3,45 \times 10^{-17}$	848,3 s
4	26	$3,25 \times 10^{-17}$	1049,4 s	4	27	$2,45 \times 10^{-17}$	661,7 s
5	100	$3,52 \times 10^{-01}$	3908,5 s	5	54	$2,87 \times 10^{-17}$	1362,2 s

C.7.1 Experimento final 7, computador 1, instancia 1

Tabla 68. Pruebas del experimento 7, instancia 1, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	$\exists p$?	$\exists t$?	$\exists k$?
1	p	75	0	2	0	75	73	0	0	0
	t	5	62	8	70	13	67	0	0	0
	k	0	6	69	75	69	6	0	0	0
2	p	75	0	6	0	75	69	0	0	0
	t	3	63	9	72	12	66	0	0	0
	k	0	9	67	75	66	8	0	0	0
3	p	75	0	8	0	75	67	0	0	0
	t	2	63	10	73	12	65	0	0	0
	k	0	1	73	75	73	1	0	1	1
4	p	75	0	7	0	75	68	0	0	0
	t	3	63	9	72	12	66	0	0	0
	k	0	4	71	75	71	4	0	0	0
5	p	75	0	7	0	75	68	0	0	0
	t	3	58	12	72	16	62	0	1	1
	k	0	6	69	75	69	6	0	0	0

C.7.2 Experimento final 7, computador 1, instancia 2

Tabla 69. Pruebas del experimento 7, instancia 2, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	$\exists p$?	$\exists t$?	$\exists k$?
1	p	73	0	0	2	75	75	0	0	0
	t	13	46	7	62	29	68	0	0	0
	k	7	8	50	68	67	25	0	0	0
2	p	73	0	0	2	75	75	0	0	0
	t	17	42	5	58	33	70	0	0	0
	k	5	9	47	70	66	28	0	0	0
3	p	73	0	0	2	75	75	0	0	0
	t	7	52	3	68	23	72	0	0	0
	k	11	6	40	63	69	35	1	0	0
4	p	68	0	0	7	75	75	0	0	0
	t	10	40	3	65	35	72	0	0	0
	k	8	5	53	67	70	22	0	0	0
5	p	72	0	0	3	75	75	0	0	0
	t	9	44	4	65	31	71	1	0	0
	k	10	6	51	64	69	24	1	0	0

C.7.3 Experimento final 7, computador 1, instancia 3

Tabla 70. Pruebas del experimento 7, instancia 3, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	75	75	0	0	0
	t	5	67	3	70	8	72	0	0	0
	k	0	12	61	73	63	14	2	0	0
2	p	75	0	0	0	75	75	0	0	0
	t	3	70	2	72	5	73	0	0	0
	k	0	9	66	75	66	9	0	0	0
3	p	75	0	0	0	75	75	0	0	0
	t	2	70	3	73	5	72	0	0	0
	k	0	11	62	74	64	12	1	0	1
4	p	75	0	0	0	75	75	0	0	0
	t	3	68	4	72	7	71	0	0	0
	k	0	10	62	73	65	12	2	0	1
5	p	75	0	0	0	75	75	0	0	0
	t	3	68	4	72	7	71	0	0	0
	k	1	10	62	73	65	11	1	0	2

C.7.4 Experimento final 7, computador 1, instancia 4

Tabla 71. Pruebas del experimento 7, instancia 4, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	75	75	0	0	0
	t	3	65	12	70	10	63	2	0	0
	k	0	3	72	75	72	3	0	0	0
2	p	75	0	0	0	75	75	0	0	0
	t	0	67	11	73	8	62	2	0	2
	k	0	4	71	75	71	4	0	0	0
3	p	75	0	0	0	75	75	0	0	0
	t	2	66	9	73	9	66	0	0	0
	k	0	3	72	75	72	3	0	0	0
4	p	75	0	0	0	75	75	0	0	0
	t	0	65	12	72	10	63	3	0	0
	k	0	4	71	75	71	4	0	0	0
5	p	75	0	0	0	75	75	0	0	0
	t	1	66	11	72	9	64	2	0	0
	k	0	3	72	75	72	3	0	0	0

C.7.5 Experimento final 7, computador 1, instancia 5

Tabla 72. Pruebas del experimento 7, instancia 5, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	75	75	0	0	0
	t	75	0	1	0	75	56	0	0	18
	k	75	0	63	0	75	6	0	0	6
2	p	75	0	0	0	75	75	0	0	0
	t	75	0	4	0	75	50	0	0	21
	k	75	0	58	0	75	10	0	0	7
3	p	75	0	0	0	75	75	0	0	0
	t	75	0	1	0	75	64	0	0	10
	k	75	0	59	0	75	11	0	0	5
4	p	75	0	0	0	75	75	0	0	0
	t	75	0	3	0	75	59	0	0	13
	k	75	0	58	0	75	10	0	0	7
5	p	75	0	0	0	75	75	0	0	0
	t	75	0	2	0	75	58	0	0	15
	k	75	0	57	0	75	4	0	0	14

C.7.6 Experimento final 7, computador 2, instancia 1

Tabla 73. Pruebas del experimento 7, instancia 1, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	75	75	0	0	0
	t	5	66	4	70	9	71	0	0	0
	k	3	6	66	72	69	9	0	0	0
2	p	75	0	0	0	75	75	0	0	0
	t	3	67	3	72	6	70	0	2	2
	k	4	8	63	71	67	12	0	0	0
3	p	75	0	0	0	75	75	0	0	0
	t	2	69	4	73	6	71	0	0	0
	k	1	7	67	74	68	8	0	0	0
4	p	75	0	0	0	75	75	0	0	0
	t	3	65	5	72	8	68	0	2	2
	k	1	5	68	74	69	6	0	1	1
5	p	75	0	0	0	75	75	0	0	0
	t	3	67	4	72	7	70	0	1	1
	k	2	6	67	73	69	8	0	0	0

C.7.7 Experimento final 7, computador 2, instancia 2

Tabla 74. Pruebas del experimento 7, instancia 2, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	3	0	75	72	0	0	0
	t	5	63	4	70	12	71	0	0	0
	k	0	3	72	75	72	3	0	0	0
2	p	75	0	1	0	75	74	0	0	0
	t	3	67	3	72	8	72	0	0	0
	k	0	4	71	75	71	4	0	0	0
3	p	75	0	3	0	75	72	0	0	0
	t	2	62	8	73	13	67	0	0	0
	k	0	3	71	75	69	3	0	3	1
4	p	75	0	2	0	75	72	0	0	1
	t	3	66	2	72	9	73	0	0	0
	k	0	4	71	75	70	4	0	1	0
5	p	75	0	3	0	75	72	0	0	0
	t	3	65	3	72	10	72	0	0	0
	k	0	3	72	75	72	3	0	0	0

C.7.8 Experimento final 7, computador 2, instancia 3

Tabla 75. Pruebas del experimento 7, instancia 3, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	75	75	0	0	0
	t	8	72	0	67	3	74	0	0	1
	k	3	4	68	72	70	5	0	1	2
2	p	75	0	0	0	75	75	0	0	0
	t	2	72	0	71	3	71	2	0	4
	k	4	5	66	71	69	5	0	1	4
3	p	75	0	0	0	75	75	0	0	0
	t	5	70	0	70	5	73	0	0	2
	k	1	3	71	74	71	3	0	1	1
4	p	75	0	0	0	75	75	0	0	0
	t	6	71	1	68	4	72	1	0	2
	k	1	2	72	74	73	2	0	0	1
5	p	75	0	0	0	75	75	0	0	0
	t	7	69	2	68	6	71	0	0	2
	k	2	5	68	73	69	5	0	1	2

C.7.9 Experimento final 7, computador 2, instancia 4

Tabla 76. Pruebas del experimento 7, instancia 4, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	75	75	0	0	0
	t	5	67	3	70	8	72	0	0	0
	k	4	8	68	70	67	7	1	0	0
2	p	75	0	3	0	75	72	0	0	0
	t	3	67	5	72	8	70	0	0	0
	k	5	5	70	70	70	5	0	0	0
3	p	72	0	3	3	75	72	0	0	0
	t	2	65	7	73	9	67	0	1	1
	k	2	10	65	70	65	10	3	0	0
4	p	73	0	3	2	75	72	0	0	0
	t	3	68	4	72	6	71	0	1	0
	k	2	8	67	72	67	8	1	0	0
5	p	75	0	2	0	75	73	0	0	0
	t	3	62	10	72	13	65	0	0	0
	k	4	7	68	70	68	7	1	0	0

C.7.10 Experimento final 7, computador 2, instancia 5

Tabla 77. Pruebas del experimento 7, instancia 5, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	72	2	3	3	73	72	0	0	0
	t	8	66	0	67	8	75	0	1	0
	k	0	12	66	75	63	9	0	0	0
2	p	74	0	0	0	74	75	1	1	0
	t	5	69	0	70	5	75	0	1	0
	k	0	12	64	75	63	11	0	0	0
3	p	73	1	2	2	74	73	0	0	0
	t	5	70	0	70	5	75	0	0	0
	k	0	14	61	75	61	14	0	0	0
4	p	74	0	0	0	75	75	1	0	0
	t	7	68	0	68	7	75	0	0	0
	k	0	14	61	75	61	14	0	0	0
5	p	73	0	2	2	74	73	0	1	0
	t	7	68	0	68	7	75	0	0	0
	k	0	13	61	75	60	12	0	2	2

C.8 Experimento final 8: /p/, /t/ y /k/ cepstro

Cuadro 58. Experimento 8, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/p/, /t/ y /k/	512 muestras	del 2 al 15	58	35

Cuadro 59. Experimento 8, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/p/	28	150	14	75	uniforme	binario
/t/	44	150	23	75	uniforme	binario
/k/	49	150	25	75	uniforme	binario

Cuadro 60. Experimento 8, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 78. Entrenamiento del experimento 8

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	100	$6,66 \times 10^{-03}$	4218,8 s	1	21	$1,12 \times 10^{-16}$	524,8 s
2	100	$1,20 \times 10^{-01}$	4198,4 s	2	32	$1,14 \times 10^{-16}$	819,7 s
3	100	$2,59 \times 10^{-02}$	4199,2 s	3	100	$2,22 \times 10^{-01}$	2489,8 s
4	24	$3,68 \times 10^{-17}$	952,4 s	4	27	$5,51 \times 10^{-17}$	669,2 s
5	100	$5,93 \times 10^{-03}$	4185,5 s	5	23	$1,59 \times 10^{-16}$	520,4 s

C.8.1 Experimento final 8, computador 1, instancia 1

Tabla 79. Pruebas del experimento 8, instancia 1, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	δp ?	δt ?	δk ?
1	p	67	0	0	7	75	75	1	0	0
	t	7	51	22	68	20	53	0	4	0
	k	13	10	60	62	65	15	0	0	0
2	p	67	0	0	7	75	75	1	0	0
	t	3	59	17	72	15	58	0	1	0
	k	9	15	52	65	60	22	1	0	1
3	p	57	0	0	17	75	75	1	0	0
	t	3	62	10	72	12	65	0	1	0
	k	4	12	60	71	63	15	0	0	0
4	p	65	0	0	8	75	75	2	0	0
	t	5	61	13	70	13	62	0	1	0
	k	4	12	63	71	63	12	0	0	0

	p	67	0	0	8	75	75	0	0	0
5	t	5	53	25	70	20	49	0	2	1
	k	6	12	62	69	63	13	0	0	0

C.8.2 Experimento final 8, computador 1, instancia 2

Tabla 80. Pruebas del experimento 8, instancia 2, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	65	0	0	10	75	75	0	0	0
	t	4	59	4	71	11	71	0	5	0
	k	2	15	0	73	60	75	0	0	0
2	p	61	0	0	14	75	75	0	0	0
	t	3	55	2	72	17	73	0	3	0
	k	6	13	0	69	61	75	0	1	0
3	p	57	0	0	17	75	75	1	0	0
	t	1	62	5	74	9	70	0	4	0
	k	4	8	0	71	66	75	0	1	0
4	p	56	0	0	19	75	75	0	0	0
	t	4	60	7	71	9	68	0	6	0
	k	3	12	0	72	63	75	0	0	0
5	p	61	0	0	14	75	75	0	0	0
	t	6	55	8	69	16	67	0	4	0
	k	3	12	0	72	62	75	0	1	0

C.8.3 Experimento final 8, computador 1, instancia 3

Tabla 81. Pruebas del experimento 8, instancia 3, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	69	6	8	6	69	67	0	0	0
	t	0	62	16	75	13	59	0	0	0
	k	3	5	58	72	70	17	0	0	0
2	p	72	3	0	3	72	75	0	0	0
	t	0	70	11	75	5	64	0	0	0
	k	4	11	54	71	64	21	0	0	0
3	p	66	9	7	9	66	68	0	0	0
	t	0	69	15	75	6	60	0	0	0
	k	4	4	62	71	71	12	0	0	1
4	p	67	8	6	8	67	69	0	0	0
	t	0	65	18	75	10	56	0	0	1
	k	0	5	62	75	70	13	0	0	0
5	p	70	5	2	5	70	73	0	0	0
	t	0	67	12	75	7	63	0	1	0
	k	2	5	58	73	70	17	0	0	0

C.8.4 Experimento final 8, computador 1, instancia 4

Tabla 82. Pruebas del experimento 8, instancia 4, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	75	0	0	0	75	75	0	0	0
	t	3	59	12	71	16	63	1	0	0
	k	0	1	74	75	74	1	0	0	0
2	p	75	0	0	0	75	75	0	0	0
	t	3	69	3	72	6	72	0	0	0
	k	0	3	71	75	72	3	0	0	1
3	p	75	0	0	0	75	75	0	0	0
	t	1	68	6	74	7	69	0	0	0
	k	0	4	71	75	71	4	0	0	0
4	p	75	0	0	0	75	75	0	0	0
	t	4	62	9	71	13	66	0	0	0
	k	0	6	69	75	69	6	0	0	0
5	p	75	0	0	0	75	75	0	0	0
	t	6	63	6	69	12	69	0	0	0
	k	0	3	72	75	72	3	0	0	0

C.8.5 Experimento final 8, computador 1, instancia 5

Tabla 83. Pruebas del experimento 8, instancia 5, computador 1

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	75	0	1	0	75	74	0	0	0
	t	0	63	7	75	11	68	0	1	0
	k	0	16	60	75	58	14	0	1	1
2	p	72	0	1	2	75	74	1	0	0
	t	0	72	1	75	3	74	0	0	0
	k	0	10	66	75	65	9	0	0	0
3	p	75	0	0	0	75	75	0	0	0
	t	0	70	4	75	5	71	0	0	0
	k	0	10	65	75	65	9	0	0	1
4	p	75	0	0	0	75	75	0	0	0
	t	0	69	5	75	6	69	0	0	1
	k	0	10	65	75	65	10	0	0	0
5	p	74	0	0	1	75	75	0	0	0
	t	0	70	1	75	5	74	0	0	0
	k	0	13	65	75	61	9	0	1	1

C.8.6 Experimento final 8, computador 2, instancia 1

Tabla 84. Pruebas del experimento 8, instancia 1, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	75	75	0	0	0
	t	4	63	3	71	6	72	0	6	0
	k	1	1	74	74	74	1	0	0	0
2	p	75	0	0	0	75	75	0	0	0
	t	3	67	4	72	7	71	0	1	0
	k	4	2	73	71	72	2	0	1	0
3	p	75	0	0	0	75	75	0	0	0
	t	1	71	2	74	3	73	0	1	0
	k	3	3	72	72	71	3	0	1	0
4	p	75	0	0	0	75	75	0	0	0
	t	4	66	3	71	7	72	0	2	0
	k	4	3	72	71	71	3	0	1	0
5	p	75	0	0	0	75	75	0	0	0
	t	6	62	5	69	11	70	0	2	0
	k	2	2	73	73	72	2	0	1	0

C.8.7 Experimento final 8, computador 2, instancia 2

Tabla 85. Pruebas del experimento 8, instancia 2, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ξp ?	ξt ?	ξk ?
1	p	75	0	0	0	72	75	0	3	0
	t	0	68	5	74	6	69	1	1	1
	k	0	3	73	75	72	2	0	0	0
2	p	75	1	0	0	73	75	0	1	0
	t	2	73	1	73	2	74	0	0	0
	k	0	2	73	75	73	2	0	0	0
3	p	75	1	0	0	70	75	0	4	0
	t	0	74	1	74	1	74	1	0	0
	k	0	3	72	75	72	3	0	0	0
4	p	75	5	0	0	69	75	0	1	0
	t	1	71	2	73	3	73	1	1	0
	k	0	3	72	75	72	3	0	0	0
5	p	75	2	0	0	71	75	0	2	0
	t	1	69	3	73	5	71	1	1	1
	k	0	3	72	75	72	2	0	0	1

C.8.8 Experimento final 8, computador 2, instancia 3

Tabla 86. Pruebas del experimento 8, instancia 3, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	70	2	75	5	73	0	0	0	0
	t	0	56	75	75	14	0	0	5	0
	k	0	11	75	75	63	0	0	1	0
2	p	69	0	75	6	74	0	0	1	0
	t	2	61	75	73	11	0	0	3	0
	k	0	17	75	75	57	0	0	1	0
3	p	70	0	75	5	75	0	0	0	0
	t	2	61	75	73	10	0	0	4	0
	k	0	17	75	75	57	0	0	1	0
4	p	70	1	75	5	74	0	0	0	0
	t	1	58	75	72	15	0	2	2	0
	k	0	14	75	75	61	0	0	0	0
5	p	68	0	75	7	75	0	0	0	0
	t	1	56	75	74	16	0	0	3	0
	k	0	10	75	75	64	0	0	1	0

C.8.9 Experimento final 8, computador 2, instancia 4

Tabla 87. Pruebas del experimento 8, instancia 4, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	ζp ?	ζt ?	ζk ?
1	p	75	0	0	0	75	75	0	0	0
	t	4	68	3	71	7	72	0	0	0
	k	0	10	65	75	65	10	0	0	0
2	p	75	0	0	0	75	75	0	0	0
	t	3	68	4	72	7	71	0	0	0
	k	0	6	69	75	68	6	0	1	0
3	p	75	0	0	0	75	75	0	0	0
	t	1	72	2	74	3	73	0	0	0
	k	0	7	68	75	67	7	0	1	0
4	p	75	0	0	0	75	75	0	0	0
	t	4	68	3	71	7	72	0	0	0
	k	0	13	62	75	62	10	0	0	3
5	p	75	0	0	0	75	75	0	0	0
	t	6	64	5	69	11	70	0	0	0
	k	0	6	69	75	69	6	0	0	0

C.8.10 Experimento final 8, computador 2, instancia 5

Tabla 88. Pruebas del experimento 8, instancia 5, computador 2

Prueba	Entrada	Aceptación			Rechazo			Indecisión		
		p	t	k	$\neg p$	$\neg t$	$\neg k$	\hat{p} ?	\hat{t} ?	\hat{k} ?
1	p	74	0	1	0	75	74	1	0	0
	t	0	64	13	75	11	59	0	0	3
	k	0	9	66	75	66	9	0	0	0
2	p	75	0	0	0	75	75	0	0	0
	t	0	69	8	75	6	66	0	0	1
	k	0	5	70	75	69	5	0	1	0
3	p	74	0	0	0	75	74	1	0	1
	t	0	71	4	75	4	71	0	0	0
	k	0	5	70	75	70	5	0	0	0
4	p	75	0	0	0	75	75	0	0	0
	t	0	67	8	75	8	66	0	0	1
	k	0	7	68	75	68	7	0	0	0
5	p	74	0	1	0	75	74	1	0	0
	t	0	64	10	75	11	63	0	0	2
	k	0	6	69	75	69	6	0	0	0

C.9 Experimento final 9: /a/ y /e/ wavelets

Cuadro 61. Experimento 9, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 62. Experimento 9, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	54	150	27	75	aleatoria	aleatorio
/e/	64	150	32	75	aleatoria	aleatoria

Cuadro 63. Experimento 9, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 89. Entrenamiento del experimento 9

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	100	$1,68 \times 10^{-02}$	1124,5 s	1	100	$1,02 \times 10^{-02}$	682,1 s
2	100	$6,69 \times 10^{-03}$	1099,6 s	2	100	$3,33 \times 10^{-03}$	665,8 s

3	92	$3,33 \times 10^{-03}$	993,3 s	3	95	$9,14 \times 10^{-18}$	617,8 s
4	100	$1,01 \times 10^{-02}$	1107,0 s	4	100	$1,98 \times 10^{-16}$	662,9 s
5	66	$6,67 \times 10^{-03}$	702,8 s	5	100	$6,36 \times 10^{-02}$	678,8 s

C.9.1 Experimento final 9, computador 1, instancia 1

Tabla 90. Pruebas del experimento 9, instancia 1, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	50	25	25	50	0	0
	e	26	47	47	26	2	2
2	a	53	20	20	53	2	2
	e	16	56	56	16	3	3
3	a	50	24	24	50	1	1
	e	26	49	49	26	0	0
4	a	61	13	13	61	1	1
	e	24	51	51	24	0	0
5	a	57	18	18	57	0	0
	e	19	56	56	19	0	0

C.9.2 Experimento final 9, computador 1, instancia 2

Tabla 91. Pruebas del experimento 9, instancia 2, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	54	20	20	54	1	1
	e	27	48	48	27	0	0
2	a	49	26	26	49	0	0
	e	22	52	52	22	1	1
3	a	50	24	24	50	1	1
	e	20	53	53	20	2	2
4	a	59	12	12	59	4	4
	e	22	51	51	22	2	2
5	a	63	12	12	63	0	0
	e	17	58	58	17	0	0

C.9.3 Experimento final 9, computador 1, instancia 3

Tabla 92. Pruebas del experimento 9, instancia 3, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	51	23	23	51	1	1
	e	18	57	57	18	0	0
2	a	51	22	22	51	2	2
	e	12	63	63	12	0	0
3	a	46	27	27	46	2	2

	e	18	55	55	18	2	2
4	a	54	20	20	54	1	1
	e	17	56	57	17	1	2
5	a	53	21	21	53	1	1
	e	16	58	58	16	1	1

C.9.4 Experimento final 9, computador 1, instancia 4

Tabla 93. Pruebas del experimento 9, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	60	12	12	60	3	3
	e	26	47	47	26	2	2
2	a	52	21	21	52	2	2
	e	21	52	51	22	3	1
3	a	55	18	18	55	2	2
	e	19	52	52	20	4	3
4	a	62	11	11	62	2	2
	e	19	53	53	19	3	3
5	a	56	17	18	56	1	2
	e	17	56	56	17	2	2

C.9.5 Experimento final 9, computador 1, instancia 5

Tabla 94. Pruebas del experimento 9, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	55	20	20	55	0	0
	e	21	54	54	21	0	0
2	a	52	23	23	52	0	0
	e	19	55	55	19	1	1
3	a	47	28	28	47	0	0
	e	17	58	58	17	0	0
4	a	62	13	13	62	0	0
	e	19	56	56	19	0	0
5	a	54	21	21	54	0	0
	e	11	62	62	11	2	2

C.9.6 Experimento final 9, computador 2, instancia 1

Tabla 95. Pruebas del experimento 9, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	55	20	20	55	0	0
	e	25	48	48	25	2	2
2	a	51	22	22	51	2	2

	e	17	57	57	17	1	1
3	a	49	23	23	49	3	3
	e	21	52	52	21	2	2
4	a	63	10	10	63	2	2
	e	20	53	53	20	2	2
5	a	59	15	15	59	1	1
	e	16	59	59	16	0	0

C.9.7 Experimento final 9, computador 2, instancia 2

Tabla 96. Pruebas del experimento 9, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	57	17	17	57	1	1
	e	33	40	40	33	2	2
2	a	52	22	22	52	1	1
	e	27	48	48	27	0	0
3	a	57	17	17	57	1	1
	e	28	46	45	29	2	0
4	a	66	9	9	66	0	0
	e	28	47	47	28	0	0
5	a	62	13	13	62	0	0
	e	20	53	53	20	2	2

C.9.8 Experimento final 9, computador 2, instancia 3

Tabla 97. Pruebas del experimento 9, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	48	26	26	48	1	1
	e	21	53	53	21	1	1
2	a	48	27	27	48	0	0
	e	17	58	58	17	0	0
3	a	45	30	30	45	0	0
	e	18	57	57	18	0	0
4	a	52	20	20	52	3	3
	e	16	58	58	16	1	1
5	a	52	22	22	52	1	1
	e	19	56	56	19	0	0

C.9.9 Experimento final 9, computador 2, instancia 4

Tabla 98. Pruebas del experimento 9, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	56	19	19	56	0	0

	e	22	52	51	22	2	1
2	a	57	17	17	57	1	1
	e	24	50	50	24	1	1
3	a	53	22	22	53	0	0
	e	17	57	57	17	1	1
4	a	60	14	14	60	1	1
	e	23	50	50	23	2	2
5	a	58	17	17	58	0	0
	e	19	55	54	19	2	1

C.9.10 Experimento final 9, computador 2, instancia 5

Tabla 99. Pruebas del experimento 9, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	50	13	22	61	3	1
	e	19	48	55	25	1	2
2	a	48	19	26	55	1	1
	e	14	44	60	28	1	3
3	a	53	17	22	56	0	2
	e	12	53	62	21	1	1
4	a	63	9	12	62	0	4
	e	20	52	54	21	1	2
5	a	55	15	18	58	2	2
	e	12	55	61	16	2	4

C.10 Experimento final 10: /a/ y /e/ wavelets

Cuadro 64. Experimento 10, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 65. Experimento 10, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	54	150	27	75	uniforme	aleatorio
/e/	64	150	32	75	uniforme	aleatoria

Cuadro 66. Experimento 10, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 100. Entrenamiento del experimento 10

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	100	$5,96 \times 10^{-03}$	1115,2 s	1	94	$4,54 \times 10^{-18}$	616,2 s
2	56	$1,33 \times 10^{-02}$	601,2 s	2	74	$1,00 \times 10^{-02}$	488,0 s
3	100	$1,33 \times 10^{-02}$	1309,5 s	3	100	$6,67 \times 10^{-03}$	668,5 s
4	73	$1,54 \times 10^{-18}$	794,4 s	4	93	$6,67 \times 10^{-03}$	610,9 s
5	89	$2,85 \times 10^{-17}$	974,2 s	5	100	$1,32 \times 10^{-02}$	680,2 s

C.10.1 Experimento final 10, computador 1, instancia 1**Tabla 101. Pruebas del experimento 10, instancia 1, computador 1**

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	53	20	20	53	2	2
	e	23	49	49	23	3	3
2	a	49	26	26	49	0	0
	e	27	47	47	27	1	1
3	a	53	21	20	53	2	1
	e	20	54	54	20	1	1
4	a	63	11	11	63	1	1
	e	29	46	46	29	0	0
5	a	47	26	26	47	2	2
	e	24	51	51	24	0	0

C.10.2 Experimento final 10, computador 1, instancia 2**Tabla 102. Pruebas del experimento 10, instancia 2, computador 1**

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	61	13	13	61	1	1
	e	18	57	57	18	0	0
2	a	57	18	17	56	1	1
	e	13	61	61	13	1	1
3	a	63	12	12	62	0	1
	e	13	62	62	13	0	0
4	a	59	15	15	59	1	1
	e	21	52	52	21	2	2
5	a	60	12	13	60	2	3
	e	19	56	56	19	0	0

C.10.3 Experimento final 10, computador 1, instancia 3**Tabla 103. Pruebas del experimento 10, instancia 3, computador 1**

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	57	18	18	57	0	0

	e	16	57	57	17	2	1
2	a	56	19	19	56	0	0
	e	12	63	63	12	0	0
3	a	51	24	24	51	0	0
	e	13	62	62	13	0	0
4	a	51	22	22	51	2	2
	e	22	52	52	22	1	1
5	a	53	21	21	53	1	1
	e	16	59	59	16	0	0

C.10.4 Experimento final 10, computador 1, instancia 4

Tabla 104. Pruebas del experimento 10, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	50	24	24	50	1	1
	e	22	52	52	22	1	1
2	a	54	20	20	54	1	1
	e	24	51	51	24	0	0
3	a	50	24	24	50	1	1
	e	22	53	53	22	0	0
4	a	51	24	24	51	0	0
	e	22	53	53	22	0	0
5	a	58	17	17	58	0	0
	e	27	47	47	27	1	1

C.10.5 Experimento final 10, computador 1, instancia 5

Tabla 105. Pruebas del experimento 10, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	64	11	11	64	0	0
	e	22	52	52	22	1	1
2	a	60	15	15	60	0	0
	e	19	55	55	19	1	1
3	a	52	23	23	52	0	0
	e	18	56	56	18	1	1
4	a	58	17	17	58	0	0
	e	20	55	55	20	0	0
5	a	59	16	16	59	0	0
	e	25	49	49	25	1	1

C.10.6 Experimento final 10, computador 2, instancia 1

Tabla 106. Pruebas del experimento 10, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	59	15	15	59	1	1
	e	25	50	50	25	0	0
2	a	57	16	16	58	2	1
	e	20	55	55	20	0	0
3	a	52	21	21	52	2	2
	e	19	56	56	19	0	0
4	a	52	23	22	51	1	1
	e	23	52	52	23	0	0
5	a	50	25	25	50	0	0
	e	25	50	50	25	0	0

C.10.7 Experimento final 10, computador 2, instancia 2

Tabla 107. Pruebas del experimento 10, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	60	14	15	60	0	1
	e	19	56	56	19	0	0
2	a	55	19	19	55	1	1
	e	25	48	48	25	2	2
3	a	58	17	17	58	0	0
	e	19	54	54	19	2	2
4	a	59	16	16	59	0	0
	e	22	53	53	22	0	0
5	a	57	18	18	57	0	0
	e	23	52	52	23	0	0

C.10.8 Experimento final 10, computador 2, instancia 3

Tabla 108. Pruebas del experimento 10, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	49	20	20	49	6	6
	e	20	55	55	20	0	0
2	a	54	19	19	54	2	2
	e	21	51	52	21	2	3
3	a	54	20	20	54	1	1
	e	16	58	57	16	2	1
4	a	57	18	18	57	0	0
	e	18	52	52	18	5	5

5	a	56	17	17	56	2	2
	e	16	59	59	16	0	0

C.10.9 Experimento final 10, computador 2, instancia 4

Tabla 109. Pruebas del experimento 10, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	55	20	20	55	0	0
	e	26	49	49	26	0	0
2	a	55	20	20	55	0	0
	e	25	50	50	25	0	0
3	a	55	19	19	55	1	1
	e	15	60	60	15	0	0
4	a	56	18	18	56	1	1
	e	24	51	51	24	0	0
5	a	53	21	21	53	1	1
	e	28	47	47	28	0	0

C.10.10 Experimento final 10, computador 2, instancia 5

Tabla 110. Pruebas del experimento 10, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	57	17	17	57	1	1
	e	18	57	57	18	0	0
2	a	53	20	20	53	2	2
	e	22	51	51	22	2	2
3	a	59	14	14	59	2	2
	e	21	54	54	21	0	0
4	a	55	20	20	55	0	0
	e	22	53	53	22	0	0
5	a	54	21	21	54	0	0
	e	19	56	56	19	0	0

C.11 Experimento final 11: /a/ y /e/ wavelets

Cuadro 67. Experimento 11, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 68. Experimento 11, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		

/a/	54	150	27	75	aleatorio	binario
/e/	64	150	32	75	aleatorio	binario

Cuadro 69. Experimento 11, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 111. Entrenamiento del experimento 10

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	60	$1,33 \times 10^{-02}$	643,6 s	1	100	$3,50 \times 10^{-02}$	673,1 s
2	100	$2,81 \times 10^{-01}$	1118,2 s	2	100	$1,05 \times 10^{-02}$	672,1 s
3	67	$3,33 \times 10^{-03}$	727,5 s	3	100	$3,03 \times 10^{-01}$	641,0 s
4	92	$6,67 \times 10^{-03}$	987,6 s	4	100	$1,36 \times 10^{-02}$	676,0 s
5	100	$2,91 \times 10^{-01}$	1103,9 s	5	100	$7,91 \times 10^{-03}$	687,0 s

C.11.1 Experimento final 11, computador 1, instancia 1**Tabla 112. Pruebas del experimento 11, instancia 1, computador 1**

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	57	18	18	57	0	0
	e	28	45	45	28	2	2
2	a	58	16	16	58	1	1
	e	21	53	53	21	1	1
3	a	51	23	23	51	1	1
	e	23	51	51	23	1	1
4	a	56	19	19	56	0	0
	e	24	51	51	24	0	0
5	a	55	19	19	55	1	1
	e	25	49	49	25	1	1

C.11.2 Experimento final 11, computador 1, instancia 2**Tabla 113. Pruebas del experimento 11, instancia 2, computador 1**

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	75	15	0	56	0	4
	e	75	48	0	15	0	12
2	a	75	10	0	56	0	9
	e	75	50	0	12	0	13
3	a	75	15	0	48	0	12
	e	75	45	0	17	0	13
4	a	75	11	0	59	0	5
	e	75	49	0	14	0	12

5	a	75	14	0	53	0	8
	e	75	48	0	15	0	12

C.11.3 Experimento final 11, computador 1, instancia 3

Tabla 114. Pruebas del experimento 11, instancia 3, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	59	16	16	59	0	0
	e	25	50	50	25	0	0
2	a	61	14	14	61	0	0
	e	21	54	54	21	0	0
3	a	57	18	18	57	0	0
	e	17	56	56	17	2	2
4	a	70	5	5	70	0	0
	e	20	54	54	20	1	1
5	a	60	15	15	60	0	0
	e	17	55	55	17	3	3

C.11.4 Experimento final 11, computador 1, instancia 4

Tabla 115. Pruebas del experimento 11, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	57	18	18	57	0	0
	e	30	42	42	30	3	3
2	a	60	15	15	60	0	0
	e	26	49	49	26	0	0
3	a	59	16	16	59	0	0
	e	25	49	49	25	1	1
4	a	63	12	12	63	0	0
	e	29	45	45	29	1	1
5	a	57	17	17	57	1	1
	e	25	50	50	25	0	0

C.11.5 Experimento final 11, computador 1, instancia 5

Tabla 116. Pruebas del experimento 11, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	57	75	17	0	1	0
	e	9	75	54	0	12	0
2	a	56	75	17	0	2	0
	e	9	75	57	0	9	0
3	a	52	75	16	0	7	0
	e	10	75	53	0	12	0

4	a	64	75	7	0	4	0
	e	10	75	56	0	9	0
5	a	53	75	16	0	6	0
	e	13	75	53	0	9	0

C.11.6 Experimento final 11, computador 2, instancia 1

Tabla 117. Pruebas del experimento 11, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	57	13	14	57	4	5
	e	24	48	48	24	3	3
2	a	58	13	13	58	4	4
	e	14	57	57	14	4	4
3	a	56	14	14	57	5	4
	e	17	53	53	17	5	5
4	a	61	10	11	63	3	2
	e	8	57	57	8	10	10
5	a	59	12	11	58	5	5
	e	15	50	50	15	10	10

C.11.7 Experimento final 11, computador 2, instancia 2

Tabla 118. Pruebas del experimento 11, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	60	15	14	60	1	0
	e	22	48	48	21	5	6
2	a	64	10	10	63	1	2
	e	19	54	54	19	2	2
3	a	60	14	14	60	1	1
	e	19	53	53	19	3	3
4	a	62	11	11	62	2	2
	e	14	60	60	14	1	1
5	a	59	15	15	59	1	1
	e	13	59	59	13	3	3

C.11.8 Experimento final 11, computador 2, instancia 3

Tabla 119. Pruebas del experimento 11, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	53	75	17	0	5	0
	e	11	75	35	0	29	0
2	a	57	75	9	0	9	0
	e	7	75	52	0	16	0

3	a	54	75	14	0	7	0
	e	11	75	41	0	23	0
4	a	59	75	4	0	12	0
	e	6	75	40	0	29	0
5	a	50	75	16	0	9	0
	e	8	75	43	0	24	0

C.11.9 Experimento final 11, computador 2, instancia 4

Tabla 120. Pruebas del experimento 11, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	52	21	21	52	2	2
	e	15	56	58	16	2	3
2	a	53	18	18	54	4	3
	e	14	54	54	16	7	5
3	a	54	18	20	55	1	2
	e	22	46	49	24	4	5
4	a	58	15	16	58	1	2
	e	26	45	45	26	4	4
5	a	51	21	21	51	3	3
	e	21	47	52	24	2	4

C.11.10 Experimento final 11, computador 2, instancia 5

Tabla 121. Pruebas del experimento 11, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	64	11	11	64	0	0
	e	24	50	50	24	1	1
2	a	61	14	14	61	0	0
	e	15	57	57	15	3	3
3	a	61	14	14	61	0	0
	e	18	56	56	18	1	1
4	a	60	14	14	60	1	1
	e	17	56	56	16	2	3
5	a	59	16	16	59	0	0
	e	20	55	55	20	0	0

C.12 Experimento final 12: /a/ y /e/ wavelets

Cuadro 70. Experimento 12, descripción del experimento

Fonemas	Tamaño de la ventana	Onditas			Número de frases	
		Ondita	Niveles	Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	<i>Haar</i> (<i>Doubachies</i> 1)	7	energía por nivel	58	35

Cuadro 71. Experimento 12, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	54	150	27	75	uniforme	binario
/e/	64	150	32	75	uniforme	binario

Cuadro 72. Experimento 12, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 122. Entrenamiento del experimento 10

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	59	$3,33 \times 10^{-03}$	625,3 s	1	76	$3,33 \times 10^{-03}$	498,7 s
2	100	$6,67 \times 10^{-03}$	1116,3 s	2	87	$3,47 \times 10^{-18}$	572,2 s
3	82	$3,33 \times 10^{-03}$	889,7 s	3	100	$6,67 \times 10^{-03}$	676,4 s
4	100	$6,67 \times 10^{-03}$	1110,7 s	4	41	$1,61 \times 10^{-17}$	263,6 s
5	100	$5,96 \times 10^{-03}$	1113,6 s	5	100	$1,72 \times 10^{-02}$	662,8 s

C.12.1 Experimento final 12, computador 1, instancia 1**Tabla 123. Pruebas del experimento 12, instancia 1, computador 1**

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	63	12	12	63	0	0
	e	18	57	57	18	0	0
2	a	65	10	10	65	0	0
	e	20	54	54	20	1	1
3	a	66	9	9	66	0	0
	e	18	56	56	18	1	1
4	a	68	7	7	68	0	0
	e	18	57	57	18	0	0
5	a	68	7	7	68	0	0
	e	19	55	55	19	1	1

C.12.2 Experimento final 12, computador 1, instancia 2**Tabla 124. Pruebas del experimento 12, instancia 2, computador 1**

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	62	13	13	62	0	0
	e	18	56	56	18	1	1
2	a	57	17	17	58	1	0
	e	23	51	51	23	1	1

3	a	60	15	15	60	0	0
	e	21	52	51	21	3	2
4	a	67	8	8	67	0	0
	e	19	54	54	19	2	2
5	a	67	8	8	67	0	0
	e	25	50	50	25	0	0

C.12.3 Experimento final 12, computador 1, instancia 3

Tabla 125. Pruebas del experimento 12, instancia 3, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	61	14	14	61	0	0
	e	15	57	57	15	3	3
2	a	54	20	21	54	0	1
	e	22	52	52	22	1	1
3	a	56	19	19	56	0	0
	e	23	51	51	23	1	1
4	a	63	11	12	63	0	1
	e	17	57	57	17	1	1
5	a	62	12	12	62	1	1
	e	22	52	52	22	1	1

C.12.4 Experimento final 12, computador 1, instancia 4

Tabla 126. Pruebas del experimento 12, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	56	19	19	56	0	0
	e	19	55	55	19	1	1
2	a	54	20	20	54	1	1
	e	23	50	50	23	2	2
3	a	56	19	19	56	0	0
	e	23	52	52	23	0	0
4	a	56	19	19	56	0	0
	e	14	61	61	14	0	0
5	a	59	15	15	59	1	1
	e	17	57	57	17	1	1

C.12.5 Experimento final 12, computador 1, instancia 5

Tabla 127. Pruebas del experimento 12, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	58	16	16	58	1	1
	e	19	53	53	19	3	3

2	a	58	17	17	58	0	0
	e	27	46	46	27	2	2
3	a	56	19	19	56	0	0
	e	22	51	51	22	2	2
4	a	60	13	13	60	2	2
	e	13	61	61	13	1	1
5	a	60	14	14	60	1	1
	e	22	52	51	21	2	2

C.12.6 Experimento final 12, computador 2, instancia 1

Tabla 128. Pruebas del experimento 12, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	64	11	11	64	0	0
	e	21	53	53	21	1	1
2	a	58	17	17	58	0	0
	e	27	48	48	27	0	0
3	a	60	15	15	60	0	0
	e	22	52	52	22	1	1
4	a	61	14	14	61	0	0
	e	16	58	58	16	1	1
5	a	61	13	13	61	1	1
	e	21	53	53	21	1	1

C.12.7 Experimento final 12, computador 2, instancia 2

Tabla 129. Pruebas del experimento 12, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	59	16	16	59	0	0
	e	20	54	54	20	1	1
2	a	63	12	12	63	0	0
	e	27	48	48	27	0	0
3	a	60	15	15	60	0	0
	e	30	44	44	30	1	1
4	a	64	11	11	64	0	0
	e	18	57	57	18	0	0
5	a	66	9	9	66	0	0
	e	29	45	45	29	1	1

C.12.8 Experimento final 12, computador 2, instancia 3

Tabla 130. Pruebas del experimento 12, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$

1	a	57	17	17	57	1	1
	e	20	55	55	20	0	0
2	a	55	20	20	55	0	0
	e	18	53	53	18	4	4
3	a	59	16	16	59	0	0
	e	14	59	59	14	2	2
4	a	64	9	9	65	2	1
	e	18	54	54	18	3	3
5	a	60	14	14	60	1	1
	e	18	54	54	18	3	3

C.12.9 Experimento final 12, computador 2, instancia 4

Tabla 131. Pruebas del experimento 12, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	62	12	12	62	1	1
	e	31	42	42	31	2	2
2	a	58	17	17	58	0	0
	e	28	45	45	28	2	2
3	a	63	11	11	63	1	1
	e	33	41	41	33	1	1
4	a	69	6	6	69	0	0
	e	17	57	57	17	1	1
5	a	62	13	13	62	0	0
	e	28	46	46	28	1	1

C.12.10 Experimento final 12, computador 2, instancia 5

Tabla 132. Pruebas del experimento 12, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	61	14	14	61	0	0
	e	19	52	52	19	4	4
2	a	55	19	19	55	1	1
	e	24	50	50	24	1	1
3	a	56	18	18	56	1	1
	e	24	49	49	24	2	2
4	a	63	12	12	63	0	0
	e	24	48	48	24	3	3
5	a	55	18	18	55	2	2
	e	21	53	53	21	1	1

C.13 Experimento final 13: /a/ y /e/ cepstro

Cuadro 73. Experimento 13, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro		Número de frases	
		Coeficientes		entrenamiento	pruebas
/a/ y /e/	512 muestras	del 2 al 15		58	35

Cuadro 74. Experimento 13, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	54	150	27	75	aleatorio	aleatorio
/e/	64	150	32	75	aleatorio	aleatorio

Cuadro 75. Experimento 13, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 133. Entrenamiento del experimento 13

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	24	$4,23 \times 10^{-17}$	413,4 s	1	21	$9,33 \times 10^{-17}$	211,7 s
2	32	$3,92 \times 10^{-17}$	572,0 s	2	23	$5,96 \times 10^{-17}$	233,8 s
3	20	$1,19 \times 10^{-16}$	341,5 s	3	100	$2,33 \times 10^{-02}$	1073,2 s
4	64	$1,00 \times 10^{-02}$	1165,5 s	4	21	$5,54 \times 10^{-17}$	211,1 s
5	24	$3,44 \times 10^{-17}$	411,3 s	5	21	$2,96 \times 10^{-17}$	208,7 s

C.13.1 Experimento final 13, computador 1, instancia 1

Tabla 134. Pruebas del experimento 13, instancia 1, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	2	2	73	0	0
	e	6	69	69	5	0	1
2	a	72	3	3	72	0	0
	e	2	73	73	2	0	0
3	a	73	2	2	73	0	0
	e	8	67	67	8	0	0
4	a	73	2	2	73	0	0
	e	5	70	70	5	0	0
5	a	74	1	1	74	0	0
	e	4	71	71	4	0	0

C.13.2 Experimento final 13, computador 1, instancia 2

Tabla 135. Pruebas del experimento 13, instancia 2, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	72	3	3	72	0	0
	e	5	70	70	5	0	0
2	a	71	4	4	71	0	0
	e	3	72	72	3	0	0
3	a	72	3	3	72	0	0
	e	9	66	66	9	0	0
4	a	74	1	1	74	0	0
	e	5	70	70	5	0	0
5	a	74	1	1	73	0	1
	e	3	72	72	3	0	0

C.13.3 Experimento final 13, computador 1, instancia 3

Tabla 136. Pruebas del experimento 13, instancia 3, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	71	2	2	72	2	1
	e	4	71	71	4	0	0
2	a	73	2	2	73	0	0
	e	0	75	75	0	0	0
3	a	72	2	3	72	0	1
	e	5	69	69	5	1	1
4	a	74	1	1	74	0	0
	e	1	74	74	1	0	0
5	a	74	0	0	74	1	1
	e	1	74	74	1	0	0

C.13.4 Experimento final 13, computador 1, instancia 4

Tabla 137. Pruebas del experimento 13, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	70	4	5	71	0	0
	e	1	75	73	0	1	0
2	a	72	3	3	72	0	0
	e	3	74	72	0	0	1
3	a	71	5	4	70	0	0
	e	7	74	67	1	1	0
4	a	73	4	2	71	0	0
	e	11	70	64	5	0	0

5	a	70	5	4	70	1	0
	e	4	73	70	2	1	0

C.13.5 Experimento final 13, computador 1, instancia 5

Tabla 138. Pruebas del experimento 13, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	72	3	3	72	0	0
	e	4	71	71	4	0	0
2	a	72	2	2	72	1	1
	e	0	75	75	0	0	0
3	a	72	3	3	72	0	0
	e	6	69	69	6	0	0
4	a	73	2	2	73	0	0
	e	4	70	70	3	1	2
5	a	73	2	2	73	0	0
	e	4	71	71	4	0	0

C.13.6 Experimento final 13, computador 2, instancia 1

Tabla 139. Pruebas del experimento 13, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	1	1	74	0	0
	e	10	63	63	10	2	2
2	a	73	2	2	73	0	0
	e	6	69	69	6	0	0
3	a	73	2	2	73	0	0
	e	11	64	64	10	0	1
4	a	73	2	2	73	0	0
	e	10	65	65	10	0	0
5	a	73	2	2	73	0	0
	e	10	64	65	10	0	1

C.13.7 Experimento final 13, computador 2, instancia 2

Tabla 140. Pruebas del experimento 13, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	1	1	74	1	0
	e	2	73	73	2	0	0
2	a	72	3	3	72	0	0
	e	2	73	73	2	0	0
3	a	73	2	2	73	0	0
	e	3	72	72	3	0	0

4	a	72	1	3	72	0	2
	e	1	74	74	1	0	0
5	a	73	1	1	74	1	0
	e	1	74	74	1	0	0

C.13.8 Experimento final 13, computador 2, instancia 3

Tabla 141. Pruebas del experimento 13, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	0	1	75	0	0
	e	3	63	72	12	0	0
2	a	74	1	1	74	0	0
	e	0	72	75	3	0	0
3	a	74	1	1	74	0	0
	e	5	58	70	17	0	0
4	a	75	0	0	75	0	0
	e	3	58	72	16	0	1
5	a	73	2	2	73	0	0
	e	1	64	74	11	0	0

C.13.9 Experimento final 13, computador 2, instancia 4

Tabla 142. Pruebas del experimento 13, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	2	2	73	0	0
	e	3	72	72	3	0	0
2	a	73	2	2	73	0	0
	e	0	74	74	0	1	1
3	a	75	0	0	75	0	0
	e	6	69	69	6	0	0
4	a	73	2	2	73	0	0
	e	3	72	72	3	0	0
5	a	73	2	2	73	0	0
	e	3	72	72	2	0	1

C.13.10 Experimento final 13, computador 2, instancia 5

Tabla 143. Pruebas del experimento 13, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	2	2	73	0	0
	e	3	72	72	3	0	0
2	a	72	3	3	72	0	0
	e	2	73	73	2	0	0

3	a	71	4	4	71	0	0
	e	8	66	66	8	1	1
4	a	72	3	3	72	0	0
	e	3	72	71	3	1	0
5	a	72	3	3	72	0	0
	e	3	71	71	3	1	1

C.14 Experimento final 14: /a/ y /e/ cepstro

Cuadro 76. Experimento 14, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro		Número de frases	
		Coefficientes	del 2 al 15	entrenamiento	pruebas
/a/ y /e/	512 muestras			58	35

Cuadro 77. Experimento 14, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	54	150	27	75	uniforme	aleatorio
/e/	64	150	32	75	uniforme	aleatorio

Cuadro 78. Experimento 14, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 144. Entrenamiento del experimento 14

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	26	$2,73 \times 10^{-17}$	447,0 s	1	23	$9,21 \times 10^{-17}$	233,2 s
2	23	$9,82 \times 10^{-17}$	396,4 s	2	22	$1,41 \times 10^{-16}$	254,7 s
3	24	$3,68 \times 10^{-17}$	418,3 s	3	82	$1,20 \times 10^{-16}$	838,5 s
4	33	$1,02 \times 10^{-16}$	594,2 s	4	23	$1,88 \times 10^{-16}$	231,3 s
5	26	$6,22 \times 10^{-17}$	455,1 s	5	21	$2,10 \times 10^{-16}$	210,5 s

C.14.1 Experimento final 14, computador 1, instancia 1

Tabla 145. Pruebas del experimento 14, instancia 1, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\xi a?$	$\xi e?$
1	a	74	1	1	74	0	0
	e	4	70	70	4	1	1
2	a	73	1	1	73	1	1
	e	2	71	71	2	2	2
3	a	74	0	0	75	1	0
	e	2	72	73	2	0	1
4	a	74	1	1	74	0	0

	e	1	72	72	1	2	2
5	a	74	0	0	74	1	1
	e	3	72	72	3	0	0

C.14.2 Experimento final 14, computador 1, instancia 2

Tabla 146. Pruebas del experimento 14, instancia 2, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	75	0	0	75	0	0
	e	4	70	70	4	1	1
2	a	75	0	0	75	0	0
	e	7	68	68	7	0	0
3	a	74	1	1	74	0	0
	e	3	72	71	3	1	0
4	a	74	1	1	74	0	0
	e	1	74	74	1	0	0
5	a	74	1	1	74	0	0
	e	3	72	72	3	0	0

C.14.3 Experimento final 14, computador 1, instancia 3

Tabla 147. Pruebas del experimento 14, instancia 3, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	72	2	2	72	1	1
	e	1	74	74	1	0	0
2	a	74	1	1	74	0	0
	e	3	72	72	3	0	0
3	a	74	1	1	74	0	0
	e	1	74	74	1	0	0
4	a	74	1	1	74	0	0
	e	0	75	75	0	0	0
5	a	70	5	5	70	0	0
	e	1	74	74	1	0	0

C.14.4 Experimento final 14, computador 1, instancia 4

Tabla 148. Pruebas del experimento 14, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	2	2	72	0	1
	e	4	71	71	4	0	0
2	a	73	1	1	73	1	1
	e	5	70	70	5	0	0
3	a	75	0	0	75	0	0

	e	4	71	71	4	0	0
4	a	74	1	1	74	0	0
	e	3	72	72	3	0	0
5	a	74	1	1	74	0	0
	e	4	71	71	4	0	0

C.14.5 Experimento final 14, computador 1, instancia 5

Tabla 149. Pruebas del experimento 14, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	2	2	73	0	0
	e	2	73	73	2	0	0
2	a	73	2	2	73	0	0
	e	1	74	74	1	0	0
3	a	74	1	1	74	0	0
	e	1	74	74	1	0	0
4	a	74	1	1	74	0	0
	e	0	75	75	0	0	0
5	a	72	1	1	72	2	2
	e	2	73	73	2	0	0

C.14.6 Experimento final 14, computador 2, instancia 1

Tabla 150. Pruebas del experimento 14, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	1	1	74	0	0
	e	5	70	70	5	0	0
2	a	73	2	2	73	0	0
	e	6	69	69	6	0	0
3	a	75	0	0	75	0	0
	e	4	71	71	4	0	0
4	a	73	2	2	73	0	0
	e	2	72	72	2	1	1
5	a	73	2	2	73	0	0
	e	2	72	72	3	1	0

C.14.7 Experimento final 14, computador 2, instancia 2

Tabla 151. Pruebas del experimento 14, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	1	1	74	0	0
	e	5	70	70	5	0	0
2	a	73	2	2	73	0	0

	e	5	70	70	5	0	0
3	a	75	0	0	75	0	0
	e	4	71	71	4	0	0
4	a	73	2	2	73	0	0
	e	1	74	74	1	0	0
5	a	74	1	1	74	0	0
	e	3	72	72	3	0	0

C.14.8 Experimento final 14, computador 2, instancia 3

Tabla 152. Pruebas del experimento 14, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	70	5	5	70	0	0
	e	1	74	74	1	0	0
2	a	71	4	4	71	0	0
	e	4	71	71	4	0	0
3	a	74	1	1	74	0	0
	e	0	74	75	1	0	0
4	a	74	1	1	74	0	0
	e	0	75	75	0	0	0
5	a	71	4	4	71	0	0
	e	0	73	75	2	0	0

C.14.9 Experimento final 14, computador 2, instancia 4

Tabla 153. Pruebas del experimento 14, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	72	3	2	72	1	0
	e	5	70	70	5	0	0
2	a	74	1	1	74	0	0
	e	4	71	71	4	0	0
3	a	75	0	0	75	0	0
	e	3	72	71	3	1	0
4	a	74	1	1	74	0	0
	e	2	73	72	2	1	0
5	a	75	0	0	75	0	0
	e	2	73	73	2	0	0

C.14.10 Experimento final 14, computador 2, instancia 5

Tabla 154. Pruebas del experimento 14, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	1	1	74	0	0

	e	2	73	73	2	0	0
2	a	74	1	1	74	0	0
	e	3	72	72	3	0	0
3	a	74	1	1	74	0	0
	e	2	73	73	2	0	0
4	a	73	2	2	73	0	0
	e	0	75	75	0	0	0
5	a	73	2	2	73	0	0
	e	3	72	72	3	0	0

C.15 Experimento final 15: /a/ y /e/ cepstro

Cuadro 79. Experimento 15, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro		Número de frases	
		Coeficientes	entrenamiento	pruebas	
/a/ y /e/	512 muestras	del 2 al 15	58	35	

Cuadro 80. Experimento 5, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	54	150	27	75	aleatorio	binario
/e/	64	150	32	75	aleatorio	binario

Cuadro 81. Experimento 5, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 155. Entrenamiento del experimento 15

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	27	$3,85 \times 10^{-16}$	602,1 s	1	100	$7,33 \times 10^{-02}$	1089,3 s
2	23	$2,22 \times 10^{-16}$	391,2 s	2	21	$1,11 \times 10^{-16}$	248,8 s
3	100	$3,33 \times 10^{-03}$	1886,8 s	3	100	$1,33 \times 10^{-02}$	1091,4 s
4	22	$1,97 \times 10^{-16}$	380,5 s	4	46	$8,03 \times 10^{-16}$	488,9 s
5	24	$1,50 \times 10^{-16}$	411,5 s	5	92	$6,67 \times 10^{-03}$	988,7 s

C.15.1 Experimento final 15, computador 1, instancia 1

Tabla 156. Pruebas del experimento 15, instancia 1, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	69	6	6	69	0	0
	e	3	72	72	3	0	0
2	a	68	7	7	68	0	0
	e	5	70	70	5	0	0

3	a	70	5	5	70	0	0
	e	3	71	71	3	1	1
4	a	73	2	2	73	0	0
	e	2	73	73	2	0	0
5	a	72	3	3	72	0	0
	e	4	71	71	4	0	0

C.15.2 Experimento final 15, computador 1, instancia 2

Tabla 157. Pruebas del experimento 15, instancia 2, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	70	5	5	70	0	0
	e	4	71	71	4	0	0
2	a	72	2	3	72	0	1
	e	7	68	68	7	0	0
3	a	70	4	4	70	1	1
	e	5	70	70	5	0	0
4	a	74	1	1	74	0	0
	e	3	72	72	3	0	0
5	a	73	2	2	73	0	0
	e	4	70	70	4	1	1

C.15.3 Experimento final 15, computador 1, instancia 3

Tabla 158. Pruebas del experimento 15, instancia 3, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	69	6	6	69	0	0
	e	3	70	72	5	0	0
2	a	67	8	8	67	0	0
	e	4	70	71	5	0	0
3	a	70	5	5	70	0	0
	e	5	70	70	5	0	0
4	a	73	2	2	73	0	0
	e	1	73	74	2	0	0
5	a	71	4	4	71	0	0
	e	3	69	71	5	1	1

C.15.4 Experimento final 15, computador 1, instancia 4

Tabla 159. Pruebas del experimento 15, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	1	1	74	0	0
	e	3	72	72	3	0	0

2	a	75	0	0	75	0	0
	e	3	72	72	3	0	0
3	a	72	2	2	72	1	1
	e	2	72	72	2	1	1
4	a	75	0	0	75	0	0
	e	2	73	73	2	0	0
5	a	74	1	1	74	0	0
	e	2	72	72	2	1	1

C.15.5 Experimento final 15, computador 1, instancia 5

Tabla 160. Pruebas del experimento 5, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	69	6	6	69	0	0
	e	3	72	72	3	0	0
2	a	67	8	8	67	0	0
	e	7	68	68	7	0	0
3	a	70	5	5	70	0	0
	e	6	69	69	6	0	0
4	a	73	2	2	73	0	0
	e	3	72	72	3	0	0
5	a	72	3	3	72	0	0
	e	4	71	71	4	0	0

C.15.6 Experimento final 15, computador 2, instancia 1

Tabla 161. Pruebas del experimento 15, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	1	1	74	0	0
	e	11	68	63	7	1	0
2	a	74	1	1	74	0	0
	e	9	70	66	5	0	0
3	a	75	0	0	75	0	0
	e	8	70	67	5	0	0
4	a	75	2	0	73	0	0
	e	4	72	71	3	0	0
5	a	75	0	0	75	0	0
	e	7	71	68	4	0	0

C.15.7 Experimento final 15, computador 2, instancia 2

Tabla 162. Pruebas del experimento 15, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$

1	a	70	5	5	70	0	0
	e	4	70	70	4	1	1
2	a	69	6	6	69	0	0
	e	5	70	70	5	0	0
3	a	70	5	5	70	0	0
	e	3	72	72	3	0	0
4	a	73	2	2	73	0	0
	e	3	72	72	3	0	0
5	a	72	3	3	72	0	0
	e	6	69	69	6	0	0

C.15.8 Experimento final 15, computador 2, instancia 3

Tabla 163. Pruebas del experimento 15, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	69	6	6	69	0	0
	e	0	73	75	2	0	0
2	a	71	4	4	71	0	0
	e	2	71	73	4	0	0
3	a	71	4	4	71	0	0
	e	1	72	74	3	0	0
4	a	72	3	3	72	0	0
	e	0	70	75	5	0	0
5	a	73	2	2	73	0	0
	e	2	71	73	4	0	0

C.15.9 Experimento final 15, computador 2, instancia 4

Tabla 164. Pruebas del experimento 15, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	69	5	5	70	1	0
	e	3	71	72	4	0	0
2	a	68	7	7	68	0	0
	e	3	71	72	3	0	1
3	a	70	5	5	70	0	0
	e	2	73	73	2	0	0
4	a	72	2	3	73	0	0
	e	2	73	73	2	0	0
5	a	70	3	4	72	1	0
	e	2	73	73	2	0	0

C.15.10 Experimento final 15, computador 2, instancia 5

Tabla 165. Pruebas del experimento 15, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	2	2	73	0	0
	e	6	69	69	6	0	0
2	a	73	2	2	73	0	0
	e	7	68	68	7	0	0
3	a	72	3	3	72	0	0
	e	8	67	67	8	0	0
4	a	75	0	0	75	0	0
	e	5	70	70	5	0	0
5	a	74	1	1	74	0	0
	e	6	69	69	6	0	0

C.16 Experimento final 16: /a/ y /e/ cepstro

Cuadro 82. Experimento 13, descripción del experimento

Fonemas	Tamaño de la ventana	Cepstro	Número de frases	
		Coefficientes	entrenamiento	pruebas
/a/ y /e/	512 muestras	del 2 al 15	58	35

Cuadro 83. Experimento 13, selección y ajuste de ventanas

Fonemas	Cantidad para entrenar		Cantidad para probar		Selección de fonema	Ajuste de ventana
	fonemas en frases	ventanas tomadas	fonemas en frases	ventanas tomadas		
/a/	54	150	27	75	uniforme	binario
/e/	64	150	32	75	uniforme	binario

Cuadro 84. Experimento 13, características de la red

Tipo de red	Número de neuronas			Función de adaptación			Algoritmo de entrenamiento
	nivel 1	nivel 2	nivel 3	nivel 1	nivel 2	nivel 3	
perceptrón multinivel <i>feedforward</i>	10	15	3	<i>tansig</i>	<i>tansig</i>	<i>logsig</i>	<i>Levenberg-Marquardt</i>

Tabla 166. Entrenamiento del experimento 10

Instancia	Computador 1			Instancia	Computador 2		
	épocas	MSE final	tiempo		épocas	MSE final	tiempo
1	29	$2,50 \times 10^{-01}$	480,8 s	1	19	$1,38 \times 10^{-16}$	186,0 s
2	35	$2,00 \times 10^{-16}$	629,7 s	2	27	$8,72 \times 10^{-17}$	269,2 s
3	25	$2,50 \times 10^{-01}$	433,9 s	3	24	$1,03 \times 10^{-16}$	240,7 s
4	22	$2,40 \times 10^{-16}$	367,0 s	4	27	$1,18 \times 10^{-17}$	275,8 s
5	100	$3,33 \times 10^{-03}$	1879,8 s	5	25	$5,65 \times 10^{-17}$	240,9 s

C.16.1 Experimento final 16, computador 1, instancia 1

Tabla 167. Pruebas del experimento 16, instancia 1, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	71	75	4	0	0	0
	e	5	75	70	0	0	0
2	a	68	75	7	0	0	0
	e	5	75	68	0	2	0
3	a	72	75	3	0	0	0
	e	6	75	69	0	0	0
4	a	73	75	1	0	1	0
	e	5	75	70	0	0	0
5	a	73	75	2	0	0	0
	e	4	75	70	0	1	0

C.16.2 Experimento final 16, computador 1, instancia 2

Tabla 168. Pruebas del experimento 16, instancia 2, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	71	4	4	71	0	0
	e	8	67	67	8	0	0
2	a	68	7	7	67	0	1
	e	7	68	68	7	0	0
3	a	72	3	3	72	0	0
	e	6	69	69	6	0	0
4	a	73	3	2	72	0	0
	e	4	71	71	4	0	0
5	a	73	2	2	73	0	0
	e	4	71	71	4	0	0

C.16.3 Experimento final 16, computador 1, instancia 3

Tabla 169. Pruebas del experimento 16, instancia 3, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	75	4	0	71	0	0
	e	75	69	0	6	0	0
2	a	75	7	0	68	0	0
	e	75	73	0	2	0	0
3	a	75	3	0	72	0	0
	e	75	73	0	2	0	0
4	a	75	2	0	73	0	0
	e	75	74	0	1	0	0

5	a	75	2	0	73	0	0
	e	75	74	0	1	0	0

C.16.4 Experimento final 16, computador 1, instancia 4

Tabla 170. Pruebas del experimento 16, instancia 4, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	71	4	4	71	0	0
	e	6	69	69	6	0	0
2	a	69	6	6	69	0	0
	e	2	73	73	2	0	0
3	a	73	2	2	72	0	1
	e	2	73	73	2	0	0
4	a	74	1	1	74	0	0
	e	1	74	74	1	0	0
5	a	74	1	1	74	0	0
	e	1	74	74	1	0	0

C.16.5 Experimento final 16, computador 1, instancia 5

Tabla 171. Pruebas del experimento 16, instancia 5, computador 1

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	71	4	4	71	0	0
	e	8	67	67	8	0	0
2	a	69	6	6	69	0	0
	e	5	70	70	3	0	2
3	a	73	2	2	73	0	0
	e	7	68	68	7	0	0
4	a	74	1	1	74	0	0
	e	3	72	72	2	0	1
5	a	74	1	1	74	0	0
	e	5	70	70	5	0	0

C.16.6 Experimento final 16, computador 2, instancia 1

Tabla 172. Pruebas del experimento 16, instancia 1, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	71	4	4	71	0	0
	e	9	66	66	9	0	0
2	a	68	7	7	68	0	0
	e	7	68	68	7	0	0
3	a	72	3	3	72	0	0
	e	6	68	68	6	1	1

4	a	74	1	1	74	0	0
	e	4	71	71	4	0	0
5	a	73	2	2	73	0	0
	e	4	71	71	4	0	0

C.16.7 Experimento final 16, computador 2, instancia 2

Tabla 173. Pruebas del experimento 16, instancia 2, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	73	1	1	73	1	1
	e	4	71	71	4	0	0
2	a	70	4	4	69	1	2
	e	2	73	73	2	0	0
3	a	73	2	2	73	0	0
	e	2	73	73	2	0	0
4	a	73	1	1	73	1	1
	e	0	75	75	0	0	0
5	a	73	2	2	73	0	0
	e	1	74	74	1	0	0

C.16.8 Experimento final 16, computador 2, instancia 3

Tabla 174. Pruebas del experimento 16, instancia 3, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	72	3	3	72	0	0
	e	7	68	68	7	0	0
2	a	68	5	5	68	2	2
	e	3	72	72	3	0	0
3	a	73	2	2	73	0	0
	e	2	73	73	2	0	0
4	a	74	1	1	74	0	0
	e	2	72	72	2	1	1
5	a	74	1	1	74	0	0
	e	1	73	73	1	1	1

C.16.9 Experimento final 16, computador 2, instancia 4

Tabla 175. Pruebas del experimento 16, instancia 4, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	1	1	74	0	0
	e	2	73	73	2	0	0
2	a	72	3	3	72	0	0
	e	0	75	75	0	0	0

3	a	75	0	0	75	0	0
	e	0	75	75	0	0	0
4	a	74	1	1	74	0	0
	e	1	74	74	1	0	0
5	a	74	1	1	74	0	0
	e	0	75	75	0	0	0

C.16.10 Experimento final 16, computador 2, instancia 5

Tabla 176. Pruebas del experimento 16, instancia 5, computador 2

Prueba	Entrada	Aceptación		Rechazo		Indecisión	
		a	e	$\neg a$	$\neg e$	$\zeta a?$	$\zeta e?$
1	a	74	2	1	73	0	0
	e	5	69	70	5	0	1
2	a	71	4	4	71	0	0
	e	6	68	69	6	0	1
3	a	73	3	2	72	0	0
	e	4	70	70	5	1	0
4	a	74	2	1	73	0	0
	e	4	71	71	4	0	0
5	a	73	2	2	73	0	0
	e	5	70	70	5	0	0

NEXO D. PROGRAMAS PARA EXTRACCIÓN DE LA BASE DE DATOS

D.1 Programa phons para extracción de difonemas

```

/*****
 * difon.c
 * programa de conversión fonética y
 * extracción de difonemas de una lista
 * de palabras
 *****/

#include <stdio.h>
#include <string.h>

char conv[256] = {
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0xA, 0, 0, 0xD, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 'a', 'B', 'C', 'D', 'e', 'f', 'G',
    'H', 'I', 'x', 'k', 'L', 'm', 'N', 'o',
    'p', 'k', 'R', 's', 't', 'U', 'B', 'W',
    'X', 'Y', 's', 0, 0, 0, 0, 0,

    0, 'a', 'B', 'C', 'D', 'e', 'f', 'G',
    'H', 'I', 'x', 'k', 'L', 'm', 'N', 'o',
    'p', 'k', 'R', 's', 't', 'U', 'B', 'W',
    'X', 'Y', 's', 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a',
    'e', 'e', 'e', 'e', 'i', 'i', 'i', 'i',
    'D', 'J', 'o', 'o', 'o', 'o', 'o', 0,
    'o', 'u', 'u', 'u', 'w', 'i', 'f', 's',

    'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a',
    'e', 'e', 'e', 'e', 'i', 'i', 'i', 'i',
    'D', 'J', 'o', 'o', 'o', 'o', 'o', 0,
    'o', 'u', 'u', 'u', 'w', 'i', 'f', 'Y'
};

```

```

#define Conv(x) conv[(x)&0xff]

int main (int argc, char* argv[])
{
    FILE *ifp, *ofp;
    char linea[16384],c,d;
    int f,i;

    if((ofp=(argc<3)?stdout:fopen(argv[2],"wt"))==NULL)
    {
        fprintf(stderr,"%s error: archivo de salida %s no puede abrirse\n", argv[0],
argv[2]);
        exit(1);
    }

    if((ifp=(argc<2)?stdin:fopen(argv[1],"rt"))==NULL)
    {
        fprintf(stderr,"%s error: archivo de entrada %s no puede abrirse\n", argv[0],
argv[1]);
        exit(1);
    }

    while((fgets(linea,16284,ifp)!=NULL)
    {
        f=1;
        for(i=0;linea[i];i++)
        {
            c=conv[linea[i]&0xff];
            switch(c)
            {
                case 'B': case 'D':
                    d=c+0x20;
                    break;
                case 'C':
                    if(strchr("Iei",Conv(linea[i+1])) && Conv(linea[i+1]))
                        d='s';
                    else if (linea[i+1]=='h' || linea[i+1]=='H')
                        d='C';
                    else
                        d='k';
                    break;
                case 'G':
                    if(strchr("Iei",Conv(linea[i+1])) && Conv(linea[i+1]))
                        d='x';
                    else
                        d='g';
                    break;
                case 'H':
                    d=' ';
                    break;
                case 'I':
                    if (strchr("AEOUaeou",Conv(linea[i+1])) && Conv(linea[i+1]))
                        d='j';
                    else if (i && strchr("AEOaeo",Conv(linea[i-1])) && Conv(linea[i-1]))
                        d='j';
                    else
                        d='i';
                    break;
                case 'L':
                    if (strchr("Ll",linea[i+1]))
                        { d='y'; i++; }
                    else
                        d='l';
                    break;
                case 'N':
                    if (strchr("Bp",Conv(linea[i+1])) && Conv(linea[i+1]))
                        d='m';
                    else if (strchr("Gkx",Conv(linea[i+1])) && Conv(linea[i+1]))
                        d='N';
            }
        }
    }
}

```

```

        else
            d='n';
            break;
    case 'R':
        if (strchr("Rr",linea[i+1]))
            { d='r'; i++; }
        else if (i && strchr("BCDGIUaeikoptuw",Conv(linea[i-1])) && Conv(linea[i-1])
            && strchr("IUaeiouw",Conv(linea[i+1])) && Conv(linea[i+1]))
            d='4';
        else
            d='r';
            break;
    case 'U':
        if (!i || strchr("Qq",linea[i-1]))
            d=' ';
        else if (i && strchr("Gg",linea[i-1]) && strchr("Iei",Conv(linea[i+1])))
            d=' ';
        else if (strchr("AEIOaeio",Conv(linea[i+1])) && Conv(linea[i+1]))
            d='w';
        else if (i && strchr("AEOaeo",Conv(linea[i-1])) && Conv(linea[i-1]))
            d='w';
        else
            d='u';
            break;
    case 'W':
        if (!i || strchr("Nm",Conv(linea[i-1])))
            putc('g',ofp);
        else if (!strchr("IUaeiou",Conv(linea[i+1])))
            putc('G',ofp);
            d='w';
            break;
    case 'X':
        if (Conv(linea[i+1]))
            putc('k',ofp);
            d='s';
            break;
    case 'Y':
        if (strchr("AEOUaeou",Conv(linea[i+1])) && Conv(linea[i+1]))
            d='y';
        else if (i && strchr("AEOaeo",Conv(linea[i-1])) && Conv(linea[i-1]))
            d='j';
        else
            d='i';
            break;
    default:
        d=c;
    }
    if(d==0)
    {
        if(f) putc(' ',ofp);
        f=0;
    }
    else
    {
        f=1;
        if(d!=' ') putc(d,ofp);
    }
}
}

fclose(ofp);

fclose(ifp);

return 0;
}

```

D.2 Programa di fon para extracción de difonemas

```

/*****
 * difon.c
 * programa de conversión fonética y
 * extracción de difonemas de una lista
 * de palabras
 *****/

#include <stdio.h>
#include <string.h>

char conv[256] = {
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0xA, 0, 0, 0xD, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    ' ', 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    '0', '1', '2', '3', '4', '5', '6', '7',
    '8', '9', 0, 0, 0, 0, 0, 0,

    0, 'a', 'b', 'c', 'd', 'e', 'f', 'g',
    'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
    'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
    'x', 'y', 'z', 0, 0, 0, 0, 0,

    0, 'a', 'b', 'c', 'd', 'e', 'f', 'g',
    'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
    'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
    'x', 'y', 'z', 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    'a', 'á', 'a', 'a', 'a', 'a', 'æ', 'c',
    'e', 'é', 'e', 'e', 'i', 'í', 'i', 'i',
    'd', 'ñ', 'o', 'ó', 'o', 'o', 'o', 0,
    'o', 'u', 'ú', 'u', 'ü', 'y', 'þ', 'ÿ',

    'a', 'á', 'a', 'a', 'a', 'a', 'æ', 'c',
    'e', 'é', 'e', 'e', 'i', 'í', 'i', 'i',
    'd', 'ñ', 'o', 'ó', 'o', 'o', 'o', 0,
    'o', 'u', 'ú', 'u', 'ü', 'y', 'þ', 'ÿ'
};

typedef struct _nodopal
{
    char* w;
    unsigned long q;
    struct _nodopal *n;
} nodopal;

char* addword(char*, nodopal**);
char* noword(char*);

int main (int argc, char* argv[])
{
    FILE *ifp, *ofp;
    nodopal *L=NULL, *n;
    char linea[16384], *s, *t;

```



```

int f;

if (argc != 3)
{
    fprintf(stderr,"%s: número incorrecto de argumentos\n", argv[0]);
    fprintf(stderr,"Uso:\n\t%s <archivo de entrada> <archivo de salida>\n", argv[0]);
    exit(1);
}

if((ifp=fopen(argv[1],"rt"))==NULL)
{
    fprintf(stderr,"%s error: archivo de entrada %s no puede abrirse\n", argv[0],
argv[1]);
    exit(1);
}

while((fgets(linea,16284,ifp))!=NULL)
{
    for(s=noword(linea);s;=noword(s))
        s = addword(s,&L);
}

fclose(ifp);

if((ofp=(argc<3)?stdout:fopen(argv[2],"wt"))==NULL)
{
    fprintf(stderr,"%s error: archivo de salida %s no puede abrirse\n", argv[0],
argv[2]);
    exit(1);
}

for (n=L;n!=NULL;n=n->n)
{
    fprintf(ofp,"%s,%d\n",n->w,n->q);
}

fclose(ofp);

return 0;
}

#define ISALPHA(c) (conv[(c)&0xff]&0xc0)

char* addword(char*s,nodopal**pL)
{
    char *t,pal[40];
    nodopal *n,*m;

    for(t=pal;ISALPHA(*s);*t++=conv[( *s++)&0xff]);
    *t=0;
    for(n=*pL,m=NULL;n!=NULL;n=n->n)
    {
        if(strcmp(pal,n->w)>=0)
            m=n;
        else
            break;
    }
    if(m==NULL)
    {
        n=(nodopal*)malloc(sizeof(nodopal));
        n->w = strdup(pal);
        n->q = 1;
        n->n = *pL;
        *pL = n;
    }
    /*printf("%s", */
    }
    else if(strcmp(pal,m->w)==0)
    {
        m->q++;
    }
}

```

```

    }
    else
    {
        n=(nodopal*)malloc(sizeof(nodopal));
        n->w = strdup(pal);
        n->q = 1;
        n->n = m->n;
        m->n = n;
    }

    return s;
}

char* noword(char* s)
{
    while(*s && !ISALPHA(*s)) s++;
    return *s? s: NULL;
}

```

D.3 Programa di phon para extracción de difonemas

```

#include <stdio.h>

char conv[256] = {
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    '0', '1', '2', '3', '4', '5', '6', '7',
    '8', '9', 0, 0, 0, 0, 0, 0,

    '@', 'A', 'B', 'C', 'D', 'E', 'F', 'G',
    'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O',
    'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W',
    'X', 'Y', 'Z', 0, 0, 0, 0, 0,

    0, 'a', 'b', 'c', 'd', 'e', 'f', 'g',
    'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
    'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
    'x', 'y', 'z', 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0
};

typedef struct _lnode

```

```

{
    unsigned char a, b;
    int c;
    struct _lnode *n;
} lnode;

int add (lnode*, unsigned char, unsigned char);

int main (int argc, char* argv[])
{
    FILE *ifp, *ofp;
    unsigned char c, bc;
    lnode L={0,0,NULL}, *p;

    if((ifp=(argc<2)?stdin:fopen(argv[1], "rt"))==NULL)
    {
        fprintf(stderr, "%s error: archivo de entrada %s no puede abrirse\n", argv[0],
argv[1]);
        exit(1);
    }

    for(bc=0; !feof(ifp); bc=c)
    {
        c=getc(ifp);
        if(conv[c] && conv[bc]) add(&L, conv[bc], conv[c]);
    }

    fclose(ifp);

    if((ofp=(argc<3)?stdout:fopen(argv[2], "wt"))==NULL)
    {
        fprintf(stderr, "%s error: archivo de entrada %s no puede abrirse\n", argv[0],
argv[2]);
        exit(1);
    }

    fprintf(ofp, "hex,di,count\n");
    for(p=L.n; p; p=p->n)
        fprintf(ofp, "[%02X;%02X],%c%c,%d\n", p->a, p->b, p->a, p->b, p->c);

    fclose(ofp);

    return 0;
}

int add (lnode* pL, unsigned char a, unsigned char b)
{
    lnode *p, *s;
    for(p=pL; p; p=p->n)
        if(a>p->a || (a==p->a && b>p->b)) s=p;
        else break;
    if(s->a==a && s->b==b)
    {
        s->c++;
    }
    else
    {
        p=(lnode*)malloc(sizeof(lnode));
        p->a=a;
        p->b=b;
        p->c=1;
        p->n=s->n;
        s->n=p;
    }
}

```

D.4 Programa di graph para extracción de digrafos

```

#include <stdio.h>

char conv[256] = {
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 'a','b','c','d','e','f','g',
    'h','i','j','k','l','m','n','o',
    'p','q','r','s','t','u','v','w',
    'x','y','z', 0, 0, 0, 0, 0,

    0, 'a','b','c','d','e','f','g',
    'h','i','j','k','l','m','n','o',
    'p','q','r','s','t','u','v','w',
    'x','y','z', 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    'á','á','a','a','a','a','æ','c',
    'é','é','e','e','i','í','i','i',
    'd','ñ','ó','ó','o','o','o', 0,
    'o','ú','ú','u','ü','y','þ','ÿ',

    'a','á','a','a','a','a','æ','c',
    'e','é','e','e','i','í','i','i',
    'd','ñ','o','ó','o','o','o', 0,
    'o','u','ú','u','ü','y','þ','y'
};

typedef struct _lnode
{
    unsigned char a, b;
    int c;
    struct _lnode *n;
} lnode;

int add (lnode*,unsigned char,unsigned char);

int main (int argc,char* argv[])
{
    FILE *ifp, *ofp;
    unsigned char c, bc;
    lnode L={0,0,NULL},*p;

    if((ifp=(argc<2)?stdin:fopen(argv[1],"rt"))==NULL)
    {
        fprintf(stderr,"%s error: archivo de entrada %s no puede abrirse\n", argv[0],
        argv[1]);
        exit(1);
    }

    for(bc=0; !feof(ifp); bc=c)

```

```

    {
        c=getc(ifp);
        if(conv[c] && conv[bc]) add(&L,conv[bc],conv[c]);
    }

fclose(ifp);

if((ofp=(argc<3)?stdout:fopen(argv[2],"wt"))==NULL)
    {
        fprintf(stderr,"%s error: archivo de entrada %s no puede abrirse\n", argv[0],
argv[2]);
        exit(1);
    }

fprintf(ofp,"hex,di,count\n");
for(p=L.n; p; p=p->n)
    fprintf(ofp,"[%02X;%02X],%c%c,%d\n",p->a,p->b,p->a,p->b,p->c);

fclose(ofp);

return 0;
}

int add (lnode* pL,unsigned char a,unsigned char b)
{
    lnode *p,*s;
    for(p=pL;p;p=p->n)
        if(a>p->a || (a==p->a && b>p->b)) s=p;
        else break;
    if(s->a==a && s->b==b)
        {
            s->c++;
        }
    else
        {
            p=(lnode*)malloc(sizeof(lnode));
            p->a=a;
            p->b=b;
            p->c=1;
            p->n=s->n;
            s->n=p;
        }
}

```

D.5 Programa frases para extracción de frases de hasta cuatro palabras

```

/*****
*****/

#include <stdio.h>
#include <string.h>

char conv[256] = {
    0, 0, 0, 0, 0, 0, 0, 0,
    0, ' ',0xA, 0, 0, 0xD, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,

    ' ','.',',',0, 0, 0, 0, '.',
    ' ','-',',',',',',',',
    '0','1','2','3','4','5','6','7',
    '8','9',',',',',',',0, 0, 0, '.',

    0, 'a','b','c','d','e','f','g',
    'h','i','j','k','l','m','n','o',
    'p','q','r','s','t','u','v','w',
    'x','y','z', 0, 0, 0, 0, 0,

```

```

    0, 'a', 'b', 'c', 'd', 'e', 'f', 'g',
    'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
    'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
    'x', 'y', 'z', 0, 0, 0, 0, 0,

    0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0,

    ' ', '.', 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, '.',

    'a', 'á', 'a', 'a', 'a', 'a', 'æ', 'c',
    'e', 'é', 'e', 'e', 'i', 'í', 'i', 'i',
    'd', 'ñ', 'o', 'ó', 'o', 'o', 'o', 0,
    'o', 'u', 'ú', 'u', 'ü', 'y', 'þ', 'ß',

    'a', 'á', 'a', 'a', 'a', 'a', 'æ', 'c',
    'e', 'é', 'e', 'e', 'i', 'í', 'i', 'i',
    'd', 'ñ', 'o', 'ó', 'o', 'o', 'o', 0,
    'o', 'u', 'ú', 'u', 'ü', 'y', 'þ', 'y'
};

#define Conv(c) conv[(c)&0xff]

char* noword(char*);
char* siword(char*, char*);
int doword(char*, FILE*);

int main (int argc, char* argv[])
{
    FILE *ifp, *ofp;
    char linea[16384], *s, *t, pal[256];
    int f;

    if((ifp=(argc<2)?stdin:fopen(argv[1], "rt"))==NULL)
    {
        fprintf(stderr, "%s error: archivo de entrada %s no puede abrirse\n", argv[0],
        argv[1]);
        exit(1);
    }

    if((ofp=(argc<3)?stdout:fopen(argv[2], "wt"))==NULL)
    {
        fprintf(stderr, "%s error: archivo de salida %s no puede abrirse\n", argv[0],
        argv[2]);
        exit(1);
    }

    while((fgets(linea, 16384, ifp))!=NULL)
    {
        for(s=noword(linea); s;s=noword(s))
        {
            s=siword(pal, s);
            doword(pal, ofp);
        }
    }

    fclose(ofp);

    fclose(ifp);

    return 0;
}

#define ISALPHA(c) (!strchr(" \n\r", conv[(c)&0xff]))

```

```

char* noword(char* s)
{
    if(!s) return NULL;
    while(*s && !ISALPHA(*s)) s++;
    return *s? s: NULL;
}

char* siword(char* p, char* s)
{
    while(*s && ISALPHA(*s)) *p++=*s++;
    *p=0;
    return *s? s: NULL;
}

int doword(char* p, FILE* fp)
{
    static char ps[4][256];
    static int n=0;
    int i;
    while(*p)
    {
        if(Conv(*p)=='.')
        {
            if(n<4)
                fprintf(fp,n==0?"\n":n==1?"%s\n":n==2?"%s %s\n":"%s %s
%s\n",ps[0],ps[1],ps[2]);
            n=0;
            while(Conv(*p)=='.') p++;
        }
        if(*p && Conv(*p)!='.')
        {
            if(n>=4)
            {
                fprintf(fp,"%s %s %s %s\n",ps[0],ps[1],ps[2],ps[3]);
                strcpy(ps[0],ps[1]);
                strcpy(ps[1],ps[2]);
                strcpy(ps[2],ps[3]);
                for(i=0;*p && Conv(*p)!='.';i++) ps[3][i]=Conv(*p++);
                ps[3][i]=0;
            }
            else
            {
                for(i=0;*p && Conv(*p)!='.';i++) ps[n][i]=Conv(*p++);
                ps[n][i]=0;
                n++;
            }
        }
    }
    return 0;
}

```

NEXO E. PROGRAMAS PARA MANIPULACIÓN DE ARCHIVOS . WAV

E.1 Programa VIEWWAV para ver la estructura de un archivo . wav

```

#include <stdio.h>
#include <stdlib.h>

typedef
struct _riffchunk
{
    char riff[4];
    long int paksize;
    char wave[4];
} riffchunk;

typedef
struct _formatchunk
{
    char fmt[4];
    long int fmtlen;
    short int _0x01;
    short int channels;
    long int sampps;
    long int bps;
    short int bpsamp;
    short int bitps;
} formatchunk;

typedef
struct _datachunk
{
    char data[4];
    long int datasize;
} datachunk;

int main (int argc, char* argv[])
{
    FILE *fp;
    riffchunk r;
    formatchunk f;
    datachunk d;
    char s[5];

    if ( argc == 2 )
    {
        if ( (fp=fopen(argv[1],"rb")) == NULL )
        {
            fprintf ( stderr, "Error: Archivo %s no pude ser abierto.\n" , argv[1] );
            exit(1);
        }
    }
    else
    {
        fprintf ( stderr, "Uso: viewwav file.wav.\n" );
        exit(0);
    }
}

```



```

    }

    fread(&r,sizeof(r),1,fp);
    fread(&f,sizeof(f),1,fp);
    fread(&d,sizeof(d),1,fp);

    fclose(fp);

    printf( "cadena: %s\n", strncpy(s,r.riff,4) );
    printf( "tamaño: %ld\n", r.paksize );
    printf( "cadena: %s\n", strncpy(s,r.wave,4) );
    printf( "cadena: %s\n", strncpy(s,f.fmt,4) );
    printf( "tamaño formato: %ld\n", f.fmtlen );
    printf( "dato reservado: %hd\n", f._0x01 );
    printf( "No. canales: %hd\n", f.channels );
    printf( "muestras p seg: %ld\n", f.sampps );
    printf( "bytes p seg: %ld\n", f.bps );
    printf( "bytes p muestra: %hd\n", f.bpsamp );
    printf( "bits p muestra: %hd\n", f.bitps );
    printf( "cadena: %s\n", strncpy(s,d.data,4) );
    printf( "tamaño de datos: %ld\n", d.datasize );

    return 0;
}

```

E.2 Programa partir para ver la estructura de un archivo .wav

```

/*****
 * Programa partir
 * separa un archivo .wav grande con muchas frases
 * en archivos .wav individuales por frases.
 *****/
 * Autores: Carlos Andrés Moncayo Ramírez
 * Carlos Eugenio Thomspón Pinzón
 * Pontificia Universidad Javeriana, Bogotá
 * © 1999-2000
 *****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* Definición de estructuras para manejo de archivos .wav
 */
typedef /* "Pedazo" (chunk) de archivo */
struct _riffchunk
{
    char riff[4]; /* firma: "RIFF" */
    long int paksize; /* tamaño del archivo */
    char wave[4]; /* firma: "wave" */
} riffchunk;

typedef /* "Pedazo" (chunk) de formato */
struct _formatchunk
{
    char fmt[4]; /* firma: "fmt " */
    long int fmtlen; /* posición relativa del siguiente chunk */
    short int _0x01; /* reservado */
    short int channels; /* número de canales */
    long int sampps; /* número de muestras por segundo */
    long int bps; /* número de bytes por segundo */
    short int bpsamp; /* bytes por muestra */
    short int bitps; /* bits por muestra */
} formatchunk;

typedef /* "Pedazo" (chunk) de datos */
struct _datachunk

```

```

{
    char data[4];          /* firma "data" */
    long int datasize; /* posición relativa del siguiente chunk */
} datachunk;

/* Definición de estados para determinar los puntos de
 * quiebre de las frases.
 */
typedef
enum _status
{
    sample, silence, noyet, begin, ending
} status;

typedef
union _dataitem
{
    char c[16];
    short int s[8];
    long int l[4];
} dataitem;

char datablock [1000000L];
unsigned long Z=0;

int main (int argc, char* argv[])
/* Rutina principal.
 * toma argumentos usuales de C para la rutina principal
 */
{
    FILE *fp, *ofp;
    riffchunk r;
    formatchunk f;
    datachunk d;
    char s[5],filename[256];
    long int ss, nn, ii, jj, cc, d0, ds;
    status tt;
    dataitem dd;
    int dif=0, cn=0;

    if ( argc >= 2 )
        { /* intenta abrir el archivo dado en el comando */
            sprintf(filename,"%s.wav",argv[1]); /* agrega la extensión al nombre del archivo */
            if ( (fp=fopen(filename,"rb")) == NULL )
                { /* error si no se puede abrir el archivo */
                    fprintf ( stderr, "Error: Archivo %s no pude ser abierto.\n" , filename );
                    exit(1);
                }
        }
    else /* numero insuficiente de comandos */
        {
            fprintf ( stderr, "Uso: partir file.wav.\n" );
            exit(0);
        }

    fread(&r,sizeof(r),1,fp);
    fread(&f,sizeof(f),1,fp);

    if(f.fmtlen > sizeof(f)-8)
        fread(&dd,dif = (8 + f.fmtlen - sizeof(f)),1,fp);

    fread(&d,sizeof(d),1,fp);

    printf( "cadena: %s\n", strncpy(s,r.riff,4) );
    printf( "tamaño: %ld\n", r.paksize );
    printf( "cadena: %s\n", strncpy(s,r.wave,4) );
    printf( "cadena: %s\n", strncpy(s,f.fmt,4) );
    printf( "tamaño formato: %ld\n", f.fmtlen );
    printf( "dato reservado: %6hd\n", f._0x01 );
    printf( "No. canales: %6hd\n", f.channels );

```

```

printf( "muestras p seg: %6ld\n", f.sampps );
printf( "bytes p seg:      %6ld\n", f.bps );
printf( "bytes p muestra:%6hd\n", f.bpsamp );
printf( "bits p muestra:  %6hd\n", f.bitps );
printf( "cadena: %s\n", strncpy(s,d.data,4) );
printf( "tamaño de datos:%6ld\n", ds=d.datasize );

ss = (long int) (((argc>=3)? atof(argv[2])      : 0.5 ) *      f.sampps);
nn = (long int) (((argc>=4)? atof(argv[3])/100 : 0.01 ) * (1L << f.bitps));
tt = begin;
cc = 0;

if ( f.bitps<=8 )
{
  for(ii=0; ii<ds; ii++)
  {
    if(!fread(&dd,f.bpsamp,1,fp))
    {
      fprintf(stderr,"Archivo terminado antes de tiempo\n");
      exit(1);
    }
    if(tt==begin)
    {
      d0 = dd.c[0];
      tt = noyet;
      jj=0;
    }
    else if ((tt==noyet || tt==ending) && abs(d0-dd.c[0])>nn)
    {
      tt = sample;
      d0 = dd.c[0];
      cc = 0;
    }
    else if (tt==sample && abs(d0-dd.c[0])<nn)
    {
      tt = ending;
    }
    else if (tt==sample)
    {
      d0 = dd.c[0];
    }
    else if (tt==ending)
    {
      cc ++;
    }
  }
  if (cc>=ss)
  {
    r.paksize += jj - d.datasize;
    d.datasize = jj;
    sprintf(filename,"%s_%03d.wav",argv[1],++cn);
    if ((ofp = fopen(filename,"wb")) != NULL)
    {
      fwrite(&r,sizeof(r),1,ofp);
      fwrite(&f,sizeof(f),1,ofp);
      if(dif)fwrite(&Z,dif,1,ofp);
      fwrite(&d,sizeof(d),1,ofp);
      fwrite(datablock,jj,1,ofp);
      fclose(ofp);
      printf("archivo %s hasta la posición %ld\n", filename, ii);
    }
    else
    {
      printf("archivo %s no pudo crearse\n", filename );
    }
    tt = begin;
    cc=0;
  }
  datablock[jj++] = dd.c[0];
  if ( f.channels >= 2 )
  {

```

```

        datablock[jj++] = dd.c[1];
    }
}
else if ( f.bitps<=16 )
{
    for(ii=0; ii<ds; ii+=2)
    {
        if(!fread(&dd,f.bpsamp,1,fp))
        {
            fprintf(stderr,"Archivo terminado antes de tiempo\n");
            exit(1);
        }
        ;if(tt==begin)
        {
            d0 = dd.s[0];
            tt = noyet;
            jj=0;
        }
        else if ((tt==noyet || tt==ending) && abs(d0-dd.s[0])>nn)
        {
            tt = sample;
            d0 = dd.s[0];
            cc = 0;
        }
        else if (tt==sample && abs(d0-dd.s[0])<nn)
        {
            tt = ending;
        }
        else if (tt==sample)
        {
            d0 = dd.s[0];
        }
        else if (tt==ending)
        {
            cc ++;
        }
        if (tt!=noyet)
        {
            datablock[jj++] = dd.c[0];
            datablock[jj++] = dd.c[1];
            if ( f.channels >= 2 )
            {
                datablock[jj++] = dd.c[2];
                datablock[jj++] = dd.c[3];
            }
        }
    }
    if (cc>=ss)
    {
        jj -= cc;
        tt = begin;
        cc = 0;
        if (jj < 25600L)
        {
            printf("Archivo muy pequeño\n");
            continue;
        }
        r.paksize += jj - d.datasize;
        d.datasize = jj;
        sprintf(filename,"%s_%03d.wav",argv[1],++cn);
        if ((ofp = fopen(filename,"wb")) != NULL)
        {
            fwrite(&r,sizeof(r),1,ofp);
            fwrite(&f,sizeof(f),1,ofp);
            if(dif)fwrite(&Z,dif,1,ofp);
            fwrite(&d,sizeof(d),1,ofp);
            fwrite(datablock,jj,1,ofp);
            fclose(ofp);
            printf("archivo %s hasta la posición %ld\n", filename, ii);
        }
    }
}

```

```

        else
        {
            printf("archivo %s no pudo crearse\n", filename );
        }
    }
}
else if ( f.bitps<=32 )
{
    for(ii=0; ii<ds; ii+=4)
    {
        if(!fread(&dd,f.bpsamp,1,fp))
        {
            fprintf(stderr,"Archivo terminado antes de tiempo\n");
            exit(1);
        }
        if(tt==begin)
        {
            d0 = dd.l[0];
            tt = noyet;
            jj=0;
        }
        else if ((tt==noyet || tt==ending) && abs(d0-dd.l[0])>nn)
        {
            tt = sample;
            d0 = dd.l[0];
            cc = 0;
        }
        else if (tt==sample && abs(d0-dd.l[0])<nn)
        {
            tt = ending;
        }
        else if (tt==sample)
        {
            d0 = dd.l[0];
        }
        else if (tt==ending)
        {
            cc ++;
        }
        if (cc>=ss)
        {
            r.paksize += jj - d.datasize;
            d.datasize = jj;
            sprintf(filename,"%s_%03d.wav",argv[1],++cn);
            if ((ofp = fopen(filename,"wb")) != NULL)
            {
                fwrite(&r,sizeof(r),1,ofp);
                fwrite(&f,sizeof(f),1,ofp);
                if(dif)fwrite(&Z,dif,1,ofp);
                fwrite(&d,sizeof(d),1,ofp);
                fwrite(datablock,jj,1,ofp);
                fclose(ofp);
                printf("archivo %s hasta la posici n %ld\n", filename, ii);
            }
            else
            {
                printf("archivo %s no pudo crearse\n", filename );
            }
            tt = begin;
            cc=0;
        }
        datablock[jj++] = dd.c[0];
        datablock[jj++] = dd.c[1];
        datablock[jj++] = dd.c[2];
        datablock[jj++] = dd.c[3];
        if ( f.channels >= 2 )
        {
            datablock[jj++] = dd.c[4];
            datablock[jj++] = dd.c[5];
        }
    }
}

```

```

                datablock[jj++] = dd.c[6];
                datablock[jj++] = dd.c[7];
            }
        }
    }
    fclose(fp);
    return 0;
}

```

E.3 Programa rotinwav para ver la estructura de un archivo .wav

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef
struct _chunk
{
    char sign[4];
    long size;
} chunk;

typedef
struct _riff
{
    char wave[4];
} riff;

typedef
struct _limits
{
    long int begin;
    long int length;
} limits;

int main (int argc, char* argv[])
{
    FILE *wfp, *rfp;
    chunk rc,ec={"etic",0},xc[20];
    riff rr;
    char ss[50], line[4096], ph[6];
    limits ll[50];
    void *xx[20];
    char wfn[256], rfn[256];
    long size, begin, end;
    int i,j,n, dum;
    short m;

    if ( argc >= 2 )
    {
        sprintf(wfn,"%s.wav",argv[1]);
        sprintf(rfn,"%s.rot",argv[1]);
    }
    else
    {
        fprintf ( stderr, "Uso: partir file.wav.\n" );
        exit(0);
    }

    if ( (wfp=fopen(wfn,"rb")) == NULL )
    {
        fprintf ( stderr, "Error: Archivo %s no pude ser abierto.\n" , wfn );
        exit(1);
    }
}

```

```

fread(&rc,sizeof(rc),1,wfp);
if (strncmp(rc.sign,"RIFF",4)!=0)
{
fprintf ( stderr, "Error: Archivo %s no es un formato v lido.\n" , wfn );
exit(1);
}
fread(&rr,sizeof(rr),1,wfp);
for(i=0;fread(&xc[i]),sizeof(xc[i]),1,wfp)==1;i++)
{
if (strncmp(xc[i].sign,"etic",4)==0)
fprintf ( stderr, "Advertencia: El archivo %s ya fue convertido.\n" , wfn );
size = xc[i].size&1 ? xc[i].size+1 : xc[i].size;
xx[i]=malloc(size);
fread(xx[i],1,size,wfp);
}
n=i;

fclose( wfp );

if( (rfp=fopen(rfn,"rt")) == NULL )
{
fprintf( stderr, "Error: no se puede abrir %s para lectura\n", rfn );
exit(1);
}

fgets( line, 4096, rfp ); // lee encabezado: firma
fgets( line, 4096, rfp ); // nombre del archivo
fgets( line, 4096, rfp ); // frase en español
fgets( line, 4096, rfp ); // transcripción fonética

i=0;
while( fgets( line, 4096, rfp ) )
{
sscanf( line, "%d [%s ( %d , %d ) ( %ld , %ld )",
&dum, ph, &dum, &dum, &begin, &end );
for(j=0;ph[j]!='\0';j++)
{
ss[i]=ph[j];
ll[i].begin=2*begin;
ll[i].length=2*(end-begin);
i++;
}
}
m=i;
ss[i]=0;

fclose( rfp );

ec.size = sizeof(m)+(i+1)*sizeof(char)+i*sizeof(limits);
rc.size = sizeof(riff);
for(i=0;i<n;i++)
if (strncmp(xc[i].sign,"etic",4)!=0)
rc.size += sizeof(xc[i])+ ( xc[i].size&1 ? xc[i].size+1 : xc[i].size
);
rc.size += sizeof(ec)+( ec.size&1 ? ec.size+1: ec.size&1 );

if( (wfp=fopen(wfn,"wb")) == NULL )
{
fprintf( stderr, "Error: no se puede abrir %s para escritura\n", wfn );
exit(1);
}

fwrite(&rc,sizeof(rc),1,wfp);
fwrite(&rr,sizeof(rr),1,wfp);

for(i=0;i<n;i++)
{
if (strncmp(xc[i].sign,"etic",4)==0)
{
fwrite(&ec,sizeof(ec),1,wfp);
}
}

```

```

        fwrite(&m,sizeof(m),1,wfp);
        fwrite(ss,sizeof(char),m+1,wfp);
        fwrite(ll,sizeof(limits),m,wfp);
        if(ec.size&1)
            fwrite("",sizeof(char),1,wfp);
    }
    else
    {
        fwrite(xc+i,sizeof(xc[i]),1,wfp);
        fwrite(xx[i],1, xc[i].size&1 ? xc[i].size+1 : xc[i].size , wfp);
    }
}

fclose( wfp );

return 0;
}

```

E.4 archivo rot2tgt. c, programa para crear un archivo de *targets* a partir de la rotulación

```

/*****
* Programa rot2tgt
* Crea un archivo MTh de objetivos a partir de
* un archivo de rótulos. Toma sólo los fonemas
* que figuran en el preinforme.
*****/
* Autores: Carlos Andrés Moncayo Ramírez
* Carlos Eugenio Thomspón Pinzón
* Pontificia Universidad Javeriana, Bogotá
* © 1999-2000
*****/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

typedef char vector[18];

int getindex(char*ph);
vector* extend(vector* pv,int os,int ns);

char muestras[18][3] =
{
    "i", "e", "a", "o", "u", "p", "t", "k", "tS",
    "f", "s", "m", "n", "\xF8", "R", "r", "l", "\xC6"
};

char asciipa[18] =
{
    'i', 'e', 'a', 'o', 'u', 'p', 't', 'k', 'C',
    'f', 's', 'm', 'n', 'N', 'R', 'r', 'l', 'J'
};

int bcd ( int arg );

typedef
struct _myfileh
{
    char sign [4];
    char type [4];
    char reserved;
    char sec, min, hour, day, month;
    short year;
} myfileh;

```



```

typedef
struct _mydatah
{
    long cols, rows;
    short itemSize;
    short flags;
    long dataSize;
} mydatah;

int main( int C, char* A[] )
{
    FILE *ifp, *ofp;
    char ifn[1024], ofn[1024], line[4096];
    unsigned wnd_s, jmp_s, q, r, wnd_b, wnd_e;
    double perm;
    vector *phonv;
    char ph[6];
    long begin, end;
    volatile int dum;
    int c;
    myfileh fh = { "MTh", "tgt", 0, 0, 0, 0, 0x21, 0x1, 0x2000 };
    mydatah dh = { 18, 0, 1, 0, 0 };
    time_t timer;
    struct tm * tmp;

    time(&timer);
    tmp = localtime(&timer);
    fh.sec   = bcd( tmp->tm_sec );
    fh.min   = bcd( tmp->tm_min );
    fh.hour  = bcd( tmp->tm_hour );
    fh.day   = bcd( tmp->tm_mday );
    fh.month = bcd( tmp->tm_mon+1 );
    fh.year  = bcd( tmp->tm_year + 1900 );

    if( C>=2 )
    {
        strcpy( ifn, A[1] );
        strcat( ifn, ".rot" );
        strcpy( ofn, A[1] );
        strcat( ofn, ".tgt" );
    }
    else
    {
        fprintf( stderr, "Error: argumentos insuficientes\n", ifn );
        exit(1);
    }

    wnd_s = C>=3 ? atoi( A[2] ) : 1024;
    jmp_s = C>=4 ? atoi( A[3] ) : wnd_s/2;
    perm  = C>=5 ? atof( A[4] ) : .75;
    wnd_b = (1-perm) * wnd_s;
    wnd_e = perm * wnd_s;

    if( (ifp=fopen(ifn,"rt")) == NULL )
    {
        fprintf( stderr, "Error: no se puede abrir %s para lectura\n", ifn );
        exit(1);
    }

    fgets( line, 4096, ifp ); // lee encabezado: firma
    fgets( line, 4096, ifp ); // nombre del archivo
    fgets( line, 4096, ifp ); // frase en español
    fgets( line, 4096, ifp ); // transcripción fonética

    phonv = extend( NULL, 0, q=1 );

    while( fgets( line, 4096, ifp ) )
    {
        sscanf( line, "%d [%s ( %d , %d ) ( %ld , %ld )",
                &dum, ph, &dum, &dum, &begin, &end );
    }
}

```

```

    c = getindex(ph);
    if( end > q*jmp_s + wnd_b )
    {
        r = q;
        while( end > q*jmp_s + wnd_b ) q++;
        phonv = extend( phonv, r, q );
    }
    if( c>=0 )
    {
        for( r=0; begin > r*jmp_s + wnd_e; r++ );
        for( ; end > r*jmp_s + wnd_b; r++ )
            phonv[r][c]=1;
    }
}

fclose( ifp );

if( (ofp=fopen(ofn,"wb")) == NULL )
{
    fprintf( stderr, "Error: no se puede abrir %s para escritura\n", ofn );
    exit(1);
}

dh.itemSize = sizeof(char);
dh.flags    = 0;
dh.rows     = q;
dh.cols     = sizeof(vector)/sizeof(char);
dh.dataSize = q*sizeof(vector);

fwrite( &fh, sizeof(fh), 1, ofp );
fwrite( &dh, sizeof(dh), 1, ofp );
fwrite( phonv, sizeof(vector), q, ofp );

fclose( ofp );

return 0;
} //end main

int getindex( char*ph )
{
    switch( *ph )
    {
        case 'i': return 0; // fonema /i/
        case 'e': return 1; // fonema /e/
        case 'a': return 2; // fonema /a/
        case 'o': return 3; // fonema /o/
        case 'u': return 4; // fonema /u/
        case 'p': return 5; // fonema /p/
        case 't': return // fonema /t/ o /tS/ <ch>
                    ph[1]!='S' ? 6 : 8;
        case 'k': return 7; // fonema /k/
        case 'f': return 9; // fonema /f/
        case 's': return 10; // fonema /s/
        case 'm': return 11; // fonema /m/
        case 'n': return 12; // fonema /n/
        case '\xF8': return 13; // fonema /J/
        case 'R': return 14; // fonema /4/
        case 'r': return 15; // fonema /rr/
        case 'l': return 16; // fonema /l/
        case '\xC6': return 17; // fonema /j\ / <y>
    }
    return -1;
}

vector* extend( vector* pv, int os, int ns )
{
    vector *ans;
    if( (ans=(vector*)malloc(sizeof(vector)*ns)) == NULL )
    {
        fprintf( stderr, "Error: memoria insuficiente\n" );
    }
}

```

```

        exit(1);
    }
    if( !os )
        memset ( ans, 0, sizeof(vector)*ns );
    else
    {
        if( ns>os )
        {
            memcpy ( ans, pv, sizeof(vector)*os );
            memset ( ans+os, 0, sizeof(vector)*(ns-os) );
        }
        else
            memcpy ( ans, pv, sizeof(vector)*ns );
        free(pv);
    }
    return ans;
}

int bcd ( int arg )
{
    int ans=0;
    ans += ( arg      %10);
    ans += ((arg/10  )%10)<<4;
    ans += ((arg/100 )%10)<<8;
    ans += ((arg/1000)%10)<<12;
    return ans;
}

```

E.5 archivo rot2tgx. c, programa para crear un archivo de *targets* a partir de la rotulación, incluyendo todos los fonemas.

```

/*****
 * Programa rot2tgx
 * Crea un archivo MTh de objetivos a partir de
 * un archivo de rótulos. Toma todos los fonemas.
 *****/
 * Autores: Carlos Andrés Moncayo Ramírez
 *          Carlos Eugenio Thomspón Pinzón
 * Pontificia Universidad Javeriana, Bogotá
 * © 1999-2000
 *****/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

#define NUMPHON 24

typedef char vector[NUMPHON];

int getindex(char*ph);
vector* extend(vector* pv,int os,int ns);

char muestras[NUMPHON][3] =
{
    "i", "e", "a", "o", "u", "p", "t", "k", "tS",
    "f", "s", "m", "n", "\xF8", "R", "r", "l", "\xC6",
    "b", "d", "g", "x", "j", "w"
};

char asciipa[NUMPHON] =
{
    'i', 'e', 'a', 'o', 'u', 'p', 't', 'k', 'C',
    'f', 's', 'm', 'n', 'N', 'R', 'r', 'l', 'J',
    'b', 'd', 'g', 'x', 'j', 'w'
}

```

```

};

int bcd ( int arg );

typedef
struct _myfileh
{
    char sign [4];
    char type [4];
    char reserved;
    char sec, min, hour, day, month;
    short year;
} myfileh;

typedef
struct _mydatah
{
    long cols, rows;
    short itemSize;
    short flags;
    long dataSize;
} mydatah;

int main( int C, char* A[] )
{
    FILE *ifp, *ofp;
    char ifn[1024], ofn[1024], line[4096];
    unsigned wnd_s, jmp_s, q, r, wnd_b, wnd_e;
    double perm;
    vector *phonv;
    char ph[6];
    long begin, end;
    volatile int dum;
    int c;
    myfileh fh = { "MTh", "tgx", 0, 0, 0, 0, 0x21, 0x1, 0x2000 };
    mydatah dh = { NUMPHON, 0, 1, 0, 0 };
    time_t timer;
    struct tm * tmp;

    time(&timer);
    tmp = localtime(&timer);
    fh.sec   = bcd( tmp->tm_sec );
    fh.min   = bcd( tmp->tm_min );
    fh.hour  = bcd( tmp->tm_hour );
    fh.day   = bcd( tmp->tm_mday );
    fh.month = bcd( tmp->tm_mon+1 );
    fh.year  = bcd( tmp->tm_year + 1900 );

    if( C>=2 )
    {
        strcpy( ifn, A[1] );
        strcat( ifn, ".rot" );
        strcpy( ofn, A[1] );
        strcat( ofn, ".tgx" );
    }
    else
    {
        fprintf( stderr, "Error: argumentos insuficientes\n", ifn );
        exit(1);
    }

    wnd_s = C>=3 ? atoi( A[2] ) : 1024;
    jmp_s = C>=4 ? atoi( A[3] ) : wnd_s/2;
    perm  = C>=5 ? atof( A[4] ) : .75;
    wnd_b = (1-perm) * wnd_s;
    wnd_e = perm * wnd_s;

    if( (ifp=fopen(ifn,"rt")) == NULL )
    {
        fprintf( stderr, "Error: no se puede abrir %s para lectura\n", ifn );
    }

```

```

    exit(1);
}

fgets( line, 4096, ifp ); // lee encabezado: firma
fgets( line, 4096, ifp ); // nombre del archivo
fgets( line, 4096, ifp ); // frase en español
fgets( line, 4096, ifp ); // transcripción fonética

phonv = extend( NULL, 0, q=1 );

while( fgets( line, 4096, ifp ) )
{
    sscanf( line, "%d [%s ( %d , %d ) ( %ld , %ld )",
            &dum, ph, &dum, &dum, &begin, &end );
    c = getindex(ph);
    if( end > q*jmp_s + wnd_b )
    {
        r = q;
        while( end > q*jmp_s + wnd_b ) q++;
        phonv = extend( phonv, r, q );
    }
    if( c>=0 )
    {
        for( r=0; begin > r*jmp_s + wnd_e; r++ );
        for( ; end > r*jmp_s + wnd_b; r++ )
            phonv[r][c]=1;
    }
}

fclose( ifp );

if( (ofp=fopen(ofn,"wb")) == NULL )
{
    fprintf( stderr, "Error: no se puede abrir %s para escritura\n", ofn );
    exit(1);
}

dh.itemSize = sizeof(char);
dh.flags    = 0;
dh.rows     = q;
dh.cols     = sizeof(vector)/sizeof(char);
dh.dataSize = q*sizeof(vector);

fwrite( &fh, sizeof(fh), 1, ofp );
fwrite( &dh, sizeof(dh), 1, ofp );
fwrite( phonv, sizeof(vector), q, ofp );

fclose( ofp );

return 0;
} //end main

int getindex( char*ph )
{
    switch( *ph )
    {
        case 'i': return 0; // fonema /i/
        case 'e': return 1; // fonema /e/
        case 'a': return 2; // fonema /a/
        case 'o': return 3; // fonema /o/
        case 'u': return 4; // fonema /u/
        case 'p': return 5; // fonema /p/
        case 't': return // fonema /t/ o /tS/ <ch>
                    ph[1]!='S' ? 6 : 8;
        case 'k': return 7; // fonema /k/
        case 'f': return 9; // fonema /f/
        case 's': return 10; // fonema /s/
        case 'm': return 11; // fonema /m/
        case 'n': return 12; // fonema /n/
        case '\xF8': return 13; // fonema /J/
    }
}

```

```

    case 'R':    return 14; // fonema /4/
    case 'r':    return 15; // fonema /rr/
    case 'l':    return 16; // fonema /l/
    case '\xC6': return 17; // fonema /j\ <y>
    case 'B':
    case 'b':    return 18; // fonema /b/
    case 'D':
    case 'd':    return 19; // fonema /d/
    case 'G':
    case 'g':    return 20; // fonema /g/
    case 'h':
    case 'x':    return 21; // fonema /x/
    case 'j':    return 22; // fonema /j/
    case 'w':    return 23; // fonema /w/
    }
    return -1;
}

vector* extend( vector* pv, int os, int ns )
{
    vector *ans;
    if( (ans=(vector*)malloc(sizeof(vector)*ns)) == NULL )
    {
        fprintf( stderr, "Error: memoria insuficiente\n" );
        exit(1);
    }
    if( !os )
        memset ( ans, 0, sizeof(vector)*ns );
    else
    {
        if( ns>os )
        {
            memcpy ( ans, pv, sizeof(vector)*os );
            memset ( ans+os, 0, sizeof(vector)*(ns-os) );
        }
        else
            memcpy ( ans, pv, sizeof(vector)*ns );
        free(pv);
    }
    return ans;
}

int bcd ( int arg )
{
    int ans=0;
    ans += ( arg      %10);
    ans += ((arg/10  )%10)<<4;
    ans += ((arg/100 )%10)<<8;
    ans += ((arg/1000)%10)<<12;
    return ans;
}

```

NEXO F. PROGRAMA DE EXTRACCIÓN ALEATORIA DE FONEMAS

F.1 Archivo fonextr.h, definición de rutinas y estructuras del programa

fonextr.

```

/*****
 * fonextr.h
 *****/

void errormsg(char*,...);
void warnmsg(char*,...);

#define LONGVEC 30

typedef unsigned long ulong;

typedef
struct _limite
{
    char*filename;
    ulong ini,lon;
} limite;

typedef
struct _listalim
{
    limite lim;
    struct _listalim *sig;
} nodolim, *listalim;

typedef
struct _listafon
{
    char fonema[4];
    listalim lista;
    struct _listafon *sig;
} nodofon, *listafon;

typedef
struct _fonent
{
    unsigned int number;
    char fon[10][4];
} fonent, *vecfonent;

typedef listalim *vecfonsal;

listafon nuevaLF(void);
listalim nuevaLL(void);
vecfonent nuevoVFE(void);
vecfonsal nuevoVFS(void);

nodolim* primerLL(listalim*);
nodofon* primerLF(listafon*);

```

```

nodolim* sigLL(listalim*,nodolim*);
nodofon* sigLF(listafon*,nodofon*);
nodolim* elimSigLL(listalim*,nodolim*);
nodofon* buscarLF(listafon*,char*);
nodolim* adicioneLL(listalim*,char*,ulong,ulong);
nodofon* adicioneLF(listafon*,char*);
ulong tamanoLL(listalim*);
int esfinLL(listalim* pLL, nodolim* pNL);
int esfinLF(listafon* pLF, nodofon* pNF);
vecfont* adicionaVFE(vecfont*, int, char*);

listafon* agregaLF(listafon* pLF, char* fon, char* filename, ulong ini, ulong lon);
listalim* agregaLL(listalim* pLL, char* filename, ulong ini, ulong lon);
listalim* concatLL(listalim* dest,listalim* src);

limite* aleatorio(listafon Lfdonde, listafon Lfcual, ulong tamanno);

int leerArchivo (listafon* pLF,char* filename);
int leerArchivoWav(listafon* pLF,char* filename);
int leerArchivoRot(listafon* pLF,char* filename);
char* agregaFonema(vecfont* pVFE, int pos, char* fon);
int agregaNumero(vecfont* pVFE, int pos, char* fon);

typedef unsigned char uint8;

int estaVacioVFE (vecfont);
int estaVacioVFS (vecfonsal);
int selAzarVFE (vecfonsal*, vecfont*, vecfonsal*, ulong);
void selAzarVFS (double*, uint8*, vecfonsal*, ulong);
void agregaLimsVFS (vecfonsal*, listafon*, char*, int);
void creaLista (vecfonsal*, vecfont*, listafon*);

void* NuevoArchSalida (char*, ulong, ulong);
void* NuevoArchTarget (char*, ulong, ulong);
int NuevaColSal (double*, ulong, void*);
int NuevaColTgt (uint8*, ulong, void*);
void CerrarArchSalida (void*, ulong, ulong);
void CerrarArchTarget (void*, ulong, ulong);

void mostrarVFS(vecfonsal);

```

F.2 Archivo fonextr.c, programa principal.

```

/*****
 * fonextr.c
 *****/

#include "fonextr.h"
#include "wavfile.h"
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define MAXLINEA 4096
#define MAXFILE 256

typedef
enum _estado { stInicio, stArchivo, stFonema, stParametros, stOtros } estado;

//TEMPORARY
void ImprimaListaLF(listafon);
void ImprimaVectorEntrada(vecfont*,int);

int main (int argc, char* argv[])
{
    FILE *sfp, *tfp, *ofp;

```



```

char linea[MAXLINEA], ofn[MAXFILE], tfn[MAXFILE], *s;
estado st;
listafon lstArch = nuevaLF();
vecfonent vecEnt = nuevoVFE();
vecfonsal vecSal = nuevoVFS(), vecInt=nuevoVFS();
int numlinea, numfonema, nummuestras;
unsigned ventana;
double * vecReal;
uint8 * vecTgt;

srand( (unsigned)time( NULL ) );

/* abrir el archivo de script */
if (argc < 2)
{
    errormsg("Numero de argumentos incorrecto");
}
else
{
    sfp = fopen (argv[1],"rt");
    if (sfp == NULL)
        errormsg("archivo de script %s no se puede abrir", argv[1]);
}

st = stInicio;
numlinea=0;
numfonema=0;
while( fgets(linea,MAXLINEA,sfp) != NULL )
{
    numlinea++;
    for( s=linea; *s && isspace(*s); s++ );
    if (!*s || *s=='#') continue;
    if ( *s=='[' )
    {
        switch(*++s)
        {
            case 'a': case 'A':
                st = stArchivo;
                break;
            case 'f': case 'F':
                st = stFonema;
                break;
            case 'p': case 'P':
                st = stParametros;
                break;
            default:
                st = stOtros;
                break;
        }
    }
    else
    {
        switch(st)
        {
            case stInicio:
                warnmsg("cl usula inv lida en la l;nea %d", numlinea);
                break;
            case stArchivo:
                if( leerArchivo(&lstArch,s) != 0 )
                    warnmsg("Archivo inv lido en la l;nea %d", numlinea);
                break;
            case stFonema:
                if( *s++ != '{' )
                    warnmsg("Descripci;n de fonema inv lida en la l;nea %d", numlinea);
                else
                {
                    while(*s && *s != '}')
                        s = agregaFonema(&vecEnt,numfonema,s);
                    if (!*s)
                }
            }
        }
    }
}

```

```

        warnmsg("Descripci3n de fonema inv lida en la l;nea %d", numlinea);
        break;
    }
    s++;
    while(*s && (isspace(*s) || *s=='=')) s++;
    if ( agregaNumero(&vecEnt,numfonema,s) != 0)
        warnmsg("Descripci3n de fonema inv lida en la l;nea %d", numlinea);
    numfonema++;
}
break;
case stParametros:
switch(*s)
{
case 'v': case 'V':
    while(*s && *s != '=') s++;
    if (!*s)
    {
        warnmsg("No hay valor de par metro en la l;nea %d", numlinea);
        break;
    }
    ventana = atoi(++s);
    break;
case 's': case 'S':
    while(*s && *s != '=') s++;
    for(*s?s++:s;*s && isspace(*s);s++);
    if (!*s)
    {
        warnmsg("No hay valor de par metro en la l;nea %d", numlinea);
        break;
    }
    strcpy(ofn,s);
    for(s=ofn;*s && !isspace(*s);s++);
    *s=0;
    break;
case 'o': case 'O':
    while(*s && *s != '=') s++;
    for(*s?s++:s;*s && isspace(*s);s++);
    if (!*s)
    {
        warnmsg("No hay valor de par metro en la l;nea %d", numlinea);
        break;
    }
    strcpy(tfn,s);
    for(s=tfn;*s && !isspace(*s);s++);
    *s=0;
    break;
default:
    warnmsg("Par metro desconocido en la l;nea %d", numlinea);
}
break;
case stOtros:
default:
    warnmsg("No debi4 haber llegado a este estado en la l;nea %d", numlinea);
}
}
}

fclose(sfp);

/* seleccionar las ventanas */
creaLista(&vecInt,&vecEnt,&lstArch);
nummuestras=0;
while( !estaVacioVFE(vecEnt) )
{
    nummuestras += selAzarVFE(&vecSal,&vecEnt,&vecInt,ventana);
}

/* armar los archivos .wnd y .tgt */
vecReal = (double*)calloc(ventana,sizeof(double));
vecTgt = (unsigned char*)calloc(numfonema,sizeof(unsigned char));

```

```

ofp = NuevoArchSalida(ofn,nummuestras,ventana);
tfp = NuevoArchTarget(tfn,nummuestras,numfonema);

while( !estaVacioVFS(vecSal) )
{
    selAzarVFS(vecReal,vecTgt,&vecSal,numfonema);
    NuevaColSal(vecReal,ventana,ofp);
    NuevaColTgt(vecTgt,numfonema,tfp);
}

CerrarArchTarget(tfp,numfonema,nummuestras);
CerrarArchSalida(ofp,ventana,nummuestras);

return 0;
}

int leerArchivo (listafon* pLF, char* str)
{
    char *wfn, *s;
    s = wfn = _strdup(str);
    if(s==NULL) return -1;
    while(*s && !isspace(*s) && *s!='#') s++;
    if (!*s) return -1;
    *s=0;

    s=strrchr(wfn, '.');
    if(s==NULL)
    {
        warnmsg("archivo %s no es .wav o .rot",wfn);
        return 1;
    }
    if(_stricmp(s+1,"wav")==0)
        return leerArchivoWav(pLF,wfn);
    if(_stricmp(s+1,"rot")==0)
        return leerArchivoRot(pLF,wfn);

    warnmsg("archivo %s no es .wav o .rot",wfn);
    return 0;
}

int leerArchivoWav(listafon* pLF, char* filename)
{
    FILE* wF;
    unsigned short n,i,j;
    char fons[128], f[128];
    struct _lim { ulong ini, lon; } lims[128];
    if( (wF=wavOpen(filename))==NULL )
    {
        warnmsg("No se pudo abrir %s",filename);
        return -1;
    }
    if( wavSeekChunk(wF,"etic") != 0)
    {
        warnmsg("No hay chunk de fon,tica \"etic\" en %s", filename);
        return -1;
    }
    if( fread( &n, sizeof(n), 1, wF ) != 1 )
    {
        warnmsg("Chunk termin  antes de tiempo");
        return -1;
    }
    if( fread( fons, sizeof(char), n+1, wF ) != (size_t)(n+1) )
    {
        warnmsg("Chunk termin  antes de tiempo");
        return -1;
    }
    if( fread(lims,sizeof(struct _lim),n, wF) != n )

```

```

    {
        warnmsg("Chunk terminç antes de tiempo");
        return -1;
    }

    for(i=0;i<n;i++)
    {
        j=0;
        f[j++]=fons[i];
        while(i<n-1 && lims[i].ini==lims[i+1].ini)
        {
            f[j++]=fons[++i];
        }
        f[j]=0;
        agregaLF(pLF,f,filename,lims[i].ini/2,lims[i].lon/2);
    }

    fclose(wF);
    return 0;
}

int leerArchivoRot(listafon* pLF, char* filename)
{
    printf("Archivos .ROT como %s, no est n implementados afn\n", filename);
    return(1);
}

char* agregaFonema (vecfont* pVFE, int nF, char* str)
{
    char *s, *t=str, fon[MAXLINEA];
    while(*t && isspace(*t)) t++;
    s=strcpy(fon,t);
    if(s==NULL) return NULL;

    while(*t && !isspace(*t) && *t!='}') t++,s++;
    *s = 0;

    adicionaVFE(pVFE,nF,fon);
    return t;
}

int agregaNumero (vecfont* pVFE, int nF, char* str)
{
    char*s=str;
    while(*s && (isspace(*s) || *s=='=')) s++;
    if(!*s) return -1;

    (*pVFE)[nF].number = atoi(s);

    return 0;
}

```

F.3 Archivo archi vo. c, rutinas para manipular archivos MTh.

```

/*****
 * archivo.c
 *****/

#include "fonextr.h"
#include <stdio.h>
#include <time.h>

int bcd ( int arg )
{
    int ans=0;
    ans += ( arg %10);
    ans += ((arg/10 )%10)<<4;
}

```

```

    ans += ((arg/100 )%10)<<8;
    ans += ((arg/1000)%10)<<12;
    return ans;
}

typedef
struct _myfileh
{
    char sign [4];
    char type [4];
    char reserved;
    char sec, min, hour, day, month;
    short year;
} myfileh;

typedef
struct _mydatah
{
    long cols, rows;
    short itemSize;
    short flags;
    long dataSize;
} mydatah;

void* NuevoArchSalida (char* filename, ulong filas, ulong cols)
{
    FILE* ofp = fopen(filename,"wb");
    time_t tt;
    struct tm *tp;
    myfileh fh = { "MTh", "wnd", 0, 0, 0, 0, 0x05, 0x4, 0x2000 };
    mydatah dh = { cols, filas, sizeof(double), 1, cols*filas*sizeof(double) };

    time(&tt);
    tp=localtime(&tt);
    fh.sec  = bcd( tp->tm_sec );
    fh.min  = bcd( tp->tm_min );
    fh.hour = bcd( tp->tm_hour );
    fh.day  = bcd( tp->tm_mday );
    fh.month = bcd( tp->tm_mon+1 );
    fh.year = bcd( tp->tm_year + 1900 );

    fwrite(&fh,sizeof(fh),1,ofp);
    fwrite(&dh,sizeof(dh),1,ofp);

    return ofp;
}

void* NuevoArchTarget (char* filename, ulong filas, ulong cols)
{
    FILE* ofp = fopen(filename,"wb");
    time_t tt;
    struct tm *tp;
    myfileh fh = { "MTh", "tgt", 0, 0, 0, 0, 0x05, 0x4, 0x2000 };
    mydatah dh = { cols, filas, sizeof(char), 0, cols*filas*sizeof(char) };

    time(&tt);
    tp=localtime(&tt);
    fh.sec  = bcd( tp->tm_sec );
    fh.min  = bcd( tp->tm_min );
    fh.hour = bcd( tp->tm_hour );
    fh.day  = bcd( tp->tm_mday );
    fh.month = bcd( tp->tm_mon+1 );
    fh.year = bcd( tp->tm_year + 1900 );

    fwrite(&fh,sizeof(fh),1,ofp);
    fwrite(&dh,sizeof(dh),1,ofp);

    return ofp;
}

```

```

int NuevaColSal (double* bufR, ulong tam, void* arch)
{
    return fwrite(bufR,sizeof(double),tam,arch);
}

int NuevaColTgt (uint8* bufT, ulong tam, void* arch)
{
    return fwrite(bufT,sizeof(uint8),tam,arch);
}

void CerrarArchSalida (void* arch, ulong filas, ulong cols)
{
}

void CerrarArchTarget (void* arch, ulong filas, ulong cols)
{
}

```

F.4 Archivo listas.c, rutinas para manipular listas y árboles.

```

/*****
 * listas.c
 *****/

#include "fonextr.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

listafon nuevaLF(void)
{
    return NULL;
}

listalim nuevaLL(void)
{
    return NULL;
}

listafon* agregaLF(listafon* pLF, char* fon, char* filename, ulong ini, ulong lon)
{
    nodofon *aux = buscarLF(pLF,fon);
    if(aux == NULL)
    {
        aux = adicioneLF(pLF,fon);
    }
    adicioneLL(&(aux->lista),filename,ini,lon);
    return pLF;
}

nodolim* primerLL(listalim* pLL)
{
    return *pLL;
}

nodofon* primerLF(listafon* pLF)
{
    return *pLF;
}

nodolim* sigLL(listalim* pLL,nodolim* pNL)
{
    return pNL->sig;
}

nodofon* sigLF(listafon* pLF,nodofon* pNF)

```

```

{
    return pNF->sig;
}

nodofon* buscarLF(listafon* pLF, char* fon)
{
    nodofon* aux;
    for (aux=primerLF(pLF);aux!=NULL && strcmp(fon,aux->fonema)!=0;aux=sigLF(pLF,aux));
    return aux;
}

nodolim* adicioneLL(listalim* pLL, char* filename,ulong ini,ulong lon)
{
    nodolim* aux = (nodolim*)malloc(sizeof(nodolim));
    if(aux) {
        aux->lim.filename = filename;
        aux->lim.ini = ini;
        aux->lim.lon = lon;
        aux->sig = primerLL(pLL);
        *pLL = aux; }
    return aux;
}

nodofon* adicioneLF(listafon* pLF, char* fon)
{
    nodofon* aux = (nodofon*)malloc(sizeof(nodofon));
    strcpy(aux->fonema, fon);
    aux->lista = nuevaLL();
    aux->sig = primerLF(pLF);
    *pLF = aux;
    return aux;
}

int esfinLL(listalim* pLL, nodolim* pNL)
{
    return pNL == NULL;
}

int esfinLF(listafon* pLF, nodofon* pNF)
{
    return pNF == NULL;
}

vecfonent nuevoVFE()
{
    vecfonent aux=(vecfonent)calloc(LONGVEC,sizeof(fonent));
    int i;
    for (i=0; i<LONGVEC; i++)
        aux[i].number = 0;
    return aux;
}

vecfonsal nuevoVFS()
{
    vecfonsal aux=(vecfonsal)calloc(LONGVEC,sizeof(listalim));
    int i;
    for (i=0; i<LONGVEC; i++)
        aux[i] = nuevaLL();
    return aux;
}

vecfonent* adicionaVFE(vecfonent* pVFE, int nF, char* fon)
{
    fonent *FE=*pVFE;
    strcpy( FE[nF].fon[FE[nF].number++], fon );
    return pVFE;
}

nodolim* elimSigLL(listalim* pLL,nodolim* pNL)
{

```

```

    nodolim* aux;
    if(!pNL)
    {
        aux = *pLL;
        *pLL = aux->sig;
        free(aux);
        return *pLL;
    }
    else
    {
        aux = pNL->sig;
        pNL->sig = aux->sig;
        free(aux);
    }
    return pNL->sig;
}

```

F.5 Archivo mensaje.c, rutinas para escribir mensajes de error.

```

/*****
 * mensaje.c
 *****/

#include "fonextr.h"
#include <stdarg.h>
#include <stdlib.h>
#include <stdio.h>

void errormsg(char* msg,...)
{
    va_list argptr;
    va_start(argptr,msg);
    fprintf(stderr,"error: ");
    vfprintf(stderr,msg,argptr);
    fprintf(stderr,"\n");
    exit(1);
}

void warnmsg(char* msg,...)
{
    va_list argptr;
    va_start(argptr,msg);
    fprintf(stderr,"advertencia: ");
    vfprintf(stderr,msg,argptr);
    fprintf(stderr,"\n");
}

```

F.6 Archivo selec.c, rutinas para seleccionar aleatoriamente los fonemas.

```

/*****
 * selec.c
 *****/

#include "fonextr.h"
#include "wavfile.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

/*#define wnd_RANDOM*/
/*#define wnd_CENTRED*/
/*#define wnd_NORMAL*/

/*#define phn_RANDOM*/
#define phn_UNIFORM

```



```

ulong tamanovFE (vecfontent vFE)
{
    int i;
    ulong m;
    for(i=0,m=0;i<LONGVEC;i++)
        m += vFE[i].number;
    return m;
}

ulong tamanovFS (vecfonsal vFS)
{
    int i;
    ulong m;
    for(i=0,m=0;i<LONGVEC;i++)
        m += tamanoll(&vFS[i]);
    return m;
}

int estaVacioVFE (vecfontent vFE)
{
    int i,m;
    for(i=0,m=0;i<LONGVEC;i++)
        m += vFE[i].number;
    return m==0;
}

int estaVacioVFS (vecfonsal vFS)
{
    int i,ans = 1;
    for (i=0; i<LONGVEC && ans; i++)
        if(vFS[i]) ans=0;
    return ans;
}

ulong random(ulong n)
{
    double f=((double)rand())/RAND_MAX;
    return (ulong)(f*n);
}

double nrand(void)
{
    double a=0.0;
    a = rand()+rand()+rand()+rand()+rand()+rand();
    return (a/6)/RAND_MAX;
}

ulong gray(ulong n)
{
    return (n>>1)^n;
}

double invreal(ulong n)
{
    double r=0.0;
    ulong b;
    for (b=1; b<=n; b<=&1)
        if (b&n)
            r += 1.0/b;
    return r/2;
}

double urand(void)
{
    static int n=0;
    if (!n) n=rand();
    return invreal(gray(n++));
}

```

```

int selAzarVFE (vecfonsal* psDst, vecfonent* pFE, vecfonsal* psSrc, ulong nW)
{
    ulong n,k,i,j,l;
    nodolim *tmp;
    k=random(n=tamanoVFE(*pFE));
    for(i=0;k>=(*pFE)[i].number;i++)
    {
        k -= (*pFE)[i].number;
    }
    j=random(n=tamanoLL(&(*psSrc)[i]));
    if(n>0)
    {
        tmp=primerLL(&(*psSrc)[i]);
        for(l=0;l<j;l++) tmp=sigLL(&(*psSrc)[i],tmp);
        if(tmp->lim.lon < nW)
        {
            #if defined(wnd_RANDOM)
                l=random(nW - tmp->lim.lon);
            #elif defined(wnd_CENTRED)
                l=(nW - tmp->lim.lon)/2;
            #elif defined(wnd_NORMAL)
                l=(int)((nW - tmp->lim.lon)*nrand())/2;
            #else
                l=0;
            #endif
            adicioneLL(&(*psDst)[i],tmp->lim.filename,tmp->lim.ini-l,nW);
        }
        else if(tmp->lim.lon > nW)
        {
            #if defined(wnd_RANDOM)
                l=random(tmp->lim.lon - nW);
            #elif defined(wnd_CENTRED)
                l=(tmp->lim.lon - nW)/2;
            #elif defined(wnd_NORMAL)
                l=(int)((tmp->lim.lon - nW)*nrand())/2;
            #else
                l=0;
            #endif
            adicioneLL(&(*psDst)[i],tmp->lim.filename,tmp->lim.ini+l,nW);
        }
        else
        {
            adicioneLL(&(*psDst)[i],tmp->lim.filename,tmp->lim.ini,nW);
        }
        (*pFE)[i].number--;
        return l;
    }
    (*pFE)[i].number--;
    return 0;
}

void selAzarVFS (double* vR, uint8* vT, vecfonsal* pFS, ulong numfon)
{
    ulong n,k,i,j;
    nodolim *tmp, *aux;
    n=tamanoVFS(*pFS);
    #if defined(phn_RANDOM)
        k=random(n);
    #elif defined(phn_UNIFORM)
        k=(ulong)(urand()*n);
    #else
        k=0;
    #endif
    #endif
    for(i=0;k>=tamanoLL(&(*pFS)[i]);i++)
    {
        k -= tamanoLL(&(*pFS)[i]);
    }
    tmp=primerLL(&(*pFS)[i]); aux=NULL;
    for(j=0;j<k;j++)
        tmp=sigLL(&(*pFS)[i],aux=tmp);
}

```

```

wavSegment(vR,tmp->lim.filename,tmp->lim.ini,tmp->lim.lon);
memset(vT,0,numfon);
vT[i]=1;

elimSigLL(&(*pFS)[i],aux);

return;
}

listalim* concatLL(listalim* dest,listalim* src)
{
    nodolim *aux;
    while(*src!=NULL)
    {
        aux = *src;
        *src = aux->sig;
        aux->sig = *dest;
        *dest = aux;
    }
    return dest;
}

void agregaLimsVFS(vecfonsal* pFS, listafon* pLF, char* fon, int pos)
{
    nodofon *aux;
    for(aux=primerLF(pLF);!esfinLF(pLF,aux);aux=sigLF(pLF,aux))
    {
        if(fon[0]=='*' || strcmp(fon,(*aux).fonema)==0)
            concatLL(&(*pFS)[pos],&(*aux).lista);
    }
}

void creaLista (vecfonsal* pFS, vecfonent* pFE, listafon* pLF)
{
    int i,j,k = -1;
    for(i=0;i<LONGVEC;i++)
    {
        for(j=0;j<10;j++)
        {
            if((*pFE)[i].fon[j][1]!='')////////////////////////*****
            {
                agregaLimsVFS(pFS,pLF,(*pFE)[i].fon[j],i);
            }
            else k=i;
        }
    }
    if (k>=0) agregaLimsVFS(pFS,pLF,"*",k);
}

void mostrarVFS(vecfonsal vfS)
{
    int i;
    nodolim *tmp;
    for(i=0;i<LONGVEC;i++)
    {
        printf("Fonema #%d:\n",i);
        for(tmp=primerLL(&vfS[i]);!esfinLL(&vfS[i],tmp);tmp=sigLL(&vfS[i],tmp))
        {
            printf("archivo: %s de %d a %d\n", tmp->lim.filename,
                tmp->lim.ini, tmp->lim.ini+tmp->lim.lon-1 );
        }
    }
}

ulong tamanoLL(listalim* pLL)
{
    ulong i=0;
    nodolim*tmp;
    for(tmp=primerLL(pLL);!esfinLL(pLL,tmp);tmp=sigLL(pLL,tmp))
        i++;
}

```

```

    return i;
}

```

F.7 Archivo `wavfile.h`, definición de rutinas para manipular los archivos

```

    .wav.

/*****
 * wavfile.h
 *****/

#ifndef __WAVFILE_H
#define __WAVFILE_H

typedef
struct _chunk
{
    char sign[4];
    long size;
} chunk;

typedef
struct _wavfmt
{
    short int _0x01;
    short int chanel;
    long int sampps;
    long int bps;
    short int bpsamp;
    short int bitps;
} wavfmt;

void* wavOpen(char* filename);
int wavSeekChunk(void* wavfile, char* sign);
double* wavSegment(double* vec, char* filename, long ini, long lon);

#endif /* __WAVFILE_H */

```

F.8 Archivo `wavfile.c`, rutinas para manipular los archivos `.wav`.

```

/*****
 * wavfile.c
 *****/

void warnmsg(char*,...);

#include "wavfile.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void* wavOpen(char* filename)
{
    return fopen(filename,"rb");
}

int wavSeekChunk(void* wfp, char* sign)
{
    chunk x;
    fseek(wfp,sizeof(chunk)+4*sizeof(char),SEEK_SET);
    while(1)
    {
        if ( fread(&x,sizeof(x),1,wfp) != 1)
            break;
        if (strncmp(x.sign,sign,4)==0)
            return 0;
    }
}

```

```

        fseek(wfp, x.size&1? x.size+1: x.size, SEEK_CUR);
    }
    return -1;
}

double* wavSegment(double* vec,char* filename,long ini,long lon)
{
    wavfmt wavFmt;
    void* buf;
    FILE* wfp = wavOpen(filename);
    int i;

    if (wfp == NULL)
    {
        warnmsg("no se pudo abrir %s\n", filename );
        return vec;
    }
    if ( wavSeekChunk(wfp,"fmt ") != 0 )
    {
        warnmsg("No se encontr  chunk fmt en %s\n", filename );
        return vec;
    }
    if ( fread(&wavFmt,sizeof(wavFmt),1,wfp) != 1 )
    {
        warnmsg("chunk fmt incorrecto en %s\n", filename );
        return vec;
    }
    if ( wavSeekChunk(wfp,"data") != 0 )
    {
        warnmsg("No se encontr  chunk data en %s\n", filename );
        return vec;
    }

    if( fseek(wfp,ini*wavFmt.bpsamp,SEEK_CUR) )
    {
        warnmsg("salto incompleto en %s", filename );
        return vec;
    }
    buf = calloc(lon,wavFmt.bpsamp);
    if ( fread(buf,wavFmt.bpsamp,lon,wfp) != (unsigned)lon )
    {
        warnmsg("termin  de leer %s antes de tiempo", filename );
        return vec;
    }
    switch(wavFmt.bitps)
    {
    case 8:
        for(i=0;i<lon;i++)
        {
            vec[i] = *((unsigned char*)buf + i*wavFmt.channels) - 128;
            vec[i] /= 128;
        }
    case 16:
        for(i=0;i<lon;i++)
        {
            vec[i] = *((short*)buf + i*wavFmt.channels);
            vec[i] /= 0x8000;
        }
    }
    free(buf);
    return vec;
}

```

NEXO G. PROGRAMAS DE MATLAB PARA MANIPULACIÓN DE ARCHIVOS

G.1 Archivo FileRead.m, abre un archivo MTh para lectura.

```
function [fid,type,msg] = FileRead(FileName)
%   Función para abrir un archivo MTh para lectura
%   [fid,type,msg]=FileRead(FileName)
%   Donde
%       FileName     es el nombre del archivo a abrir
%       fid          Identificador del archivo abierto
%       type         El subtipo de archivo
%       msg          mensaje de error si fid==-1
%
%   Ver también: NewFile, FileGet.
%
%   © 2000, Pontificia Universidad Javeriana, Bogotá
%   Escrito por
%       Carlos Andrés Moncayo Ramírez
%       Carlos Eugenio Thompson Pinzón
[fid,msg] = fopen(FileName,'rb');

if fid<0,
    sign=fread(fid,[4,2],'char');
    if strcmp(sign(1,:)','MTh' 0),
        type=sign(2,:);
        fread(fid,8,'uint8');
    else
        fid = -1;
        msg = 'No es un archivo MTh válido';
    end
end
```

G.2 Archivo FileGet.m, lee una matriz o un conjunto de matrices de un archivo MTh.

```
function [M,fid] = FileGet(arg1,arg2)
%   Función para leer una matriz desde un archivo MTh
%   [M,fid]=FileGet(fid,N)
%   Donde
%       fid          Identificador del archivo
%       N            Índice de la matriz a leer
%       M            Matriz leída
%   si el campo N se deja en blanco hace una lectura secuencial del archivo
%
%   [C,fid]=FileGet(fid,'all')
%   Donde
%       fid          Identificador del archivo
%       C            Arreglo de celdas donde se almacenan todas las matrices en fid
%
%   [C,fid]=FileGet(fid,'remind')
%   Donde
%       fid          Identificador del archivo
```

```

%           C           Arreglo de celdas donde se almacenan las matrices restantes en fid
%
%           En todos los casos, si el argumento de entrada fid es una cadena, abre el archivo
%           con un llamado a FileRead y el argumento de salida fid es el identificador del
archivo
%
%           Ver también: FileRead, FileMatrixAdd, FileMatrixHeader, FileMatrixVector.

%           © 2000, Pontificia Universidad Javeriana, Bogotá
%           Escrito por
%           Carlos Andrés Moncayo Ramírez
%           Carlos Eugenio Thompson Pinzón
if ischar(arg1)
    fid=FileRead(arg1);
else
    fid=arg1;
end

if nargin<2
    M = FileGetNext(fid);
else
    switch arg2
    case 'all'
        status = fseek(fid,16,'bof');
        M={}; I=0;
        while ~feof(fid),
            I=I+1;
            M{I} = FileGetNext(fid);
        end
    case 'remind'
        M={}; I=0;
        while ~feof(fid),
            I=I+1;
            M{I} = FileGetNext(fid);
        end
    otherwise
        N=arg2;
        I=0;
        while I<N,
            FileJumpNext(fid);
        end
        M = FileGetNext(fid);
    end
end

function [MM,status]=FileGetNext(fid)
status=0;
[ms,c]=fread(fid,[1,2],'uint32'); if c<2, status=-1;end
[bs,c]=fread(fid,1,'uint16'); if c<1, status=-1;end
[fl,c]=fread(fid,1,'uint16'); if c<1, status=-1;end
[ds,c]=fread(fid,1,'uint32'); if c<1, status=-1;end

fmt=FileFlagFmt(bs,fl);

[MM,c]=fread(fid,ms,fmt); if c<prod(ms), status=-1;end
return

function status=FileJumpNext(fid)
[ms,c]=fread(fid,[1,2],'uint32'); if c<2, status=-1;end
[bs,c]=fread(fid,1,'uint16'); if c<1, status=-1;end
[fl,c]=fread(fid,1,'uint16'); if c<1, status=-1;end
[ds,c]=fread(fid,1,'uint32'); if c<1, status=-1;end
if status==0,
    status = fseek(fid,ds,'cof');
end
return

```

G.3 Archivo NewFile.m, crea un archivo MTh nuevo.

```
function [fid,msg]=NewFile(FileNameBase,FileType)
%   Función para crear un nuevo archivo MTh
%   [fid,msg]=NewFile(FNB,FT)
%   Donde
%       FNB     es la base del nombre del archivo (nombre y path sin extensión)
%       FT      es el subtipo de archivo. Se añade a FileNameBase a modo de
extensión
%       fid     es el identificador de archivo
%       msg     posible mensaje de error si fid==-1
%
%   Ver también: FileRead, FileMatrixAdd, FileMatrixHeader, FileMatrixVector.
%
%   © 2000, Pontificia Universidad Javeriana, Bogotá
%   Escrito por
%       Carlos Andrés Moncayo Ramírez
%       Carlos Eugenio Thompson Pinzón
[fid,msg]=fopen( [FileNameBase,',' ,FileType], 'wb' );

if fid>0
    fwrite(fid,['MTh' 0],'char');

    if length(FileType)<4
        FileType(4)=0;
    end
    fwrite(fid,FileType(1:4),'char');

    t=now; tv=datevec(t); tvb=double(fliplr(bcd(tv)));
    fwrite(fid,0,'uint8');
    fwrite(fid, tvb(1:5), 'uint8');
    fwrite(fid, tvb(6), 'uint16');
else
    if isempty(msg),
        msg = ['No se puede abrir ',FileNameBase,',' ,FileType,' para escritura'];
    end
end
end
```

G.4 Archivo FileMatrixAdd.m, adiciona una matriz en un archivo MTh.

```
function fid = FileMatrixAdd(fid,M,fmt)
%   Función para escribir una matriz incluyendo encabezado y datos
%   fid = FileMatrixAdd(fid,M,fmt)
%   Donde
%       fid     Identificador del archivo
%       M       Matriz a guardar
%       fmt     Formato de datos
%   Para una descripción de los formatos de datos, ver FREAD.
%
%   Si el argumento de entrada fid es una cadena, hace un llamado
%   a NewFile usando 'MTh' como subtipo. El argumento de salida fid
%   es el identificador del archivo abierto
%
%   Ver también: NewFile, FileMatrixHeader, FileMatrixVector, FileGet.
%
%   © 2000, Pontificia Universidad Javeriana, Bogotá
%   Escrito por
%       Carlos Andrés Moncayo Ramírez
%       Carlos Eugenio Thompson Pinzón
fid = FileMatrixHeader(fid,size(M),fmt);
fwrite(fid,M,fmt);
```


G.5 Archivo FileMatrixHeader. m, escribe el encabezado de una matriz en un archivo MTh.

```
function fid = FileMatrixHeader(arg1,arg2,arg3,arg4)
%   Función para escribir el encabezado de una matriz en un archivo MTh
%       fid=FileMatrixHeader(fid,H,W,fmt)
%       fid=FileMatrixHeader(fid,[H,W],fmt)
%   Donde
%       fid   Identificador del archivo
%       H     Número de filas (altura) de la matriz
%       W     Número de columnas (ancho) de la matriz
%       fmt   Formato de datos
%   Para una descripción de los formatos de datos, ver FREAD.
%
%   Si el argumento de entrada fid es una cadena, hace un llamado
%   a NewFile usando 'MTh' como subtipo. El argumento de salida fid
%   es el identificador del archivo abierto
%
%   Ver también: NewFile, FileMatrixVector, FileMatrixAdd, FileGet.
%
%   © 2000, Pontificia Universidad Javeriana, Bogotá
%   Escrito por
%       Carlos Andrés Moncayo Ramírez
%       Carlos Eugenio Thompson Pinzón
if ischar(arg1)
    fid=NewFile(arg1,'MTh');
else
    fid=arg1;
end

if nargin<4,
    fmt=arg3;
    W=arg2(2);
    H=arg2(1);
else
    fmt=arg4;
    W=arg3;
    H=arg2;
end

[bsize,flfmt] = FileFmtFlag(fmt);

fwrite(fid,[H,W],'uint32');
fwrite(fid,[bsize,flfmt],'uint16');
fwrite(fid,H*W*bsize,'uint32');
```

G.6 Archivo FileMatrixVector. m, escribe una columna de una matriz en un archivo MTh.

```
function fid = FileMatrixVector(fid,vector,fmt)
%   Función para escribir datos en un archivo MTh, una vez se haya escrito el encabezado
%       fid=FileMatrixVector(fid,V,fmt)
%   Donde
%       fid   Identificador del archivo
%       V     Vector a guardar
%       fmt   Formato de datos
%   Para una descripción de los formatos de datos, ver FREAD.
%
%   Si el argumento de entrada fid es una cadena, se traduce como un llamado
%   a FileMatrixAdd. El argumento de salida fid es el identificador del archivo
%   abierto
%
%   Ver también: NewFile, FileMatrixAdd, FileMatrixHeader, FileMatrixAdd, FileGet.
```

```

%      © 2000, Pontificia Universidad Javeriana, Bogotá
%      Escrito por
%          Carlos Andrés Moncayo Ramírez
%          Carlos Eugenio Thompson Pinzón
if ischar(fid),
    fid=FileMatrixAdd(fid,vector,fmt);
else
    fwrite(fid,vector,fmt);
end

```

G.7 Archivo FileFlagFmt.m, genera los parámetros de almacenamiento para un archivo mth del formato de una matriz a guardar.

```

function fmt=FileFlagFmt(bs,fl)
%      Retorna una cadena de formato, dado el tamaño de los datos y las banderas
%          fmt=FileFlagFmt(bs,fl)
%      donde
%          bs      tamaño en bytes de los datos
%          fl      banderas: 0=entero, 1=punto flotante, 2=sin signo
%          fmt      Cadena de formato. Ver FREAD
%
%      Ver también FileGet, FileFmtFlag
%
%      © 2000, Pontificia Universidad Javeriana, Bogotá
%      Escrito por
%          Carlos Andrés Moncayo Ramírez
%          Carlos Eugenio Thompson Pinzón
base = 'int';
if bitand(fl,4)
    base = 'char';
end
if bitand(fl,2)
    base = ['u' base];
end
if bitand(fl,1)
    base = 'float';
end

tail = num2str(ceil(bs)*8,'%2d');

fmt = [base tail];

```

G.8 Archivo FileFmtFlag.m, devuelve el formato en forma de cadena de la información contenida en un archivo MTh.

```

function [bs,fl]=FileFmtFlag(fmt)
%      Retorna una cadena de formato, dado el tamaño de los datos y las banderas
%          [bs,fl]=FileFmtFlag(fmt)
%      donde
%          fmt      Cadena de formato. Ver FREAD
%          bs      tamaño en bytes de los datos
%          fl      banderas: 0=entero, 1=punto flotante, 2=sin signo
%
%      Ver también FileMatrixHeader, FileMatrixAdd, FileFlagFmt
%
%      © 2000, Pontificia Universidad Javeriana, Bogotá
%      Escrito por
%          Carlos Andrés Moncayo Ramírez
%          Carlos Eugenio Thompson Pinzón
bs=8;
fl=1;

efmt=fmt((length(fmt)-1):length(fmt));

```

```

switch efmt,
case {'t8', '*1', 'ar'},
    bs=1;
case {'16', '*2', 'rt'},
    bs=2;
case {'32', '*4', 'nt'},
    bs=4;
case {'64', '*8'},
    bs=8;
case 'ng',
    bs=4; %cambiar a 8 para compatibilidad con procesadores de 64 bits
end

switch fmt(1:2),
case {'in', 'si'},
    fl=0;
case {'ui', 'us', 'ul', 'un'},
    fl=2;
case {'fl', 're', 'do'},
    fl=1;
case {'ch', 'sc'},
    fl=4;
case 'uc',
    fl=6;
end

```

G.9 archivo bcd. m, programa para convertir de decimal a BCD.

```

function y=bcd(x)
uni=floor(mod(x,10));
dec=floor(mod(x/10,10));
cen=floor(mod(x/100,10));
mil=floor(mod(x/1000,10));
y=(uni+16*dec+256*cen+4096*mil);
%y=int32(uni+16*dec+256*cen+4096*mil);

```

NEXO H. PROGRAMAS DE MATLAB PARA EL PROCESO DE ENTRENAMIENTO Y PRUEBAS DEFINITIVOS

H.1 archivo `aciert_error.m`, programa para calcular las matrices de aceptación y rechazo.

```
function [ac,re,in]=aciert_error(T,Tsim,delta),
if nargin<3,
    delta=0.2;
end
Tac=(Tsim>1-delta);
Tre=(Tsim<delta);
Tin=1-Tac-Tre;
ac=T*Tac';
re=T*Tre';
in=T*Tin';
```

H.2 archivo `entae.m`, programa para entrenar una red de dos salidas con los fonemas /a/ y /e/.

```
%function entae(tipo,trans,n)
%
% Función entae: crea una red feedforward para el entrenamiento de los fonemas /a/ y /e/
%
%     tipo = tipo de extraccion de características: 'aa' 'au' 'na' 'nu'
%     trans = transformada utilizada para el procesamiento de los datos
%     n = numero del archivo de entrenamiento y por consiguiente consecutivo de la red
%
% Archivos relacionados:
%
%     entae entae5 entae5x5 entae5x5x4
%
function entae(tipo,trans,n)
file=[tipo,'aee',num2str(n)];
ae=file2matrix([file, '.',trans]);
aetgt=file2matrix([file, '.tgt']);
if trans=='wlet'
    aecoeff=matwlet2wcoef(ae);
elseif trans=='ceps'
    aecoeff=ae(2:15,:);
end
ta='tansig';
lo='logsig';
eval(['net' file trans '=newff(minmax(aecoeff),[10,15,2],{ta ta lo});']);
eval(['net' file trans '.trainparam.show=2;']);
eval(['net' file trans '.trainparam.min_grad=1e-12;']);
tic
eval(['net' file trans ',TR]=train(net' file trans ',aecoeff,aetgt);'])
```

```
t=toc
save(['net',file,trans],['net',file,trans],'t','TR')
```

H.3 archivo entae4. m, programa para automatizar entae. m en combinaciones de selección y ajuste de ventanas.

```
function entae4(n,trans),
['red_aa_ae_',num2str(n)]
entae('aa',trans,n);
['red_au_ae_',num2str(n)]
entae('au',trans,n);
['red_na_ae_',num2str(n)]
entae('na',trans,n);
['red_nu_ae_',num2str(n)]
entae('nu',trans,n);
```

H.4 archivo entae5. m, programa para automatizar entae. m por las cinco instancias de entrenamiento.

```
function entae5(tipo,trans),
for i=1:5,
    ['red_',tipo,'_aee_',num2str(i)]
    entae(tipo,trans,i);
end
```

H.5 archivo entae5x4. m, producto cartesiano de entae4. m y entae5. m.

```
function entae5x4(trans),
for i=1:5,
    entae4(i,trans);
end
```

H.6 archivo entptkceps. m, programa para entrenar una red de tres salidas con los fonemas /p/, /t/ y /k/ con cepstro.

```
function entptkceps(tipo,n)
file=[tipo,'ptke',num2str(n)];
ptkeceps=file2matrix([file,'.ceps']);
ptkecepscoef=ptkeceps(2:15,:);
ptketgt=file2matrix([file,'.tgt']);
ta='tansig';
lo='logsig';
eval(['net' file 'ceps' '=newff(minmax(ptkecepscoef),[10,15,3],{ta ta lo});']);
eval(['net' file 'ceps' '.trainparam.show=2;']);
eval(['net' file 'ceps' '.trainparam.min_grad=1e-12;']);
tic
eval(['net' file 'ceps,TR' '=train(net' file 'ceps,ptkecepscoef,ptketgt);'])
t=toc
save(['net',file,'ceps'],['net',file,'ceps'],'t','TR')
```

H.7 archivo entptkceps4. m, programa para automatizar entptkceps. m.

```
function entptkceps4(n),
```

```

['red_aa_ptke_',num2str(n)]
entptkceps('aa',n);
['red_au_ptke_',num2str(n)]
entptkceps('au',n);
['red_na_ptke_',num2str(n)]
entptkceps('na',n);
['red_nu_ptke_',num2str(n)]
entptkceps('nu',n);

```

H.8 archivo entptkceps5. m, programa para automatizar entptkceps. m.

```

function entptkceps5(tipo),
for i=1:5,
    ['red_',tipo,'_ptke_',num2str(i)]
    entptkceps(tipo,i);
end

```

H.9 archivo entptkceps5x4. m, programa para automatizar entptkceps. m.

```

function entptkceps5x4,
for i=1:5,
    entptkceps4(i);
end

```

H.10 archivo entptkwlet. m, programa para entrenar una red de tres salidas con los fonemas /p/, /t/ y /k/ con onditas.

```

function entptkwlet(file)
ptkewlet=file2matrix([file,'.wlet']);
ptkewcoef=matwlet2wcoef(ptkewlet);
ptketgt=file2matrix([file,'.tgt']);
ta='tansig';
lo='logsig';
eval(['net' file 'wlet' '=newff(minmax(ptkewcoef),[10,15,3],{ta ta lo});']);
eval(['net' file 'wlet' '.trainparam.show=2;']);
eval(['net' file 'wlet' '.trainparam.min_grad=1e-12;']);
tic
eval(['net' file 'wlet,TR' '=train(net' file 'wlet,ptkewcoef,ptketgt);'])
t=toc
save(['net',file,'wlet'],['net',file,'wlet'],'t','TR')

```

H.11 archivo entptkwlet5x4. m, programa para automatizar entptkwlet. m.

```

function entptkwlet5x4,
for i=1:5,
    ['red_aaptke',num2str(i)]
    entptkwlet(['aaptke',num2str(i)]);
    ['red_auptke',num2str(i)]
    entptkwlet(['auptke',num2str(i)]);
    ['red_naptke',num2str(i)]
    entptkwlet(['naptke',num2str(i)]);
    ['red_nuptke',num2str(i)]
    entptkwlet(['nuptke',num2str(i)]);
end

```

H.12 archivo perfsi mptk. m, programa para calcular el MSE de la simulación.

```
%function perf=perfsimptk(file,trans,n),
%
%   file = Nombre del archivo que se pretende simular por la red 'aaptkp1' o 'aaptke2'
%   trans = transformada de la red a evaluar 'wlet' o 'ceps'
%   n     = Número de la red a evaluar ej: netaaptk2 donde n=2
%
%   perf = MSE calculado por prueba por red
%
% Archivos relacionados:
%   perfsimptk5 perfsimptk5x5 perfsimptk5x5x4
%
function perf=perfsimptk(file,trans,n),
ptk=file2matrix([file,',' ,trans]);
ptktgt=file2matrix([file,',' ,tgt]);
if trans=='wlet'
    ptkcoef=matwlet2wcoef(ptk);
elseif trans=='ceps'
    ptkcoef=ptk(2:15,:);
end
[i,j]=size(file);
eval(['load net' file(1:j-2) 'e' num2str(n) trans]);
eval(['ptksim=sim(net' file(1:j-2) 'e' num2str(n) trans ',' ,ptkcoef);']);
perf(2,1)=mse(ptktgt-ptksim);
```

H.13 archivo perfsi mptk5. m, programa para automatizar perfsi mptk. m.

```
%function PERF=perfsimptk5(file,trans,n)
%
% Función perfsimptk5: Recopila todos(5) los errores MSE de las
%                     simulaciones de los archivos de pruebas 'p'
%
%   file = Nombre del archivo a probar puede ser de entrenamiento: terminado en 'e'
%           o de pruebas: terminado en 'p' ej: 'aaptkp' o 'nuptke'
%   trans = Transformada utilizada: 'wlet' o 'ceps'
%   n     = Número de la red entrenada a evaluar
%
%   PERF = Matriz de 15x1 con los resultados de los MSE de cada una de las 5 redes
%           y los demás van ceros.
%
% Archivos relacionados:
%   perfsimptk5 perfsimptk5x5 perfsimptk5x5x4
%
function PERF=perfsimptk5(file,trans,n),
PERF=[];
for i=1:5,
    perf=perfsimptk([file,num2str(i)],trans,num2str(n));
    PERF=[PERF;perf];
end
```

H.14 archivo perfsi mptk5x5. m, programa para automatizar perfsi mptk. m.

```
%function perfsimptk5x5(file,trans)
%
% funcion perfsimptk5x5: Recopila los datos de MSE de todas las redes
%                       especificadas en file y trans.
%
%   file = ej: 'aaptkp'
%   trans = ej: 'wlet' o 'ceps'
%
% Archivos relacionados:
%   perfsimptk5 perfsimptk5x5 perfsimptk5x5x4
%
```

```
function perfsimptk5x5(file,trans),
for i=1:5,
    m=perfsimptk5(file,trans,i);
    eval([file trans num2str(i) 'm'])
end
```

H.15 archivo perfsimptk5x5x4. m, programa para automatizar perfsimptk. m.

```
%function perfsimptk5x5x4(clase,fon,trans),
%
% Función perfsimptk5x5x4: Recopila todos los datos de error MSE de la
% simulacion con los 5 archivos de cada uno de
% los tipos en la clase indicada: 'p' o 'e'
%
% clase = Pruebas o entrenamiento: 'p' o 'e'
% fon = Fonemas: 'ae' o 'ptk'
% trans = Transformada: 'wlet' o 'ceps'
%
% Archivos relacionados:
%
% perfsimptk5 perfsimptk5x5 perfsimptk5x5x4
%
function perfsimptk5x5x4(clase,fon,trans),
perfsimptk5x5(['aa',fon,clase],trans)
perfsimptk5x5(['au',fon,clase],trans)
perfsimptk5x5(['na',fon,clase],trans)
perfsimptk5x5(['nu',fon,clase],trans)
```

H.16 archivo rec. m, programa para recuperar los datos de entrenamiento en una red almacenada.

```
%function [Ep,Pe,Ti]=rec(file),
%
% file = Nombre de archivo de red: 'netaaptkelwlet' 'netnuae5ceps'
%
% Ep = epocas de entrenamiento
% Pe = performance error: MSE último
% Ti = tiempo que tardó la red en llegar a esas epocas
%
function [Ep,Pe,Ti]=rec(file),
eval(['load ' file ' TR t']);
[a,b]=size(TR.epoch);
Ep=TR.epoch(b);
Pe=TR.perf(b);
Ti=t;
```

H.17 archivo rec5. m, programa para automatizar rec. m.

```
%function ept=rec5(tipo,fon,trans,n),
%
% tipo = Tipo de escogencia de los archivos: 'aa' 'au' 'na' 'nu'
% fon = Fonemas: 'ae' 'ptk'
% trans = Transformada utilizada 'wlet' o 'ceps'
% n = Número de archivo de red: (1 a 5)
%
function ept=rec5(tipo,fon,trans,n),
Ep=[];
Ti=[];
Pe=[];
num=[];
for i=1:n
    [e,p,t]=rec(['net',tipo,fon,'e',num2str(i),trans]);
```



```

    num=[num;i];
    Ep=[Ep;e];
    Ti=[Ti;t];
    Pe=[Pe;p];
    end
format short g
ept=[num Ep Pe Ti];

```

H.18 archivo rec5x.m, programa para automatizar rec. m.

```

function E=rec5x(tipo,trans,n),
Ti=[];
num=[];
for i=1:n
    t=recx(['net',tipo,'ptke',num2str(i),trans]);
    num=[num;i];
    Ti=[Ti;t];
end
format short g
E=[num Ti];

```

H.19 archivo rec5x4.m, programa para automatizar rec. m.

```

%function [aa,au,na,nu]=rec5x4(trans,fon,n),
%
%      trans = transformada utilizada: 'wlet' 'ceps'
%      fon   = fonemas: 'ae' o 'ptk'
%      n     = Número de archivo de red (1 a 5)
%
function [aa,au,na,nu]=rec5x4(trans,fon,n),
aa=rec5('aa',fon,trans,n);
au=rec5('au',fon,trans,n);
na=rec5('na',fon,trans,n);
nu=rec5('nu',fon,trans,n);
save(['rec' fon '4x5' trans])

```

H.20 archivo rec5x4x.m, programa para automatizar rec. m.

```

function [aa,au,na,nu]=rec5x4x(trans,n),
aa=rec5x('aa',trans,n);
au=rec5x('au',trans,n);
na=rec5x('na',trans,n);
nu=rec5x('nu',trans,n);
save(['recptk4x5' trans])

```

H.21 archivo simptk.m, programa para simular una red con un conjunto de pruebas.

```

%function [ac,re,in]=simptk(file,trans,n),
%
%      file = Nombre del archivo que se pretende simular por la red 'aaptkpl' o 'aaptke2'
%      trans = transformada utilizada en la red a evaluar 'wlet' o 'ceps'
%      n     = Número de la red a evaluar ej: netaaptk2 donde n=2
%
%      ac = Matriz de aciertos.
%      re = Matriz de rechazos.
%      in = Matriz de indecisiones.
%
%archivos relacionados:
%      simptk5 simptk5x5 simptk5x5x4
%

```

```

function [ac,re,in]=simptk(file,trans,n),
ptk=file2matrix([file,',' ,trans]);
ptktgt=file2matrix([file,',' ,tgt]);
if trans=='wlet'
    ptkcoef=matwlet2wcoef(ptk);
elseif trans=='ceps'
    ptkcoef=ptk(2:15,:);
end
[i,j]=size(file);
eval(['load net' file(1:j-2) 'e' num2str(n) trans]);
eval(['ptksim=sim(net' file(1:j-2) 'e' num2str(n) trans ',ptkcoef);']);
[ac,re,in]=acierto_error(ptktgt,ptksim,.3);

```

H.22 archivo si mptk5. m, programa para automatizar si mptk. m.

```

%function res=simptk5(file,trans,n)
%
%funcion simptk5: Recopila todas(5) las simulaciones de los archivos de pruebas 'p'
%
%    file = Nombre del archivo a probar puede ser de entrenamiento: terminado en 'e'
%           o de pruebas: terminado en 'p' ej: 'aaptkp' o 'nuptke'
%    trans = Tipo de transformada utilizada en la red entrenada a
%            evaluar: 'wlet' o 'ceps'
%    n      = Numero de la red entrenada a evaluar
%
%    res    = Matriz de 15x9 van aciertos rechazos e indecisiones pegados verticalmente
%
%Archivos relacionados:
%    simptk5 simptk5x5 simptk5x5x4
%
function res=simptk5(file,trans,n),
AC=[];
RE=[];
IN=[];
for i=1:5,
    [ac,re,in]=simptk([file,num2str(i)],trans,num2str(n));
    AC=[AC;ac];
    RE=[RE;re];
    IN=[IN;in];
end
res=[AC RE IN];

```

H.23 archivo si mptk5x5. m, programa para automatizar si mptk. m.

```

%function simptk5x5(file,trans)
%
%funcion simptk5x5: Recopila los datos de todas las redes especificadas en file y trans ej:
%
%    file = ej: 'aaptkp'
%    trans = ej: 'wlet' o 'ceps'
%
%Archivos relacionados:
%    simptk5 simptk5x5 simptk5x5x4
%
function simptk5x5(file,trans),
for i=1:5,
    m=simptk5(file,trans,i);
    eval([file trans num2str(i) '=m'])
end

```

H.24 archivo si mptk5x5x4. m, programa para automatizar si mptk. m.

```

%function simptk5x5x4(clase,fon,trans),
%

```

```

% funcion simptk5x5x4: Recopila todos los datos de la simulacion
%                   con los 5 archivos de cada uno de los tipos
%                   en la clase indicada: 'p' o 'e'
%
%                   clase = especifica el conjunto con el cual se va a simular
%                   'p' : conjunto pruebas y 'e' : conjunto entrenamiento
%                   trans = 'wlet' o 'ceps'
%                   fon   = 'ae' o 'ptk'
%
% Archivos relacionados:
%
%                   simptk5 simptk5x5 simptk5x5x4
%
function simptk5x5x4(clase,fon,trans),
simptk5x5(['aa',fon,clase],trans)
simptk5x5(['au',fon,clase],trans)
simptk5x5(['na',fon,clase],trans)
simptk5x5(['nu',fon,clase],trans)

```

H.25 archivo wnd2ceps. m, programa para automatizar wnd2ceps. m.

¡Error!Imposible abrir el archivo.

H.26 archivo wnd2ceps5x16. m, programa para automatizar wnd2ceps. m.

```

function wnd2ceps5x16,
for i=1:5,
    wnd2ceps(['aaptke',num2str(i)]);
    ['aaptke',num2str(i)]
    wnd2ceps(['aaptkp',num2str(i)]);
    ['aaptkp',num2str(i)]
    wnd2ceps(['auptke',num2str(i)]);
    ['auptke',num2str(i)]
    wnd2ceps(['auptkp',num2str(i)]);
    ['auptkp',num2str(i)]
    wnd2ceps(['naptke',num2str(i)]);
    ['naptke',num2str(i)]
    wnd2ceps(['naptkp',num2str(i)]);
    ['naptkp',num2str(i)]
    wnd2ceps(['nuptke',num2str(i)]);
    ['nuptke',num2str(i)]
    wnd2ceps(['nuptkp',num2str(i)]);
    ['nuptkp',num2str(i)]
    wnd2ceps(['aaeee',num2str(i)]);
    ['aaeee',num2str(i)]
    wnd2ceps(['aaaep',num2str(i)]);
    ['aaaep',num2str(i)]
    wnd2ceps(['auaee',num2str(i)]);
    ['auaee',num2str(i)]
    wnd2ceps(['auaep',num2str(i)]);
    ['auaep',num2str(i)]
    wnd2ceps(['naeee',num2str(i)]);
    ['naeee',num2str(i)]
    wnd2ceps(['naaep',num2str(i)]);
    ['naaep',num2str(i)]
    wnd2ceps(['nuaaa',num2str(i)]);
    ['nuaaa',num2str(i)]
    wnd2ceps(['nuaaep',num2str(i)]);
    ['nuaaep',num2str(i)]
end

```

H.27 archivo wnd2wl et. m, programa para automatizar wnd2wl et. m.

¡Error!Imposible abrir el archivo.

H.28 archivo wnd2wl et 5x16. m, programa para automatizar wnd2wl et. m.

```
function wnd2wlet5x16,
for i=1:5,
    wnd2wlet(['aaptke',num2str(i)]);
    ['aaptke',num2str(i)]
    wnd2wlet(['aaptkp',num2str(i)]);
    ['aaptkp',num2str(i)]
    wnd2wlet(['auptke',num2str(i)]);
    ['auptke',num2str(i)]
    wnd2wlet(['auptkp',num2str(i)]);
    ['auptkp',num2str(i)]
    wnd2wlet(['naptke',num2str(i)]);
    ['naptke',num2str(i)]
    wnd2wlet(['naptkp',num2str(i)]);
    ['naptkp',num2str(i)]
    wnd2wlet(['nuptke',num2str(i)]);
    ['nuptke',num2str(i)]
    wnd2wlet(['nuptkp',num2str(i)]);
    ['nuptkp',num2str(i)]
    wnd2wlet(['aaaae',num2str(i)]);
    ['aaaae',num2str(i)]
    wnd2wlet(['aaaep',num2str(i)]);
    ['aaaep',num2str(i)]
    wnd2wlet(['auaee',num2str(i)]);
    ['auaee',num2str(i)]
    wnd2wlet(['auaep',num2str(i)]);
    ['auaep',num2str(i)]
    wnd2wlet(['naaee',num2str(i)]);
    ['naaee',num2str(i)]
    wnd2wlet(['naaep',num2str(i)]);
    ['naaep',num2str(i)]
    wnd2wlet(['nuaaee',num2str(i)]);
    ['nuaaee',num2str(i)]
    wnd2wlet(['nuaaep',num2str(i)]);
    ['nuaaep',num2str(i)]
end
```

H.29 archivo Wl et 2Wcoef. m, programa para automatizar wnd2ceps. m.

```
function Wlet2Wcoef(file)
y=file2matrix([file, '.wlet']);
newfile=[file, '.Wcoef'];

load l_1024_db03;
l=l_1024_db03;

ncoef=2;

% ncoef Es el numero de coeficientes que se quieren,
% el superior incluye los anteriores.
% 1 en energía
% 2 st desviación estándar
% 3 mx máximo
% 4 mn mínimo
% 5 me media

%formato = 'int16'; bsize = 2; escala=10000; flfmt=0;
%formato = 'int32'; bsize = 4; escala=1e+8; flfmt=0;
formato = 'float32'; bsize = 4; escala=1; flfmt=1;
%formato = 'float64'; bsize = 8; escala=1; flfmt=1;

[fid,msg]=fopen(newfile,'wb');
fwrite(fid,['MTh' 0], 'char');
fwrite(fid,'wcoe', 'char');

t=now; tv=datevec(t); tvb=double(fliplr(bcd(tv)));
fwrite(fid,0,'uint8');
```

```

fwrite(fid, tvb(1:5), 'uint8');
fwrite(fid, tvb(6), 'uint16');

vec=[];
jj=0;
for ii=1:length(l)-1,
    x = y(jj+1:(jj+1(ii)),:);
    en = sqrt(sum(x.*x))/l(ii);
    st = std(x,1);
    mx = max(x);
    mn = min(x);
    me = mean(x);
    if ncoef==5,
        vec=[vec;en;st;mx;mn;me];
    end
    if ncoef==4,
        vec=[vec;en;st;mx;mn];
    end
    if ncoef==3,
        vec=[vec;en;st;mx];
    end
    if ncoef==2,
        vec=[vec;en;st];
    end
    jj = jj+1(ii);
end

[files,cols] = size(vec);
fwrite(fid,[files,cols], 'uint32');
fwrite(fid,[bsize,flfmt], 'uint16');
fwrite(fid,bsize*files*cols,formato);
fwrite(fid,vec,formato);

fclose(fid);

```

