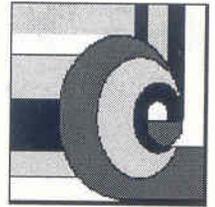




**INSTITUTO POLITECNICO NACIONAL**  
**CENTRO DE INVESTIGACION EN COMPUTACION**



I. P. N.  
CENTRO DE INVESTIGACION EN  
COMPUTACION  
BIBLIOTECA

Estudio estadístico comparativo entre Memorias Asociativas  
Clásicas, Memorias Morfológicas y Memorias Alfa-Beta  
para el caso binario

T E S I S

que para obtener el grado de

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

presenta

**JULIO CÉSAR SALGADO RAMÍREZ**

Director de tesis: Dr. Cornelio Yáñez Márquez  
Codirector de tesis: Dr. Juan Luis Díaz de León Santiago

## Índice de Gráficas.

|  |     |
|--|-----|
| gráfica 5.1 Patrones recuperados por las memorias asociativas para el conjunto fundamental 1 sin ruido.....                                | 71  |
| gráfica 5.2. Recuperación de patrones por conjunto fundamental sin ruido.....  | 74  |
| gráfica 5.3. Porcentaje de efectividad en la recuperación de patrones por conjunto fundamental sin ruido.....                              | 75  |
| gráfica 5.4. Patrones recuperados por las memorias ante el ruido aditivo para el conjunto fundamental 1.....                               | 76  |
| gráfica 5.5. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1 afectado con ruido aditivo.....       | 77  |
| gráfica 5.6. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1 afectado con ruido sustractivo..... | 79  |
| gráfica 5.7. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1 afectado con ruido sustractivo.....   | 80  |
| gráfica 5.8. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1 afectado con ruido mezclado.....    | 81  |
| gráfica 5.9. Porcentaje de efectividad en la recuperación de patrones afectados con ruido mezclado.....                                    | 82  |
| gráfica 5.10. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo.....                                 | 83  |
| gráfica 5.11. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 2 afectado con ruido aditivo.....      | 84  |
| gráfica 5.12. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo aditivo.....                         | 85  |
| gráfica 5.13. Porcentaje de efectividad de patrones recuperados para el conjunto fundamental 2 afectado con ruido sustractivo.....         | 86  |
| gráfica 5.14. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido mezclado.....                                | 87  |
| gráfica 5.15. Porcentaje de efectividad de las memorias heteroasociativas en el conjunto fundamental 2 afectado con ruido mezclado.....    | 88  |
| gráfica 5.16 Total de patrones recuperados por las memorias heteroasociativas en el conjunto fundamental 3 afectado con ruido aditivo..... | 89  |
| gráfica 5.17 Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 3 afectado con ruido aditivo.....       | 90  |
| gráfica 5.18 Total de patrones recuperados en el conjunto fundamental 3 afectado con ruido sustractivo.....                                | 91  |
| gráfica 5.19. Porcentajes de efectividad en la recuperación de patrones en el conjunto fundamental 3 afectado con ruido sustractivo.....   | 92  |
| gráfica 5.20. Total de patrones recuperados en el conjunto fundamental 3 alterado con ruido mezclado.....                                  | 93  |
| gráfica 5.21. Porcentaje de efectividad de patrones recuperados en el conjunto fundamental 3 alterado con ruido mezclado.....              | 94  |
| gráfica 5.22. Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido aditivo.....                                   | 95  |
| gráfica 5.23. Porcentaje de efectividad de recuperación de patrones en el conjunto fundamental 4 afectado con ruido aditivo.....           | 96  |
| gráfica 5.24 Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido sustractivo.....                                | 97  |
| gráfica 5.25. Porcentaje de efectividad en la recuperación de patrones del conjunto fundamental 4 afectado con ruido sustractivo.....      | 98  |
| gráfica 5.26. Total de patrones recuperados en el conjunto fundamental 4 afectado con ruido mezclado.....                                  | 99  |
| gráfica 5.27. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 4 afectado con ruido mezclado.....       | 100 |
| gráfica 5.28. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido aditivo.....                                   | 101 |
| gráfica 5.29. Porcentaje de efectividad de las memorias heteroasociativas para el conjunto fundamental 5 alterado con ruido aditivo.....   | 102 |
| gráfica 5.30. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido sustractivo.....                               | 103 |

|   |     |
|---|-----|
| gráfica 5.31. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 5 afectado por ruido sustractivo.....   | 104 |
| gráfica 5.32. Total de patrones recuperados en el conjunto fundamental 5 afectado con ruido mezclado.....   | 105 |
| gráfica 5.33. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 5 alterado con ruido mezclado.....  | 106 |
| gráfica 5.34. Total de patrones recuperados por la memoria morfológica min en todos los conjuntos fundamentales.....  | 107 |
| gráfica 5.35. Total de patrones recuperados por la memoria Alfa Beta min en todos los conjuntos fundamentales.....  | 107 |
| gráfica 5.36. Total de patrones recuperados por la memoria morfológica max en todos los conjuntos fundamentales.....  | 108 |
| gráfica 5.37. Total de patrones recuperados por la memoria Alfa Beta max en todos los conjuntos fundamentales.....  | 108 |
| gráfica 5.38. Total de patrones recuperados por la memoria morfológica heurística en todos los conjuntos fundamentales.....   | 109 |
| gráfica 5.39. Total de patrones recuperados por la memoria Alfa Beta heurística en todos los conjuntos fundamentales.....   | 109 |
| gráfica 5.40. Total de patrones recuperados por memorias autoasociativa para los conjuntos fundamentales sin ruido.....   | 113 |
| gráfica 5.41. Porcentaje de efectividad de recuperación de patrones obtenido por las memorias autoasociativas para todos los conjuntos fundamentales.....                         | 114 |
| gráfica 5.42. Total de patrones recuperados por las memorias autoasociativas para el conjunto fundamental 1 afectado con ruido aditivo.....                                       | 115 |
| gráfica 5.43. Porcentaje de efectividad en la recuperación de patrones afectados con ruido aditivo.....   | 116 |
| gráfica 5.44. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido sustractivo.....  | 118 |
| gráfica 5.45. Porcentaje de la efectividad de recuperación de patrones ante ruido sustractivo para el conjunto fundamental 1.....   | 118 |
| gráfica 5.46. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido mezclado.....   | 119 |
| gráfica 5.47. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 1 afectado por ruido mezclado.....  | 120 |
| gráfica 5.48. Total de patrones recuperados en el conjunto fundamental 2 afectado con ruido aditivo.....  | 121 |
| gráfica 5.49. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.....                | 123 |
| gráfica 5.50. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.....            | 124 |
| gráfica 5.51. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.....               | 125 |
| gráfica 5.52. Total de patrones recuperados por la memoria morfológica min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.....     | 126 |
| gráfica 5.53. Total de patrones recuperados por la memoria Alfa Beta min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.....       | 127 |
| gráfica 5.54. Total de patrones recuperados por la memoria morfológica max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.....         | 128 |
| gráfica 5.55. Total de patrones recuperados por la memoria Alfa Beta max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.....           | 129 |
| gráfica 5.56. Total de patrones recuperados por la memoria morfológica heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado..... | 130 |
| gráfica 5.57. Total de patrones recuperados por la memoria Alfa Beta heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.....   | 131 |
| gráfica 5.58. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación por modelos de memorias asociativas para 26 patrones.....                                | 136 |
| gráfica 5.59. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación de memorias asociativas para 9 patrones.....   | 137 |

## Índice de Tablas.

|   |     |
|---|-----|
| tabla 5.1. Total de patrones recuperados por intentos de recuperación, para el conjunto fundamental 1.....                                  | 71  |
| tabla 5.2. Recuperación de patrones por conjunto fundamental.....   | 73  |
| tabla 5.3. Porcentaje de efectividad en la recuperación de patrones por conjunto fundamental sin ruido.....                                 | 74  |
| tabla 5.4. Patrones recuperados con los diferentes porcentaje de ruido aditivo para el conjunto<br>fundamental 1.....                       | 76  |
| tabla 5.5. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1<br>afectado con ruido aditivo.....       | 77  |
| tabla 5.6. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1<br>afectado con ruido sustractivo..... | 78  |
| gráfica 5.7. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1<br>afectado con ruido sustractivo..... | 80  |
| tabla 5.8. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1<br>afectado con ruido mezclado.....    | 81  |
| tabla 5.9. Porcentaje de efectividad en la recuperación de patrones afectados con ruido mezclado.....                                       | 82  |
| tabla 5.10. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo.....                                    | 83  |
| tabla 5.11. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 2<br>afectado con ruido aditivo.....      | 84  |
| tabla 5.12. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo.....                                    | 85  |
| tabla 5.13. Porcentaje de efectividad de patrones recuperados para el conjunto fundamental 2 afectado con<br>ruido sustractivo.....         | 86  |
| tabla 5.14. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido mezclado.....                                   | 87  |
| tabla 5.15. Porcentaje de efectividad de las memorias heteroasociativas en el conjunto fundamental 2<br>afectado con ruido mezclado.....    | 88  |
| tabla 5.16 Total de patrones recuperados por las memorias heteroasociativas en el conjunto fundamental 3<br>afectado con ruido aditivo..... | 89  |
| tabla 5.17 Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 3<br>afectado con ruido aditivo.....       | 90  |
| tabla 5.18 Total de patrones recuperados en el conjunto fundamental 3 afectado con ruido sustractivo.....                                   | 91  |
| tabla 5.19. Porcentajes de efectividad en la recuperación de patrones en el conjunto fundamental 3<br>afectado con ruido sustractivo.....   | 92  |
| tabla 5.20. Total de patrones recuperados en el conjunto fundamental 3 alterado con ruido mezclado.....                                     | 93  |
| tabla 5.21. Porcentaje de efectividad de patrones recuperados en el conjunto fundamental 3 alterado con<br>ruido mezclado.....              | 94  |
| tabla 5.22. Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido aditivo.....                                      | 95  |
| tabla 5.23. Porcentaje de efectividad de recuperación de patrones en el conjunto fundamental 4 afectado<br>con ruido aditivo.....           | 96  |
| tabla 5.24 Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido sustractivo.....                                   | 97  |
| tabla 5.25. Porcentaje de efectividad en la recuperación de patrones del conjunto fundamental 4 afectado<br>con ruido sustractivo.....      | 98  |
| tabla 5.26. Total de patrones recuperados en el conjunto fundamental 4 afectado con ruido mezclado.....                                     | 99  |
| tabla 5.27. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 4 afectado<br>con ruido mezclado.....       | 100 |
| tabla 5.28. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido aditivo.....                                      | 101 |
| tabla 5.29. Porcentaje de efectividad de las memorias heteroasociativas para el conjunto fundamental 5<br>alterado con ruido aditivo.....   | 102 |
| tabla 5.30. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido sustractivo.....                                  | 103 |
| tabla 5.31. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 5 afectado<br>por ruido sustractivo.....    | 104 |
| tabla 5.32. Total de patrones recuperados en el conjunto fundamental 5 afectado con ruido mezclado.....                                     | 105 |
| tabla 5.33. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 5<br>alterado con ruido mezclado.....     | 106 |
| tabla 5.34 a. Total de patrones recuperados por las memorias heteroasociativas cuando $m \neq n$ .....                                      | 110 |
| tabla 5.34 b. Total de patrones recuperados por las memorias heteroasociativas cuando $m \neq n$ .....                                      | 111 |
| tabla 5.35. Recuperación de patrones por conjunto fundamental sin ruido.....  | 112 |

|   |     |
|---|-----|
| tabla 5.36. Porcentaje de efectividad de recuperación de patrones por las memorias autoasociativas para todos los conjuntos fundamentales.....                                  | 113 |
| tabla 5.37. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido aditivo.....  | 115 |
| tabla 5.38. Total de patrones recuperados para el conjunto fundamental 1 afectado con ruido sustractivo.....  | 117 |
| tabla 5.39. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido mezclado.....   | 119 |
| tabla 5.40. Total de patrones recuperados en el conjunto fundamental 2 afectado con ruido aditivo.....  | 121 |
| tabla 5.41. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.....                | 122 |
| tabla 5.42. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.....            | 123 |
| tabla 5.43. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.....               | 124 |
| tabla 5.44. Total de patrones recuperados por la memoria morfológica min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.....     | 125 |
| tabla 5.45. Total de patrones recuperados por la memoria Alfa Beta min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.....       | 126 |
| tabla 5.46. Total de patrones recuperados por la memoria morfológica max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.....         | 127 |
| tabla 5.47. Total de patrones recuperados por la memoria Alfa Beta max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.....           | 128 |
| tabla 5.48. Total de patrones recuperados por la memoria morfológica heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado..... | 129 |
| tabla 5.49. Total de patrones recuperados por la memoria Alfa Beta heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.....   | 130 |
| tabla 5.50. Tabla que muestra el total de recuperaciones perfectas de los patrones 1 al 9 con tipo de ruido aditivo aleatorio.....  | 134 |
| tabla 5.51. Tabla que muestra el total de recuperaciones perfectas de los patrones 1 al 9 con tipo de ruido sustractivo aleatorio.....  | 135 |
| tabla 5.52. Tabla que muestra el total de recuperaciones perfectas de los patrones 1 al 9 con tipo de ruido mezclado aleatorio.....   | 135 |
| tabla 5.53. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación por modelos de memorias asociativas para 26 patrones.....                                | 136 |
| tabla 5.54. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación de memorias asociativas para 9 patrones.....   | 137 |

## Índice de figuras.

|  |     |
|--|-----|
| figura 1.1 ejemplo de cómo se puede ver la memoria asociativa.....   | 1   |
| figura 1.2 Patrones afectados con aditivo, sustractivo y mezclado del 40%.....   | 3   |
| figura 4.1. Conjunto fundamental a procesar.....   | 37  |
| figura 4.2 resultado del algoritmo que genera el ruido aditivo.....  | 39  |
| figura 4.3 resultado del algoritmo que genera el ruido sustractivo.....  | 40  |
| figura 4.4 resultado del algoritmo que genera el ruido mezclado.....   | 41  |
| figura 4.5. Resultado del algoritmo de dilatación.....   | 42  |
| figura 4.6. Resultado del algoritmo de erosión.....  | 43  |
| figura 4.7. Diagrama que muestra el proceso de las memorias heurísticas.....   | 58  |
| figura 4.8. Interfase gráfica del estudio comparativo.....   | 60  |
| figura 4.9. Interfase del componente Chart.....  | 65  |
| figura 4.10. Interfase que muestra el tipo de gráficas que pueden usarse con el componente Chart.....  | 65  |
| Conjuntos fundamentales para memorias heteroasociativas.....   | 69  |
| figura 5.1. Apariencia de 15 patrones después de haber sido recuperados por las memorias de tipo min.....  | 72  |
| figura 5.2. Apariencia de 9 patrones después de haber sido recuperados por las memorias de tipo max.....   | 72  |
| figura 5.3. Apariencia del primer patrón recuperado por el Linear Associator.....  | 73  |
| figura 5.4. Porcentajes de ruido del 5,10,20,40,50,60,80,90 y 99% de ruido aditivo, sustractivo y mezclado respectivamente.....  | 75  |
| figura 5.4. Apariencia de 12 patrones recuperados por las memorias de tipo $\vee$ con un 40% de ruido aditivo.....   | 77  |
| figura 5.5. Apariencia de 12 patrones recuperados con las memorias heurísticas, después de haber estado afectados con el 40% de ruido sustractivo.....   | 79  |
| figura 5.6. Apariencia de patrones recuperados por las memorias heurísticas, después de haber estado afectados con el 60% de ruido mezclado.....   | 82  |
| Los cinco conjuntos fundamentales de las memorias autoasociativas.....   | 111 |
| figura 5.7. Apariencia de los 5 patrones recuperados por la memoria Hopfield.....  | 113 |
| figura 5.8. 10 patrones recuperados que estaban afectados con el 50% de ruido aditivo.....   | 116 |
| figura 5.9 Apariencia de 10 patrones recuperados con las memorias de tipo min después de que los patrones estaban afectados con el 90% de ruido sustractivo.....   | 117 |
| figura 5.10. Apariencia después de cuatro patrones recuperados por la memoria Hopfield.....  | 120 |
| figura 5.11. Patrones afectados con el 20% de ruido sustractivo.....   | 132 |
| figura 5.12. Patrones recuperados por las memorias de tipo min realizando la primera iteración.....  | 132 |
| figura 5.13. Patrones recuperados por las memorias de tipo min realizando la segunda iteración.....  | 133 |
| figura 5.14. Apariencia de los 4 primeros patrones recuperados por las memorias de tipo max.....   | 133 |
| figura 5.15. Apariencia de los patrones recuperados por las memorias de tipo max. Los patrones presentados como entrada son los patrones que recupero las memorias de tipo min.....  | 133 |
| figura 5.16. Apariencia de los patrones de salida, después de haber sido recuperados por las memorias de tipo min con un 20% de ruido aditivo., y ser nuevamente haber sido iterados con las memorias de tipo min y max..... | 134 |

## Resumen

En este trabajo de tesis se realiza un estudio experimental comparativo entre las memorias asociativas clásicas, las memorias asociativas morfológicas y las memorias asociativas alfa-beta, para el caso binario.

Ante la necesidad imperante de saber qué tipo de memoria asociativa se puede usar para ciertas aplicaciones de reconocimiento de patrones, es deseable conocer las ventajas con las que se cuenta al hacer uso de una memoria asociativa en particular; también, es deseable saber en qué momentos o situaciones específicas dicha memoria presentará problemas, o bajará su rendimiento. Teniendo a priori el conocimiento de las virtudes y defectos de las memorias asociativas, es posible hacer una selección adecuada.

Después de una exhaustiva investigación documental, fue evidente para el autor de esta tesis que en la actualidad no existen estudios experimentales comparativos para los modelos más importantes de memorias asociativas. El descubrimiento de este hecho dió lugar a la motivación para realizar este trabajo de tesis, donde se muestran tanto las virtudes como los defectos de los diferentes modelos de memorias asociativas y se detectan, además, casos particulares de patrones recuperados de manera no perfecta que puedan procesarse por algún método heurístico o por el mismo modelo de memoria asociativa que los arrojó, y hacer que estos patrones sean recuperados en forma perfecta. Pero el trabajo de tesis va más allá: se hace uso de métodos heurísticos para realizar recuperaciones perfectas donde fallan los modelos puros.

Para la realización de este estudio experimental comparativo se desarrolló un software que permite como entrada un conjunto fundamental típico de patrones, para que todos los modelos de memorias asociativas aprendan y realicen sus recuperaciones; el software muestra, además de los patrones de salida, gráficas que ilustran el rendimiento de cada modelo.

Como aportaciones adicionales de este trabajo de tesis, se desarrolló una biblioteca de funciones para los modelos de memorias asociativas clásicas, morfológicas y alfa-beta, y una biblioteca de funciones para el manejo de imágenes con formato BMP.

## Abstract.

In this thesis project a comparative experimental research is done among the classic associative memories, the morphologic associative memories and the alpha - beta associative memories for the binary case.

Because of the imperative need of knowing what type of associative memory can be used for certain applications of patten recognition, it is advisable to know the advantages which we count with to make usage of a particular associative memory; it is also good to know in what moments or specific situations this memory will show problems or will decrease its performance. Having the previous knowledge of the qualities and shortcomings of the associative memories, it is possible to make a convenient choice.

After an exhausting documental research, it was noticeable for the author of this thesis that nowadays no comparative experimental studies for the most important associative models exist; the discovery of this fact led to the motivation to make this project, where the qualities as well as the shortcomings of the different model of associative memories are shown; besides, particular cases of not perfectly recovered patterns which can be processed by a heuristic method or by the same associative memory model that recovered them are also detected. In addition to this fact, the project mentioned above makes possible for this patterns to be recovered in a perfect way. However this thesis work goes further than only this. In this work, heuristic methods are used to make perfect recoveries in which pure models would fail.

For this comparative experimental statistic study to be done, a software that allows as an entry a fundamental typical pattern set was developed in order for all associative memory patterns to learn and do the recoveries. The software shows, not only the exit patters but also graphics that illustrate the endurance of each model.

As additional benefits of this project, a function library was created for the classic associative memories, morphologic and alpha - beta, and a function library to manipulate BMP images.

# Capítulo 1

## Introducción

Las *memorias asociativas* han representado, a través del tiempo, un área importante dentro del reconocimiento de patrones. Existen muchos investigadores que han dedicado su tiempo a realizar innovaciones a los modelos de memorias asociativas existentes; otros, a la creación de nuevas memorias asociativas (Willshaw, Buneman & Longuet-Higgins, 1969; Amari, 1972; Anderson, 1972; Kohonen, 1972; Nakano, 1972; Kohonen & Ruohonen, 1973; Kohonen, 1974; Little & Shaw, 1975; Anderson, Silverstein, Ritz & Jones, 1977; Amari, 1977; Hopfield, 1982; Hopfield, 1984; Austin, 1987; Kanerva, 1988; Kolen & Pollack, 1991; Buhmann, 1995; Kinser, 1995; Bandyopadhyay & Datta, 1996; Aleksander & Morton, 1997; Austin, Buckle, Kennedy, Moulds, Pack & Turner, 1997; Ritter, Sussner, Díaz-de-León, 1988; Ritter, Díaz-de-León & Sussner, 1999) hasta el punto de implementar en chips algunas de éstas (Kennedy, Austin & Cass, 1995; Krikelis & Weems, 1997; Stringht, Coffield & Brooks 1998). Los investigadores han buscado, además, que las memorias asociativas sean de una capacidad lo mayor posible en el aprendizaje de patrones y que sean robustas ante diferentes tipos de ruido (Chen & Honovar, 1995; Graham & Willshaw, 1995; Imada & Araki, 1995; Adeodato & Tylor, 1996; Storkey, 1997; Bosch & Kurfess, 1998; Jagota, Narasimhan & Regan, 1998 ).

Una memoria asociativa tiene por objetivo: ”*recuperar de manera perfecta patrones, a partir de patrones de entrada, que quizá estén alterados con algún tipo de ruido*” (Hassoun, 1993); se puede ejemplificar como una caja negra que recibe de entrada un patrón  $x$ , procesa este patrón de entrada, y genera como resultado un patrón  $y$ .



figura 1.1 Esquema de una memoria asociativa vista como caja negra.

Es natural pensar que para formar  $M$ , el patrón de entrada  $x$  debe estar relacionado de alguna manera con el patrón de salida  $y$ . Para representar dicha relación entre estos patrones se usará la notación de una pareja ordenada  $(x, y)$ .

El patrón de entrada  $x$  y el patrón de salida  $y$  son vectores; la memoria asociativa debe ser capaz de aprender un conjunto de parejas ordenadas de patrones, y recuperar los de salida a partir de los de entrada. Debido a esta afirmación, la manera de representar al conjunto de todos los patrones que se usarán para conformar la memoria  $M$  es la siguiente:

$$\{(x^\omega, y^\omega) \mid \omega \in \{1, 2, \dots, p\}\} \quad (1.1)$$

La expresión 1.1 muestra que habrá un número finito de patrones para diseñar la memoria  $M$ ; en la misma expresión,  $p$  indica la cardinalidad del conjunto mostrado. Al conjunto finito de patrones denotado en la expresión 1.1 se le conocerá en adelante como: “conjunto fundamental de patrones”, y a sus elementos “patrones fundamentales”, los cuales pueden ser de entrada o de salida.

Para hacer referencia a algún elemento de un patrón  $x$  o de un patrón  $y$ , se usará la notación:

$$\begin{array}{l} x_j^\omega \\ y_j^\omega \end{array} \quad (1.2)$$

donde  $j$  representa el índice de la posición del elemento del patrón y  $\omega$ , el índice de pareja.

**Ejemplo:**

$x_3^1$  está haciendo referencia al tercer elemento del primer patrón de entrada  $x$  aprendido.

$y_1^7$  hace referencia al primer elemento del séptimo patrón de salida  $y$  recuperado.

La caja negra mostrada en la figura 1.1, denominada como  $M$ , será conocida en este trabajo indistintamente como “la matriz de aprendizaje  $M$ ” o “la memoria asociativa  $M$ ”.  $M$  contendrá la información codificada del conjunto fundamental, y esta misma, después que haya aprendido, servirá para ser operada de cierta manera con el patrón  $x$  que se presente de entrada, el cual puede estar alterado con algún tipo de ruido, y generará un patrón  $y$  de salida.

Con lo anterior mencionado, es posible afirmar que las memorias asociativas atacan un problema que consta de dos fases (Yáñez-Márquez, 2002):

1. *La fase de aprendizaje*
2. *La fase de recuperación.*

*La fase de aprendizaje* consiste en encontrar el o los operadores necesarios, para que de alguna manera se pueda codificar la relación que existe entre el patrón de entrada  $x$  y el patrón de salida  $y$ , y con dicha codificación se genere la matriz de aprendizaje  $M$ .

*La fase de recuperación* consiste en encontrar el operador o los operadores necesarios y las condiciones suficientes para generar un patrón de salida; es decir, una vez

que se ha conformado la matriz  $M$ , se presenta un patrón de entrada  $x$  que previamente se aprendió,  $M$  se opera con el operador o los operadores necesarios bajo ciertas circunstancias con el patrón  $x$ , y se genera un patrón de salida  $y$ .

A un patrón de entrada alterado con algún tipo de ruido, se le representará como  $\tilde{x}$ . Por ejemplo, la expresión  $\tilde{x}^\omega$  representa el patrón de entrada  $x^\omega$  alterado con algún tipo de ruido.

Anteriormente se mencionó el objetivo de las memorias asociativas. La pregunta que cabe hacer es ¿qué es una recuperación perfecta?. Se dice que una recuperación es perfecta si: al operar un vector de entrada  $\tilde{x}^\omega$  con la matriz de aprendizaje  $M$ , se genera como resultado el vector  $y^\omega$ , sin ruido, que le corresponde dentro del conjunto fundamental.

Cabe mencionar que si en la fase de recuperación todos los patrones son recuperados de manera perfecta, se dice que *la memoria es perfecta*.

La pregunta que aún está latente es ¿cuáles son los tipos de ruido que pueden alterar a un patrón de entrada? La respuesta está en la siguiente clasificación: tipo de ruido aditivo, tipo de ruido sustractivo y tipo de ruido mezclado. Para ilustrar qué significan estos tipos de ruido, se hará referencia a la siguiente figura que muestra el patrón  $x$  (imagen superior), además de mostrar las versiones alteradas (imágenes inferiores) del mismo patrón  $x$  afectado por 40 % de ruido aditivo, sustractivo y mezclado, en ese orden.

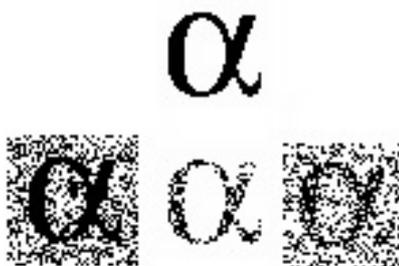


figura 1.2 Patrón afectado con 40% de ruido aditivo, sustractivo y mezclado.

Las memorias asociativas se clasifican en dos diferentes, a saber: “*memorias autoasociativas*” y “*memorias heteroasociativas*”.

Una memoria es autoasociativa si se cumple que:

$$x^\mu = y^\mu \quad \forall \mu \in \{1, 2, \dots, p\} \quad (1.3)$$

Una memoria es heteroasociativa si se cumple lo siguiente:  $\exists \mu \in \{1, 2, \dots, p\}$  para el que  $x^\mu \neq y^\mu$

Es importante también definir qué tipos de valores aceptan tanto los patrones de entrada, de salida, así como la memoria  $M$ ; arbitrariamente se llamará  $A$  al conjunto que contiene los valores asignados a las componentes de los patrones de entrada y salida, y  $B$  al conjunto de los valores que toman los elementos de la matriz  $M$ .

Los conjuntos  $A$  y  $B$  se escogen arbitrariamente, por lo que no es necesario que estos dos conjuntos sean diferentes o deban tener características especiales. Esto significa que el número de posibilidades para escoger  $A$  y  $B$  es infinito. Se presentarán algunos ejemplos de cómo definir estos conjuntos.

$A = B = \mathbb{R}$ , donde  $\mathbb{R}$  es el símbolo que representa al conjunto de los números reales.

$$A = \mathbb{R} \text{ y } B = \{0, 1\}$$

$$A = B = \{-1, 1\}$$

$$A = \mathbb{Z} \text{ y } B = \{-1, 1\}, \text{ donde } \mathbb{Z} \text{ es el conjunto de los números enteros.}$$

Cada uno de los modelos de memorias asociativas que se analizarán en el capítulo 2 posee sus propias especificaciones para los conjuntos  $A$  y  $B$ , de acuerdo con las necesidades de cada modelo.

Por último se deben establecer las dimensiones de los patrones de entrada, así como los de salida; si  $m$  y  $n$  son números enteros positivos, se denota por  $n$  la dimensión de los patrones de entrada, y por  $m$  la dimensión de los patrones de salida. Además, nada impide que  $n$  y  $m$  sean iguales.

## 1.1. Objetivo

Realizar un estudio experimental comparativo entre las memorias asociativas clásicas, las memorias asociativas morfológicas y las memorias asociativas Alfa Beta, para el caso binario, con la finalidad de mostrar las virtudes y defectos de estos modelos de memorias; además, detectar casos particulares de patrones recuperados de manera no perfecta que puedan procesarse por algún método heurístico o por el mismo modelo de memoria asociativa que los arrojó, y hacer que estos patrones sean recuperados en forma perfecta. Para la realización de este estudio estadístico comparativo se desarrollará un software que permitirá como entrada un conjunto fundamental típico de patrones, para que todos los modelos de memorias asociativas aprendan y realicen sus recuperaciones; el software deberá, además de mostrar los patrones de salida, también mostrar gráficas que ilustren el rendimiento de cada modelo.

## 1.2. Motivación

Las memorias asociativas son algoritmos que permiten cumplir con el objetivo de aprender un conjunto fundamental de patrones y recuperarlos, a pesar de que los patrones de entrada (una vez que ya se aprendieron) estén alterados por algún tipo de ruido. Los modelos de memorias que se compararán son: *la Lernmatrix de Steinbuch*, *el Linear Associator de Anderson & Kohonen*, *la memoria asociativa de Hopfield*, *la*

*memorias morfológicas de Díaz-de-León y las memorias Alfa Beta de Yáñez-Márquez & Díaz-de-León*; estos modelos de memorias serán descritos en el capítulo 2 de este trabajo.

La *Lernmatrix* fue uno de los modelos pioneros de las memorias asociativas, que fue olvidada por mucho tiempo por la comunidad científica, y que retomó vida cuando fue incluida como parte del estado del arte de la tesis doctoral "Memorias asociativas basadas en relaciones de orden y operadores binarios" (Yáñez-Márquez, 2002). La *Lernmatrix* es una memoria heteroasociativa que puede funcionar como clasificador si se escogen adecuadamente los patrones de salida; su fundamentación matemática ha sido expuesta en un reciente trabajo de tesis de maestría (Sánchez-Garfias, 2004).

Una década después de que apareció la *Lernmatrix*, se dio a conocer el modelo conocido como *Linear Associator* de Anderson & Kohonen. Esta memoria asociativa se inspiró en la *Lernmatrix* según lo afirma uno de sus autores (Kohonen, 1972); este modelo fue muy importante ya que sirvió de inspiración para los modelos de memorias que le sucedieron, tales como la memoria asociativa de Hopfield.

*La memoria de Hopfield* fue dada a conocer por John Hopfield en 1982; este trabajo fue trascendental, ya que sirvió de parteaguas, por un lado para que se trabajase con las memorias asociativas, y por otro, para recuperar el área de investigación de las redes neuronales, abandonada hacía trece años. *La memoria de Hopfield*, también conocida en la literatura como *la red de Hopfield*, es buena y eficiente para conjuntos fundamentales de cardinalidad 2, a pesar de que los patrones estén alterados con cualquier tipo de ruido, como se mostrará en los experimentos.

Otras memorias importantes por su aportación y su fundamentación matemática para garantizar recuperaciones perfectas, son *las memorias asociativas morfológicas*, que superan a la memoria autoasociativa de *Hopfield*, al *Linear Associator* y a la *Lernmatrix*. Las memorias morfológicas, entre cuyos autores se encuentra el Dr. Díaz-de-León, presentan características interesantes como el hecho de que, a través de su fundamentación matemática se indica bajo qué circunstancias son efectivas, en dónde pueden tener problemas y dónde fallarán; estas características se traducen en ventajas para un diseñador, quien será capaz de saber dónde y en qué momento se pueden usar.

En los albores del año 2002 aparecen las memorias asociativas *Alfa Beta*. Estas memorias tiene la misma efectividad y un comportamiento similar a las memorias asociativas morfológicas, a pesar de que los operadores usados para el aprendizaje y recuperación de patrones para ambas memorias son diferentes; pero las memorias asociativas *Alfa Beta* tienen, al menos, la ventaja de que la densidad aritmética es menor que la densidad aritmética de las memorias morfológicas.

Después de una exhaustiva investigación documental, fue evidente para el autor de esta tesis que en la actualidad no existen estudios experimentales comparativos para los modelos más importantes de memorias asociativas; el descubrimiento de este hecho dio lugar a la **motivación** para realizar este trabajo de tesis, donde se muestren tanto las virtudes como los defectos de los diferentes modelos de memorias asociativas.

Pero el trabajo de tesis va más allá: se hace uso de métodos heurísticos para realizar recuperaciones perfectas donde fallan los modelos puros. Cabe hacer notar que el autor de esta tesis encontró sólo un estudio comparativo de memorias asociativas (Yañez-Márquez, 2002), pero este comparativo tuvo la finalidad de demostrar la efectividad de las, hasta ese entonces, nuevas memorias asociativas Alfa Beta.

### 1.3. Planteamiento del problema

Ante la necesidad imperante de saber qué tipo de memoria asociativa se puede usar para ciertas aplicaciones de reconocimiento de patrones, es deseable conocer las ventajas con las que se cuenta al hacer uso de una memoria asociativa en particular; también es deseable saber en qué momentos o situaciones específicas dicha memoria presentará problemas, o bajará su rendimiento. Teniendo *a priori* el conocimiento de todas las virtudes y defectos de las memorias asociativas, es posible hacer una selección más adecuada para las necesidades de la aplicación de reconocimiento de patrones. En esto consiste precisamente el **problema** que se pretende atacar con este trabajo de tesis. Los resultados de esta tesis serán más significativos si se pueden comparar las memorias asociativas haciendo uso de un conjunto fundamental típico; es decir, considerar todas las memorias en condiciones similares de diseño y operación.

La realización del software es tan sólo una herramienta para realizar el estudio experimental comparativo, debido a que se llevarán a cabo muchas pruebas exhaustivas. Los resultados que arroje este software, a través del uso intensivo de imágenes y gráficas de diferentes tipos, serán analizados de tal manera que se generará un estudio experimental comparativo serio de los modelos de memorias asociativas consideradas.

### 1.4. Contribuciones

- Un estudio experimental comparativo entre los modelos clásicos de memorias asociativas, memorias Morfológicas y memorias asociativas Alfa-Beta.
- Biblioteca de funciones para la implementación de los modelos de memorias asociativas.
- Biblioteca de funciones para el manejo de imágenes con formato BMP.

### 1.5. Organización de la tesis

En las anteriores secciones se han presentado: el objetivo del trabajo de tesis, la motivación, el planteamiento del problema y las contribuciones, en ese orden.

El resto del trabajo está organizado de la siguiente manera:

El capítulo 2 está dedicado a presentar el estado del arte con respecto, tanto a los modelos matemáticos de memorias asociativas usados en este trabajo de tesis, como a los estudios comparativos experimentales de memorias asociativas que hay en la

literatura actual. En la primera sección se describen los siguientes modelos de memorias asociativas: *Lernmatrix*, *Linear Associator*, *memoria Hofield*, *memorias asociativas morfológicas* y finalmente las *memorias  $\alpha\beta$* ; y en la segunda, se hará mención de los trabajos científicos que tratan sobre estudios comparativos experimentales de modelos de memorias asociativas, y algunos trabajos relacionados.

El capítulo 3 es relevante debido a que se muestran las herramientas matemáticas utilizadas para dar cuerpo a este tema de tesis. En la sección 3.1 se destaca la importancia de las sumatorias, del operador  $\bigvee$  (máximo) y del operador  $\bigwedge$  (mínimo); la sección 3.2 presenta las operaciones matriciales que son fundamentales para este trabajo de tesis, en tanto que la sección 3.3 trata sobre el papel que desempeñan los vectores n-dimensionales para el funcionamiento de las memorias asociativas. La generación del ruido es primordial para probar las memorias asociativas en condiciones extremas, por lo que en la sección 3.4 se incluye precisamente este tema; por último, la sección 3.5 presenta a los operadores morfológicos: erosión y dilatación, operadores que son imprescindibles para el buen desempeño de las memorias heurísticas.

El desarrollo del software que genera el estudio experimental es de vital importancia para que se logre el objetivo trazado en este trabajo de tesis: el capítulo 4 está designado para mostrar este desarrollo. La importancia de este capítulo es tal, que se muestra cómo representar el planteamiento matemático de las memorias asociativas en algoritmos que posteriormente sirvan como base para codificarlos en un lenguaje de programación, y uno de sus aspectos relevantes es la creación de las memorias heurísticas. Las secciones que abarca este capítulo son: *4.1. Generación del conjunto fundamental*; *4.2. Generación de ruido*; *4.3. Operadores morfológicos*; *4.4 Implementación de las Memorias Asociativas*; *4.5 Ejecución automática de las memorias asociativas*; *4.6 Ejecución por patrón y, por último, la sección 4.7 trata sobre la generación de las gráficas en el software.*

El capítulo 5 contiene las Disquisiciones experimentales, y constituye la culminación del objetivo de este trabajo de tesis, porque presenta el total comportamiento que tuvieron los diferentes modelos de memorias asociativas, ante dos tipos de patrones: los del conjunto fundamental por un lado, y los afectados con cierto porcentaje de ruido, por el otro. Las secciones que incluye este capítulo son: *5.1 Estudio comparativo experimental para los modelos de memorias heteroasociativas*; *5.2 Estudio comparativo experimental para los modelos de memorias autoasociativas*; y *5.3 Casos especiales*. En estas tres secciones están contenidos todos los resultados que constituyen el estudio comparativo experimental de las memorias asociativas.

En la parte final de esta tesis se muestra el apéndice A, donde se presenta la simbología usada en este trabajo escrito y, finalmente, la bibliografía.

Además, se anexa un CD que contiene el software diseñado como parte fundamental de este trabajo de tesis, y dos apéndices adicionales: el apéndice B que consta del manual de uso del software, y el apéndice C que contiene la biblioteca de funciones para el manejo de imágenes con formato BMP, la cual fue creada, diseñada y desarrollada por el autor.

## Capítulo 2

# Estado del Arte

A lo largo de este capítulo se presenta el estado del arte con respecto a dos tópicos: primero, con respecto a los modelos matemáticos de memorias asociativas usados en este trabajo de tesis y, segundo, con respecto a los estudios comparativos experimentales de memorias asociativas que hay en la literatura actual.

En la primera sección se describirán, siguiendo un orden cronológico, los modelos de memorias asociativas que han surgido con el pasar del tiempo; dichos modelos serán presentados como soluciones al problema planteado en la sección 1.3. Se hablará acerca del o los autores y la procedencia de éstos, además de las operaciones que dan vida a los modelos. Los modelos de memorias asociativas que serán analizados son: *Lernmatrix*, *Linear Associator*, *memoria Hofield*, *memorias asociativas morfológicas* y finalmente las *memorias  $\alpha\beta$* .

Es importante enfatizar que para la fase de aprendizaje de todas las memorias, se considerará el conjunto fundamental  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$  y para la fase de recuperación se hará uso de los patrones  $\tilde{x}^\omega$ , los cuales se obtienen tras alterar con ruido los patrones fundamentales  $x^\omega$ . Cabe mencionar que si el ruido es cero, entonces los patrones  $\tilde{x}^\omega$  son precisamente los patrones fundamentales de entrada  $x^\omega$ .

En la segunda sección del capítulo se hará mención de los pocos trabajos que existen actualmente en la literatura, que tratan sobre estudios comparativos experimentales de modelos de memorias asociativas, y algunos trabajos relacionados con esta tarea.

### 2.1. Estado del arte en memorias asociativas

#### 2.1.1. Lernmatrix

La memoria *Lernmatrix* es una memoria heteroasociativa que fue creada por el investigador alemán Karl Steinbuch al inicio de la década de los sesenta (Steinbuch, 1961; Steinbuch & Frank, 1961); de hecho, Karl Steinbuch fue uno de los pioneros en desarrollar un método para codificar información en arreglos cuadrículados que se les llegó a conocer con el nombre de *crossbar* (Simpson, 1990). Esta memoria puede operar como un clasificador si se escogen de manera adecuada los patrones de salida.

### Fase de aprendizaje

La *Lernmatrix* acepta como entrada a los patrones  $x^\mu \in A^n$ ,  $A = \{0, 1\}$  y genera como salida la clase  $y^\mu \in A^p$  que le puede corresponder de entre  $p$  clases diferentes, codificada ésta con un método simple, a saber: para representar la clase  $k \in \{1, 2, \dots, p\}$ , se asigna a las componentes del vector de salida  $y^\omega$  los siguientes valores:  $y_k^\omega = 1$ , y  $y_j^\omega = 0$  para  $j = 1, 2, \dots, k-1, k+1, \dots, p$ .

El conjunto fundamental  $(x^\mu, y^\mu) \in A^n \times A^p$  que da origen al *crossbar* en la fase de aprendizaje, puede ilustrarse de la siguiente manera:

$$\begin{array}{cccccc}
 & x_1^\mu & x_2^\mu & \cdots & x_j^\mu & \cdots & x_n^\mu \\
 y_1^\mu & m_{11} & m_{12} & \cdots & m_{1j} & \cdots & m_{1n} \\
 y_2^\mu & m_{21} & m_{22} & \cdots & m_{2j} & \cdots & m_{2n} \\
 \vdots & \vdots & \vdots & & \vdots & & \vdots \\
 y_i^\mu & m_{i1} & m_{i2} & \cdots & m_{ij} & \cdots & m_{in} \\
 \vdots & \vdots & \vdots & & \vdots & & \vdots \\
 y_p^\mu & m_{p1} & m_{p2} & \cdots & m_{pj} & \cdots & m_{pn}
 \end{array} \tag{2.1}$$

Los componentes  $m_{ij}$  de la *Lernmatrix*  $M$  tienen valor cero al iniciar el aprendizaje, y se actualizan de acuerdo con la regla  $m_{ij} + \Delta m_{ij}$ , donde:

$$\Delta m_{ij} = \begin{cases} +\varepsilon & \text{si } y_i^\mu = 1 = x_j^\mu \\ -\varepsilon & \text{si } x_j^\mu = 0 \text{ y } y_i^\mu = 1 \\ 0 & \text{en otro caso} \end{cases} \tag{2.2}$$

siendo  $\varepsilon$  una constante positiva escogida previamente.

### Fase de recuperación

La fase de recuperación consiste en encontrar la clase a la que pertenece un vector de entrada  $x^\omega \in A^n$  dado: encontrar la clase significa obtener las coordenadas del vector  $y^\omega \in A^p$  que le corresponde al patrón  $x^\omega$ .

La manera para construir el vector  $y^\omega$  que tiene este método, es la siguiente: la  $i$ -ésima coordenada  $y_i^\omega$  del vector de la clase  $y^\omega \in A^p$ , se obtiene como se indica a continuación:

$$y_i^\omega = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} \cdot x_j^\omega = \bigvee_{h=1}^p \left[ \sum_{j=1}^n m_{hj} \cdot x_j^\omega \right] \\ 0 & \text{en otro caso} \end{cases} \tag{2.3}$$

donde  $\bigvee$  es el operador *máximo*.

La *Lernmatrix* presenta el siguiente problema: cuando sucede que existen más patrones de entrada que clases, llega un momento en que esta memoria ya no es capaz de asignar la clase correcta a la que pertenece el patrón que se intenta clasificar; este problema es conocido como el fenómeno de saturación.

### 2.1.2. Linear Associator de Anderson-Kohonen

El *Linear Associator* tiene sus orígenes en los trabajos pioneros de Anderson y Kohonen publicados en el año de 1972; algo que es digno de resaltar es el siguiente hecho: James A. Anderson es neurofisiológico y Teuvo Kohonen es físico e ingeniero eléctrico; además de estas diferencias de perfiles, estos personajes no tenían noticias el uno del otro cuando publicaron sus artículos, siendo que los resultados publicados fueron asombrosamente similares (Anderson & Rosenfeld, 1990; Kohonen, 1989).

Para presentar el modelo *Linear Associator*, se debe considerar el conjunto fundamental  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$  con:

$$A = \{0, 1\}, \quad x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad \text{y} \quad y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

#### Fase de aprendizaje

La fase de aprendizaje para el Linear Associator consiste de dos etapas:

**Primera etapa.** Para cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se encuentra la matriz  $y^\mu \cdot (x^\mu)^t$  de dimensión  $m \times n$

$$y^\mu \cdot (x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu, x_2^\mu, \dots, x_n^\mu) \quad (2.4)$$

$$y^\mu \cdot (x^\mu)^t = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \cdots & y_1^\mu x_j^\mu & \cdots & y_1^\mu x_n^\mu \\ y_2^\mu x_1^\mu & y_2^\mu x_2^\mu & \cdots & y_2^\mu x_j^\mu & \cdots & y_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \cdots & y_i^\mu x_j^\mu & \cdots & y_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu x_1^\mu & y_m^\mu x_2^\mu & \cdots & y_m^\mu x_j^\mu & \cdots & y_m^\mu x_n^\mu \end{pmatrix} \quad (2.5)$$

**Segunda etapa.** Se suman las  $p$  matrices para obtener la memoria

$$\mathbf{M} = \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t = [m_{ij}]_{m \times n} \quad (2.6)$$

el componente  $ij$ -ésimo de la memoria  $M$  se expresa de la siguiente manera:

$$m_{ij} = \sum_{\mu=1}^p y_i^\mu x_j^\mu \quad (2.7)$$

### Fase de recuperación

La fase de recuperación consiste en presentarle a la memoria un patrón de entrada  $x^\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$  y realizar la siguiente operación:

$$M \cdot x^\omega = \left[ \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t \right] \cdot x^\omega \quad (2.8)$$

Al desarrollar la expresión 2.8, se tiene:

$$\begin{aligned} M \cdot x^\omega &= \left[ y^1 \cdot (x^1)^t + y^2 \cdot (x^2)^t + \dots + y^\omega \cdot (x^\omega)^t + \dots + y^p \cdot (x^p)^t \right] \cdot x^\omega \\ M \cdot x^\omega &= \left[ y^1 \cdot (x^1)^t \right] \cdot x^\omega + \dots + \left[ y^\omega \cdot (x^\omega)^t \right] \cdot x^\omega + \dots + \left[ y^p \cdot (x^p)^t \right] \cdot x^\omega \\ M \cdot x^\omega &= y^1 \cdot \left[ (x^1)^t \cdot x^\omega \right] + \dots + y^\omega \cdot \left[ (x^\omega)^t \cdot x^\omega \right] + \dots + y^p \cdot \left[ (x^p)^t \cdot x^\omega \right] \end{aligned}$$

Esta última expresión puede escribirse de la siguiente manera:

$$M \cdot x^\omega = y^\omega \cdot \left[ (x^\omega)^t \cdot x^\omega \right] + \sum_{\mu \neq \omega} y^\mu \cdot \left[ (x^\mu)^t \cdot x^\omega \right] \quad (2.9)$$

La expresión 2.9 permite investigar las condiciones que se deben cumplir para que la fase de recuperación dé salidas perfectas; para que la expresión 2.9 genere salidas perfectas se deben cumplir las siguientes igualdades:

1.  $\left[ (x^\omega)^t \cdot x^\omega \right] = 1$
2.  $\left[ (x^\mu)^t \cdot x^\omega \right] = 0$  siempre que  $\mu \neq \omega$

Dado que  $\omega$  se escogió arbitrariamente, las dos igualdades deben cumplirse  $\forall \omega \in \{1, 2, \dots, p\}$ , esto indica que los vectores de entrada  $x^\mu$  deben ser ortonormales. Esta condición de ortonormalidad se puede resumir en la siguiente expresión:

$$(x^\mu)^t \cdot x^\omega = \delta_{\mu\omega} = \begin{cases} 1 & \text{si } \mu = \omega \\ 0 & \text{si } \mu \neq \omega \end{cases} \quad (2.10)$$

donde  $\delta_{\mu\omega}$  es la conocida *delta de Kronecker* (Moore, 1968).

Si se cumple la condición de la expresión 2.10, como resultado la recuperación debe ser perfecta; ahora la expresión 2.9 toma la siguiente forma:

$$M \cdot x^\omega = y^\omega \quad (2.11)$$

Si los vectores de entrada no son ortonormales, suceden dos cosas:

1. el factor  $[(x^\omega)^t \cdot x^\omega]$  no es 1
2. el término  $\sum_{\mu \neq \omega} y^\omega \cdot [(x^\mu)^t \cdot x^\omega]$  no es 0

El último término, que es llamado *cross-talk*, representa el ruido generado por la interacción entre los patrones de entrada, y tiene como consecuencia inmediata que la recuperación no sea perfecta.

### 2.1.3. Memoria autoasociativa Hopfield

El año de 1982 fue un año muy importante para el área de las memorias asociativas (y de las redes neuronales) ya que voces autorizadas del compendio *Neurocomputing* aseguran que la era moderna de estas ramas nace como consecuencia del artículo publicado por el distinguido y respetado físico John J. Hopfield (Anderson & Rosenfeld, Eds. 1990). En su artículo, Hopfield presenta su famosa memoria asociativa (Hopfield, 1982), que impactó positivamente a la comunidad científica, y actualmente es internacionalmente conocida.

Hopfield en su artículo, parte de un sistema físico descrito por un vector de estado  $x$  cuyas coordenadas son  $(x_1, x_2, \dots, x_n)$ ; se considera que el sistema tiene puntos límite localmente estables  $x^a, x^b, \dots$ . Entonces, si el sistema es activado en un estado suficientemente cercano a cualquier punto límite localmente estable, digamos en  $x = x^a + \Delta$ , al transcurrir el tiempo el estado del sistema cambiará hasta llegar a que  $x \approx x^a$ . El punto de arranque  $x^a + \Delta$  representa un conocimiento parcial del estado estable  $x^a$ , y a partir de ahí el sistema genera la información correcta  $x^a$ . Hopfield declara categóricamente que un sistema físico que se comporta de esta manera y que, además, sea susceptible de que cualquier *conjunto* predeterminado de estados pueda ser asignado como el conjunto de estados localmente estables, es útil como memoria asociativa.

En el modelo que originalmente propuso Hopfield, cada neurona tiene dos posibles estados:  $x_i = 0$  y  $x_i = 1$ ; sin embargo en el mismo artículo presentado por Hopfield en 1982, hace la relevante observación que la capacidad de almacenamiento de información de la memoria puede incrementarse por un factor de 2, si se escogen como posibles estados de las neuronas los valores  $x_i = -1$  y  $x_i = 1$  en lugar de los valores originales  $x_i = 0$  y  $x_i = 1$ .

La intensidad de la fuerza de conexión de la neurona  $x_i$  a la neurona  $x_j$  se representa por el valor de  $m_{ij}$ , y se considera que hay simetría, es decir,  $m_{ij} = m_{ji}$ . Si  $x_i$  no está conectada con  $x_j$ , entonces  $m_{ij} = 0$ ; en particular, no hay conexiones recurrentes de una neurona a sí misma, lo cual significa que  $m_{ii} = 0, \forall i$ . El estado instantáneo del sistema está completamente especificado por el vector columna de dimensión  $n$  cuyas coordenadas son los valores de las  $n$  neuronas.

La memoria Hopfield es autoasociativa, simétrica, con ceros en la diagonal principal. En virtud de que la memoria es autoasociativa, el conjunto fundamental para la

memoria Hopfield es  $\{(x^\mu, x^\mu) \mid \mu = 1, 2, \dots, p\}$  con

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad \text{y} \quad A = \{-1, 1\} \quad (2.12)$$

### Fase de aprendizaje

La fase de aprendizaje de la memoria Hopfield, es decir la creación de la memoria  $M$ , se realiza de la siguiente manera:

$$m_{ij} = \begin{cases} \sum_{\mu=1}^p x_i^\mu x_j^\mu & \text{si } i \neq j \\ 0 & \text{si } i = j \end{cases} \quad (2.13)$$

Operativamente la expresión 2.13 se puede obtener en tres etapas:

**Primera etapa.** Para cada una de las  $p$  asociaciones  $(x^\mu, x^\mu)$ , se encuentra la matriz  $x^\mu \cdot (x^\mu)^t$  de dimensión  $n \times n$ :

$$x^\mu \cdot (x^\mu)^t = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \cdot (x_1^\mu, x_2^\mu, \dots, x_n^\mu)$$

$$x^\mu \cdot (x^\mu)^t = \begin{pmatrix} x_1^\mu x_1^\mu & x_1^\mu x_2^\mu & \cdots & x_1^\mu x_i^\mu & \cdots & x_1^\mu x_n^\mu \\ x_2^\mu x_1^\mu & x_2^\mu x_2^\mu & \cdots & x_2^\mu x_i^\mu & \cdots & x_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ x_i^\mu x_1^\mu & x_i^\mu x_2^\mu & \cdots & x_i^\mu x_i^\mu & \cdots & x_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ x_n^\mu x_1^\mu & x_n^\mu x_2^\mu & \cdots & x_n^\mu x_i^\mu & \cdots & x_n^\mu x_n^\mu \end{pmatrix} \quad (2.14)$$

La matriz 2.14 debe tener valores 1 en su diagonal principal, porque los componentes de los patrones pertenecen al conjunto  $A = \{-1, 1\}$ , según la expresión 2.12, y para cualquiera de los valores, 1 o -1, se cumple que  $x_i^\mu x_i^\mu = 1, \forall i = 1, \dots, n$ :

$$x^\mu \cdot (x^\mu)^t = \begin{pmatrix} 1 & x_1^\mu x_2^\mu & \cdots & x_1^\mu x_i^\mu & \cdots & x_1^\mu x_n^\mu \\ x_2^\mu x_1^\mu & 1 & \cdots & x_2^\mu x_i^\mu & \cdots & x_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ x_i^\mu x_1^\mu & x_i^\mu x_2^\mu & \cdots & 1 & \cdots & x_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ x_n^\mu x_1^\mu & x_n^\mu x_2^\mu & \cdots & x_n^\mu x_i^\mu & \cdots & 1 \end{pmatrix} \quad (2.15)$$

**Segunda etapa.** A cada una de las  $p$  matrices  $x^\mu \cdot (x^\mu)^t$ , expresada en 2.15, se le resta la matriz identidad  $I$  de dimensiones  $n \times n$ , con la finalidad de hacer ceros la diagonal principal:

$$x^\mu \cdot (x^\mu)^t - I = \begin{pmatrix} 0 & x_1^\mu x_2^\mu & \cdots & x_1^\mu x_i^\mu & \cdots & x_1^\mu x_n^\mu \\ x_2^\mu x_1^\mu & 0 & \cdots & x_2^\mu x_i^\mu & \cdots & x_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ x_i^\mu x_1^\mu & x_i^\mu x_2^\mu & \cdots & 0 & \cdots & x_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ x_n^\mu x_1^\mu & x_n^\mu x_2^\mu & \cdots & x_n^\mu x_i^\mu & \cdots & 0 \end{pmatrix} \quad (2.16)$$

**Tercera etapa.** Se suman las  $p$  matrices  $x^\mu \cdot (x^\mu)^t - I$  para que finalmente se obtenga la memoria asociativa Hopfield  $M$ :

$$M = \sum_{\mu=1}^p [x^\mu \cdot (x^\mu)^t - I] = [m_{ij}]_{n \times n} \quad (2.17)$$

### Fase de recuperación

La fase de recuperación para la memoria Hopfield consiste en tres etapas; antes de hablar de éstas, se debe tener presente que: al presentar un patrón  $\tilde{x}$  a la memoria autoasociativa Hopfield, ésta cambiará su estado con el tiempo, de modo que cada neurona  $x_i$  ajuste su valor de acuerdo con el resultado que arroje la comparación de la cantidad  $\sum_{j=1}^n m_{ij}x_j$  con un valor de umbral, el cual normalmente se coloca en cero.

Se representará el estado de la memoria Hopfield en el tiempo  $t$  por  $x(t)$ ; entonces  $x_i(t)$  representa el valor de la neurona  $x_i$  en el tiempo  $t$  y  $x_i(t+1)$  el valor de  $x_i$  en el tiempo siguiente ( $t+1$ ).

*Dado un vector columna de entrada  $\tilde{x}$ :*

**Primera etapa.** Para  $t = 0$ , se hace  $x(t) = \tilde{x}$ ; es decir,  $x_i(0) = \tilde{x}_i, \forall i \in \{1, 2, 3, \dots, n\}$

**Segunda etapa.**  $\forall i \in \{1, 2, 3, \dots, n\}$  se calcula  $x_i(t+1)$  en conformidad con la siguiente condición:

$$x_i(t+1) = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) > 0 \\ x_i(t) & \text{si } \sum_{j=1}^n m_{ij}x_j(t) = 0 \\ -1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) < 0 \end{cases} \quad (2.18)$$

**Tercera etapa.** Se compara  $x_i(t+1)$  con  $x_i(t) \forall i \in \{1, 2, 3, \dots, n\}$ . Si  $x(t+1) = x(t)$  el proceso termina y el vector recuperado es  $x(0) = \tilde{x}$ . De otro modo, se realizan tantas iteraciones como sea necesario hasta llegar a un valor  $t = \tau$  para el

cual  $x_i(\tau + 1) = x_i(\tau) \forall i \in \{1, 2, 3, \dots, n\}$ ; el proceso termina y el patrón recuperado es  $x(\tau)$ .

El proceso de convergencia descrito en la tercera etapa de la fase de recuperación, indica que el sistema llega a un punto límite localmente estable en el tiempo  $\tau$ . La existencia de  $\tau$ , piedra angular en la memoria Hopfield, está garantizada a través de la demostración que hace Hopfield de que existen puntos límite localmente estables en su modelo de memoria asociativa; para ello, se define  $E$  de la siguiente manera, tomando en cuenta la condición de que  $m_{ii} = 0, \forall i$ :

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n m_{ij} x_i x_j \quad (2.19)$$

#### 2.1.4. Memorias asociativas morfológicas

Después de esta incursión en las memorias asociativas clásicas, en esta sección se analizarán *las memorias asociativas morfológicas*. *Las memorias morfológicas* rompen el esquema convencional de como venían tratándose las memorias asociativas clásicas, ya que éstas últimas se basan en las operaciones convencionales de vectores y matrices en la fase de aprendizaje y en la suma de productos para la recuperación de patrones; en contraste, las memorias asociativas morfológicas se basan en: *el uso de máximos o mínimos de sumas para la fase de aprendizaje y máximos o mínimos de sumas para la fase de recuperación* (Ritter, Sussner & Díaz-de-León, 1998; Ritter, Díaz-de-León & Sussner, 1999).

Existen dos tipos de memorias morfológicas: las memorias  $\bigvee$  (*max*), simbolizadas por  $M$ , y las memorias  $\bigwedge$  (*min*) simbolizadas por  $W$ ; ambas memorias funcionan para los modos *heteroasociativo* y *autoasociativo*. Es importante resaltar desde este momento la ventaja que tienen las memorias asociativas morfológicas sobre la memoria Hopfield, ya que las memorias morfológicas funcionan en los modos autoasociativo y heteroasociativo y la Hopfield sólo en el modo autoasociativo.

El conjunto fundamental para las memorias asociativas morfológicas es:

$$\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$$

$$A = \{0, 1\}, \quad x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad \text{y} \quad y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m \quad (2.20)$$

La memoria autoasociativa debe cumplir:  $x^\mu = y^\mu \forall \mu \in \{1, 2, \dots, p\}$ , y para que sea heteroasociativa se puede afirmar que  $\exists \mu \in \{1, 2, \dots, p\}$  para el que se cumpla que  $x^\mu \neq y^\mu$ .

Es necesario definir dos nuevas operaciones entre matrices en términos de las operaciones  $+$ ,  $\bigvee$  y  $\bigwedge$ , con el fin de expresar las fases de aprendizaje y recuperación de las memorias asociativas morfológicas (Ritter, Sussner & Díaz-de-León, 1998).

Sea  $D$  una matriz  $[d_{ij}]_{m \times r}$  y  $H$  una matriz  $[h_{ij}]_{r \times n}$  cuyos términos son números enteros.

**Definición 2.1** *El producto máximo de  $D$  y  $H$ , denotado por  $C = D \nabla H$ , es una matriz  $[c_{ij}]_{m \times n}$  cuya  $ij$ -ésima componente  $c_{ij}$  se define de la siguiente manera:*

$$c_{ij} = \bigvee_{k=1}^r (d_{ik} + h_{kj}) \quad (2.21)$$

**Definición 2.2** *El producto mínimo de  $D$  y  $H$ , denotado por  $C = D \Delta H$ , es una matriz  $[c_{ij}]_{m \times n}$  cuya  $ij$ -ésima componente  $c_{ij}$  se define así:*

$$c_{ij} = \bigwedge_{k=1}^r (d_{ik} + h_{kj}) \quad (2.22)$$

Las expresiones 2.21 y 2.22 contienen a los operadores máximo  $\bigvee$  y mínimo  $\bigwedge$ , los cuales están íntimamente ligados con los conceptos de las dos operaciones básicas de la morfología matemática: *dilatación y erosión* (Serra, 1992; Haralick, Stenberg & Zhuang, 1987; Ritter, Sussner & Díaz-de-León, 1998); el nombre de memorias asociativas *morfológicas* está inspirado precisamente en estas dos operaciones morfológicas básicas.

## Memorias asociativas morfológicas $\bigvee$

Las memorias asociativas morfológicas *max* heteroasociativas o autoasociativas utilizan el producto *mínimo* 2.22 tanto para la fase de aprendizaje como para la fase de recuperación. Además, en la fase de aprendizaje juega un papel importante el operador *máximo* 2.21; de ahí el nombre de memorias *max*.

### Fase de aprendizaje

La fase de aprendizaje para las memorias morfológicas *max*, heteroasociativa y autoasociativa, consta de dos etapas:

**Primera etapa.** En cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se usa el producto mínimo 2.22 para crear la matriz  $y^\mu \Delta (-x^\mu)^t$  de una dimensión  $m \times n$ , donde el negado transpuesto del patrón de entrada  $x^\mu$  se define como:  $(-x^\mu)^t = (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$ :

$$y^\mu \Delta (-x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \Delta (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$$

$$\begin{aligned}
y^\mu \Delta (-x^\mu)^t &= \begin{pmatrix} [\wedge (y_1^\mu - x_1^\mu)] & \cdots & \left[ \wedge \left( y_1^\mu - x_j^\mu \right) \right] & \cdots & [\wedge (y_1^\mu - x_n^\mu)] \\ [\wedge (y_2^\mu - x_1^\mu)] & \cdots & \left[ \wedge \left( y_2^\mu - x_j^\mu \right) \right] & \cdots & [\wedge (y_2^\mu - x_n^\mu)] \\ \vdots & & \vdots & & \vdots \\ [\wedge (y_i^\mu - x_1^\mu)] & \cdots & \left[ \wedge \left( y_i^\mu - x_j^\mu \right) \right] & \cdots & [\wedge (y_i^\mu - x_n^\mu)] \\ \vdots & & \vdots & & \vdots \\ [\wedge (y_m^\mu - x_1^\mu)] & \cdots & \left[ \wedge \left( y_m^\mu - x_j^\mu \right) \right] & \cdots & [\wedge (y_m^\mu - x_n^\mu)] \end{pmatrix} \\
y^\mu \Delta (-x^\mu)^t &= \begin{pmatrix} y_1^\mu - x_1^\mu & y_1^\mu - x_2^\mu & \cdots & y_1^\mu - x_j^\mu & \cdots & y_1^\mu - x_n^\mu \\ y_2^\mu - x_1^\mu & y_2^\mu - x_2^\mu & \cdots & y_2^\mu - x_j^\mu & \cdots & y_2^\mu - x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu - x_1^\mu & y_i^\mu - x_2^\mu & \cdots & y_i^\mu - x_j^\mu & \cdots & y_i^\mu - x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu - x_1^\mu & y_m^\mu - x_2^\mu & \cdots & y_m^\mu - x_j^\mu & \cdots & y_m^\mu - x_n^\mu \end{pmatrix} \quad (2.23)
\end{aligned}$$

**Segunda etapa.** Se aplica el operador  $\bigvee$ (máximo) a las  $p$  matrices para obtener la memoria morfológica  $M$ .

$$M = \bigvee_{\mu=1}^p [y^\mu \Delta (-x^\mu)^t] = [m_{ij}]_{m \times n} \quad (2.24)$$

$$m_{ij} = \bigvee_{\mu=1}^p (y_i^\mu - x_j^\mu) \quad (2.25)$$

### Fase de recuperación

La fase de recuperación consiste en realizar el producto mínimo  $\Delta$  de la memoria  $M$  con un patrón de entrada  $x^\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$ , para obtener un vector columna de dimensión  $m$ :

$$y = M \Delta x^\omega \quad (2.26)$$

La  $i$ -ésima componente del vector  $y$  es:

$$y_i = \bigwedge_{j=1}^n (m_{ij} + x_j^\omega) \quad (2.27)$$

Como se observará en el capítulo de "Disquisiciones experimentales", las memorias morfológicas  $M$  son muy robustas para el ruido aditivo hasta en porcentajes muy altos; sin embargo, los efectos son devastadores con el ruido sustractivo, aún en porcentajes muy bajos.

## Memorias asociativas morfológicas $\bigwedge$

Las memorias asociativas morfológicas *min* utilizan el producto *máximo* tanto para la fase de aprendizaje como para la de recuperación. Además, en la fase de aprendizaje el operador *mínimo* tiene un peso muy grande; de ahí, se deriva el nombre de memorias *min*.

### Fase de aprendizaje

Las memorias morfológica *min* para el modo heteroasociativo y autoasociativo, en la fase de aprendizaje tienen dos etapas para generar la memoria  $W$ .

**Primera etapa.** En cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se hace uso del producto máximo 2.21 para crear la matriz  $y^\mu \nabla (-x^\mu)$  de dimensiones  $m \times n$ , donde el negado del transpuesto del patrón de entrada  $x^\mu$  se define como:  $(-x^\mu)^t = (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$

$$y^\mu \nabla (-x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \nabla (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$$

$$y^\mu \nabla (-x^\mu)^t = \begin{pmatrix} [\vee(y_1^\mu - x_1^\mu)] & \cdots & [\vee(y_1^\mu - x_j^\mu)] & \cdots & [\vee(y_1^\mu - x_n^\mu)] \\ [\vee(y_2^\mu - x_1^\mu)] & \cdots & [\vee(y_2^\mu - x_j^\mu)] & \cdots & [\vee(y_2^\mu - x_n^\mu)] \\ \vdots & & \vdots & & \vdots \\ [\vee(y_i^\mu - x_1^\mu)] & \cdots & [\vee(y_i^\mu - x_j^\mu)] & \cdots & [\vee(y_i^\mu - x_n^\mu)] \\ \vdots & & \vdots & & \vdots \\ [\vee(y_m^\mu - x_1^\mu)] & \cdots & [\vee(y_m^\mu - x_j^\mu)] & \cdots & [\vee(y_m^\mu - x_n^\mu)] \end{pmatrix}$$

$$y^\mu \nabla (-x^\mu)^t = \begin{pmatrix} y_1^\mu - x_1^\mu & y_1^\mu - x_2^\mu & \cdots & y_1^\mu - x_j^\mu & \cdots & y_1^\mu - x_n^\mu \\ y_2^\mu - x_1^\mu & y_2^\mu - x_2^\mu & \cdots & y_2^\mu - x_j^\mu & \cdots & y_2^\mu - x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu - x_1^\mu & y_i^\mu - x_2^\mu & \cdots & y_i^\mu - x_j^\mu & \cdots & y_i^\mu - x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu - x_1^\mu & y_m^\mu - x_2^\mu & \cdots & y_m^\mu - x_j^\mu & \cdots & y_m^\mu - x_n^\mu \end{pmatrix} \quad (2.28)$$

**Segunda etapa.** Se aplica el operador mínimo  $\bigwedge$  a las  $p$  matrices para obtener la memoria  $W$ .

$$W = \bigwedge_{\mu=1}^p [y^\mu \nabla (-x^\mu)^t] = [w_{ij}]_{m \times n} \quad (2.29)$$

$$w_{ij} = \bigwedge_{\mu=1}^p (y_i^\mu - x_j^\mu) \quad (2.30)$$

### Fase de recuperación

La fase de recuperación para los modos heteroasociativo y autoasociativo de estas memorias, consiste en realizar el producto máximo  $\nabla$  de la memoria  $W$  con un patrón de entrada  $x^\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$ , para obtener un vector columna  $y$  de dimensión  $m$ :

$$y = W \nabla x^\omega \quad (2.31)$$

la  $i$ -ésima componente del vector  $y$  es:

$$y_i = \bigvee_{j=1}^n (w_{ij} + x_j^\omega) \quad (2.32)$$

#### 2.1.5. Memorias asociativas $\alpha\beta$

Las memorias asociativas  $\alpha\beta$  son las últimas que se analizarán en este capítulo. Estas memorias presentan las mismas virtudes que las memorias morfológicas, por lo tanto, de entrada son mejores que las memorias asociativas clásicas. Además las memorias asociativas  $\alpha\beta$  son memorias que funcionan para los modos heteroasociativo y autoasociativo, haciendo uso de dos nuevos operadores, llamados: el operador Alfa ( $\alpha$ ) para la fase de aprendizaje y el operador Beta ( $\beta$ ) para la fase de recuperación. Estas memorias fueron presentadas en (Yáñez-Márquez, 2002).

Se deben especificar los conjuntos  $A$  y  $B$ , que serán fundamentales para la fase de aprendizaje y de recuperación. Se definen de la siguiente manera:

$$A = \{0, 1\} \quad \text{y} \quad B = \{0, 1, 2\}$$

Además, las memorias asociativas  $\alpha\beta$  hacen uso de dos operaciones binarias llamadas  $\alpha$  y  $\beta$ . La operación binaria  $\alpha = A \times A \rightarrow B$ , está definida por la siguiente tabla:

| $x$ | $y$ | $\alpha(x, y)$ |
|-----|-----|----------------|
| 0   | 0   | 1              |
| 0   | 1   | 0              |
| 1   | 0   | 2              |
| 1   | 1   | 1              |

La operación binaria  $\beta = B \times A \longrightarrow A$ , está definida de la siguiente manera:

| $x$ | $y$ | $\beta(x, y)$ |
|-----|-----|---------------|
| 0   | 0   | 0             |
| 0   | 1   | 0             |
| 1   | 0   | 0             |
| 1   | 1   | 1             |
| 2   | 0   | 1             |
| 2   | 1   | 1             |

Las memorias asociativas  $\alpha\beta$  hacen uso de cuatro operaciones matriciales que son:

**1. Operación  $\alpha$  máx:**  $P_{m \times r} \cup_{\alpha} Q_{r \times n} = [f_{ij}^{\alpha}]_{m \times n}$ , donde

$$f_{ij}^{\alpha} = \bigvee_{k=1}^r \alpha(p_{ik}, q_{kj})$$

**2. Operación  $\beta$  máx:**  $P_{m \times r} \cup_{\beta} Q_{r \times n} = [f_{ij}^{\beta}]_{m \times n}$ , donde

$$f_{ij}^{\beta} = \bigvee_{k=1}^r \beta(p_{ik}, q_{kj})$$

**3. Operación  $\alpha$  mín:**  $P_{m \times r} \cap_{\alpha} Q_{r \times n} = [h_{ij}^{\alpha}]_{m \times n}$ , donde

$$h_{ij}^{\alpha} = \bigwedge_{k=1}^r \alpha(p_{ik}, q_{kj})$$

**4. Operación  $\beta$  mín:**  $P_{m \times r} \cap_{\beta} Q_{r \times n} = [h_{ij}^{\beta}]_{m \times n}$ , donde

$$h_{ij}^{\beta} = \bigwedge_{k=1}^r \beta(p_{ik}, q_{kj})$$

**Nota:**  $k$  es un entero positivo que puede tomar valores de entre 1 y  $r$  inclusive

**Restricciones:**

- Ninguna de las cuatro operaciones está definida si  $\exists j, k$  tales que  $q_{kj} = 2$ .
- Las operaciones  $\cup_{\alpha}$  y  $\cap_{\alpha}$  no están definidas si  $\exists i, j, k$  tales que  $p_{ik} = 2$  o  $q_{kj} = 2$ .

**Notación 2.14** El símbolo  $\boxtimes$  representa indistintamente a alguna de las dos operaciones  $\cup_{\alpha}$  o  $\cap_{\alpha}$  cuando opere un vector columna de dimensión  $m$  con un vector fila de dimensión  $n$ :

$$y \cup_{\alpha} x^t = y \boxtimes x^t = y \cap_{\alpha} x^t \quad (2.33)$$

De acuerdo con la expresión 2.33 se tiene que:

$$y \boxtimes x^t = \begin{pmatrix} \alpha(y_1, x_1) & \alpha(y_1, x_2) & \cdots & \alpha(y_1, x_n) \\ \alpha(y_2, x_1) & \alpha(y_2, x_2) & \cdots & \alpha(y_2, x_n) \\ \vdots & \vdots & & \vdots \\ \alpha(y_m, x_1) & \alpha(y_m, x_2) & \cdots & \alpha(y_m, x_n) \end{pmatrix}$$

entonces la  $ij$ -ésima componente de la matriz  $y \boxtimes x^t$  esta dada por:

$$[y \boxtimes x^t]_{ij} = \alpha(y_i, x_j) \quad (2.34)$$

es decir, para cada  $\mu = 1, 2, \dots, p$  se tiene:

$$[y^\mu \boxtimes (x^\mu)^t]_{ij} = \alpha(y_i^\mu, x_j^\mu) \quad (2.35)$$

Para este momento se tiene el bagaje de conocimientos necesario para hacer uso de las memorias como tal.

### Memorias asociativas $\alpha\beta$ de tipo $\bigvee$

Se mencionó anteriormente que las memorias asociativas  $\alpha\beta$  funcionan para los modos heteroasociativo y autoasociativo.

La fase de aprendizaje consta de dos fases; en el primero se utiliza el operador  $\boxtimes$ , y en el segundo se echa mano del operador máximo  $\bigvee$ .

#### Fase de aprendizaje

**Fase 1.** Para cada  $\mu = 1, 2, \dots, p$ , a partir de la pareja  $(x^\mu, y^\mu)$  se construye la matriz

$$[y^\mu \boxtimes (x^\mu)^t]_{m \times n} \quad (2.36)$$

**Fase 2.** Se aplica el operador binario máximo  $\bigvee$  a las matrices obtenidas en la fase 1.

$$\mathbf{V} = \bigvee_{\mu=1}^p [y^\mu \boxtimes (x^\mu)^t] \quad (2.37)$$

La entrada  $ij$ -ésima está dada por la siguiente expresión:

$$v_{ij} = \bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu) \quad (2.38)$$

y de acuerdo con la definición de  $\alpha : A \times A \longrightarrow B$ , es posible observar que  $v_{ij} \in B, \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}$ .

### Fase de recuperación

Se presenta un patrón  $x^\omega$ , con  $\omega \in \{1, 2, \dots, p\}$ , a la memoria heteroasociativa o autoasociativa  $\alpha\beta$  tipo  $\bigvee$  y se realiza la operación  $\mathbb{m}_\beta$ :

$$\mathbf{V}\mathbb{m}_\beta x^\omega \quad (2.39)$$

Las dimensiones de la matriz  $\mathbf{V}$  son  $m \times n$  y  $x^\omega$  es un vector columna de dimensión  $n$ ; debido a esto, el resultado de la operación anterior debe ser un vector columna de dimensión  $m$ , cuya  $i$ -ésima componente es posible obtener a partir de la siguiente expresión:

$$(\mathbf{V}\mathbb{m}_\beta x^\omega)_i = \bigwedge_{j=1}^n \beta(v_{ij}, x_j^\omega) \quad (2.40)$$

$$(\mathbf{V}\mathbb{m}_\beta x^\omega)_i = \bigwedge_{j=1}^n \beta \left\{ \left[ \bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu) \right], x_j^\omega \right\} \quad (2.41)$$

### Memorias asociativas $\alpha\beta$ de tipo $\bigwedge$

La memoria autoasociativa o heteroasociativa  $\alpha\beta$  de tipo *min* para realizar la fase de aprendizaje debe seguir dos etapas; en la primera se debe utilizar el operador  $\boxtimes$ , y en la segunda se hace uso del operador  $\bigwedge$ .

**Primera etapa.** Para cada  $\mu = 1, 2, \dots, p$ , a partir de la pareja  $(x^\mu, y^\mu)$  se construye la matriz

$$[y^\mu \boxtimes (x^\mu)^t]_{m \times n} \quad (2.42)$$

**Segunda etapa.** Se aplica el operador binario mínimo  $\bigwedge$  a las matrices obtenidas en la fase 1.

$$\Lambda = \bigwedge_{\mu=1}^p [y^\mu \boxtimes (x^\mu)^t] \quad (2.43)$$

La entrada  $ij$ -ésima está dada por la siguiente expresión:

$$\lambda_{ij} = \bigwedge_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu) \quad (2.44)$$

y de acuerdo con la definición de  $\alpha : A \times A \longrightarrow B$ , es posible observar que  $\lambda_{ij} \in B, \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}$ .

### Fase de recuperación

En la fase de recuperación para la memoria autoasociativa o heteroasociativa, se debe presentar un patrón  $x^\omega$ , con  $\omega \in \{1, 2, \dots, p\}$ , a la memoria asociativa  $\alpha\beta$  de tipo  $\bigwedge$  y se debe realizar la operación  $\mathbb{m}_\beta$ :

$$\Lambda \mathbb{U}_\beta x^\omega \quad (2.45)$$

Las dimensiones de la matriz  $\Lambda$  son  $m \times n$  y  $x^\omega$  es un vector columna de dimensión  $n$ ; debido a esto, el resultado de la operación anterior debe ser un vector columna dimensión  $m$ , cuya  $i$ -ésima componente es posible obtener a partir de la siguiente expresión:

$$(\Lambda \mathbb{U}_\beta x^\omega)_i = \bigvee_{j=1}^n \beta(\lambda_{ij}, x_j^\omega) \quad (2.46)$$

$$(\Lambda \mathbb{U}_\beta x^\omega)_i = \bigvee_{j=1}^n \beta \left\{ \left[ \bigwedge_{\mu=1}^n \alpha(y_i^\mu, x_j^\mu) \right], x_j^\omega \right\} \quad (2.47)$$

## 2.2. Estado del arte en estudios experimentales

Dentro del ámbito de las memorias asociativas, los trabajos más recientes realizados, están en el desarrollo de las memorias morfológicas (Ritter, Sussner, Diaz-de-Leon, 1999) y en las memorias  $\alpha\beta$  (Yáñez-Márquez, 2002). Algunas memorias asociativas pioneras han sido retomadas, como el caso de la *Lernmatrix* de Steinbuch, que aunque el autor no dejó trabajos posteriores de ésta que sustentaran matemáticamente su funcionamiento, fue tomada nuevamente como un antecedente en la tesis doctoral (Yáñez-Márquez, 2000) y parte de la fundamentación matemática fue desarrollada en la tesis de maestría (Sánchez-Garfias, 2004). Investigaciones recientes muestran que, operando la memoria *Lernmatrix* de cierta manera con la memoria asociativa *Linear Associator* da como origen un clasificador híbrido que genera resultados prometedores (Santiago-Montero, Yáñez-Márquez & Díaz-de-León, 2002; Santiago-Montero, 2003).

Las memorias asociativas, como es el caso también de muchas otras áreas de investigación, necesitan tener un punto que mida su rendimiento ante ciertas condiciones (Raudys, 2003; Davey & Hunt, 2000; Cherkassky, Fasset & Vassilas, 1991; Santiago-Montero, Yáñez-Márquez & Díaz de León, 2002; Yáñez-Márquez, 2002), con la finalidad de mostrar su capacidad, ya sea en función del almacenamiento, de la rapidez de procesamiento, de la recuperación de patrones, entre otras características (Chen & Honovar, 1995; Graham & Willshaw, 1995; Imada & Araki, 1995; Adeodato & Taylor, 1996; Storkey, 1997; Bosch & Kurfess, 1988; Jagota, Narasimhan & Regan, 1988).

Existen registros de estudios comparativos estadísticos donde muestran que las redes neuronales pueden ser vistas como memorias asociativas (Cherkassky, Fasset

& Vassilas, 1991), hay otros comparativos que muestran variaciones de memorias de correlación donde se comparan los rendimientos de éstas (Raudys, 2003); existen también comparativos donde la red Hopfield es tratada como memoria asociativa (Yáñez-Márquez & Díaz de León, 2002; Davey & Hunt, 2000); hay investigaciones en las cuales se han realizado variaciones a la memoria Hopfield y han dejado plasmados los resultados en un estudio comparativo (Davey & Hunt, 2000). Es cierto que el rendimiento puede variar por la forma de la implementación y el tamaño de las muestras con las que se experimentarán, pero es una realidad que este tipo de análisis muestra el perfil del modelo que se esté estudiando (Duin, 1996).

Un trabajo de investigación que tiene características similares de comparación para el estudio estadístico, como se realiza en este trabajo de tesis es: "Linear Algebra Approach to Neural Memories and Noise Performance of Neural Classifier" (Cherkassky, Fasset & Vassilas, 1991). Este trabajo tiene la finalidad de mostrar que las redes neuronales pueden ser usadas como memorias asociativas para el proceso de clasificación. Las memorias que son usadas en estos experimentos son: la "memoria generalizada inversa (GI)" y "la memoria de matriz de correlación (CMM)". El planteamiento es el siguiente: las memorias están fundamentadas en matrices de correlación y la manera de operarlas es a través de técnicas estándar de álgebra lineal. Se hace énfasis en el uso de modelos de memorias asociativas distribuidas (DAM).

*Las características similares que tiene este trabajo con esta tesis son:*

1. Trabajan con modelos de memorias asociativas.
2. Tienen un punto de comparación entre memorias asociativas.
3. Los patrones a recuperar están alterados con ruido.
4. Las memorias funcionan para el modo heteroasociativo y/o autoasociativo.
5. Tienen una muestra bastante grande de patrones a aprender y recuperar.

*Las diferencias que existe entre estos dos trabajos son:*

1. Que el presente trabajo de tesis muestra los resultados de once diferentes modelos de memorias asociativas y el trabajo de Cherkassky, Fasset & Vassilas, muestra tan sólo dos memorias (Cherkassky, Fasset & Vassilas, 1991).

2. El punto de comparación del desempeño entre las memorias GI y las memorias CMM se basa en lo robusta que puede resultar la matriz de aprendizaje después de realizarles algunas operaciones de álgebra lineal a las memorias, porque la forma de recuperar los patrones es la utilizada en la memoria *Linear Associator*:  $y = Mx$ . El punto de comparación para determinar el desempeño de las memorias asociativas en el presente trabajo de tesis, es la recuperación perfecta del patrón en sí.

3. Las memorias GI y CMM son sometidas solamente a ruido sustractivo del 10%, pero los vectores con ruido son preprocesados antes de intentar recuperarlos. Las

memorias analizadas en el presente trabajo de tesis son sometidas a ruido aditivo, sustractivo y mezclado, con los siguientes porcentajes: 5, 10, 20, 40, 50, 60, 80, 90 y 99% para los tres tipos de ruido; a excepción de las memorias híbridas heurísticas, todos los patrones alterados con ruido son presentados tal cual a las memorias para su recuperación.

4. Las memorias GI y CMM sólo fueron probadas para el modo autoasociativo. Las memorias que son analizadas en el presente trabajo de tesis son probadas para ambos modos, heteroasociativo y autoasociativo, con dos excepciones: la memoria Hopfield, que por su diseño original solamente funciona en el modo autoasociativo, y la Lernmatrix que de origen sólo funciona para el modo heteroasociativo.

5.

a) Las muestras usadas para la memoria GI y la memoria CMM fueron: 100 diferentes caracteres como: alfabeto inglés, alfabeto griego y el resto fueron caracteres reflejados. Los caracteres fueron representados con imágenes binarias de 8 pixeles de alto por 8 pixeles de largo, cuyos valores fueron 0 para la parte oscura y 1 para la parte blanca. El ruido fue generado de forma aleatoria.

b) Las muestras usadas en el presente trabajo fueron: 139 imágenes binarias de 50 pixeles de alto por 50 pixeles de largo, el valor cero es asignado para la parte oscura de la imagen y el valor 1 para la parte blanca de la imagen. Los caracteres usados fueron: las 26 letras del alfabeto en mayúsculas, 26 letras del alfabeto en minúsculas, 26 caracteres del alfabeto griego minúsculas, 26 caracteres del alfabeto griego en mayúsculas, 26 imágenes misceláneas y 9 caracteres que representan los números del 1 al 9. El tipo de ruido aditivo, sustractivo y mezclado fue generado aleatoriamente.

Los modelos de memorias clásicos que son analizados en este trabajo de tesis, así como las memorias morfológicas y las memorias  $\alpha\beta$  no han sido, previamente a este trabajo de tesis, comparadas a plenitud entre ellas, ante las mismas condiciones de aprendizaje y recuperación (Yañez-Márquez, 2002). El presente trabajo tiene esa finalidad.

## Capítulo 3

# Herramientas matemáticas

A lo largo de este capítulo se analizarán las herramientas matemáticas usadas para el desarrollo de este trabajo de tesis. Se destaca el hecho de que este capítulo es crucial para el cabal entendimiento del funcionamiento de las memorias asociativas. El capítulo está estructurado de la siguiente manera: en primera instancia, se hablará de las sumaciones, posteriormente se destacará el uso de las operaciones matriciales y de los vectores  $n$ -dimensionales en el ámbito de las memorias asociativas; después se describe cómo se generan los diferentes tipos de ruidos en imágenes binarias y por último se trata el tema de los operadores morfológicos usados en el presente trabajo.

### 3.1. Sumaciones

Una sumación es una serie o sucesión que es afectada por algún operador; por ejemplo, cuando se hace uso de  $\sum$  (sigma) se está hablando de una serie cuyo operador que está afectando es una suma, y se le conoce como sumatoria. En el presente trabajo no solamente se hace uso de sumatorias, sino de series afectadas por el operador  $\bigvee$  (máximo) o por el operador  $\bigwedge$  (mínimo). Este tipo de operaciones son significativas en las memorias asociativas: Lernmatrix, morfológicas y  $\alpha\beta$ . Se analizará a continuación las sumatorias, el operador máximo y el operador mínimo.

#### 3.1.1. Sumatorias

El carácter griego cuyo nombre es sigma  $\sum$ , dentro de las matemáticas se ha usado para facilitar la escritura de la suma de los elementos de una sucesión y puede definirse de la siguiente manera:

$$\sum_{i=m}^n F(i) = F(m) + F(m+1) + F(m+2) + \cdots + F(n)$$

Donde  $m$  y  $n$  son enteros y se cumple que  $m \leq n$ . El número  $m$  se denomina límite inferior de la suma y  $n$  se llama límite superior. El símbolo  $i$  es un símbolo convencional

y puede ser sustituido por cualquier otro. En el capítulo 2 se puede observar que la memoria *Lernmatrix* usa la sumatoria para el proceso de recuperación; por su parte, la memoria *Linear Associator* hace uso de las sumatorias en su proceso de aprendizaje y la memoria *Hopfield* también hace uso de las sumatorias en su fase de aprendizaje. Las sumatorias son una piedra angular en los procesos ya sea de aprendizaje o de recuperación para las memorias asociativas clásicas. Se verán algunos ejemplos de cómo hacer uso de las sumatorias.

**Ejemplo 3.1**

$$\sum_{i=4}^6 i^3 = 4^3 + 5^3 + 6^3$$

**Ejemplo 3.2** Cuando los términos de una suma contienen subíndices, la suma:

$$a_1 + a_2 + \cdots + a_n$$

se puede escribir con la notación sigma de la siguiente manera:

$$\sum_{i=1}^n a_i$$

**Ejemplo 3.3** Sea la suma de los elementos de la sucesión:

$$-3a_3 + 4a_4 - 5a_5 + 6a_6 - 7a_7 + 8a_8$$

Esta sucesión puede ser representada de la siguiente manera:

$$\sum_{i=3}^8 (-1)^i i a_i$$

### 3.1.2. Operador $\bigvee$

El operador  $\bigvee$  dentro de una sucesión significa obtener el valor máximo de los elementos que se encuentran dentro de ésta, y se define de la siguiente manera:

$$\bigvee_{i=m}^n F(i) = F(m) \bigvee F(m+1) \bigvee F(m+2) \bigvee \cdots \bigvee F(n)$$

Donde  $m$  y  $n$  son enteros y se cumple que  $m \leq n$ . El número  $m$  se denomina límite inferior del máximo y  $n$  se llama límite superior. El símbolo  $i$  es un símbolo convencional y puede ser sustituido por cualquier otro. El operador  $\bigvee$  es fundamental para el proceso de recuperación en la memoria *Lernmatrix*. Este operador es determinante, también, para la memoria morfológica de tipo max y para la memoria  $\alpha\beta$  de tipo max, de hecho el término max de estas memorias, es debido a que el operador  $\bigvee$  desempeña un papel fundamental en el buen rendimiento de estas memorias. El siguiente ejemplo muestra el funcionamiento del operador  $\bigvee$ .

**Ejemplo 3.4.** *Obtener el máximo de los 3 primeros números enteros positivos.*

$$\bigvee_{i=1}^3 i = 1 \bigvee 2 \bigvee 3 = 3$$

### 3.1.3. Operador $\bigwedge$

El operador  $\bigwedge$  dentro de una sucesión, significa obtener el valor mínimo que se encuentra dentro de ésta y se define de la siguiente manera:

$$\bigwedge_{i=m}^n F(i) = F(m) \bigwedge F(m+1) \bigwedge F(m+2) \bigwedge \cdots \bigwedge F(n)$$

Donde  $m$  y  $n$  son enteros y se cumple que  $m \leq n$ . El número  $m$  se denomina límite inferior del mínimo y  $n$  se llama límite superior. En el capítulo 2 se habla de la memoria morfológica min y la memoria  $\alpha\beta$  min, obviamente se deduce que el operador  $\bigwedge$  es determinante para el buen desempeño de éstas. El siguiente ejemplo muestra el funcionamiento del operador min:

**Ejemplo 3.5.** *Obtener el mínimo de los 3 primeros números enteros positivos.*

$$\bigwedge_{i=1}^3 i = 1 \bigwedge 2 \bigwedge 3 = 1$$

### 3.2. Operaciones matriciales

En virtud de que las memorias asociativas se representan por medio de matrices, es necesario exponer algunos conceptos ya conocidos sobre las notaciones de las matrices y vectores, que de hecho ya se usaron en el capítulo 2 y se usarán en el capítulo 4.

**Notación 3.6** Las matrices se representan con esta notación: si  $m, n, r$  son números enteros positivos,  $\mathbf{A} = [a_{ij}]_{m \times r}$  representa una matriz de dimensiones  $m \times r$ , cuya  $ij$ -ésima entrada es  $a_{ij}$ , y  $\mathbf{B} = [b_{ij}]_{r \times n}$  representa una matriz de dimensiones  $r \times n$ , cuya  $ij$ -ésima entrada es  $b_{ij}$ .

**Definición 3.7** Una matriz  $A$  de dimensiones  $m \times r$  es un arreglo rectangular, cuyos  $m \times r$  elementos están dispuestos en  $m$  renglones y  $r$  columnas.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1r} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2r} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{ir} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mr} \end{pmatrix}$$

**Definición 3.8** Suma de matrices: Sean  $A$  y  $B$  dos matrices de  $m \times n$ . Entonces la suma de  $A$  y  $B$  es la matriz de dimensión  $m \times n$ ,  $A + B$  dada por:

$$A + B = (a_{ij} + b_{ij}) = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{pmatrix}$$

es decir,  $A + B$  es la matriz  $m \times n$  que se obtiene al sumar los componentes correspondientes de  $A$  y  $B$ .

**Definición 3.9** Resta de matrices: Sean  $A$  y  $B$  dos matrices de  $m \times n$ . Entonces la suma de  $A$  y  $B$  es la matriz  $m \times n$ ,  $A + B$  dada por:

$$A - B = (a_{ij} - b_{ij}) = \begin{pmatrix} a_{11} - b_{11} & a_{12} - b_{12} & \cdots & a_{1n} - b_{1n} \\ a_{21} - b_{21} & a_{22} - b_{22} & \cdots & a_{2n} - b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} - b_{m1} & a_{m2} - b_{m2} & \cdots & a_{mn} - b_{mn} \end{pmatrix}$$

es decir,  $A - B$  es la matriz  $m \times n$  que se obtiene al restar los componentes correspondientes de  $A$  y  $B$ .

**Definición 3.10** *Máximo de matrices:* Sean  $A$  y  $B$  dos matrices de  $m \times n$ . Entonces el máximo de  $A$  y  $B$  es la matriz  $m \times n$ ,  $A \vee B$  dada por:

$$A \vee B = (a_{ij} \vee b_{ij}) = \begin{pmatrix} a_{11} \vee b_{11} & a_{12} \vee b_{12} & \cdots & a_{1n} \vee b_{1n} \\ a_{21} \vee b_{21} & a_{22} \vee b_{22} & \cdots & a_{2n} \vee b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} \vee b_{m1} & a_{m2} \vee b_{m2} & \cdots & a_{mn} \vee b_{mn} \end{pmatrix}$$

es decir,  $A \vee B$  es la matriz  $m \times n$  que se genera al obtener el máximo de los componentes correspondientes de  $A$  y  $B$ .

**Definición 3.11** *Mínimo de matrices:* Sean  $A$  y  $B$  dos matrices de  $m \times n$ . Entonces el mínimo de  $A$  y  $B$  es la matriz  $m \times n$ ,  $A \wedge B$  dada por:

$$A \wedge B = (a_{ij} \wedge b_{ij}) = \begin{pmatrix} a_{11} \wedge b_{11} & a_{12} \wedge b_{12} & \cdots & a_{1n} \wedge b_{1n} \\ a_{21} \wedge b_{21} & a_{22} \wedge b_{22} & \cdots & a_{2n} \wedge b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} \wedge b_{m1} & a_{m2} \wedge b_{m2} & \cdots & a_{mn} \wedge b_{mn} \end{pmatrix}$$

es decir,  $A \wedge B$  es la matriz  $m \times n$  que se genera al obtener el mínimo de los componentes correspondientes de  $A$  y  $B$ .

Las anteriores operaciones de matrices son usadas para la construcción de la matriz de aprendizaje de los modelos de memorias asociativas. La memoria *Linear Associator* y la memoria *Hopfield*, por ejemplo, hacen uso de la suma de matrices para crear su matriz de aprendizaje. Las memorias de tipo min hacen uso de la operación de mínimo de matrices para generar la matriz de aprendizaje y las memorias de tipo max, usan la operación de máximo de matrices para su fase de aprendizaje. Como se puede constatar en el capítulo 2, las matrices son fundamentales para todo modelo de memoria asociativa.

### 3.3. Vectores n-dimensionales

Dentro de la literatura del álgebra lineal se considera a los vectores como un tipo especial de matriz (Lang, S. 1976; Grossman, S.I, 1996; Leithold, L. 1998). Así, ciertas combinaciones en los valores de  $m$ ,  $n$  y  $r$ , dan lugar a casos específicos, los cuales vale la pena mencionar porque simplifican las operaciones:

- Si  $m = 1$  y  $r = 1$ , existe una única entrada,  $a_{11}$ . En este caso se trata de un número que indicaremos con la letra  $a$  sin subíndices.

- Si  $m = 1$  y  $r > 1$ , se trata de un vector fila con  $r$  entradas:  $(a_{11}, a_{12}, \dots, a_{1r})$ . Se eliminará el subíndice 1 que corresponde a  $m$  para simplificar la notación, y el vector fila se denotará así:  $\mathbf{a} = (a_1, a_2, \dots, a_r)$ .
- Si  $r > 1$  y  $n = 1$ , se trata de un vector columna con  $n$  entradas:

$$\mathbf{b} = \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{r1} \end{pmatrix}$$

Se eliminará el subíndice 1 que corresponde a  $r$  para simplificar la notación, y el vector columna se denotará así:

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \end{pmatrix}$$

por ser casos particulares de matrices, los vectores se sujetan a las definiciones citadas en la sección anterior.

### 3.3.1. Producto externo

Sea  $A$  una matriz de  $m \times 1$ , donde:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$$

y sea  $B$  una matriz de  $1 \times m$ , tal que:

$$\mathbf{b} = (b_1, b_2, \dots, b_n)$$

el producto externo está dado por:

$$\mathbf{a} \cdot \mathbf{b} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \cdot (b_1, b_2, \dots, b_n) = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m b_1 & a_m b_2 & \cdots & a_m b_n \end{pmatrix}$$

dando como resultado una matriz de dimensión  $m \times n$ .

El producto externo tiene incidencia en el proceso de aprendizaje para la memoria Linear Associator y la memoria Hopfield en la primera fase de aprendizaje.

En algunos casos, en lugar de usar un producto entre un vector renglón y un vector columna, se usará el operador  $\bigvee$  o el operador  $\bigwedge$ , y de igual manera se generará una matriz de  $m \times n$ , con la diferencia que el resultado estará afectado, no por un producto, sino por el operador máximo o el operador mínimo. En el capítulo 2 de este trabajo, se puede constatar este hecho, ya que las memorias morfológicas y las memorias  $\alpha/\beta$  operan de esta manera para su primera fase de aprendizaje.

### 3.4. Ruido

Se comentó en el capítulo 1 que el ruido puede clasificarse en: ruido aditivo, ruido sustractivo o ruido mezclado. En este apartado se analizará cómo se genera el ruido en el conjunto fundamental, el cual estará formado por imágenes binarias.

Es posible especificar la cantidad de ruido con que se ha alterado un patrón binario mediante la probabilidad de ruido, la cual se calcula haciendo uso de la expresión siguiente (Pitas, 2000):

$$p = \frac{r}{n} \quad (3.1)$$

donde  $r$  es la cantidad de componentes que han sido alterados por el ruido y  $n$  es la cantidad total de componentes en el patrón.

#### 3.4.1. Ruido aditivo

El ruido aditivo se puede calcular de la siguiente manera:

$$p_{rr} = \binom{r}{n_1} (100) \quad (3.2)$$

donde  $r$  es la cantidad de pixeles (con valor lógico de *background*) que han sido alterados por el ruido y  $n_1$  es la cantidad de pixeles con valor lógico de *foreground* que hay en el patrón.

**Nota:** Claramente se puede observar que si  $n$  es la cantidad total de pixeles en la imagen, y si  $n_1$  es la cantidad de pixeles con valor lógico de *foreground*, se debe cumplir que  $n_1 < n$ ; además, la cantidad de pixeles con valor lógico de *background* en el patrón será:  $n - n_1$ .

Si en un caso particular se conoce el valor de  $p_{rr}$  y se desea saber la probabilidad de ruido  $p$ , se combinan las expresiones 3.1 y 3.2 para dar lugar a la expresión 3.3:

$$p = \left( \frac{p_{rr}}{100} \right) \left( \frac{n_1}{n} \right) \quad (3.3)$$

Si un patrón está almacenado en un arreglo lineal binario  $pat[n]$  con  $n_1$  pixeles que poseen valor lógico de *foreground*, el primer paso para lograr afectar el patrón con un valor  $p_{rr}$  de ruido aditivo es calcular  $r$ , el número de pixeles que serán cambiados de valor lógico de *background* a valor lógico de *foreground*; para hacerlo, se debe calcular  $r$  de la siguiente manera:

$$r = \left( \frac{p_{rr}}{100} \right) (n_1)$$

### 3.4.2. Ruido sustractivo

Para calcular el ruido sustractivo se aplican las mismas expresiones 3.1, 3.2 y 3.3, con la diferencia de que ahora  $r$  es el número de pixeles de valor lógico de *foreground* que han sido alterados con ruido. Se debe tener en cuenta el siguiente hecho: mientras que el ruido aditivo agregado a un patrón específico depende de la diferencia que hay entre el número de pixeles con valor lógico de *background* y el número de pixeles con valor lógico de *foreground* presentes en los patrones fundamentales, ocurre que la cantidad de ruido sustractivo que se añade a un patrón específico depende sólo del número de pixeles con valor lógico de *foreground* que hay en ese patrón fundamental; es decir, el máximo porcentaje de ruido sustractivo con el que es posible alterar un patrón fundamental es 100%, porque en ese momento desaparecen los pixeles con valor lógico de *foreground* del patrón, independientemente del número de pixeles con valor lógico de *background* presentes.

Si un patrón está almacenado en un arreglo binario  $pat[n]$  con  $n_1$  pixeles que poseen valor lógico de *foreground*, el primer paso para lograr afectar el patrón con un valor  $p_{rr}$  de ruido sustractivo es calcular  $r$ , el número de pixeles que serán cambiados de valor lógico de *foreground* a valor lógico de *background*; para hacerlo se debe calcular  $r$  de la siguiente manera:

$$r = \left( \frac{p_{rr}}{100} \right) (n_1)$$

### 3.4.3. Ruido mezclado

Dado que el ruido mezclado afecta tanto a pixeles con valor lógico de *background* como pixeles con valor lógico de *foreground*, se tomará como base a expresión 3.1 para calcular el la probabilidad de ruido mezclado; además, si un patrón está almacenado en un arreglo binario  $pat[n]$ , para afectar los valores lógicos independientemente su valor se debe calcular  $r$  e invertir los valores lógicos. Para calcular  $r$ , se debe hacer de la siguiente manera:

$$r = pn$$

donde  $n$  es la cantidad total de componentes en el patrón y  $r$  es la cantidad de componentes que han sido alterados por el ruido.

## 3.5. Operadores morfológicos

La morfología matemática, que comenzó a desarrollarse a finales de los años sesenta, permanece hoy en día como una parte importante en análisis de imágenes. La morfología matemática está basada en la geometría y la forma. Las operaciones morfológicas simplifican las imágenes y preservan las formas principales de los objetos. La morfología matemática suele utilizarse en tareas como las siguientes: suavizar los bordes de alguna región debido a que las imágenes pueden presentar en estos bordes, ruido, protuberancias o valles que obviamente pueden corregirse mediante transformaciones morfológicas. También se pueden separar regiones que se encuentran juntas o juntar regiones que se encuentran separadas.

La morfología matemática aprovecha las propiedades de los conjuntos de puntos, los resultados de la geometría integral y la topología, teniendo como base 2 operaciones que son la erosión y la dilatación. Estas operaciones son las que se utilizan en este tema de tesis para preprocesar los patrones que contiene ruido mezclado y presentar los patrones preprocesados a las memorias heurísticas para que éstas puedan realizar recuperaciones perfectas.

La dilatación y la erosión, definidas en las expresiones 3.4 y 3.5 respectivamente, son dos operaciones básicas de la morfología matemática (Díaz-de-León & Yañez, 2000).

$$\text{Dilatación de } A \text{ por } B : \quad A \oplus B = \{x | (B^-)_x \cap A \neq \phi\} \quad (3.4)$$

$$\text{Erosión de } A \text{ por } B : \quad A \ominus B = \{x | (B)_x \subseteq A\} \quad (3.5)$$

En ambas operaciones,  $A$  es un conjunto cualquiera y  $B$  es un conjunto, típicamente con pocos elementos, llamado elemento estructurante. En el caso en que  $A$  y  $B$  son imágenes binarias, el elemento estructurante es una vecindad con pocos píxeles.

El conjunto  $B^-$  es el reflejado del conjunto  $B$ , y se define así:  $B^- = \{-b | b \in B\}$ , mientras que  $(B)_x$  es una operación llamada traslación del conjunto  $B$  por  $x$ , y se define de la siguiente manera:  $(B)_x = \{x + b | b \in B\}$ .

El resultado generado por dilatar una imagen (patrón) afectada por ruido mezclado, es una imagen que está afectada por ruido aditivo; ahora bien, el resultado generado por erosionar una imagen (patrón) afectada por ruido mezclado es una imagen afectada por ruido sustractivo. En el capítulo 2 se analizaron las memorias morfológicas y las memorias  $\alpha\beta$ , se especificó que las memorias de tipo min son robustas para patrones afectados por el ruido sustractivo y que las memorias de tipo max son excelentes para patrones afectados por el ruido aditivo. Como se puede deducir al hacer uso de las operaciones morfológicas: erosión y dilatación, con los patrones afectados por el ruido mezclado se puede dejar preparado el camino para que las memorias de tipo min o de tipo max pueda recuperar de manera perfecta un patrón. En el capítulo 4 se muestra cómo es generada la memoria híbrida o heurística que hace uso de la erosión y la dilatación.

## Capítulo 4

# Desarrollo

Este capítulo es de vital importancia, debido a que se mostrará el desarrollo del sistema computacional que permitió hacer el estudio estadístico comparativo entre las memorias asociativas descritas en el capítulo 2. Se presentará, además, el desarrollo de un método híbrido haciendo uso de los operadores morfológicos: erosión y dilatación; el método híbrido se aplicará tanto a las memorias morfológicas como a las memorias  $\alpha\beta$ , para que éstas puedan obtener recuperaciones perfectas ante el ruido mezclado.

El capítulo está estructurado de la siguiente manera: implementación del conjunto fundamental, implementación de la generación de los diferentes tipos de ruido; implementación, en detalle, de los operadores morfológicos: dilatación y erosión; implementación de las memorias heteroasociativas y autoasociativas y, por último, se muestra cómo se usa el control Activex Chart de C++ Builder para la creación de gráficas.

Los patrones fundamentales que serán usados para todas las memorias asociativas son imágenes binarias con formato BMP, con un ancho de 50 píxeles y con un alto de 50 píxeles; el valor asignado a los píxeles de color blanco es el valor lógico 1 (de *background*) y el valor de los píxeles de color negro es el valor lógico 0 (de *foreground*). Los píxeles negros son de interés principal ya que definen a la imagen, mientras que los píxeles blancos forman el fondo.

En el apéndice A, que se encuentra en el CD anexo, se detalla la manera en que se implementó la manipulación de las imágenes en formato BMP.

### 4.1. Generación del conjunto fundamental

El conjunto fundamental se define de la siguiente manera:  $(x^\omega, y^\omega) \mid \omega \in \{1, 2, \dots, p\}$ , según la expresión 1.1. Hay que destacar que el conjunto fundamental es imprescindible para la creación de la memoria  $M$ . En el capítulo 2, también se mostró la manera de operar el conjunto fundamental para cada modelo de memoria asociativa, así como su importancia en la fase de recuperación.

Se mencionó anteriormente que el conjunto fundamental para realizar el estudio estadístico comparativo estaría conformado por imágenes binarias en formato BMP, con un tamaño de 50 píxeles de ancho por 50 píxeles de alto. Como son imágenes cuadradas, el primer paso lógico a seguir es: convertir de alguna manera la imagen de 50 x 50 a un vector de 2500 elementos; en el apéndice A se muestra cómo se puede convertir la imagen a un vector.

Como los elementos del conjunto fundamental son archivos con formato BMP, éstos son almacenados en algún lugar del disco duro para ser operados. Cuando el conjunto fundamental es usado, no se cargan todas las imágenes que forman dicho conjunto fundamental en la memoria RAM, sino que se carga una imagen a la vez, haciendo uso de un objeto gráfico de C++ Builder que permite cargar imágenes; conforme se vayan usando las imágenes se van dibujando en la interfase gráfica, lo cual quiere decir que visualmente se ve el conjunto fundamental, pero solamente la imagen (patrón) que se está procesando se encuentra en la memoria. La figura 4.1 muestra la interfase gráfica de cierto conjunto fundamental, aquí se observan las imágenes de las letras del alfabeto, pero la única que está en memoria es la imagen que contiene la letra "Z" ya que fue la última imagen que se cargó.



figura 4.1. Conjunto fundamental a procesar.

Las memorias pueden ser heteroasociativas o autoasociativas. Si una memoria es heteroasociativa, se puede afirmar que:  $\exists \mu \in \{1, 2, \dots, p\}$  para el que se cumple que  $x^\mu \neq y^\mu$ ; es decir, que al menos una imagen  $x^\mu$  es diferente de la imagen correspondiente  $y^\mu$ , siendo igual el número de imágenes tanto  $x^\mu$  como  $y^\mu$ . Si este es el caso, se guardan dos referencias a los directorios donde están almacenados dichos conjuntos de imágenes, para ser llamados más adelante por los modelos de memorias asociativas. En caso de que sean memorias autoasociativas, sólo se escoge un único conjunto de imágenes  $x^\mu$ ;

por esta razón se guarda solamente una referencia a la ruta donde se encuentran las imágenes.

Con lo anterior mencionado, los pasos lógicos para cargar el conjunto fundamental son:

1. *Determinar si la memoria es autoasociativa o heteroasociativa*
2. *Abrir el conjunto fundamental.*

## 4.2. Generación de ruido

Se ha dejado ver a lo largo de este trabajo que una de las cosas apasionantes de las memorias asociativas es la capacidad de recuperación de patrones con cierto grado de ruido; la pregunta que cabe realizar es ¿cómo se genera el ruido? En el capítulo 3, sección 4, se mostró que se pueden generar tres diferentes tipos de ruido: el aditivo, el sustractivo y el mezclado.

### 4.2.1. Ruido aditivo

La idea principal para la generación de ruido aditivo es buscar aleatoriamente una coordenada  $(x, y)$  cuyo valor de pixel sea de *background* y cambiar el valor del pixel a *foreground*, en cantidades determinadas por la expresión 3.3, de acuerdo con el porcentaje de ruido que se desee obtener.

Para generar ruido aditivo se deben seguir los dos pasos que a continuación se detallan:

#### 1. Se obtiene el área de la imagen a través del siguiente algoritmo:

*Se obtiene el negativo de la imagen*

*area = 0*

*Para y = 0 hasta el alto de la imagen, hacer:*

*Para x = 0 hasta el ancho de la imagen, hacer:*

*Si el valor del pixel es igual a 1 entonces*

*area = area + 1*

#### 2. En función a la expresión 3.3 se obtiene la cantidad de ruido

$$\text{cantidad\_de\_ruido} = (\text{area} * \text{porcentaje}) / 100.$$

*Donde porcentaje es una variable que guarda el valor dado por un usuario.*

*Una vez calculado la cantidad de ruido se procede a realizar el siguiente algoritmo:*

*Para i = 0 hasta cantidad\_de\_ruido, hacer:*

*mientras el valor del pixel obtenido aleatoriamente sea igual a 1 repetir este paso de la coordenada x,y obtenida, que hizo que el ciclo anterior terminara, se debe asignar un valor 1 al pixel de esta misma coordenada*

*Se obtiene el negativo de la imagen operada.*

Por ejemplo, el efecto de seguir este algoritmo con un 60% de ruido aditivo es el siguiente:

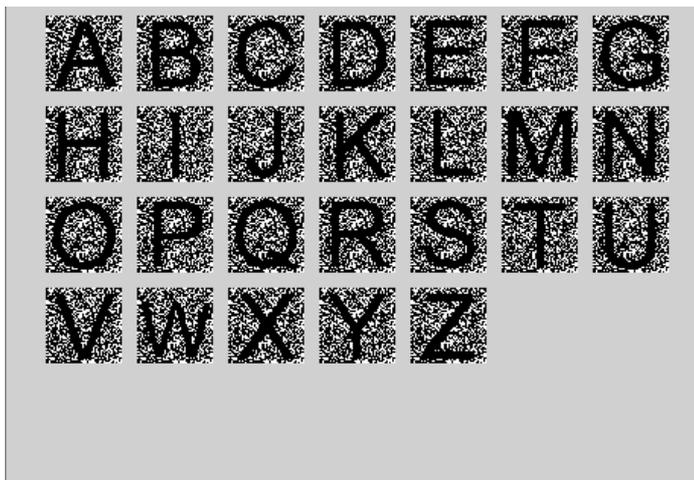


figura 4.2 resultado del algoritmo que genera el ruido aditivo.

#### 4.2.2. Ruido sustractivo

El ruido sustractivo, como lo muestra la figura 1.2 del capítulo 1, significa afectar el área oscura de la imagen, es decir, afectar los pixeles cuyos valores son de *foreground*. La idea central de la generación de ruido sustractivo es encontrar aleatoriamente coordenadas cuyos valores en los pixeles sean de *foreground* y al encontrarlos se cambian los valores de estos pixeles a *background*. Para la generación del ruido sustractivo se debe seguir el mismo procedimiento que se realizó para el ruido aditivo, pero con la diferencia de que ahora se están buscando los pixeles cuyo valor es de *foreground* para cambiarlos a *background*. El ruido sustractivo se genera de la siguiente manera:

##### 1. Se obtiene el área de la imagen a través del siguiente algoritmo:

$area = 0$

Para  $y = 0$  hasta el alto de la imagen, hacer:

Para  $x = 0$  hasta el ancho de la imagen, hacer:

Si el valor del pixel es igual a 1 entonces

$area = area + 1$

##### 2. En función a la expresión 3.3 se obtiene la cantidad de ruido

$cantidad\_de\_ruido = (area * porcentaje) / 100.$

Donde *porcentaje* es una variable que guarda el valor dado por un usuario.

Una vez calculado la cantidad de ruido se procede a realizar el siguiente algoritmo:

Para  $i = 0$  hasta  $cantidad\_de\_ruido$ , hacer:

mientras el valor del pixel obtenido aleatoriamente sea igual a 1 repetir este paso

de la coordenada  $x,y$  obtenida, que hizo que el ciclo anterior terminara, se debe asignar un valor 1 al pixel de esta misma coordenada

El resultado de este algoritmo con un 60% de ruido sustractivo es el siguiente:

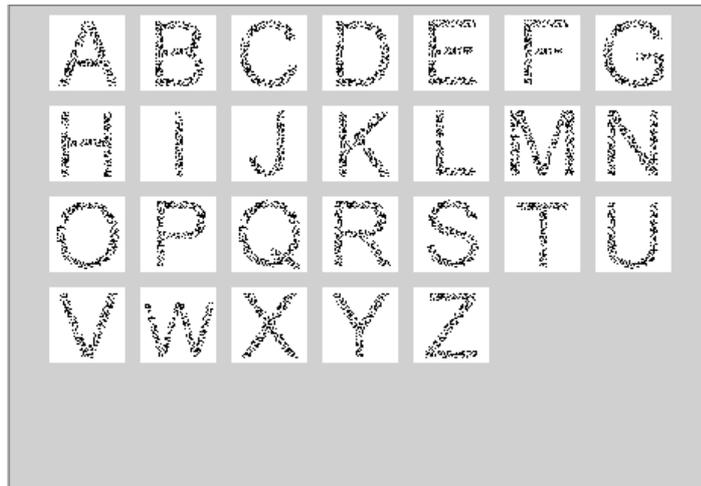


figura 4.3 resultado del algoritmo que genera el ruido sustractivo

### 4.2.3. Ruido mezclado

El ruido mezclado, como lo muestra la figura 1.2 del capítulo 1, consiste en que aleatoriamente se encuentre una coordenada  $(x,y)$  y el valor del pixel que tenga esta coordenada sea cambiado, es decir, si el valor del pixel es de *background* se cambiará por el valor de *foreground*, y si es de *foreground* se cambiará por el de *background*; se debe tomar en cuenta que si el pixel cuya coordenada a cambiar es  $(x,y)$  ya ha sido cambiada anteriormente, no se realizará el cambio de valor; es decir el ruido aleatorio es condicionado. El algoritmo para generar este tipo de ruido se puede implementar de la siguiente manera:

```

ruido = ( ancho_de_imagen * alto_de_imagen * porcentaje_a_afectar ) / 100;
para i= 0 hasta (ancho_imagen-1) * (alto_imagen -1), en incremento de 1 hacer:
  ImgAux [i] = 2;
  Para area = 0 hasta area < ruido-1 hacer
    x = rand( ) % w;
    y = rand( ) % h;
    si GetPixel(x , y) = 1 entonces
      si ImgAux[ y * ancho_imagen + x] = 2 entonces
        ImgAux[ y * ancho_imagen + x] = 0;
        SetPixel(x,y) = 1
        area = area + 1
    sino
      si ImgAux[ y * ancho_imagen + x] = 2 entonces
        ImgAux[ y * w + x] = 1;

```

```

    SetPixel(x,y) = 1
    area = area + 1
fin del GetPixel

```

El resultado de seguir este algoritmo con 40% de ruido mezclado es el siguiente:

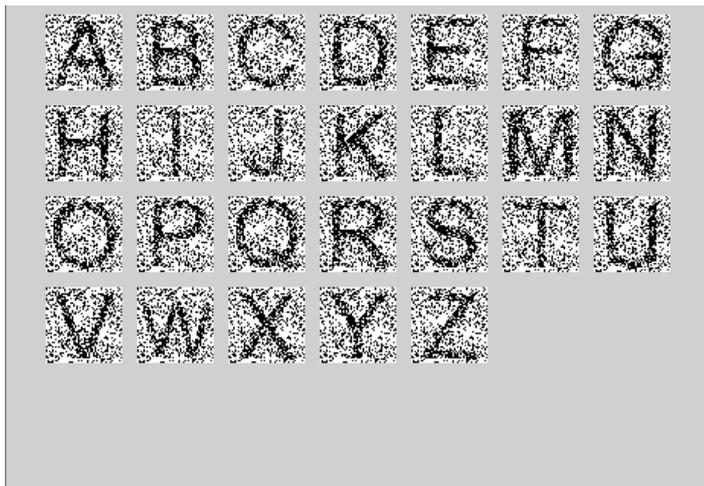


figura 4.4 resultado del algoritmo que genera el ruido mezclado.

### 4.3. Operadores morfológicos

La erosión y la dilatación son los operadores morfológicos que fueron usados en este tema de tesis como adición en las memorias asociativas morfológicas y en las memorias asociativas  $\alpha\beta$  para solucionar el problema de éstas con el ruido mezclado. El nombre que se les dio a estas memorias híbridas fue el de "*memorias heurísticas*".

En la expresión 3.4 se define la dilatación de la siguiente manera: Dilatación de  $A$  por  $B$  :  $A \oplus B = \{x | (B^-)_x \cap A \neq \phi\}$ . Para implementar la dilatación se sugiere el siguiente algoritmo:

```

k = 4
W = Ancho_Imagen_A_Dilatar
H = Alto_Imagen_A_Dilatar
centro_x = ( Ancho_ElementoEstructurante - 1 ) / 2
centro_y = ( Alto_ElementoEstructurante - 1 ) / 2
para y = 0 hasta Alto_Imagen_A_Dilatar, incremento en 1, hacer:
    para x = 0 hasta Ancho_Imagen_A_Dilatar, incremento en 1 hacer:
        min = 0;
        para yy = 0 hasta Alto_ElementoEstructurante, incremento en 1, hacer:
            para xx = 0 hasta Ancho_ElementoEstructurante, incremento en 1, hacer:
                si la coordenada (xx,yy) del ElementoEstructurante es 0 entonces:
                    W = ( 2 * Ancho_Imagen_A_Dilatar + x - ( xx - centro_x ) ) %

```

```

    Ancho_Imagen_A_Dilatar
    H = ( 2 * Alto_Imagen_A_Dilatar + y - ( yy - centro_y ) ) %
        Alto_Imagen_A_Dilatar
    si la coordenada (W,H) de la Imagen_A_Dilatar es diferente de 0 entonces
        min = min + 1
    si min > k
        la coordenada (x,y) de la Imagen_A_Dilatar es 1
    en caso contrario
        la coordenada (x,y) de la Imagen_A_Dilatar es 0

```

El resultado obtenido de este algoritmo en una imagen que contiene el 10% de ruido mezclado se muestra en la figura 4.5.



figura 4.5. Resultado del algoritmo de dilatación.

En la expresión 3.5 la erosión se define de la siguiente manera: Erosión de  $A$  por  $B$ :  $A \ominus B = \{x | (B)_x \subseteq A\}$ . Para implementar la erosión de una imagen se sugiere el siguiente algoritmo:

```

k = 4;
centro_x = ( Ancho_ElementoEstructurante - 1 ) / 2
centro_y = ( Alto_ElementoEstructurante - 1 ) / 2
para y = 0 hasta Alto_Imagen_A_Erosionar, incrementando en 1, hacer:
    para x = 0 hasta Ancho_Imagen_A_Erosionar, incremento en 1, hacer:
        min = 0;
        para yy = 0 hasta Alto_ElementoEstructurante, incemento en 1, hacer:
            para xx = 0 hasta Ancho_ElementoEstructurante, incrementeo en 1 hacer:
                si la coordenada (xx,yy) de ElementoEstructurante = 0 entonces:
                    W = ( Ancho_Imagen_A_Erosionar + x + xx - centro_x ) %
                        Ancho_Imagen_A_Erosionar
                    H = ( Alto_Imagen_A_Erosionar + y + yy - centro_y ) %
                        Alto_Imagen_A_Erosionar
                si la coordenada (W,H) de Imagen_A_Erosionar = 0 entonces:
                    min = min + 1
            si min <= k
                if( min <= k )
                    la coordenada (x,y) de Imagen_A_Erosionar es 1
                en caso contrario
                    la coordenada (x,y) de Imagen_A_Erosionar es 0

```

El resultado obtenido con este algoritmo aplicado a una imagen que contiene el 10% de ruido mezclado se muestra en la figura 4.6.



figura 4.6. Resultado del algoritmo de erosión.

El elemento estructurante del que se habló en la sección 3.5 y que se usó en la implementación de los algoritmos de la erosión y de la dilatación es el siguiente:

|   |   |   |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Las memorias morfológicas de tipo *max* son muy robustas al ruido aditivo y las memorias morfológicas *min* son muy robustas al ruido sustractivo. Lo mismo sucede con las memorias  $\alpha\beta$ . Al dilatar un patrón que contenga ruido mezclado el resultado de este patrón, es un patrón con ruido aditivo (figura 4.5), que puede ser recuperado de manera perfecta con las memorias de tipo *max* o al erosionarlo, el resultado es un patrón con ruido sustractivo (figura 4.6), que puede ser recuperado de manera perfecta con las memorias de tipo *min*.

#### 4.4. Implementación de las Memorias Asociativas

Las memorias asociativas que se someterán al estudio experimental comparativo son las que se describieron a detalle en el capítulo 2; éstas son: *la memoria Lernmatrix*, *la memoria Linear Associator*, *la memoria autoasociativa Hopfield*, *las memorias morfológicas* y *las memorias  $\alpha\beta$* .

Antes de comenzar a describir la implementación de estas memorias, es necesario definir la estructura general de una memoria asociativa. La idea se concibió de la siguiente manera: Una memoria asociativa puede ser vista como un objeto, cuyas características principales son: 1. El saber aprender un conjunto fundamental que está conformado por una asociación de vectores y 2. Saber recuperar un patrón completo a partir de un patrón de entrada. Todas las memorias asociativas analizadas en el capítulo 2 tienen estas características; teniendo esto presente, se debe definir una plantilla genérica que indique las características de las memorias, la plantilla es la siguiente:

```
class MEMORY{
    protected:
        int *Memory;
        int *x;
        int *y;
        int width;
        int height;

    public:
        virtual int Learn( int *x, int *y ) = 0;
        virtual int *Get( int *x ) = 0; };
```

Como se observa en la plantilla anterior es importante definir un espacio en la memoria RAM donde se almacenará la matriz de aprendizaje  $M$  (*int \*Memory*), pero para formar  $M$  está claro que primero se debe definir su tamaño; para asignar el tamaño de  $M$  es necesario conocer a priori la cantidad de elementos (*int width*) que contiene el vector de entrada  $x$  (*int \*x*) y la cantidad de elementos (*int height*) que contiene el vector de salida  $y$  (*int \*y*). Donde cobran sentido los vectores  $x$  y  $y$  definidos en la plantilla son en las funciones de aprendizaje y de recuperación. Al declarar las funciones de aprendizaje (*virtual int Learn( int \*x, int \*y ) = 0*) y de recuperación (*virtual int \*Get( int \*x ) = 0*) como virtualmente puras, se está indicando que estas funciones serán definidas por la memoria asociativa que herede esta plantilla; dicho de otra forma, existirán tantas funciones de aprendizaje y recuperación como memorias se implementen. Sin embargo, aunque sea de manera diferente la implementación (algoritmos), las funciones de aprendizaje y de recuperación de las memorias tendrán el mismo nombre: a este último término dentro del paradigma de la programación orientada a objetos se le conoce como "polimorfismo".

Teniendo como base la plantilla genérica de memorias asociativas se comenzará con la implementación de éstas; en las secciones siguientes sólo se hablará de las implementaciones de las funciones de aprendizaje y de recuperación.

#### 4.4.1. Implementación de la memoria asociativa Lernmatrix

##### Fase de aprendizaje

La *Lernmatrix* acepta como entrada a los siguientes patrones binarios  $x^\mu \in A^n$ ,  $A = \{0, 1\}$  y genera como salida la clase  $y^\mu \in A^p$  que le puede corresponder de entre  $p$  clases diferentes. Cada uno de los componentes  $m_{ij}$  de  $M$  de la *Lernmatrix*, tiene valor 0 al iniciar el aprendizaje, y se actualiza de acuerdo con la expresión 2.2 que contiene la regla  $m_{ij} + \Delta m_{ij}$ , donde:

$$\Delta m_{ij} = \begin{cases} +\varepsilon & \text{si } x_i^\mu = 1 = y_j^\mu \\ -\varepsilon & \text{si } x_i^\mu = 0 \text{ y } y_j^\mu = 1 \\ 0 & \text{en otro caso} \end{cases}$$

siendo  $\varepsilon$  una constante positiva escogida previamente.

##### Construcción de la memoria Lernmatrix en función de la expresión 2.1

```

LearnMatrix( width, height ){
    Learnmatrix::width = width
    LearnMatrix::height = height
    E = 1;
    numero_de_clases = alto
    Contador_de_patrones = 0;
    se crea el espacio en la memoria ram para la matriz de aprendizaje M
    en función a width y height
    se pone a 0 todos los elementos de la matriz de M
}

```

**Fase de aprendizaje en función de la expresión 2.2**

```

Learn( x, y ){
  Learnmatrix::x = x
  Learnmatrix::y = y
  Apuntador = Contador_de_patrones * LearnMatrix::width;
  para i = 0 hasta Learnmatrix::width, incremento de 1, hacer
    si Learnmatrix::x[ i ] = 1 y Learnmatrix[ Contador_de_patrones ] = 1
      Learnmatrix::Memory[ Apuntador + i ] =
        Learnmatrix::Memory[ Apuntador + i ] + E
    en caso contrario
      si LearnMatrix::x[i] = 0 y LearnMatrix::y[ Contador_de_patrones ] = 1
        LearnMatrix::Memory[Apuntador + i ] =
          LearnMatrix::Memory[ Apuntador + i ] - E;
      Contador_de_patrones = Contador_de_patrones + 1
}

```

**Nota:** el *Contador\_de\_patrones*, sirve para ubicar el renglón donde se hará la comparación que especifica la expresión 2.2 y el *Apuntador*, sirve para calcular la posición de memoria en la que se encuentra el elemento en cuestión.

**Fase de recuperación**

La  $i$ -ésima coordenada  $y_i^\omega$  del vector de clase  $y^\omega \in A^p$  se obtiene como lo indica la expresión 2.3, donde  $\bigvee$  es el operador *máximo*:

$$y_i^\omega = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} \cdot x_j^\omega = \bigvee_{h=1}^p \left[ \sum_{j=1}^n m_{hj} \cdot x_j^\omega \right] \\ 0 & \text{en otro caso} \end{cases}$$

La implementación de la expresión 2.3 es de la siguiente manera:

```

Get( x ){
  Se calcula  $\sum_{j=1}^n m_{ij} \cdot x_j^\omega$  :
    Se asigna 0 a todos los elementos de Learnmatrix::y
    LearnMatrix::x = x
    para j = 0 hasta numero_de_clases, incremento en 1, hacer:
      Apuntador = LearnMatrix::width * j
      para i = 0 hasta Learnmatrix::width, incremento en 1 hacer:
        Learnmatrix::y[ j ] = Learnmatrix::y[ j ] + ( Learnmatrix::Memory[
Apuntador + i ] * Learnmatrix::x[ i ] );
  Se calcula  $\bigvee_{h=1}^p \left[ \sum_{j=1}^n m_{hj} \cdot x_j^\omega \right]$  :
    para i = 0 hasta Learnmatrix::height, incremento de 1, hacer:
      vY[ i ] = LearnMatrix::y[ i ]
    para i = 0 hasta Learnmatrix::height, incremento de 1, hacer:
      si vY[ i ] > vY[ i + 1 ] entonces
        aux = vY[ i ]
        vY[ i ] = vY[ i + 1 ]
        vY[ i + 1 ] = aux

```

```

    Max = vY[ LearnMatrix::height - 1 ]
    para i = 0 hasta Learnmatrix::height, incremento de 1, hacer
        si Learnmatrix::y[ i ] = Max entonces
            Learnmatrix::y[ i ] = 1
        en caso contrario
            Learnmatrix::y[ i ] = 0
    devolver LearnMatrix::y
}

```

#### 4.4.2. Implementación del Linear Associator

##### Fase de aprendizaje

La fase de aprendizaje para el *Linear Associator* consiste en dos etapas:

**Primera etapa.** Para cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se encuentra la matriz  $y^\mu \cdot (x^\mu)^t$  de dimensión  $m \times n$  (expresiones 2.5 y 2.6):

**Segunda etapa.** Se suman las  $p$  matrices para obtener la memoria (expresión 2.7):

$$M = \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t = [m_{ij}]_{m \times n}$$

La implementación de la fase de aprendizaje es la siguiente:

##### Se crea la memoria

```

LinearAssociator( width, height ){
    LinearAssociator::width = width;
    LineraAssociator::height = height;
    Se reserva espacio en memoria RAM para M en función de widht y height
    Se pone en 0 todos los elementos de M
}

```

##### Learn( x, y ){

```

    LinearAssociator::x = x
    LinearAssociator::y = y
Se calcula    M =  $\sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t = [m_{ij}]_{m \times n}$ 
    para j = 0 hasta LinearAssociator::height, incremento de,1 hacer:
        para i = 0 hasta LinearAssociator::width, incremento de 1, hacer:
            LinearAssociator::Memory[ j * LinearAssociator::width + i ] =
            LinearAssociator::Memory[ j * LinearAssociator::width + i ] +
            ( LinearAssociator::y[ j ] * LinearAssociator::x[ i ] )
}

```

##### Fase de recuperación

La fase de recuperación consiste en presentarle a la memoria un patrón de entrada  $x^\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$  y realizar la siguiente operación (expresión 2.9):

$$M \cdot x^\omega = \left[ \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t \right] \cdot x^\omega$$

La implementación para la fase de recuperación es la siguiente:

```

Get( x ){
  LinearAssociator::x = x
  Se calcula  $\left[ \sum_{\mu=1}^p y^{\mu} \cdot (x^{\mu})^t \right] \cdot x^{\omega}$  :
  para i = 0 hasta LinearAssociator::height, incremento de 1, hacer
    LinearAssociator::y[ i ] = 0;
  para j = 0 hasta LinearAssociator::height, incremento de 1, hacer:
    para i = 0 hasta LinearAssociator::width, incremento de 1, hacer:
      LinearAssociator::y[ j ] = LinearAssociator::y[ j ] +
      LinearAssociator::Memory[ j * LinearAssociator::width + i ]*
      LinearAssociator::x[ i ]
    regresar LinearAssociator::y
}

```

#### 4.4.3. Implementación de la memoria autoasociativa Hopfield

La memoria Hopfield es autoasociativa, simétrica, con ceros en la diagonal principal. En virtud de que la memoria es autoasociativa, el conjunto fundamental para la memoria Hopfield es (expresión 2.14)  $\{(x^{\mu}, y^{\mu}) \mid \mu = 1, 2, \dots, p\}$  con:

##### Fase de aprendizaje

La fase de aprendizaje de Hopfield, es decir la creación de la memoria  $M$  se genera de la siguiente manera (expresión 2.15):

$$m_{ij} = \begin{cases} \sum_{\mu=1}^p x_i^{\mu} x_j^{\mu} & \text{si } i \neq j \\ 0 & \text{si } i = j \end{cases}$$

Operativamente la fase de aprendizaje se puede implementar en tres etapas:

**Primera etapa.** Para cada una de las  $p$  asociaciones  $(x^{\mu}, y^{\mu})$ , se encuentra la matriz  $x^{\mu} \cdot (x^{\mu})^t$  de una dimensión  $n \times n$ :

**Segunda etapa.** A cada una de las  $p$  matrices  $x^{\mu} \cdot (x^{\mu})^t$  se le resta la matriz identidad  $I$  de dimensiones  $n \times n$ , con la finalidad de hacer ceros la diagonal principal:

**Tercera etapa.** Se suman las  $p$  matrices  $x^{\mu} \cdot (x^{\mu})^t - I$  para que finalmente se obtenga la memoria asociativa Hopfield  $M$ :

$$M = \sum_{\mu=1}^p [x^{\mu} \cdot (x^{\mu})^t - \mathbf{I}] = [m_{ij}]_{n \times n}$$

La implementación de la fase de aprendizaje es de la siguiente manera:

##### Construcción de la memoria

**Hopfield**( *width*, *height* ){

*Hopfield::width* = *width*

*Hopfield::height* = *height*

Se crea el espacio en la memoria ram para la matriz de aprendizaje  $M$  en función al *width* y *height*

Se crea espacio en la memoria ram para  $x1$  ( servirá como  $x(t)$  )

Se pone en 0 los elementos de la matriz de aprendizaje  $M$

*Hopfield::Iteracion* = 0;

```

}
Learn( x, y ){
  Hopfield::x = x
  Hopfield::y = y
Se asegura que el conjunto fundamental sea  $A = \{-1, 1\}$ 
  para  $j = 0$  hasta Hopfield::height, incremento de 1, hacer:
    si Hopfield::x[ j ] > 0 entonces:
      Hopfield::x[ j ] = 1
    en caso contrario
      Hopfield::x[ j ] = -1
    Hopfield::y[ j ] = Hopfield::x[ j ]
Se calcula  $x^\mu \cdot (x^\mu)^t$  :
  para  $j = 0$  hasta Hopfield::height, incremento de 1, hacer:
    Apuntador = j * Hopfield::width
  para  $i = 0$  hasta Hopfield::width, incremento de 1, hacer:
    Hopfield::Memory[ Apuntador + i ] = Hopfield::Memory[ Apuntador + i ] +
      ( Hopfield::y[ j ] * Hopfield::x[ i ] )
Se calcula  $\sum_{\mu=1}^p [x^\mu \cdot (x^\mu)^t - I] = [m_{ij}]_{n \times n}$  :
  para  $j = 0$  hasta Hopfield::height, incremento de 1, hacer:
    Hopfield::Memory[ j * Hopfield::width + j ] -= 1
}

```

#### Fase de recuperación

La fase de recuperación para Hopfield consiste en tres etapas.

Dado un vector columna de entrada  $\tilde{x}$  :

**Primera etapa.** Para  $t = 0$ , se hace  $x(t) = \tilde{x}$ ; es decir,  $x_i(0) = \tilde{x}_i, \forall i \in \{1, 2, 3, \dots, n\}$

**Segunda etapa.**  $\forall i \in \{1, 2, 3, \dots, n\}$  se calcula  $x_i(t+1)$  en conformidad con la siguiente condición:

$$x_i(t+1) = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) > 0 \\ x_i(t) & \text{si } \sum_{j=1}^n m_{ij}x_j(t) = 0 \\ -1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) < 0 \end{cases}$$

**Tercera etapa.** Se compara  $x_i(t+1)$  con  $x_i(t) \forall i \in \{1, 2, 3, \dots, n\}$ . Si  $x(t+1) = x(t)$  el proceso termina y el vector recuperado es  $x(0) = \tilde{x}$ . De otro modo, el proceso continúa de la siguiente manera: los pasos 2 y 3 se iteran tantas veces como sea necesario hasta llegar a un valor  $t = \tau$  para el cual  $x_i(\tau+1) = x_i(\tau) \forall i \in \{1, 2, 3, \dots, n\}$ ; el proceso termina y el patrón recuperado es  $x(\tau)$ .

La implementación para estas tres etapas es de la siguiente manera:

```

Get( x ){
  Hopfield::x = x
Se asegura que  $x \in A$ , donde  $A = \{-1, 1\}$ 
  para  $j = 0$  hasta Hopfield::height, incremento de 1, hacer:
    si Hopfield::x[ j ] > 0 entonces:
      Hopfield::x[ j ] = 1
    en caso contrario

```

$Hopfield::x[j] = -1$

**Primera etapa:**  
 $Hopfield::x0 = Hopfield::x;$

**Segunda etapa:**  
 $Iterar = true;$   
 $Hopfield::Iteracion = 0;$   
 Mientras(  $Iterar$  ), hacer:  
   para  $j = 0$  hasta  $Hopfield::height$ , incremento de 1, hacer:  
     Apuntador =  $j * Hopfield::width$   
     para  $i = 0$  hasta  $Hopfield::width$ , incremento de 1, hacer:  
        $Hopfield::x1[j] = Hopfield::x1[j] +$   
         (  $Hopfield::Memory[ Apuntador + i ] *$   
 $Hopfield::x0[i]$  )

**Se calcula**  $x_i(t+1) = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) > 0 \\ x_i(t) & \text{si } \sum_{j=1}^n m_{ij}x_j(t) = 0 \\ -1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) < 0 \end{cases}$

  para  $j = 0$  hasta  $Hopfield::height$ , incremento de 1, hacer:  
     si  $Hopfield::x1[j] > 0$  entonces  
        $Hopfield::x1[j] = 1$   
     en caso contrario  
     si  $Hopfield::x1[j] < 0$  entonces  
        $Hopfield::x1[j] = -1$

**Tercera etapa:**  
 para  $j = 0$  hasta  $Hopfield::height$ , incremento de 1, hacer:  
   si  $Hopfield::x0[j] = Hopfield::x1[j]$  entonces:  
      $Iterar = falso$   
   en caso contrario  
      $Iterar = verdadero$   
   si  $Iterar = verdadero$ , entonces  
     para  $i = 0$  hasta  $Hopfield::height$ , incremento de 1, hacer:  
        $Hopfield::x0[i] = Hopfield::x1[i]$   
 fin del mientras(  $Iterar$  )  
 para  $j = 0$  hasta  $Hopfield::height$ , incremento de 1, hacer:  
   si  $Hopfield::x1[j] > 0$  entonces:  
      $Hopfield::x1[j] = 1$   
   en caso contrario  
      $Hopfield::x1[j] = 0$   
 regresar  $Hopfield::x1$   
 }

#### 4.4.4. Implementación de la memoria morfológica max

##### Fase de aprendizaje

La fase de aprendizaje para las memorias heteroasociativas y autoasociativas max consta de dos etapas, de acuerdo con las expresiones 2.33, 2.34 y 2.35:

**Primera etapa.** En cada una de las  $p$  asociaciones  $(x^\mu y^\mu)$  se usa el producto mínimo 2.32 para crear la matriz  $y^\mu \Delta (-x^\mu)^t$  de una dimensión  $m \times n$ , donde el negado transpuesto del patrón de entrada  $x^\mu$  se define como:  $(-x^\mu)^t = (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$ :

**Segunda etapa.** Se aplica el operador máximo  $\bigvee$  a las  $p$  matrices para obtener la memoria morfológica  $M$ .

$$m_{ij} = \bigvee_{\mu=1}^p (y_i^\mu - x_j^\mu)$$

*La implementación de la fase de aprendizaje es de la siguiente manera:*

**Se construye la memoria:**

**MorfologicaMax**( *width*, *height* ) {

*MorfologicaMax::width* = *width*;

*MorfologicaMax::height* = *height*;

*EsPriemVezQueAprende* = verdadero

*se crea el espacio en memoria RAM para M en función a width y height*

*se pone en 0 a todos los elementos de la matriz de aprendizaje M*

}

**Learn**( *x*, *y* ) {

*MorfologicaMax::x* = *x*

*MorfologicaMax::y* = *y*

*si EsPriemVezQueAprende* = verdadero entonces

*para j = 0 hasta MorfologicaMax::height, incremento de 1, hacer:*

*Apuntador* = *MorfologicaMax::width* \* *j*

*para i = 0 hasta MorfologicaMax::width, incremento de 1, hacer:*

**Se calcula**  $y^\mu \Delta (-x^\mu)^t$  :

*MorfologicaMax::Memory*[ *Apuntador* + *i* ] =

*MorfologicaMax::y*[ *j* ] - *MorfologicaMax::x*[ *i* ] ;

*EsPrimeraVezQueAprende* = falso

*en caso contrario*

*para j = 0 hasta MorfologicaMax::height, incremento de 1, hacer:*

*Apuntador* = *MorfologicaMax::width* \* *j*

*para i = 0 hasta MorfologicaMax::width, incremento de 1, hacer:*

**Se calcula**  $y^\mu \Delta (-x^\mu)^t$  :

*LastValue* = *MorfologicaMax::Memory*[ *Apuntador* + *i* ]

*MorfologicaMax::Memory*[ *Apuntador* + *i* ] =

*MorfologicaMax::y*[ *j* ] - (*MorfologicaMax::x*[ *i* ]

**Se calcula**  $\bigvee_{\mu=1}^p [y^\mu \Delta (-x^\mu)^t] = [m_{ij}]_{m \times n}$  :

*si LastValue* > *MorfologicaMax::Memory*[ *Apuntador* + *i* ] entonces

*MorfologicaMax::Memory*[ *Apuntador* + *i* ] = *LastValue*

*en caso contrario*

*LastValue* = *MorfologicaMax::Memory*[ *Apuntador* + *i* ]

}

### Fase de recuperación

La fase de recuperación consiste en realizar el producto mínimo  $\Delta$  de la memoria  $M$  con un patrón de entrada  $x^\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$ , para obtener un vector columna de dimension  $m$ , de acuerdo con las expresiones 2.36 y 2.37:

$$y_i = \bigwedge_{j=1}^n (m_{ij} + x_j^\omega)$$

La implementación para la expresión 2.37 es de la siguiente manera:

```

Get( x ) {
  MorfologicaMax::x = x
  para j = 0 hasta MorfologicaMax::height, incremento de 1, hacer:
    Apuntador = MorfologicaMax::width * j
    LastValue = MorfologicaMax::Memory[ Apuntador ] + MorfologicaMax::x[ 0
]
  para i = 0 hasta MorfologicaMax::width, incremento de 1, hacer:
Se calcula  $\bigwedge_{j=1}^n (m_{ij} + x_j^\omega)$  :
    MorfologicaMax::y[ j ] = MorfologicaMax::Memory[ Apuntador + i ] +
      MorfologicaMax::x[ i ]
    si MorfologicaMax::y[ j ] > LastValue entonces
      MorfologicaMax::y[ j ] = LastValue
    en caso contrario
      LastValue = MorfologicaMax::y[ j ]
    regresar MorfologicaMax::y
}

```

#### 4.4.5. Implementación de la memoria morfológica min

##### Fase de aprendizaje

Las memorias heteroasociativas y autoasociativas morfológicas min, en la fase de aprendizaje tiene dos etapas para generar la memoria  $M$ , de acuerdo con las expresiones 2.40, 2.41 y 2.42.

##### Primera fase

En cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se hace uso del operador máximo 2.31 para crear la matriz  $y^\mu \nabla (-x^\mu)$  de dimensiones  $m \times n$ , donde el negado del transpuesto del patrón de entrada  $x^\mu$  se define como:  $(-x^\mu)^t = (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$

##### Segunda fase

Se aplica el operador mínimo  $\bigwedge$  a las  $p$  matrices para obtener la memoria  $W$ .

$$w_{ij} = \bigwedge_{\mu=1}^p (y_i^\mu - x_j^\mu)$$

La implementación de la fase de aprendizaje es de la siguiente manera:

**Se construye la memoria de aprendizaje  $W$**

**MorfologicaMin**( *width*, *height* ) {

*MorfologicaMin::width* = *width*;

*MorfologicaMin::height* = *height*;

*EsPriemVezQueAprende* = verdadero

*se crea espacio en memoria RAM para  $W$  en función a *width* y *height**

*se pone en 0 a todos los elementos de la matriz de aprendizaje  $W$*

}

**Learn**( *x*, *y* ) {

*MorfologicaMin::x* = *x*

*MorfologicaMin::y* = *y*

*si EsPriemVezQueAprende* = verdadero entonces

*para j = 0 hasta MorfologicaMin::height, incremento de 1, hacer:*

*Apuntador = MorfologicaMin::width \* j*

*para i = 0 hasta MorfologicaMin::width, incremento de 1, hacer:*

**Se calcula**  $y^\mu \Delta (-x^\mu)^t$  :

*MorfologicaMin::Memory[ Apuntador + i ] =*

*MorfologicaMin::y[ j ] - MorfologicaMin::x[ i ];*

*EsPrimeraVezQueAprende* = falso

*en caso contrario*

*para j = 0 hasta MorfologicaMin::height, incremento de 1, hacer:*

*Apuntador = MorfologicaMin::width \* j*

*para i = 0 hasta MorfologicaMin::width, incremento de 1, hacer:*

**Se calcula**  $y^\mu \Delta (-x^\mu)^t$  :

*LastValue = MorfologicaMin::Memory[ Apuntador + i ]*

*MorfologicaMin::Memory[ Apuntador + i ] =*

*MorfologicaMin::y[ j ] - (MorfologicaMin::x[ i ]*

**Se calcula**  $\bigwedge_{\mu=1}^p [y^\mu \nabla (-x^\mu)^t] = [w_{ij}]_{m \times n}$  :

*si LastValue < MorfologicaMin::Memory[ Apuntador + i ] entonces*

*MorfologicaMin::Memory[ Apuntador + i ] = LastValue*

*en caso contrario*

*LastValue = MorfologicaMin::Memory[ Apuntador + i ]*

}

**Fase de recuperación**

La fase de recuperación consiste en realizar el producto máximo  $\nabla$  de la memoria  $W$  con un patrón de entrada  $x^\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$ , para obtener un vector columna  $y$  de dimensión  $m$ , de acuerdo con las expresiones 2.42 y 2.44:

$$y_i = \bigvee_{j=1}^n (w_{ij} + x_j^\omega)$$

La implementación de la fase de recuperación es de la siguiente manera:

**Get**( *x* ) {

*MorfologicaMin::x* = *x*

para  $j = 0$  hasta  $MorfologicaMin::height$ , incremento de 1, hacer:

$Apuntador = MorfologicaMin::width * j$

$LastValue = MorfologicaMin::Memory[ Apuntador ] + MorfologicaMin::x[ 0 ]$

para  $i = 0$  hasta  $MorfologicaMin::width$ , incremento de 1, hacer:

Se calcula  $\bigvee_{j=1}^n (w_{ij} + x_j^\omega)$  :

$MorfologicaMin::y[ j ] = MorfologicaMin::Memory[ Apuntador + i ] + MorfologicaMin::x[ i ]$

si  $MorfologicaMin::y[ j ] < LastValue$  entonces

$LastValue = MorfologicaMin::y[ j ]$

en caso contrario

$MorfologicaMin::y[ j ] = LastValue$

regresar  $MorfologicaMin::y$

}

#### 4.4.6. Implementación de la memoria asociativa $\alpha\beta$ max

Para conocer estas memorias asociativas se deben especificar los conjuntos  $A$  y  $B$ , que serán fundamentales para la fase de aprendizaje y de recuperación. Se definen de la siguiente manera:

$$A = \{0, 1\} \quad \text{y} \quad B = \{0, 1, 2\}$$

Además las memorias asociativas  $\alpha\beta$  hacen uso de dos operaciones binarias  $\alpha = A \times A \longrightarrow B$  y  $\beta = B \times A \longrightarrow A$ .

La operación binaria  $\alpha = A \times A \longrightarrow B$ , está definida por la siguiente tabla:

| $x$ | $y$ | $\alpha(x, y)$ |
|-----|-----|----------------|
| 0   | 0   | 1              |
| 0   | 1   | 0              |
| 1   | 0   | 2              |
| 1   | 1   | 1              |

La operación binaria  $\beta = B \times A \longrightarrow A$ , está definida de la siguiente manera:

| $x$ | $y$ | $\beta(x, y)$ |
|-----|-----|---------------|
| 0   | 0   | 0             |
| 0   | 1   | 0             |
| 1   | 0   | 0             |
| 1   | 1   | 1             |
| 2   | 0   | 1             |
| 2   | 1   | 1             |

La fase de aprendizaje consta de dos fases, de acuerdo con las expresiones 2.48, 2.49 y 2.50; en la primera se utiliza el operador  $\boxtimes$ , y en la segunda se echa mano del operador máximo  $\bigvee$ .

**Fase de aprendizaje**

**Fase 1** Para cada  $\mu = 1, 2, \dots, p$ , a partir de la pareja  $(x^\mu, y^\mu)$  se construye la matriz

$$[y^\mu \boxtimes (x^\mu)^t]_{m \times n}$$

**Fase 2** Se aplica el operador binario máximo  $\bigvee$  a las matrices obtenidas en la fase 1.

$$v_{ij} = \bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu)$$

La implementación de la fase de aprendizaje es de la siguiente manera:

**Construcción de la memoria  $\bigvee$** 

**AlfaBetaMax**( width, height ){

    AlfaBetaMax::width = width

    AlfaBetaMax::height = height

    EsPrimVezQueAprende = verdadero

*se crea el espacio en la memoria ram para la matriz de aprendizaje M en función a width y height*

*se pone en 0 a todos los elementos de la matriz de aprendizaje M*

}

**Se construyen las funciones Alfa y Beta, para las memorias de tipo  $\bigvee$  y**

**$\bigwedge$**

**ALFA**( x, y ){

    xAlfa = x

    yAlfa = y

    AlfaResult = ALFA\_ERROR;

*si xAlfa = 0 y yAlfa = 0 entonces*

        AlfaResult = 1

*en caso contrario*

*si xAlfa = 0 y yAlfa > 0 entonces*

            AlfaResult = 0

*en caso contrario*

*si xAlfa > 0 y yAlfa = 0 entonces*

            AlfaResult = 2

*en caso contrario*

*si xAlfa > 0 y yAlfa > 0 entonces*

            AlfaResult = 1

    regresa AlfaResult

}

**BETA**( x, y ){

    xBeta = x

    yBeta = y

    BetaResult = BETA\_ERROR;

*si xBeta = 0 y yBeta = 0 entonces*

```

        BetaResult = 0;
    en caso contrario
        si xBeta = 0 y yBeta > 0 entonces
            BetaResult = 0
        en caso contrario
            si xBeta = 1 y yBeta = 0 entonces
                BetaResult = 0
            en caso contrario
                si xBeta = 1 y yBeta > 0 entonces
                    BetaResult = 1
                en caso contrario
                    si xBeta = 2 y yBeta == 0 entonces
                        BetaResult = 1
                    en caso contrario
                        si xBeta = 2 y yBeta > 0 entonces
                            BetaResult = 1

    regresa BetaResult
}
Learn( x, y ){
    AlfaBetaMax::x = x
    AlfaBetaMax::y = y
    si EsPrimeraVezQueAprende = verdadero entonces
        para j = 0 hasta AlfaBetaMax::height, incremento de 1, hacer:
            Apuntador = AlfaBetaMax::width * j
            para i = 0 hasta AlfaBetaMax::width, incremento de 1, hacer:
Se calcula  $[y^\mu \boxtimes (x^\mu)^t]_{m \times n}$  :
                AlfaBetaMax::Memory[ Apuntador + i ] =
                    ALFA( AlfaBetaMax::y[ j ], AlfaBetaMax::x[
i ] )
                EsPrimeraVezQueAprende = falso
            en caso contrario
                para j = 0 hasta AlfaBetaMax::height, incremento de 1, hacer:
                    Apuntador = AlfaBetaMax::width * j
                    para i = 0 hasta AlfaBetaMax::width, incremento de 1, hacer:
Se calcula  $[y^\mu \boxtimes (x^\mu)^t]_{m \times n}$  :
                        LastValue = AlfaBetaMax::Memory[ Apuntador + i ]
                        AlfaBetaMax::Memory[ Apuntador + i ] =
                            ALFA( AlfaBetaMax::y[ j ], AlfaBetaMax::x[ i ] )
Se calcula  $\mathbf{V} = \bigvee_{\mu=1}^p [y^\mu \boxtimes (x^\mu)^t]$  :
                            si AlfaBetaMax::Memory[ Apuntador + i ] > LastValue entonces
                                LastValue = AlfaBetaMax::Memory[ Apuntador + i ]
                            en caso contrario
                                AlfaBetaMax::Memory[ Apuntador + i ] = LastValue
    }

```

**Fase de recuperación**

De acuerdo con las expresiones 2.51, 2.52 y 2.53, se presenta un patrón  $x^\omega$ , con  $\omega \in \{1, 2, \dots, p\}$ , a la memoria heteroasociativa  $\alpha\beta$  tipo  $\bigvee$  y se realiza la operación  $\mathbb{M}_\beta$ :

$$(\mathbf{V}\mathbb{M}_\beta x^\omega)_i = \bigwedge_{j=1}^n \beta(v_{ij}, x_j^\omega)$$

La fase de recuperación se implementa de la siguiente manera:

```

Get( x ) {
  AlfaBetaMax::x = x
  para j = 0 hasta AlfaBetaMax::height, incremento de 1, hacer:
    Apuntador = AlfaBetaMax::width * j
    LastValue = BETA( AlfaBetaMax::Memory[ Apuntador ], AlfaBetaMax::x[
0 ] )
    para i = 0 hasta AlfaBetaMax::width, incremento de 1, hacer:
      Se calcula  $\bigwedge_{j=1}^n \beta(v_{ij}, x_j^\omega)$  :
        AlfaBetaMax::y[ j ] = BETA( AlfaBetaMax::Memory[ Apuntador +
i ],
                                AlfaBetaMax::x[ i ] )
        si AlfaBetaMax::y[ j ] < LastValue entonces
          LastValue = AlfaBetaMax::y[ j ]
        en caso contrario
          AlfaBetaMax::y[ j ] = LastValue
      regresar AlfaBetaMax::y
    }
}

```

**4.4.7. Implementación de la memoria asociativa  $\alpha\beta$  Min**

De acuerdo con las expresiones 2.56, 2.57 y 2.58, las memorias asociativas  $\alpha\beta$  de tipo *min*, para realizar la fase de aprendizaje deben seguir dos etapas; en la primera se debe utilizar el operador  $\boxtimes$ , y en la segunda se hace uso del operador  $\bigwedge$ .

**Fase 1.** Para cada  $\mu = 1, 2, \dots, p$ , a partir de la pareja  $(x^\mu, y^\mu)$  se construye la matriz

$$[y^\mu \boxtimes (x^\mu)^t]_{m \times n}$$

**Fase 2.** Se aplica el operador binario mínimo  $\bigwedge$  a las matrices obtenidas en la fase 1

$$\lambda_{ij} = \bigwedge_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu)$$

La implementación de la fase de aprendizaje es de la siguiente manera:

**Construcción de la memoria  $\bigwedge$** 

```

AlfaBetaMin( width, height ) {
  AlfaBetaMin::width = width
  AlfaBetaMin::height = height
  EsPriemVezQueAprende = verdadero
  se crea espacio en memoria RAM para M en función a width y height
}

```

se pone en 0 a todos los elementos de la matriz de aprendizaje  $M$

}  
**Learn**(  $x, y$  ){  
 AlfaBetaMin:: $x = x$   
 AlfaBetaMin:: $y = y$   
 si EsPrimeraVezQueAprende = verdadero entonces  
 para  $j = 0$  hasta AlfaBetaMin:: $height$ , incremento de 1, hacer:  
 Apuntador = AlfaBetaMin:: $width * j$   
 para  $i = 0$  hasta AlfaBetaMin:: $width$ , incremento de 1, hacer:  
**Se calcula**  $[y^\mu \boxtimes (x^\mu)^t]_{m \times n}$  :  
 AlfaBetaMin:: $Memory[ Apuntador + i ] =$   
 ALFA( AlfaBetaMin:: $y[ j ], AlfaBetaMin:: $x[ i$$

) )  
 EsPrimeraVezQueAprende = falso  
 en caso contrario  
 para  $j = 0$  hasta AlfaBetaMin:: $height$ , incremento de 1, hacer:  
 Apuntador = AlfaBetaMin:: $width * j$   
 para  $i = 0$  hasta AlfaBetaMin:: $width$ , incremento de 1, hacer:  
**Se calcula**  $[y^\mu \boxtimes (x^\mu)^t]_{m \times n}$  :  
 LastValue = AlfaBetaMin:: $Memory[ Apuntador + i ]$   
 AlfaBetaMin:: $Memory[ Apuntador + i ] =$   
 ALFA( AlfaBetaMin:: $y[ j ], AlfaBetaMin:: $x[ i ]$$

**Se calcula**  $\bigwedge_{\mu=1}^p [y^\mu \boxtimes (x^\mu)^t]$  :  
 si AlfaBetaMin:: $Memory[ Apuntador + i ] > LastValue$  entonces  
 AlfaBetaMin:: $Memory[ Apuntador + i ] = LastValue$

}

#### Fase de recuperación

De acuerdo con las expresiones 2.59, 2.60 y 2.61, en la fase de recuperación se debe presentar un patrón  $x^\omega$ , con  $\omega \in \{1, 2, \dots, p\}$ , a la memoria asociativa  $\alpha\beta$  de tipo  $\bigwedge$  y se debe realizar la operación  $\boxtimes_\beta$  :

$$(\Lambda \boxtimes_\beta x^\omega)_i = \bigvee_{j=1}^n \beta(\lambda_{ij}, x^\omega)$$

La fase de recuperación se implementa de la siguiente manera:

**Get**(  $x$  ){  
 AlfaBetaMin:: $x = x$   
 para  $j = 0$  hasta AlfaBetaMin:: $height$ , incremento de 1, hacer:  
 Apuntador = AlfaBetaMin:: $width * j$   
 LastValue = 0  
 para  $i = 0$  hasta AlfaBetaMin:: $width$ , incremento de 1, hacer:  
**Se calcula**  $\bigvee_{j=1}^n \beta(\lambda_{ij}, x^\omega)$  :  
 AlfaBetaMin:: $y[ j ] = BETA( AlfaBetaMin:: $Memory[ Apuntador + i ],$$

```

        AlfaBetaMin::x[ i ] )
    si AlfaBetaMin::y[ j ] < LastValue entonces
        AlfaBetaMin::y[ j ] = LastValue
    en caso contrario
        LastValue = AlfaBetaMin::y[ j ]
    regresar AlfaBetaMin::y
}

```

#### 4.4.8. Implementación de las memorias asociativas heurísticas

Hasta este momento se ha visto la implementación de todos los modelos de memorias asociativas descritas en el capítulo 2. En este trabajo se ha hablado de métodos heurísticos que no se han definido plenamente. El propósito de esta sección es esa, la de definir los métodos heurísticos o híbridos para las memorias asociativas morfológicas y las memorias asociativas  $\alpha\beta$ . ¿Por qué solamente de estas memorias?: porque éstas son excelentes para la recuperación de patrones con un alto contenido de ruido aditivo o sustractivo, pero en cuanto al ruido mezclado su desempeño baja notablemente; pero, como se especificó en la sección 4.3 de este capítulo, si un patrón alterado con ruido mezclado, que en este caso es una imagen, se dilata, el patrón resultante es una imagen con ruido aditivo: es obvio pensar que la memoria morfológica max y la memoria  $\alpha\beta$  max tienen posibilidades de recuperar este patrón de manera perfecta. Si el mismo patrón es erosionado, el patrón resultante es una imagen con ruido sustractivo y, en este caso, la memoria morfológica min y la memoria  $\alpha\beta$  min, tienen posibilidades de recuperar de manera perfecta este patrón. La figura 4.7 ilustra este hecho.

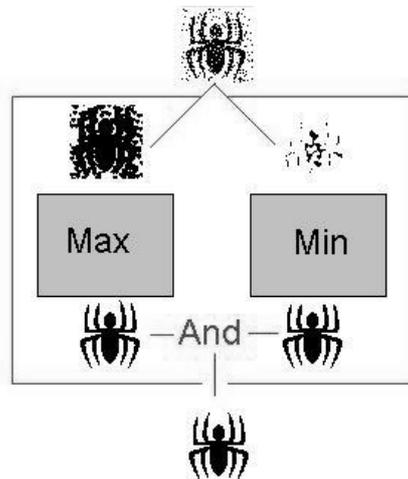


figura 4.7. Diagrama que muestra el proceso de las memorias heurísticas.

#### Morfológica Max

*Se crea una memoria MorfológicaMax  
se convierte en un vector la imagen sin ruido que fungirá como patrón de entrada  
se convierte en un vector la imagen sin ruido que fungirá como patrón de salida*

*se invoca la función Learn de la memoria MorfológicaMax y se pasan como parámetros los dos vectores convertidos*  
*se altera con ruido mezclado el patrón que se desea recuperar*  
*se dilata el patrón alterado con ruido mezclado*  
*el patrón resultante de la dilatación se convierte a un vector*  
*se invoca la función Get de la memoria MorfológicaMax pasando como parámetro el vector a recuperar*  
*el vector devuelto se convierte a una imagen.*

### **Morfológica Min**

*Se crea una memoria MorfológicaMin*  
*se convierte en un vector la imagen sin ruido que fungirá como patrón de entrada*  
*se convierte en un vector la imagen sin ruido que fungirá como patrón de salida*  
*se invoca la función Learn de la memoria MorfológicaMin y se pasan como parámetros los dos vectores convertidos*  
*se altera con ruido mezclado el patrón que se desea recuperar*  
*se erosiona el patrón alterado con ruido mezclado*  
*el patrón resultante de la erosión se convierte a un vector*  
*se invoca la función Get de la memoria MorfológicaMin pasando como parámetro el vector a recuperar*  
*el vector devuelto se convierte a una imagen.*  
*Se realiza un AND lógico entre el patrón que recuperó la memoria morfológica max y el patrón que recuperó la memoria morfológica min, el resultado de este AND es el patrón recuperado por la memoria morfológica heurística.*

### **$\alpha\beta$ Max**

*Se crea una memoria AlfaBetaMax*  
*se convierte en un vector la imagen sin ruido que fungirá como patrón de entrada*  
*se convierte en un vector la imagen sin ruido que fungirá como patrón de salida*  
*se invoca la función Learn de la memoria AlfaBetaMax y se pasan como parámetros los dos vectores convertidos*  
*se altera con ruido mezclado el patrón que se desea recuperar*  
*se dilata el patrón alterado con ruido mezclado*  
*el patrón resultante de la dilatación se convierte a un vector*  
*se invoca la función Get de la memoria AlfaBetaMax pasando como parámetro el vector a recuperar*  
*el vector devuelto se convierte a una imagen*

### **$\alpha\beta$ Min**

*Se crea una memoria AlfaBetaMin*  
*se convierte en un vector la imagen sin ruido que fungirá como patrón de entrada*  
*se convierte en un vector la imagen sin ruido que fungirá como patrón de salida*  
*se invoca la función Learn de la memoria AlfaBetaMin y se pasan como parámetros los dos vectores convertidos*

*se altera con ruido mezclado el patrón que se desea recuperar  
 se erosiona el patrón alterado con ruido mezclado  
 el patrón resultante de la erosión se convierte a un vector  
 se invoca la función Get de la memoria AlfaBetaMin pasando como parámetro el  
 vector a recuperar  
 el vector devuelto se convierte a una imagen.  
 Se realiza un AND lógico entre el patrón que recuperó la memoria  $\alpha\beta$  max y el patrón que recuperó la memoria  $\alpha\beta$  min, el resultado de este AND es el patrón recuperado por la memoria  $\alpha\beta$  heurística.*

#### 4.5. Ejecución automática de las memorias asociativas

En esta sección se detalla la manera en que se implementó la ejecución automática de las memorias asociativas. La ejecución puede ser sin ruido, con ruido aditivo, con ruido sustractivo o con ruido mezclado.

La idea general del funcionamiento de la ejecución automática, es ésta: primero se aprenderá un patrón y se intentará la recuperación de éste, después se aprenderá un segundo patrón y se intentará recuperar los dos patrones aprendidos; así sucesivamente, se repetirá hasta que se hayan aprendido todos los patrones y se haya tratado de recuperar la totalidad de estos.

Antes de hablar de la implementación de la ejecución automática, es necesario indicar qué hay detrás del funcionamiento del software; es decir, toda la estructura interna que no se ve, pero que es necesaria para generar los resultados, los cuales servirán de base para realizar el estudio estadístico comparativo; lo que está detrás es una estructura de directorios que es el lugar donde se almacenarán los patrones de salida físicamente.

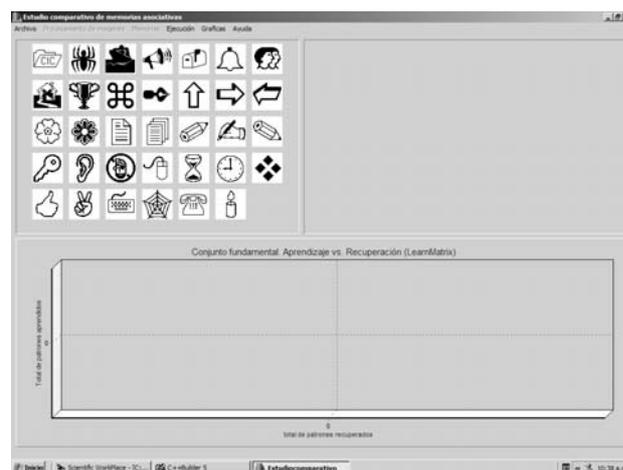


figura 4.8. Interfase gráfica del estudio comparativo.

Tomando como base el párrafo dos de esta sección, se creará un conjunto de directorios con el nombre de cada modelo de memoria asociativa, con la finalidad de tener

un control de lo que hace cada memoria. Dentro de los directorios con los nombres de las memorias asociativas se crean un conjunto de 50 subdirectorios, que tienen por nombre los números del 1 al 50, esto se hace con la finalidad de guardar un historial de las recuperaciones que se hicieron en función a los patrones que se aprendieron; es decir, el directorio 1 de la memoria asociativa en cuestión, guardará una sola imagen, ya que según el párrafo dos, si una es la memoria que se ha aprendido, ésta misma es la que se intenta recuperar y por lo tanto se almacenará una imagen en este subdirectorio; en el caso del subdirectorio con el nombre 2, tendrá almacenadas dos imágenes, así sucesivamente hasta que se hayan aprendido todos los patrones. Por último, existe un directorio que se llama "aditivo" dentro de la ruta "c:\\Hopfield", este directorio guarda las imágenes que hayan sido o no afectadas con ruido antes de comenzar la fase de aprendizaje.

El por qué solamente 50 subdirectorios, no es porque los algoritmos de las memorias asociativas no soporten más, no es esta la razón, la razón se relaciona con el diseño de la interfase visual (ver figura 4.8).

#### Implementación de la ejecución automática

1. Se deben especificar las rutas de directorios donde se encuentran las imágenes (conjunto fundamental).

2. Se debe elegir el tipo de ejecución: sin ruido, con ruido aditivo, con ruido sustractivo o con ruido mezclado. Si se elige la ejecución con algún tipo de ruido se manda a llamar el correspondiente algoritmo. Los algoritmos usados se definieron en la sección anterior.

3. Se copian de manera automática las imágenes sean o no afectadas o por algún tipo de ruido, a la ruta c:\\Hopfield\\aditivo.

4. Se ejecuta el algoritmo que a continuación se muestra:

Nota: En el siguiente algoritmo se comentarán cosas genéricas, por ejemplo, "MemoriaEnCuestión", esto significa que se puede usar cualquier modelo de memoria asociativa que se desee, Otro ejemplo es NombreDeMemoriaEnCuestion + "\\ + i +\\" + NombreDeImagenEncontradaEnLaPosicion[j], esto indica que se está haciendo referencia al directorio de la memoria en cuestión, a este directorio se le está adicionando el carácter "\\" que indica subdirectorio, *i*, indica el número de subdirectorio y finalmente: NombreDeImagenEncontradaEnLaPosicion[j], es el nombre del archivo que se encuentra ordenado en la posición *j*.

#### \*Creación de memoria:

*EsComienzoDeMemoria = verdadero*

*Path = RutaDeLocalizacionDelConjuntoDeImagenesCargadaPrimero*

*Path1 = RutaDeLocalizacionDelConjuntoDeImagenesCargadaSegundo*

*Path2 = RutaDeLocalizacionDeLasImagenesAfectadasConRuido*

*para i = 1 hasta NumeroDePatronesEnConjuntoFundamenta, incremento de 1, hacer*

*Archivo = Path + "\\ + NombreDeImagenEncontradaEnLaPosicion[i]*

```

    Bmp = CargarImagen( Archivo )
    Imagen = objeto BMP( Bmp )
    si EsComienzoDeMemoria) entonces
        tam = Imagen.Ancho * Imagen.Alto
        memoria = objeto MemoriaEnCuestion( tam, tam )
        EsComienzoDeMemoria = falso
    fin si
*Proceso de aprendizaje:
    Imagen.ImagenAVector( Vector );
*Si la memoria es heteroasociativa
    si EsMemoriaHeteroAsociativa = verdadero
        Archivo4 = Path1 + "\ " + NombreDeImagenEncontradaEn-
LaPosicion[i]
        bmp = CargarImagen( Archivo4 )
        Imagen2 = objeto BMP( bmp )
        Imagen2->ImagenAVector( Vector2 );
        memoria.Learn( Vector, Vector2 );
    en caso contrario
        memoria.Learn( Vector, Vector );
    fin del si
*Proceso de recuperación:
    PatronesRecuperados = 0;
    para j=1 hasta i, incremento de 1, hacer:
        Archivo = Path2 + NombreDeImagenEncontradaEnLaPosicion[j]
*Si la memoria es heteroasociativa
    si EsMemoriaHeteroAsociativa = verdadero entonces
        Archivo2 = Path1 + NombreDeImagenEncontradaEnLaPosicion[j]
    en caso contrario
        Archivo2 = Path + NombreDeImagenEncontradaEnLaPosicion[j]
    fin si
        Archivo3 = NombreDeMemoriaEnCuestion + "\ " + i + "\ " +
NombreDeImagenEncontradaEnLaPosicion[j]
        Bmp = CargarImagen( Archivo )
        Im = objeto BMP( Bmp );
        bmp = CargarImagen( Archivo2 )
        Im2 = objeto BMP( bmp )
        Im.ImagenAVector( v )
        Im2.ImagenAVector( v2 )
        ve = memoria->Get( v )
        Im.VectorAImage( ve )
        DibujarImagen( columna, renglon, Bmp )
        Bmp.SalvarEnArchivo( Archivo3 )
    si v2 = ve entonces
        PatronesRecuperados = PatronesRecuperados + 1
    fin del si

```

```

    fin del para j
    se grafican PatronesRecuperados
  fin del para i

```

**Observación final:** El carácter “#” indica que se hará uso de los algoritmos de dilatación o erosión en los modelos heurísticos. Y el algoritmo que se sustituye en esa parte es:

```

  si EsHeuristica entonces
    Archivo3 = MemoriaHeuristica + "\ " + i + "\ " + NombreDeImagenEncontradaEnLaPosicion[ j ]
    EE = CargarImagen( ElementosEstructurante )
    Elemento = objeto BMP( EE );
    para x = 0; hasta 7, incremento de 1 hacer:
      Im.Erosion( Elemento ) o Im.Dilatacion( Elemento ), dependiendo
  si es memoria min o max
    Im.ImagenA( v )
  fin del si

```

## 4.6. Ejecución por patrón

En esta parte se detalla la implementación de la ejecución por patrón, en la cual se intenta recuperar un patrón  $n$  veces. Este proceso tiene la finalidad de mostrar qué tan efectiva es la memoria para recuperar un patrón ante cualquier tipo de ruido; debido a que el ruido es generado aleatoriamente, puede afectar el desempeño de la memoria asociativa, de tal forma que en cierto momento el patrón en cuestión no sea recuperado, pero en una siguiente ocasión este mismo patrón con el mismo porcentaje de ruido sí logre ser recuperado de manera perfecta.

La implementación es la siguiente:

```

  porcen = PORCENTAJE_DE_RUIDO
  memoria = objeto de memoria en cuestion
  para i = 1 hasta NumeroDePatronesEnConjuntoFundamenta, incremento de 1,
  hacer
    Archivo = Path + "\ " + NombreDeImagenEncontradaEnLaPosicion[i]
    Bmp = CargarImagen( Archivo )
    DibujarImagen( Bmp )
    Imagen = objeto BMP( Bmp )
    Imagen.ImagenAVector( Vector )
    Imagen.ImagenAVector( v )
    memoria.learn( Vector, Vector )
  si = 0
  no = 0
  para j = 0 hasta ITERAR, incremento de 1, hacer:
    Imagen.VectorAImagen( v )
    en caso TIPO_DE_RUIDO
      caso ADITIVO:
        Imagen.RuidoAditivo( porcen )

```

```

    caso SUSTRACTIVO:
        Imagen.RuidoSustractivo( porcen )
    caso MEZCLADO:
        Imagen.RuidoMezclado( porcen )
    fin en caso
    DibujarImagen( Bmp )
    Imagen.ImagenAVector( Vector )
    vec = memoria.Get( Vector )
    Imagen.VectorAImagen( vec )
    DibujarImagen( Bmp )
    si vec = v entonces
        si = si + 1
    en caso contrario
        no = no + 1
    fin para j
    Graficar si y no
    fin para i

```

#### 4.7. Generación de las gráficas en el software

Las gráficas que el software muestra fueron hechas con el componente Activex "Chart" de C++ Builder. Se tomó por convención el tipo de gráficas de barras para representar los datos arrojados por las memorias asociativas. Las diferentes gráficas que muestra el software son: Patrones aprendidos vs. patrones recuperados, tiempos de recuperación, iteraciones de la memoria Hopfield para converger, gráfica comparativa de todas las memorias (patrones aprendidos vs patrones recuperados), graficas de repetición por patrón.

En la implementación de las ejecuciones automáticas no se especificó cómo se realiza el proceso de graficación, sólo fue llamada como "graficar". La sintaxis que C++ Builder usa para generar gráficas es la siguiente:

```

    Form1->ABHeurisChart->Series[0]->AddXY( i , PatronesRecuperados, "", clTee-
Color);

```

donde:

**Form1** indica la forma(ventana) activa donde se encuentra ubicada la gráfica

**AbHeuristiChart** indica el nombre de la gráfica

**Series** indica las series que se graficarán, el índice 0 indica que es la primera serie de  $n$  series que se dibujarán en esa gráfica.

**AddXY( i , PatronesRecuperados, "", clTeeColor)** indica que se agregará una barra cuyas coordenadas son  $(1, Patronesrecuperados)$ , las "" indican que se agregará el valor que le corresponde al eje de las "Y's" y **clTeeColor**, indica que toma un color asignado automáticamente por el control Chart, este color es distinto para cada barra.

Además, es necesario configurar el componente Chart. La figura 4.9 muestra la interfase del componente Chart.

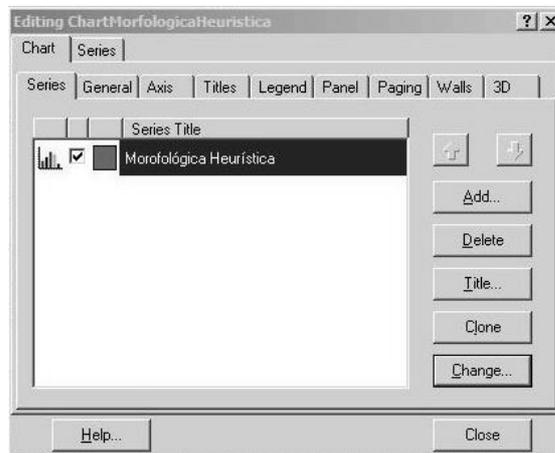


figura 4.9. Interfase del componente Chart.

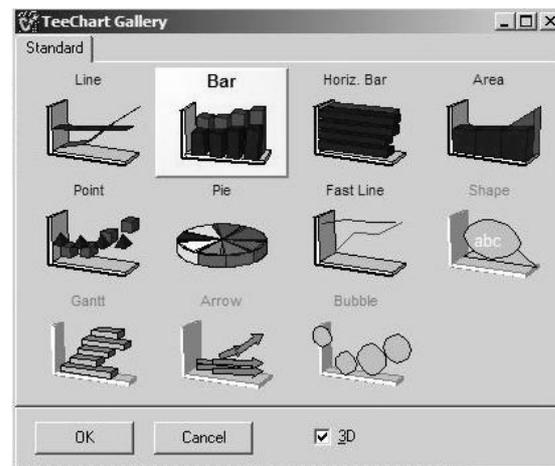


figura 4.10. Interfase que muestra el tipo de gráficas que pueden usarse con el componente Chart.

Para invocar la interfase del componente Chart (figura 4.9), debe pulsarse 2 veces el botón izquierdo del ratón sobre el control Activex Chart en la fase de diseño. Para agregar una gráfica debe pulsarse una vez, con el botón izquierdo del ratón, el botón Add (figura 4.9). Una vez presionado este botón aparecerá la figura 4.10. Ahora se podrá definir el tipo de gráfica a usarse; basta con presionar una vez el botón izquierdo del ratón sobre la gráfica deseada y posteriormente pulsar el botón OK con el botón izquierdo del ratón. Prácticamente eso será todo para crear una gráfica con C++ Builder. Si se observa en la parte inferior de la figura 4.10 hay un cuadro palomeado con la leyenda 3D. Si está seleccionado el recuadro, la apariencia de las gráficas será con profundidad; si no lo está, la apariencia de éstas será plana.

## Capítulo 5

# Disquisiciones experimentales

En este capítulo quedará plasmado el estudio comparativo experimental, como se ilustra en el esquema de la página siguiente. La importancia de este capítulo es tal, que las memorias asociativas mostrarán claramente su comportamiento ante conjuntos fundamentales específicos y ante patrones de entrada alterados con diferentes tipos de ruido, estando sometidas a las mismas condiciones; se derivarán puntos muy interesantes y como resultado se generará un ambiente que será muy ilustrativo para dejar sin lugar a dudas las ventajas y desventajas de cada modelo de memoria asociativa.

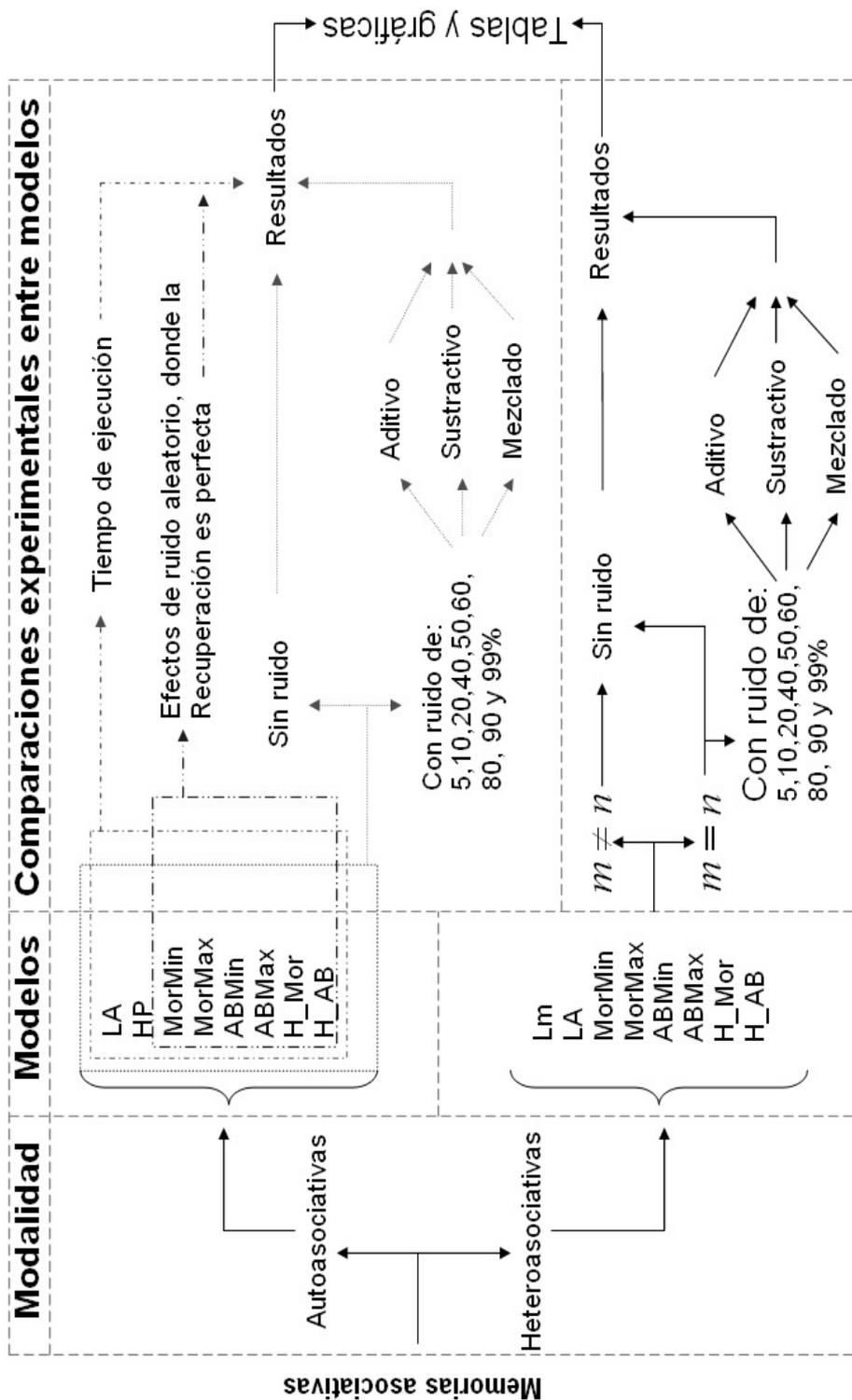
Los aspectos de interés que se tomaron en cuenta para realizar este comparativo son los mismos que se han registrado en la literatura sobre memorias asociativas a través de los años: capacidad de aprendizaje, condiciones suficientes para recuperar de manera perfecta el conjunto fundamental, cantidades y tipo de ruido que soportan, tipos y cantidad de ruido a que son sensibles.

El conjunto de patrones que se creó para formar los conjuntos fundamentales, consta de un total de 139 imágenes de 50 pixeles de ancho por 50 pixeles de alto: 26 letras mayúsculas, 26 letras minúsculas del alfabeto usual, 26 mayúsculas y 26 minúsculas del alfabeto griego, los números del uno al nueve y las restantes son una miscelánea de imágenes. Todas las imágenes están en formato BMP binario. Un pixel negro tiene el valor lógico 0 y un pixel blanco tiene el valor lógico 1.

A partir de este momento se comenzará a observar el desempeño de las memorias asociativas en condiciones que probarán su efectividad y mostrarán sus limitaciones, y es muy importante destacar la cantidad de pruebas en las que serán sometidas las memorias asociativas; por ejemplo, son 26 patrones por conjunto fundamental, a excepción del conjunto fundamental 5 que consta tan sólo de 9 patrones, lo cual significa que se está hablando de un total de 139 patrones diferentes que cada memoria asociativa aprenderá e intentará recuperar.

Además, cada proceso al que estará sometida la memoria en cuestión es: aprender el primer patrón y tratar de recuperarlo, aprender el segundo patrón e intentar recuperar los dos aprendidos, proceso que se realizará hasta haber aprendido los 26 patrones e intentar recuperar estos mismos; al finalizar todo este proceso de aprendizaje y recuperación, se tienen un total de 351 intentos de recuperación de patrones.

### Esquema experimental



Ahora bien, los experimentos se harán sin ruido, con ruido aditivo, sustractivo y mezclado. Los porcentajes con los que se harán los experimentos son: 5, 10, 20, 40, 50, 60, 80, 90 y 99%; por lo tanto, se generan 10 procesos diferentes que se harán para las memorias heteroasociativas y 10 procesos para las memorias autoasociativas, dando un total de 7020 intentos de recuperación por modelo de memoria, sin contar los casos especiales y el caso en que la memoria heteroasociativa sea  $m \neq n$ .

Por cuestiones de espacio, en este capítulo no se mostrará toda la información generada. Se mostrarán los resultados por el total de conjuntos fundamentales en los modelos de memorias asociativas considerados en esta tesis, con base a la recuperación de patrones y porcentaje efectividad en la recuperación, salvo en casos donde se indique lo contrario. Si se desea ver todas las tablas y gráficas que se obtuvieron como consecuencia de este estudio comparativo, por favor diríjase al CD anexo a este trabajo en la ruta \Resultados.

El software fue desarrollado en C++ Builder 5.0 de Borland. La ejecución de este programa se llevó a cabo en una computadora Compaq presario 2100, con 256 MB en RAM, disco duro de 40 GB y un procesador AMD Athlon a 853 MH, bajo el sistema operativo Windows XP.

Este capítulo consta de tres secciones. En la primera se analizan los modelos de memorias asociativas descritas en el capítulo 2 de este trabajo de tesis para el modo heteroasociativo; en la segunda, se analizan los mismos modelos de memorias asociativas pero para el modo autoasociativo, y en la tercera se analizan casos especiales de las memorias asociativas. Todas las secciones consistirán en aprender el mismo conjunto fundamental y tratar de recuperarlo en las siguientes circunstancias: sin ruido, con ruido aditivo, ruido sustractivo y ruido mezclado.

**NOTA:** Es digno de destacar que se considerará como *recuperación perfecta* si el patrón recuperado es idéntico al patrón aprendido, es decir que con un pixel de diferencia se considera que la recuperación no es perfecta.

### 5.1. Estudio comparativo experimental para los modelos de memorias heteroasociativas

Una memoria es heteroasociativa si sucede que:  $\exists \mu \in \{1, 2, \dots, p\}$  para el que se cumple que  $x^\mu \neq y^\mu$

Habrán dos maneras diferentes para realizar el estudio comparativo de las memorias heteroasociativas: la primera será de patrones donde  $n = m$ , y la segunda donde  $n \neq m$ . Además, este estudio experimental comparativo hará uso de 5 conjuntos fundamentales diferentes.

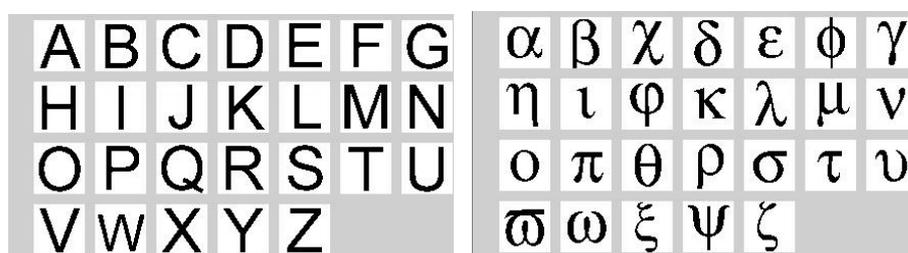
Para cada uno de los cuatro primeros conjuntos fundamentales  $(x^\omega, y^\omega) \mid \omega \in \{1, 2, \dots, p\}$ ,  $p = 26$ . Para el conjunto fundamental 5,  $p = 9$ .

### 5.1.1. Resultados obtenidos con las memorias heteroasociativas cuando $m = n$

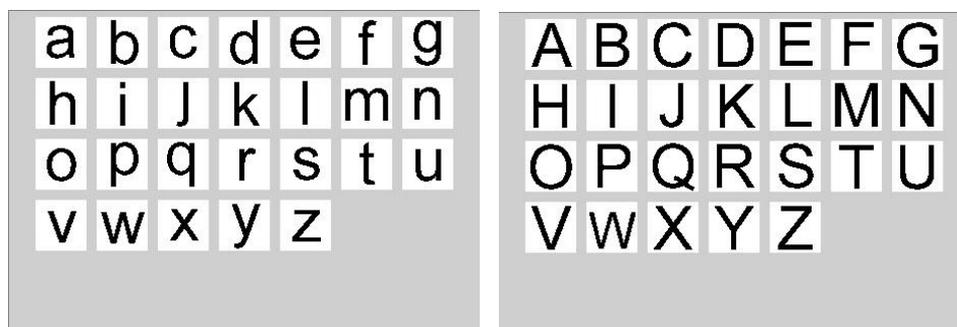
Los cinco conjuntos fundamentales se definen como sigue:

Nota: Las imágenes que se encuentran ubicadas en el lado izquierdo representan los patrones  $x^\mu$  de entrada y las imágenes de lado derecho representan los patrones  $y^\mu$  de salida.

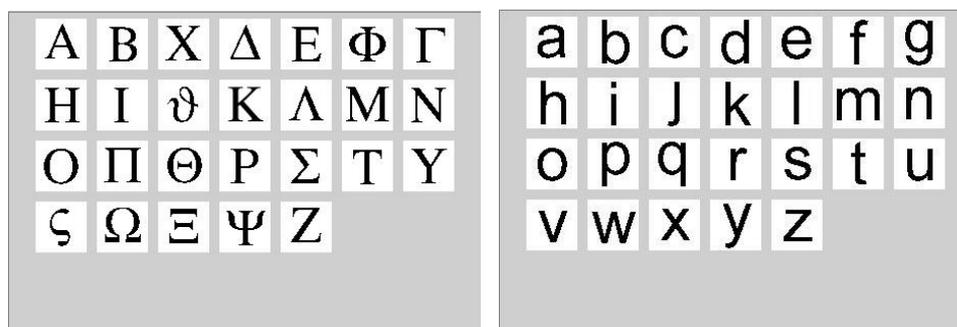
Conjunto fundamental 1:



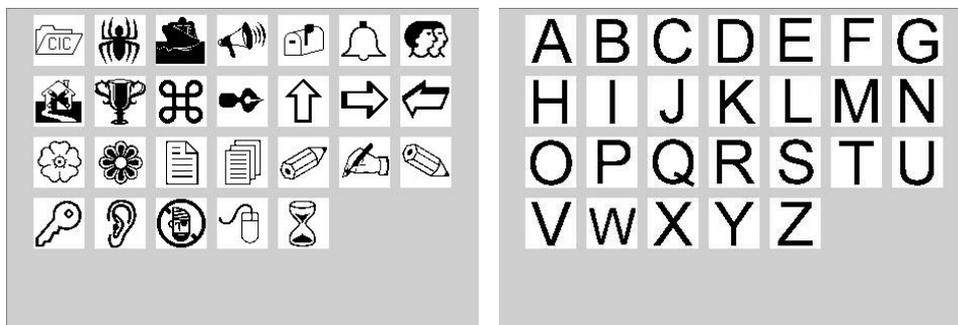
Conjunto fundamental 2:



Conjunto fundamental 3:



Conjunto fundamental 4:



Conjunto fundamental 5:



### 5.1.2. Resultados obtenidos con las memorias heteroasociativas sin ruido

A manera de ilustración se mostrará la tabla y la gráfica que se obtuvieron con el primer conjunto fundamental en el proceso de aprender e intentar recuperar todos los patrones aprendidos. Cada columna de la tabla indica tanto el número de patrones aprendidos como el número de intentos de recuperación. Después de esta tabla y gráfica, se mostrará sólo el máximo de patrones recuperados por memoria asociativa en cada conjunto fundamental.

#### Resultados obtenidos con el conjunto fundamental 1

|                        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| LernMatrix             | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
| Linear Associator      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  |
| Morfológica Min        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 10 |
| Morfológica Max        | 1 | 2 | 3 | 4 | 4 | 4 | 4 | 3 | 1 | 0  | 0  | 0  | 0  |
| Alfa Beta Min          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 10 |
| Alfa Beta Max          | 1 | 2 | 3 | 4 | 4 | 4 | 4 | 3 | 1 | 0  | 0  | 0  | 0  |
| Morfológica Heurística | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Alfa Beta Heurística   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |



La tabla 5.1 y la gráfica 5.1, ilustran que las memorias heurísticas recuperaron todos los patrones de manera perfecta; las memorias de tipo  $\wedge$ , haciendo referencia a las morfológicas y  $\alpha\beta$ , recuperaron 11 patrones. Es digno de resaltar que en las memorias de tipo  $\wedge$ , a pesar de que tienen problemas a partir del patrón 12, los patrones que no son recuperados de manera perfecta, son los mismos patrones aprendidos sólo que éstos contienen ruido sustractivo.

Las memorias de tipo  $\vee$ , haciendo referencia a las morfológicas y  $\alpha\beta$ , recuperaron 4 patrones, y a partir del quinto patrón comenzaron a degradarse. A estas memorias les sucede algo similar que a las memorias de tipo  $\wedge$ , es decir, que los patrones recuperados de manera no perfecta, son los mismos patrones aprendidos, sólo que éstos están afectados con ruido aditivo.

La razón de que las memorias de tipo min tienen mayor recuperación de patrones en comparación a las memorias de tipo max, es que la memoria de tipo min, en la fase de aprendizaje agrega los valores mínimos de patrones, es decir que almacena más ceros que unos, por esta misma situación cuando las memorias de tipo min aprenden cierta cantidad de patrones, estos patrones aprendidos comienzan a perder información de sus características; debido a esto, el patrón cuando es recuperado contiene ruido sustractivo, es decir, el patrón es el aprendido pero con un faltante de información. Por el lado de las memorias de tipo max al conformarse éstas, como se basan en el operador máximo, almacenan más unos que ceros, por lo tanto cuando estas memorias han aprendido cierta cantidad de patrones, éstas se saturan de unos provocando que en la fase de recuperación se obtengan los patrones aprendidos pero con ruido aditivo. Resumiendo lo anterior, la recuperación de las memorias min es mayor que las de tipo max, porque en las memorias min la saturación de unos es menor que en las memorias de tipo max.

La figura 5.1 y la figura 5.2 ilustran la apariencia que tienen los patrones recuperados por las memorias de tipo  $\wedge$  y de tipo  $\vee$ .



figura 5.1. Patrones recuperados por las memorias de tipo min.

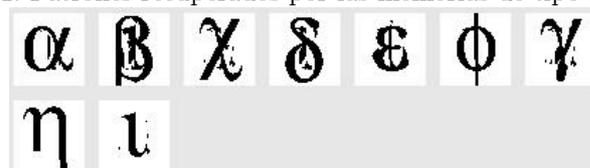


figura 5.2. Patrones recuperados por las memorias de tipo max.

*La memoria Lernmatrix de Steinbuch y la memoria Linear Associator de Anderson & Kohonen no fueron capaces de recuperar ningún patrón, e inclusive, la apariencia de los patrones recuperados no tienen ningún parecido con los patrones aprendidos, a excepción del primer patrón recuperado por la memoria Linear Associator.*



figura 5.3. Apariencia del primer patrón recuperado por el Linear Associator.

*Por las razones comentadas, esta será la última vez que se mencione a la memoria Lernmatrix y la memoria Linear Associator, en las comparaciones experimentales de esta sección. Sin embargo, los resultados obtenidos por estas memorias estarán incluidos en las tablas y gráficas.*

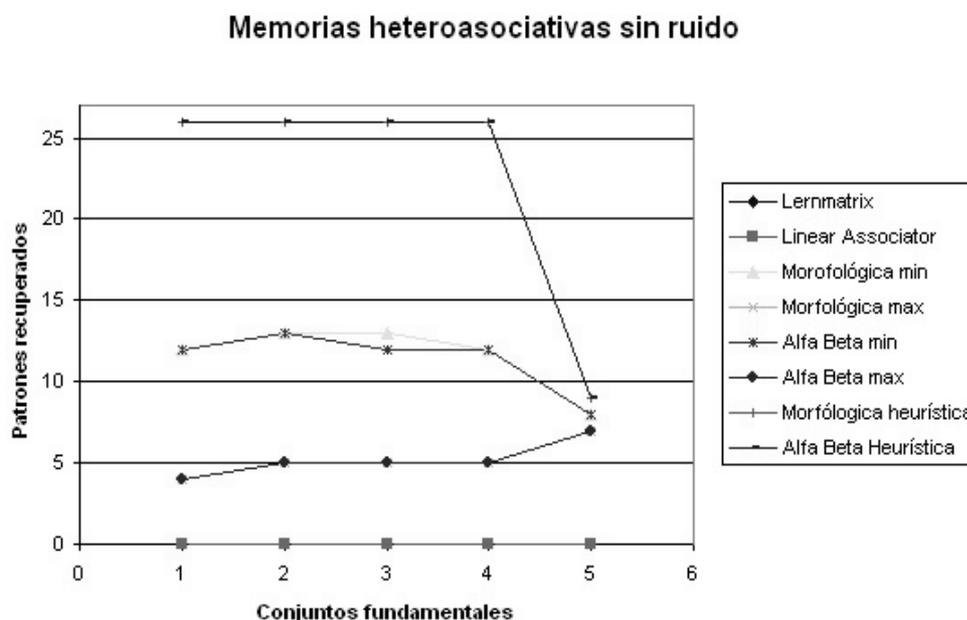
Las memorias  $\alpha\beta$  tienen un comportamiento muy similar a las memorias morfológicas, debido a que estas memorias se basan en los operadores  $\bigvee$  (memorias de tipo max) y  $\bigwedge$  (memorias de tipo min) para el aprendizaje y recuperación de patrones, por esta razón los resultados son muy parecidos; cabe mencionar que sí habrá algunas diferencias entre éstas, debido a que las memorias morfológicas se basan en máximos o mínimos de sumas para el aprendizaje y máximos o mínimos de sumas para la recuperación y las memorias  $\alpha\beta$ , se basan en máximos o mínimos de relaciones de orden de operadores binarios (operador  $\alpha$ ) para el aprendizaje y en máximos o mínimos de relaciones de orden de operadores binarios (operador  $\beta$ ) para el proceso de recuperación.

### Resultados obtenidos con todos los conjuntos fundamentales sin ruido

La tabla 5.2 y la gráfica 5.2 ilustran el total de patrones recuperados por las memorias heteroasociativas sin ruido por cada conjunto fundamental.

|                   | Conj.1 | Conj.2 | Conj.3 | Conj.4 | Conj.5 |
|-------------------|--------|--------|--------|--------|--------|
| Lernmatrix        | 0      | 0      | 0      | 0      | 0      |
| Linear Associator | 0      | 0      | 0      | 0      | 0      |
| Morfológica min   | 12     | 13     | 13     | 12     | 8      |
| Morfológica max   | 4      | 5      | 5      | 5      | 7      |
| Alfa Beta min     | 12     | 13     | 12     | 12     | 8      |
| Alfa Beta max     | 4      | 5      | 5      | 5      | 7      |
| Morf. heurist.    | 26     | 26     | 26     | 26     | 9      |
| AB heurist.       | 26     | 26     | 26     | 26     | 9      |

tabla 5.2. Recuperación de patrones por conjunto fundamental.



gráfica 5.2. Recuperación de patrones por conjunto fundamental sin ruido.

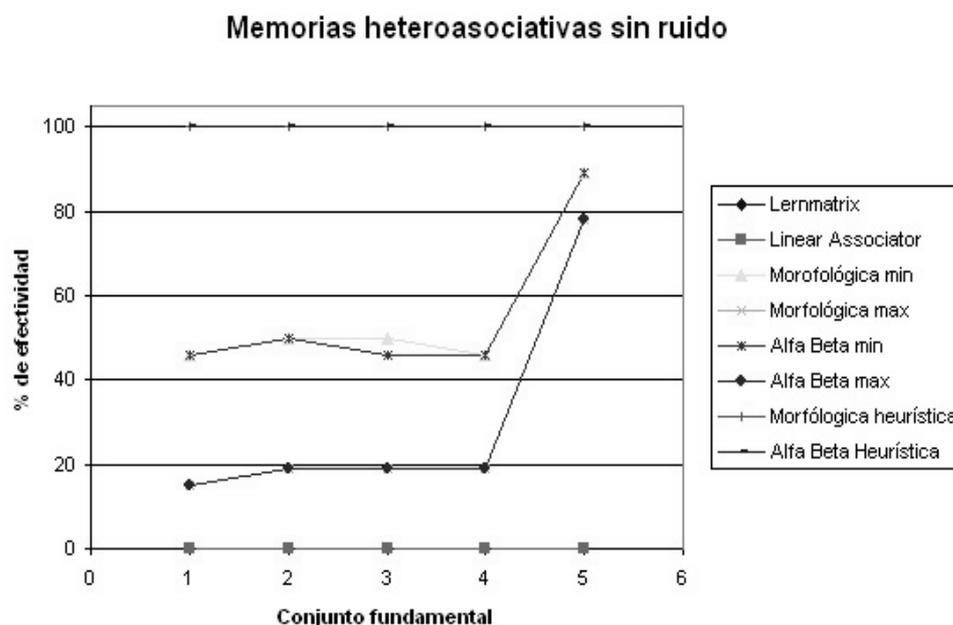
Las memorias de tipo  $\vee$  obtuvieron los mismo resultados.

Ahora bien, se puede observar que en las memorias de tipo  $\wedge$  en el conjunto fundamental 3 hay una diferencia, ya que la memoria morfológica recuperó un patrón más que la memoria  $\alpha\beta$ . Respecto de esta situación, cabe recordar que se comentó en la sección anterior que no es una regla el que las memorias de tipo  $\wedge$  siempre obtengan los mismos resultados, porque finalmente éstas son diferentes a las memorias max.

De nuevo, las memorias heurísticas recuperaron todos los patrones.

|                   | Conj.1 | Conj.2 | Conj.3 | Conj.4 | Conj.5 |
|-------------------|--------|--------|--------|--------|--------|
| Lernmatrix        | 0      | 0      | 0      | 0      | 0      |
| Linear Associator | 0      | 0      | 0      | 0      | 0      |
| Morfológica min   | 46     | 50     | 50     | 46     | 89     |
| Morfológica max   | 15     | 19     | 19     | 19     | 78     |
| Alfa Beta min     | 46     | 50     | 46     | 46     | 89     |
| Alfa Beta max     | 15     | 19     | 19     | 19     | 78     |
| Morf. heurist.    | 100    | 100    | 100    | 100    | 100    |
| AB heurist.       | 100    | 100    | 100    | 100    | 100    |

tabla 5.3. Porcentaje de efectividad en la recuperación de patrones por conjunto fundamental sin ruido.



gráfica 5.3. Porcentaje de efectividad en la recuperación de patrones por conjunto fundamental sin ruido.

### 5.1.3. Resultados obtenidos con las memorias heteroasociativas afectadas con algún tipo de ruido

El porcentaje de ruido que contendrán los patrones a recuperar serán del 5, 10, 20, 40, 50, 60, 80, 90 y 99%. Un ejemplo de la apariencia de los patrones afectados por estos porcentajes de ruido se ilustra en la figura 5.4.

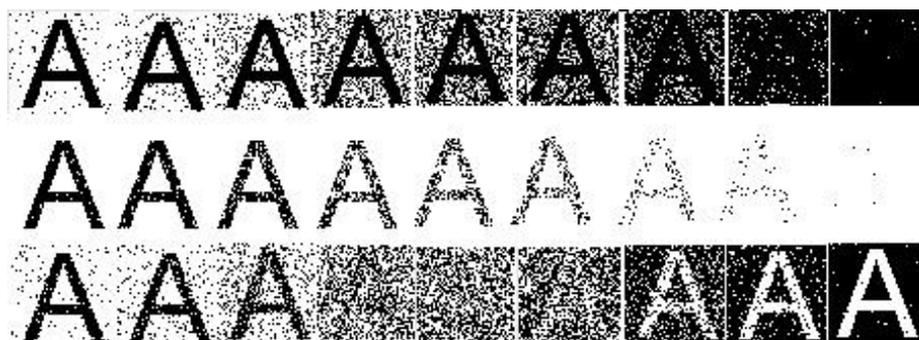


figura 5.4. Porcentajes de ruido del 5,10,20,40,50,60,80,90 y 99 % de ruido aditivo, sustractivo y mezclado respectivamente.

Los resultados se mostrarán por conjunto fundamental. Cada conjunto fundamental presentará una tabla y una gráfica de patrones recuperados, así como una tabla y una gráfica de porcentajes de efectividad de recuperación de los diferentes modelos de memorias heteroasociativas. El orden en que aparecen las tablas y las gráficas son por: ruido aditivo, ruido sustractivo y ruido mezclado.

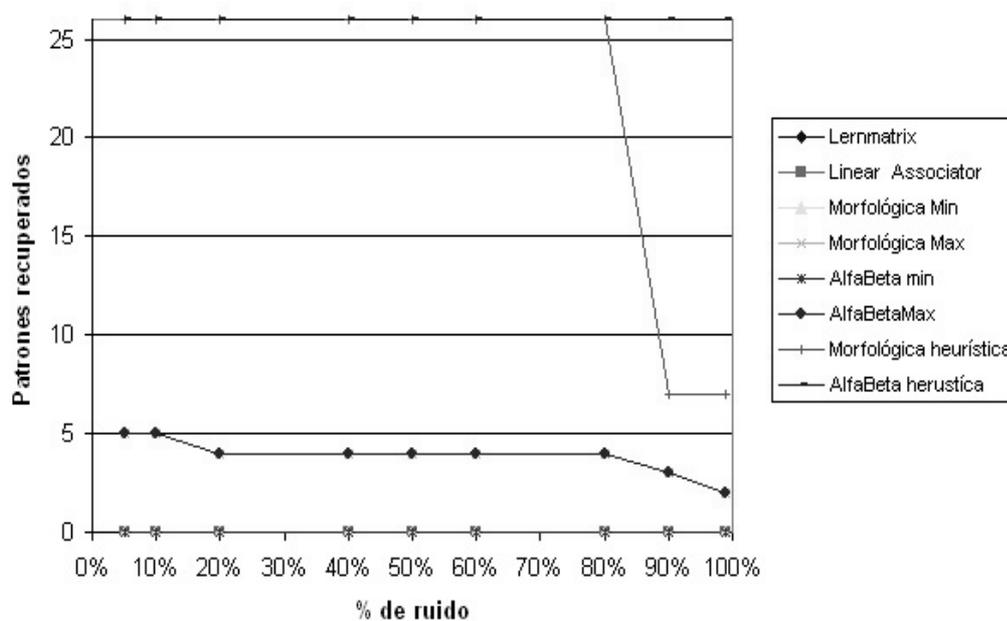
## Conjunto fundamental 1.

El conjunto fundamental 1 afectado por ruido aditivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 5   | 5    | 4    | 4    | 4    | 4    | 4    | 3    | 2    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 5   | 5    | 4    | 4    | 4    | 4    | 4    | 3    | 2    |
| Mor.heurística | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 7    | 7    |
| AB heurística  | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |

tabla 5.4. Patrones recuperados con los diferentes porcentaje de ruido aditivo para el conjunto fundamental 1.

## Patrones recuperados con ruido aditivo



gráfica 5.4. Patrones recuperados por las memorias ante el ruido aditivo para el conjunto fundamental 1.

Las memorias de tipo  $\wedge$  son muy sensibles al ruido aditivo por lo tanto nos capaces de recuperar ningún patrón aunque el porcentaje de ruido aditivo sea muy bajo. La tabla 5.4 y la gráfica 5.4 corroboran este hecho. En contraste las memorias de tipo  $\vee$

son robustas al ruido aditivo. La apariencia de 12 patrones aprendidos e intentados recuperar con un 40% de ruido aditivo por las memorias de tipo  $\nabla$ , es la siguiente:

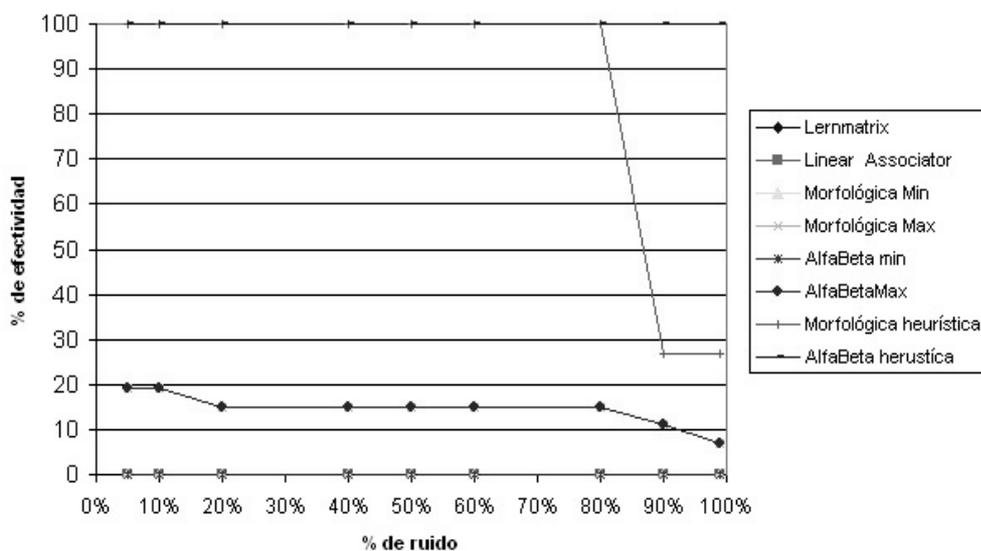


figura 5.4. Apariencia de 12 patrones recuperados por las memorias de tipo  $\nabla$  con un 40% de ruido aditivo.

|                | 5%  | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 99% |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Lernmatrix     | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Linear Assoc.  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Morf. min      | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Morf. max      | 19  | 19  | 15  | 15  | 15  | 15  | 15  | 11  | 7   |
| AlfaBeta min   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| AlfaBeta max   | 19  | 19  | 15  | 15  | 15  | 15  | 15  | 11  | 7   |
| Mor.heurística | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 27  | 27  |
| AB heurística  | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

tabla 5.5. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1 afectado con ruido aditivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido aditivo



gráfica 5.5. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1 afectado con ruido aditivo.

Claramente la tabla 5.5 y la gráfica 5.5 ilustran que las memorias heurísticas tuvieron el mejor porcentaje de recuperación que las memorias de tipo min y max, sin embargo, la efectividad se ve afectada por el tiempo de proceso debido a que las memorias heurísticas hacen las siguientes actividades:

1. Aprender el patrón con las memorias de tipo max
2. Aprender el patrón con las memorias de tipo min
3. En la fase de recuperación, las memorias heurísticas hacen:
  - 3.1. La dilatación del patrón a recuperan y lo intentan recuperar con las memorias de tipo max
  - 3.2. La erosión del patrón en cuestión y lo intentan recuperar con las memorias de tipo min
  - 3.3. Para los patrones recuperados, por las memorias de tipo min y de tipo max se realiza un AND lógico binario entre estos patrones.

Al hacer hincapié en estos procesos de las memorias heurísticas, se puede constatar que el tiempo computacional que requiere es mayor que las memorias de tipo min y de tipo max. En este mismo capítulo se podrán observar los tiempos computacionales consumidos por las memorias en sus fases de aprendizaje y recuperación; es importante destacar que las memorias heurísticas son favorecidas, debido a que los patrones afectados con ruido se procesan por medio de la erosión y dilatación, creando las condiciones necesarias para que se realicen recuperaciones perfectas, no sucede así para el resto de modelos de memorias asociativas. Por esta razón las memorias heurísticas obtienen mejores recuperaciones.

**El conjunto fundamental 1 afectado por ruido sustractivo, arrojó los siguientes resultados:**

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 12  | 12   | 11   | 11   | 11   | 11   | 10   | 7    | 5    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 12  | 12   | 11   | 11   | 11   | 11   | 10   | 7    | 5    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 26   | 17   | 6    | 1    | 0    | 0    | 0    | 0    |
| AB heurística  | 26  | 26   | 17   | 2    | 0    | 0    | 0    | 0    | 0    |

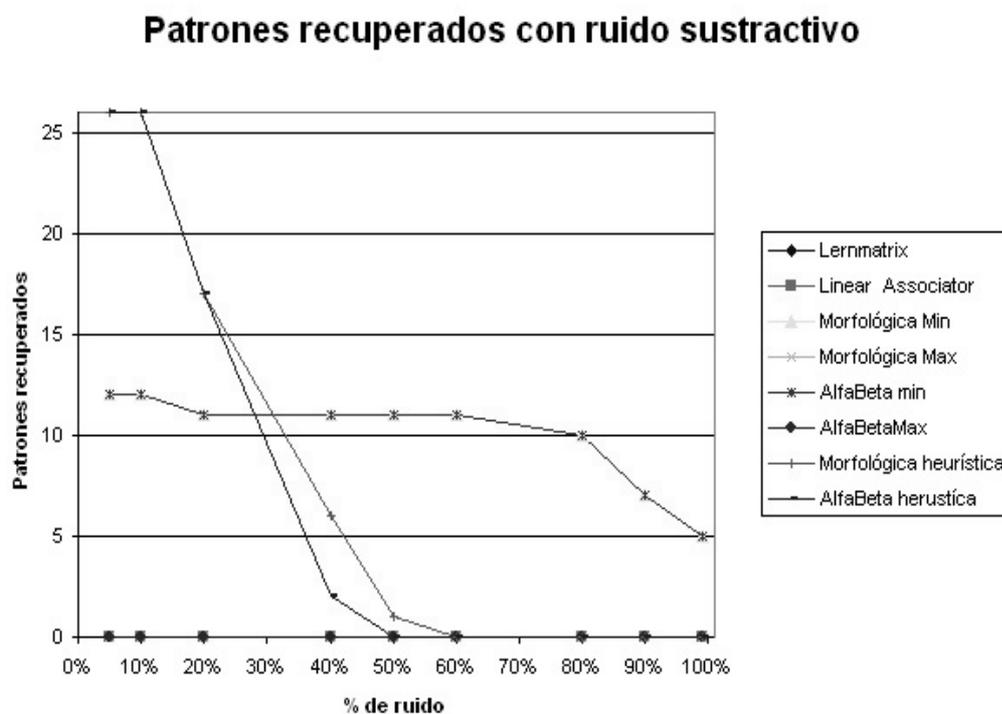
tabla 5.6. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1 afectado con ruido sustractivo.

La tabla 5.6 y la gráfica 5.6 ilustran las bondades que tienen las memorias de tipo  $\bigwedge$  para la recuperación de patrones afectados con ruido sustractivo. Algo relevante que es digno de mencionar es que las memorias heurísticas se ven muy afectadas en su rendimiento ante el ruido sustractivo y esto tiene mucho sentido, debido a que, en la fase de recuperación de éstas, hace que se erosione el patrón a recuperar, se intenta

recuperar el patrón erosionado con la memoria de tipo min, generando la posibilidad de que dicho patrón a recuperar contenga ruido sustractivo, por lo tanto, puede ser que no haya recuperación perfecta; ahora bien, cuando se dilata el patrón en cuestión y se recupera con la memoria de tipo max, a pesar de que tenga ruido aditivo de más, al momento de hacer el AND lógico, el patrón que se quiere recuperar tendrá ruido sustractivo; como consecuencia la recuperación no es perfecta. La figura 5.5 ilustra lo comentado para las memorias heurísticas, con un 40% de ruido sustractivo para 12 patrones aprendidos y recuperados.



figura 5.5. Apariencia de 12 patrones recuperados con las memorias heurísticas, después de haber estado afectados con el 40% de ruido sustractivo.

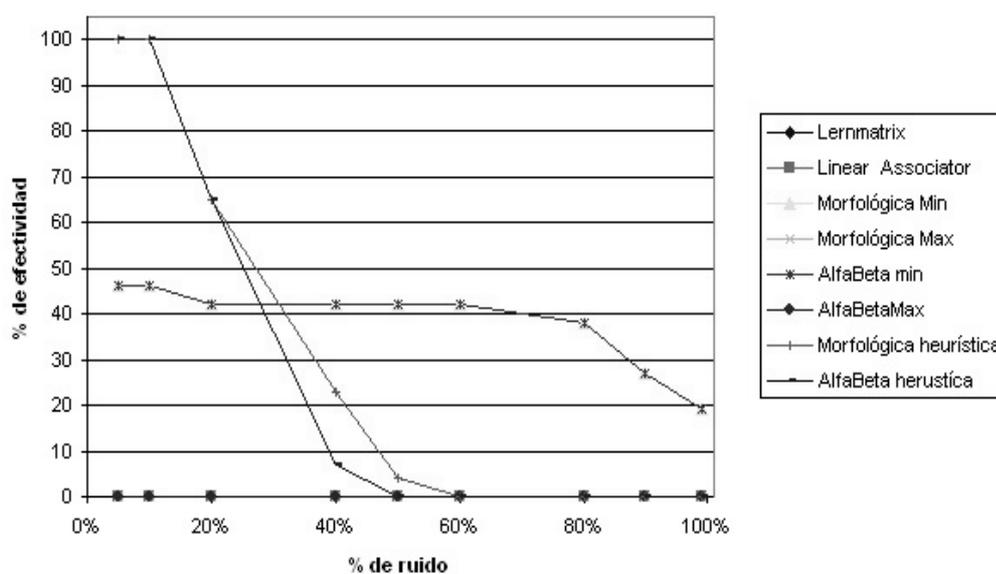


gráfica 5.6. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1 afectado con ruido sustractivo.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 46  | 46   | 42   | 42   | 42   | 42   | 38   | 27   | 19   |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 46  | 46   | 42   | 42   | 42   | 42   | 38   | 27   | 19   |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 65   | 23   | 4    | 0    | 0    | 0    | 0    |
| AB heurística  | 100 | 100  | 65   | 7    | 0    | 0    | 0    | 0    | 0    |

tabla 5.7. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1 afectado con ruido sustractivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido sustractivo.



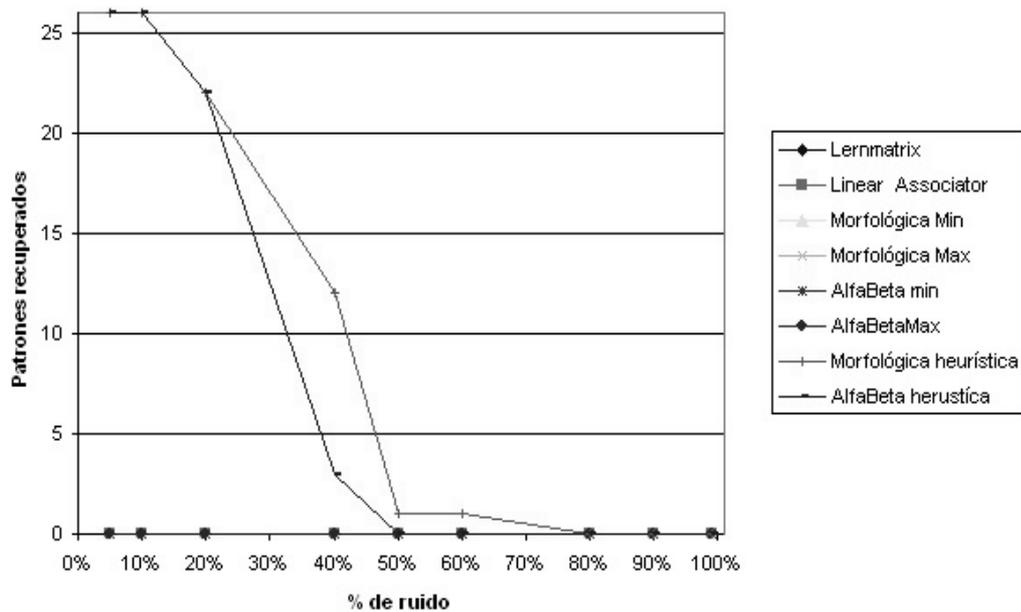
gráfica 5.7. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 1 afectado con ruido sustractivo.

El conjunto fundamental 1 afectado por ruido mezclado, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 26   | 22   | 12   | 1    | 1    | 0    | 0    | 0    |
| AB heurística  | 26  | 26   | 22   | 3    | 0    | 0    | 0    | 0    | 0    |

tabla 5.8. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1 afectado con ruido mezclado.

### Patrones recuperados con ruido mezclado



gráfica 5.8. Total de patrones recuperados por memoria heteroasociativa para el conjunto fundamental 1 afectado con ruido mezclado.

Tanto La tabla y la gráfica 5.8, ilustran que las memorias de tipo  $\wedge$  y de tipo  $\vee$  son ineficaces para la patrones con ruido mezclado a pesar que éste sea mínimo. Por el lado de las memorias heurísticas también se corrobora que fueron las únicas memorias capaces de recuperar patrones con ruido mezclado. Es digno de resaltar que

las memorias heurísticas recuperan patrones de manera perfecta aún con porcentajes altos de ruido mezclado, hablar de un 40 % de ruido mezclado es hablar de una cantidad impresionante de este tipo de ruido; estas memorias no son capaces de recuperar ningún patrón con más del 60 % de ruido mezclado, pero esto tiene sentido, porque entre mayor sea el porcentaje de ruido mezclado, el patrón se acerca a su negativo, es decir que se está hablando de dos patrones diferentes. La figura 5.6 ilustra 12 patrones aprendidos y recuperados con 60 % de ruido mezclado.

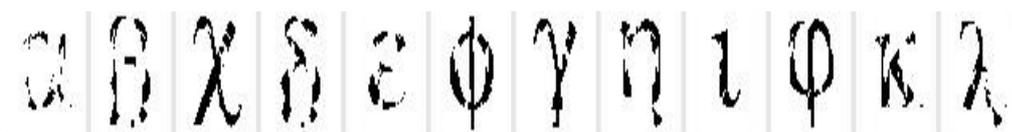
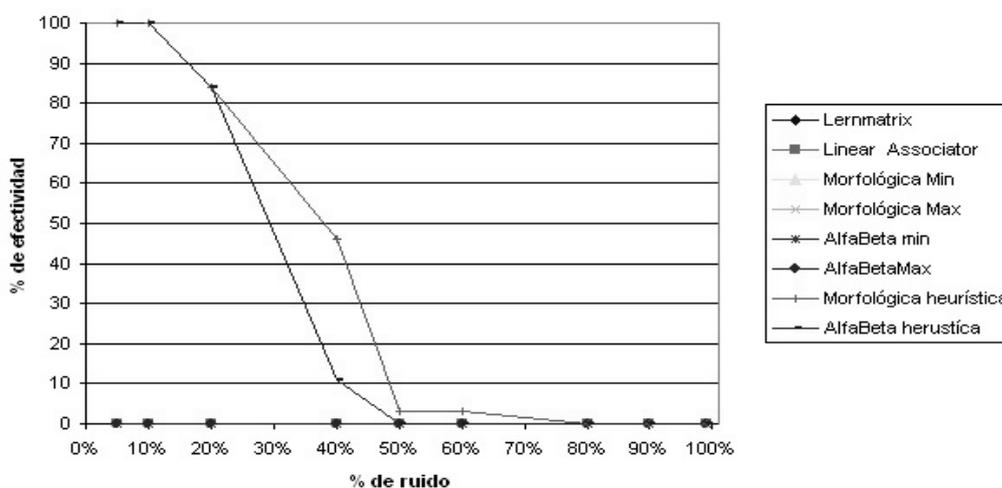


figura 5.6. Apariencia de patrones recuperados por las memorias heurísticas, después de haber estado afectados con el 60 % de ruido mezclado.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 84   | 46   | 3    | 3    | 0    | 0    | 0    |
| AB heurística  | 100 | 100  | 84   | 11   | 0    | 0    | 0    | 0    | 0    |

tabla 5.9. Porcentaje de efectividad en la recuperación de patrones afectados con ruido mezclado.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido mezclado



gráfica 5.9. Porcentaje de efectividad en la recuperación de patrones afectados con ruido mezclado.

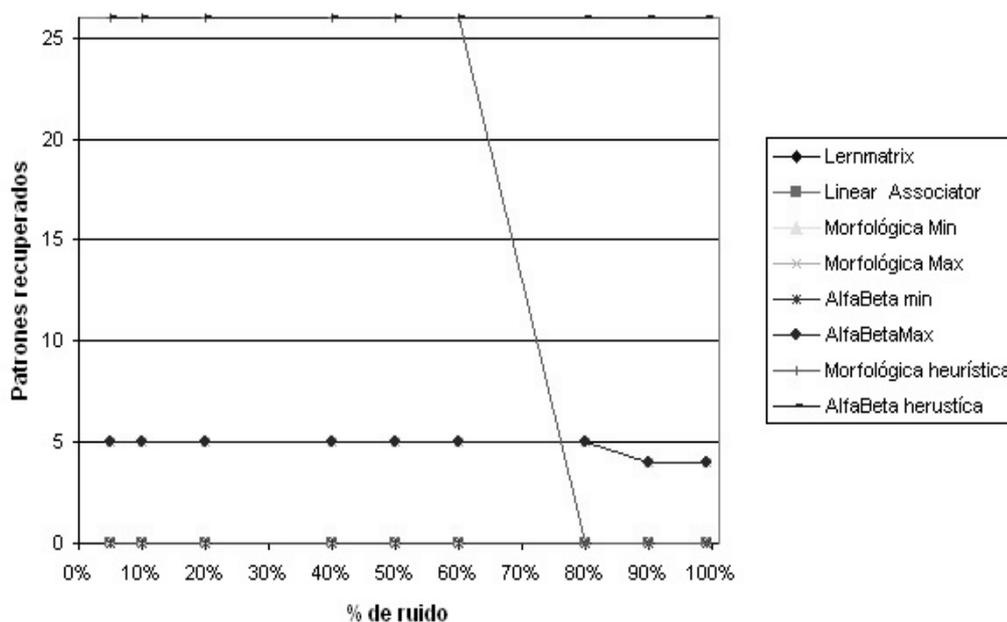
## Conjunto fundamental 2.

El conjunto fundamental 2 afectado por ruido aditivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 5   | 5    | 5    | 5    | 5    | 5    | 5    | 4    | 4    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 5   | 5    | 5    | 5    | 5    | 5    | 5    | 4    | 4    |
| Mor.heurística | 26  | 26   | 26   | 26   | 26   | 26   | 0    | 0    | 0    |
| AB heurística  | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |

tabla 5.10. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo.

## Patrones recuperados con ruido aditivo



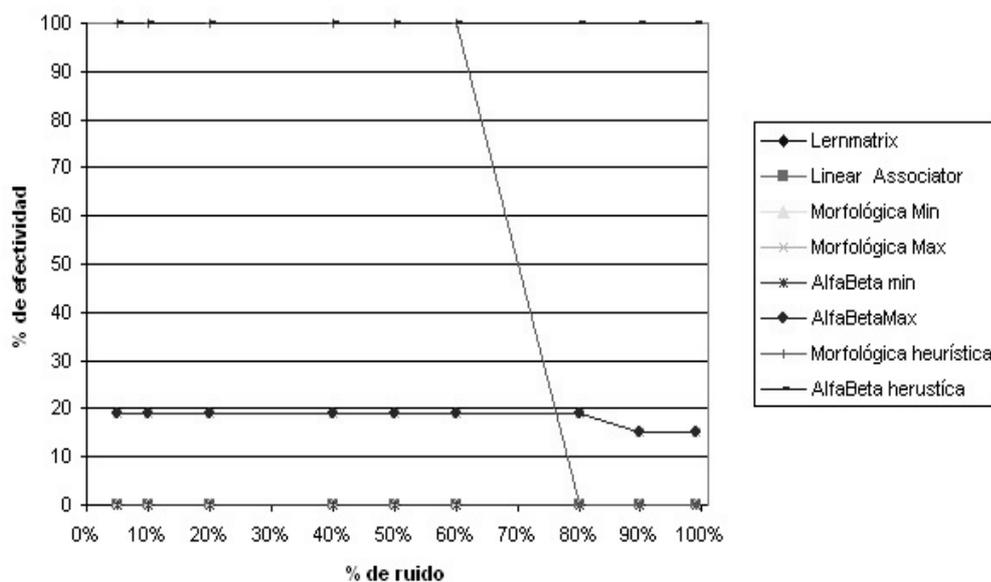
gráfica 5.10. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo.

El conjunto fundamental 2 para el ruido aditivo tiene una tendencia similar de comportamiento al primer conjunto fundamental. Se puede observar que las memorias de tipo max y las memorias heurísticas son las que recuperaron patrones de manera perfecta.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 19  | 19   | 19   | 19   | 19   | 19   | 19   | 15   | 15   |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 19  | 19   | 19   | 19   | 19   | 19   | 19   | 15   | 15   |
| Mor.heurística | 100 | 100  | 100  | 100  | 100  | 100  | 0    | 0    | 0    |
| AB heurística  | 100 | 100  | 100  | 100  | 100  | 100  | 100  | 100  | 100  |

tabla 5.11. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 2 afectado con ruido aditivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido aditivo



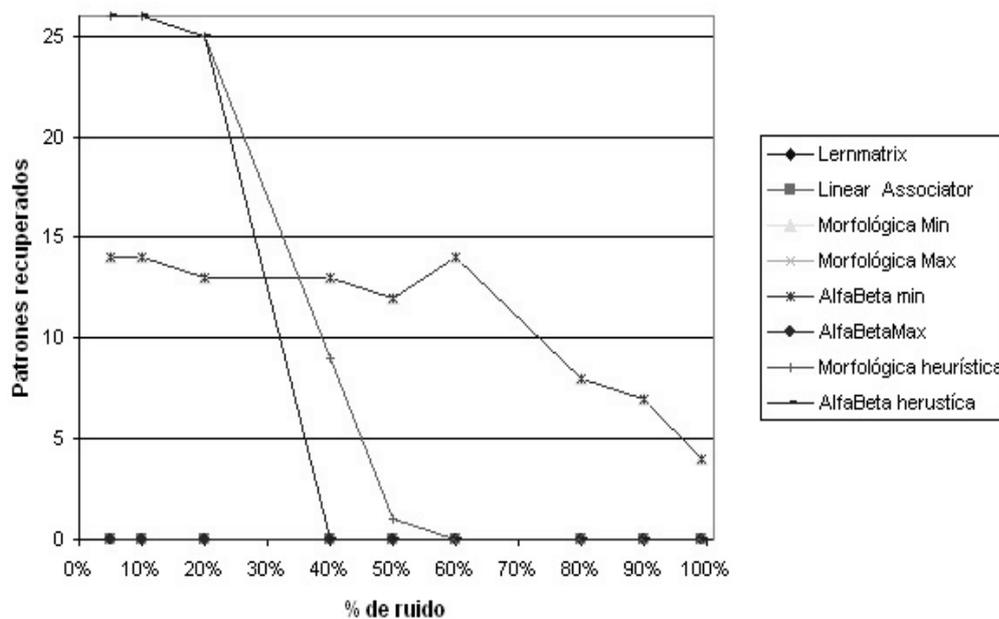
gráfica 5.11. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 2 afectado con ruido aditivo.

El conjunto fundamental 2 afectado por ruido sustractivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 14  | 14   | 13   | 13   | 12   | 14   | 8    | 7    | 4    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 14  | 14   | 13   | 13   | 12   | 14   | 8    | 7    | 4    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 26   | 25   | 9    | 1    | 0    | 0    | 0    | 0    |
| AB heurística  | 26  | 26   | 25   | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.12. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo.

### Patrones recuperados con ruido sustractivo



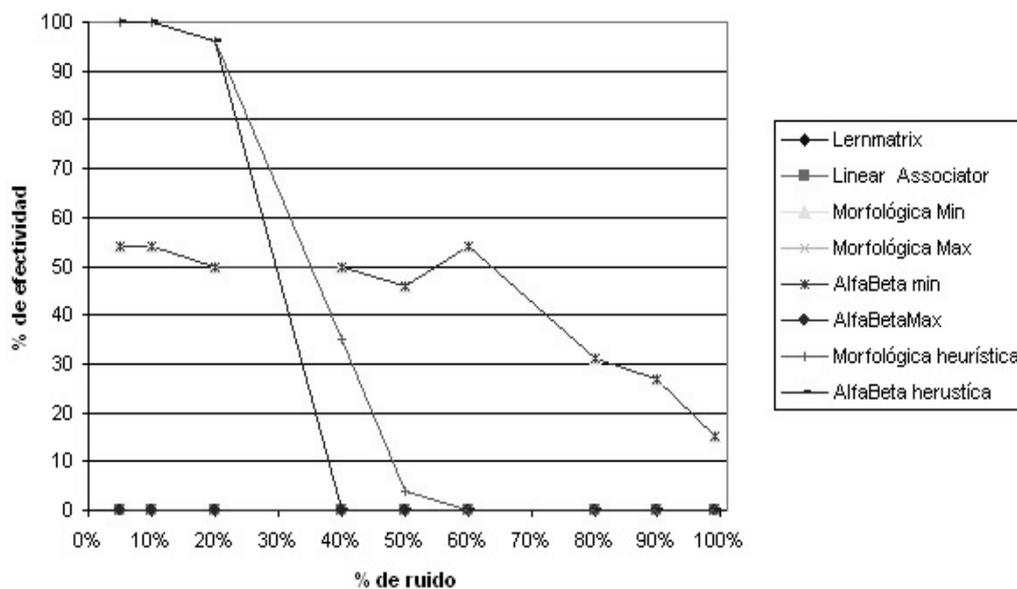
gráfica 5.12. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido aditivo.

Se puede observar tanto en la tabla y la gráfica 5.12 que las memorias de tipo  $\wedge$  son las que destacaron con recuperaciones perfectas.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 54  | 54   | 50   | 50   | 46   | 54   | 31   | 27   | 15   |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 54  | 54   | 50   | 50   | 46   | 54   | 31   | 27   | 15   |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 96   | 35   | 4    | 0    | 0    | 0    | 0    |
| AB heurística  | 100 | 100  | 96   | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.13. Porcentaje de efectividad de patrones recuperados para el conjunto fundamental 2 afectado con ruido sustractivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido sustractivo



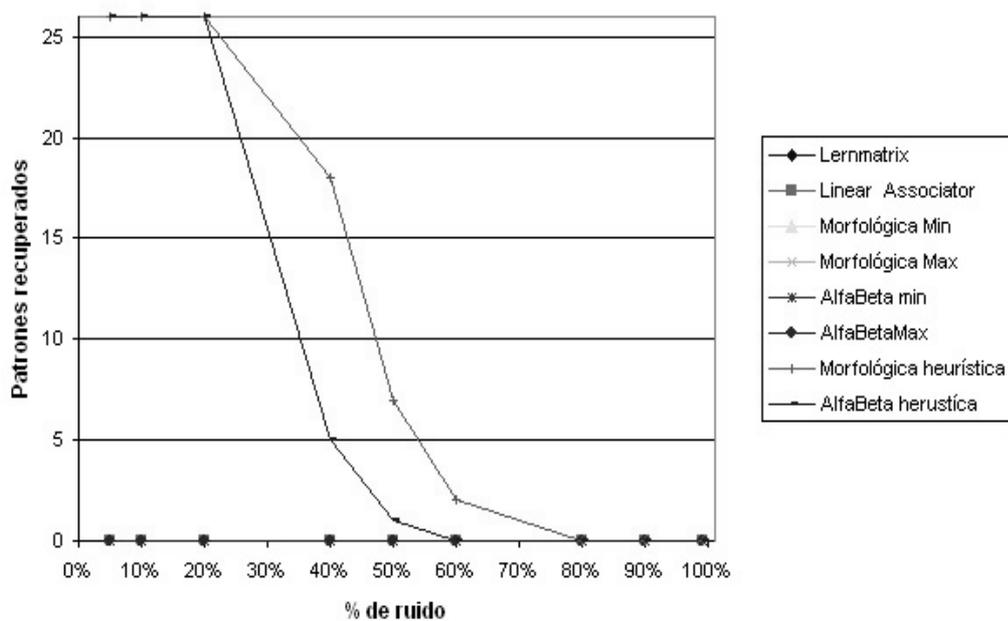
gráfica 5.13. Porcentaje de efectividad de patrones recuperados para el conjunto fundamental 2 afectado con ruido sustractivo.

El conjunto fundamental 2 afectado por ruido mezclado, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 26   | 26   | 18   | 7    | 2    | 0    | 0    | 0    |
| AB heurística  | 26  | 26   | 26   | 5    | 1    | 0    | 0    | 0    | 0    |

tabla 5.14. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido mezclado.

### Patrones recuperados con ruido mezclado

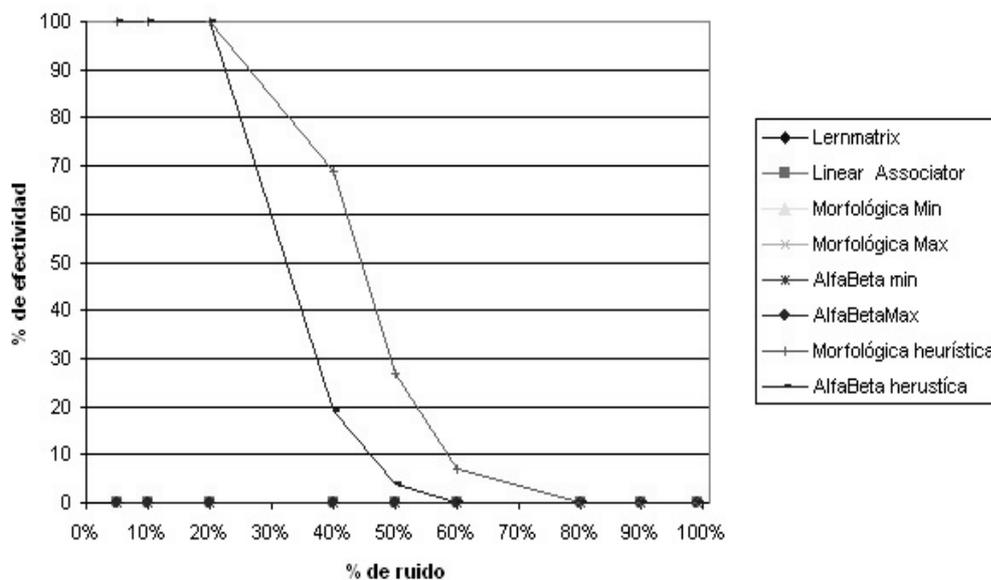


gráfica 5.14. Total de patrones recuperados para el conjunto fundamental 2 afectado con ruido mezclado.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 100  | 69   | 27   | 7    | 0    | 0    | 0    |
| AB heurística  | 100 | 100  | 100  | 19   | 4    | 0    | 0    | 0    | 0    |

tabla 5.15. Porcentaje de efectividad de las memorias heteroasociativas en el conjunto fundamental 2 afectado con ruido mezclado.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido mezclado



gráfica 5.15. Porcentaje de efectividad de las memorias heteroasociativas en el conjunto fundamental 2 afectado con ruido mezclado.

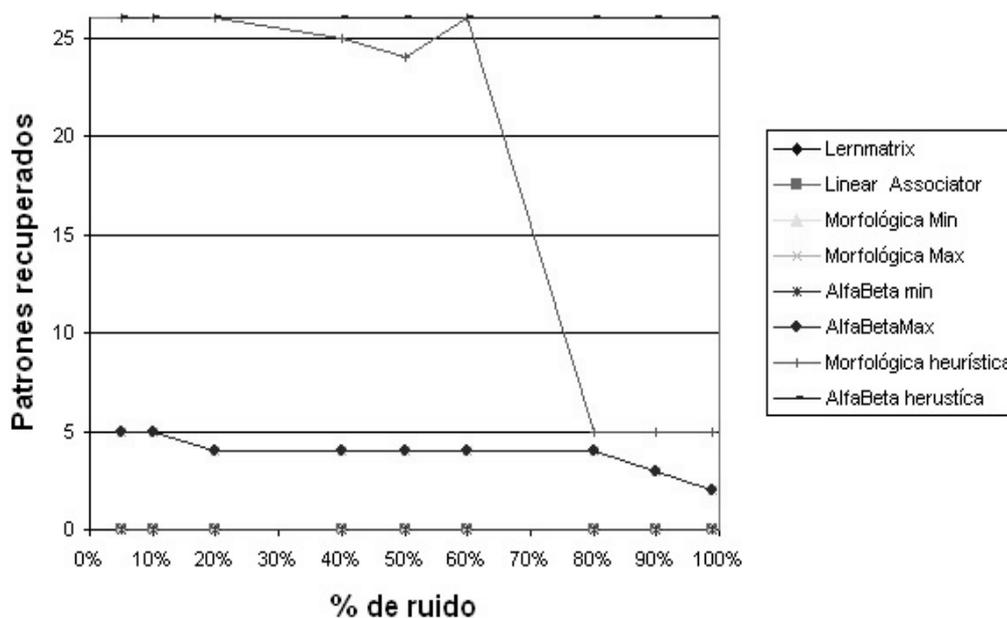
## Conjunto fundamental 3.

El conjunto fundamental 3 afectado por ruido aditivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 5   | 5    | 4    | 4    | 4    | 4    | 4    | 3    | 2    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 5   | 5    | 4    | 4    | 4    | 4    | 4    | 3    | 2    |
| Mor.heurística | 26  | 26   | 26   | 25   | 24   | 26   | 5    | 5    | 5    |
| AB heurística  | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |

tabla 5.16 Total de patrones recuperados por las memorias heteroasociativas en el conjunto fundamental 3 afectado con ruido aditivo.

## Patrones recuperados con ruido aditivo



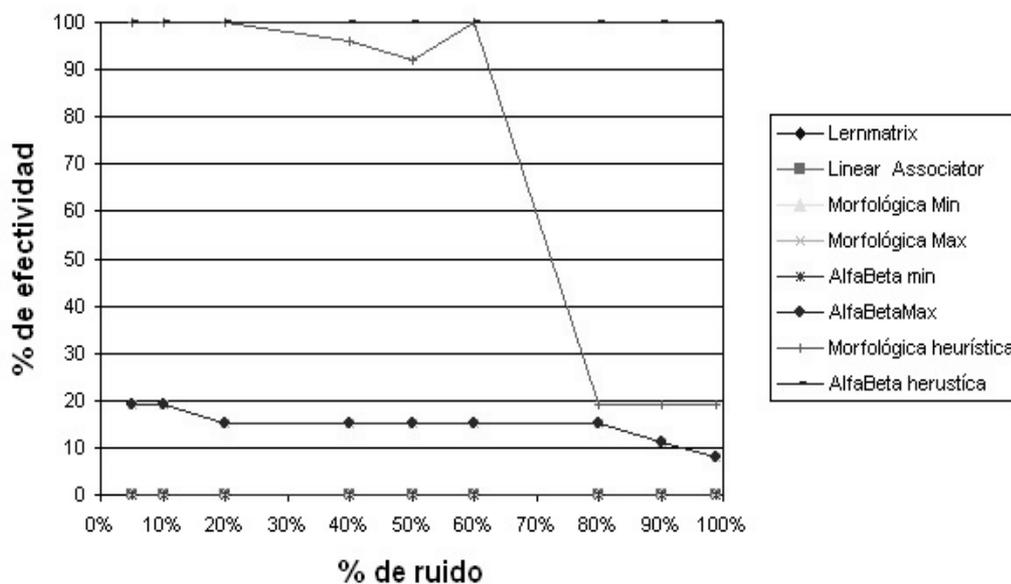
gráfica 5.16 Total de patrones recuperados por las memorias heteroasociativas en el conjunto fundamental 3 afectado con ruido aditivo.

La tabla 5.16 y la grafica 5.17 constatan de que las memorias de tipo max y las heurísticas fueron capaces de recuperar patrones afectados con ruido aditivo.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 19  | 19   | 15   | 15   | 15   | 15   | 15   | 11   | 8    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 19  | 19   | 15   | 15   | 15   | 15   | 15   | 11   | 8    |
| Mor.heurística | 100 | 100  | 100  | 96   | 92   | 100  | 19   | 19   | 19   |
| AB heurística  | 100 | 100  | 100  | 100  | 100  | 100  | 100  | 100  | 100  |

tabla 5.17 Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 3 afectado con ruido aditivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido aditivo



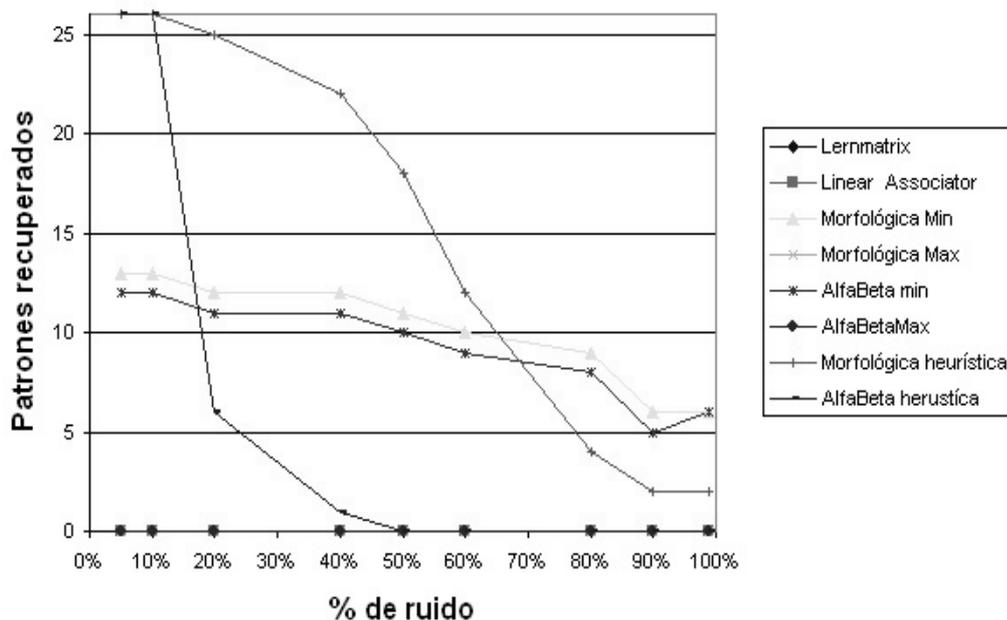
gráfica 5.17 Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 3 afectado con ruido aditivo.

El conjunto fundamental 3 afectado por ruido sustractivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 13  | 13   | 12   | 12   | 11   | 10   | 9    | 6    | 6    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 12  | 12   | 11   | 11   | 10   | 9    | 8    | 5    | 6    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 26   | 25   | 22   | 18   | 12   | 4    | 2    | 2    |
| AB heurística  | 26  | 26   | 6    | 1    | 0    | 0    | 0    | 0    | 0    |

tabla 5.18 Total de patrones recuperados en el conjunto fundamental 3 afectado con ruido sustractivo.

### Patrones recuperados con ruido sustractivo



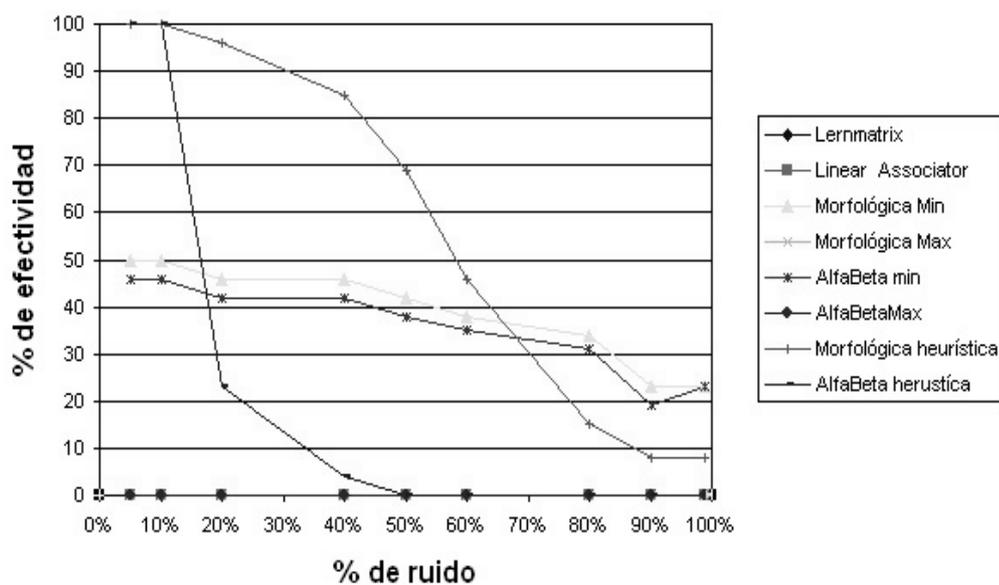
gráfica 5.18 Total de patrones recuperados en el conjunto fundamental 3 afectado con ruido sustractivo.

La tabla 5.18 y la gráfica 5.18 ilustran que las memorias de tipo min, son las que obtuvieron las mejores recuperaciones.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 50  | 50   | 46   | 46   | 42   | 38   | 34   | 23   | 23   |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 46  | 46   | 42   | 42   | 38   | 35   | 31   | 18   | 23   |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 96   | 85   | 69   | 46   | 15   | 8    | 8    |
| AB heurística  | 100 | 100  | 23   | 4    | 0    | 0    | 0    | 0    | 0    |

tabla 5.19. Porcentajes de efectividad en la recuperación de patrones en el conjunto fundamental 3 afectado con ruido sustractivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido sustractivo

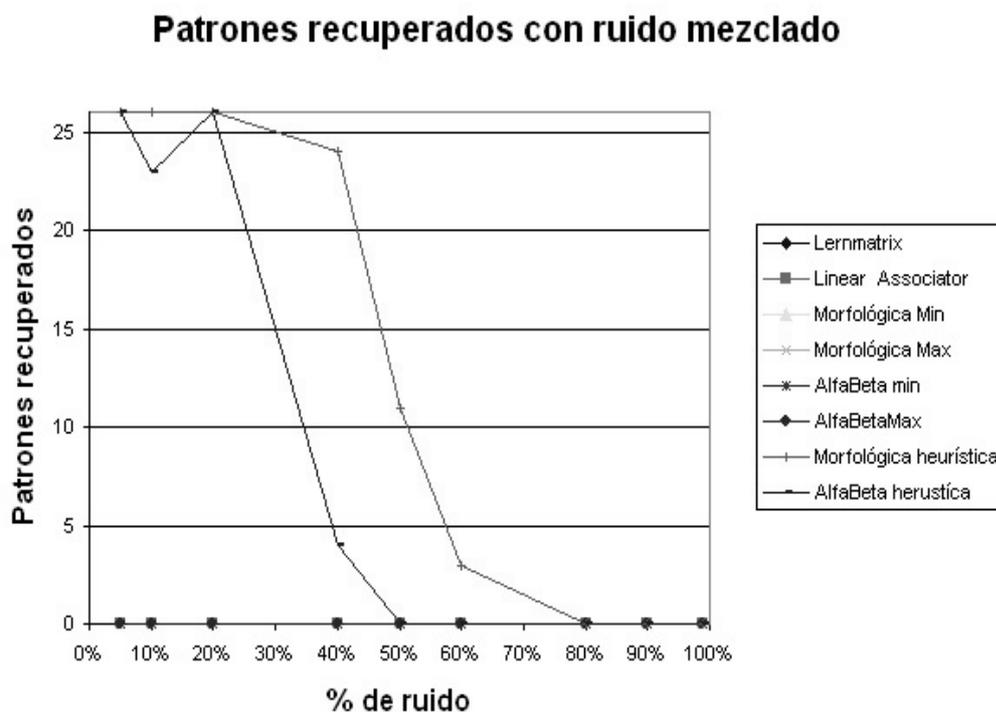


gráfica 5.19. Porcentajes de efectividad en la recuperación de patrones en el conjunto fundamental 3 afectado con ruido sustractivo.

El conjunto fundamental 3 afectado por ruido mezclado, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 26   | 26   | 24   | 11   | 3    | 0    | 0    | 0    |
| AB heurística  | 26  | 23   | 26   | 4    | 0    | 0    | 0    | 0    | 0    |

tabla 5.20. Total de patrones recuperados en el conjunto fundamental 3 alterado con ruido mezclado.



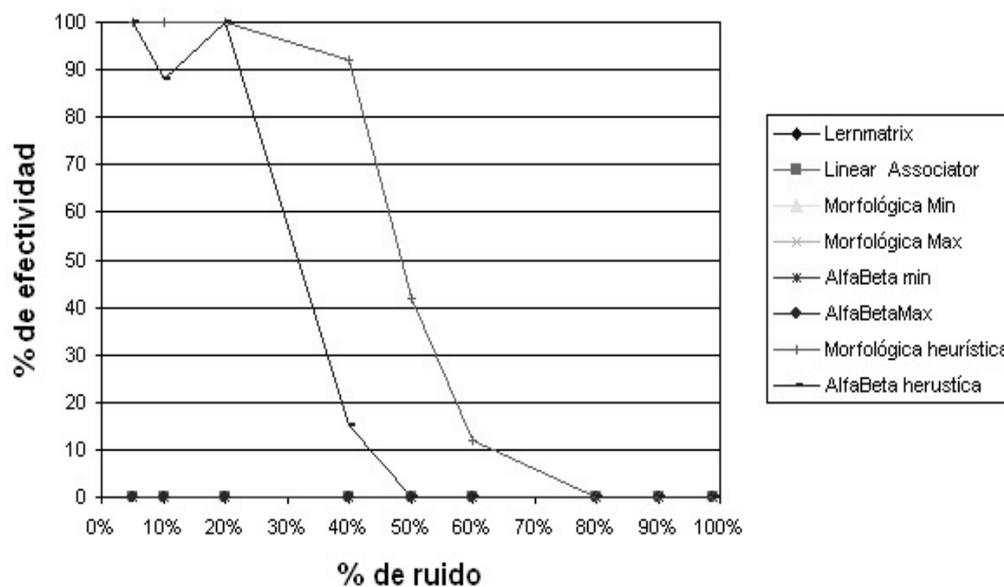
gráfica 5.20. Total de patrones recuperados en el conjunto fundamental 3 alterado con ruido mezclado.

La tendencia en cuanto al ruido mezclado procesado con las memorias heurísticas es muy similar a la tendencia del conjunto fundamental 1 y 2. Como se puede observar las memorias heurísticas son las únicas que pudieron recuperar de manera perfecta patrones.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 100  | 92   | 42   | 12   | 0    | 0    | 0    |
| AB heurística  | 100 | 88   | 100  | 15   | 0    | 0    | 0    | 0    | 0    |

tabla 5.21. Porcentaje de efectividad de patrones recuperados en el conjunto fundamental 3 alterado con ruido mezclado.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido mezclado



gráfica 5.21. Porcentaje de efectividad de patrones recuperados en el conjunto fundamental 3 alterado con ruido mezclado.

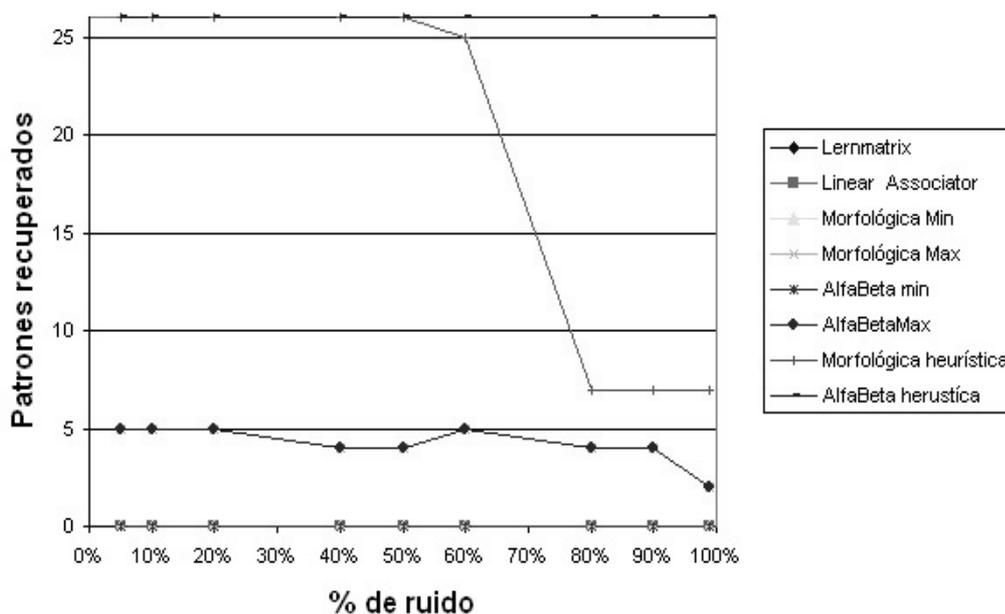
## Conjunto fundamental 4.

El conjunto fundamental 4 afectado por ruido aditivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 5   | 5    | 5    | 4    | 4    | 5    | 4    | 4    | 2    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 5   | 5    | 5    | 4    | 4    | 5    | 4    | 4    | 2    |
| Mor.heurística | 26  | 26   | 26   | 26   | 26   | 25   | 7    | 7    | 7    |
| AB heurística  | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |

tabla 5.22. Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido aditivo.

## Patrones recuperados con ruido aditivo



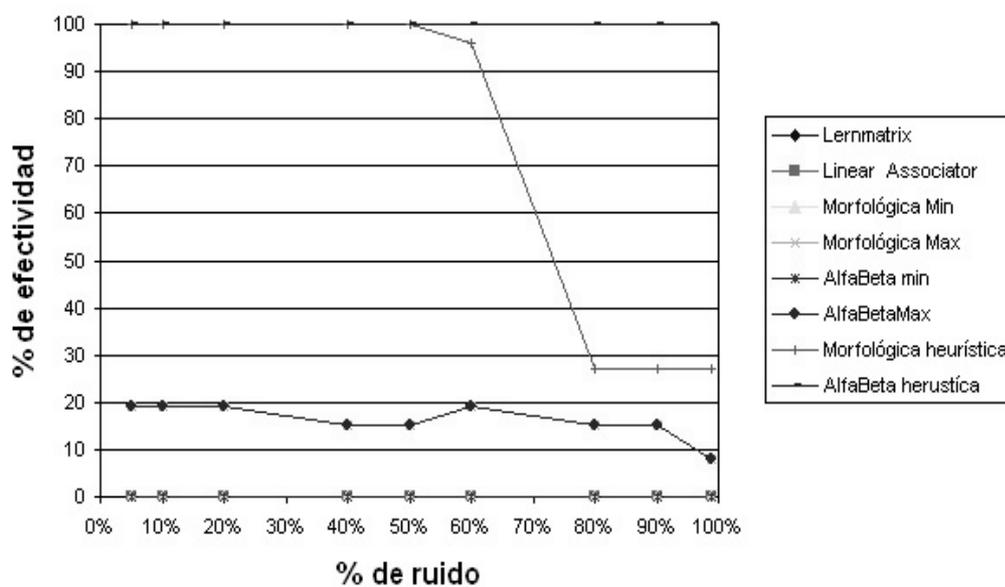
gráfica 5.22. Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido aditivo.

Las memorias de tipo max y las heurísticas siguen mostrando la misma tendencia que las tendencias obtenidas con los conjuntos fundamentales anteriores a este.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 19  | 19   | 19   | 15   | 15   | 19   | 15   | 15   | 8    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 19  | 19   | 19   | 15   | 15   | 19   | 15   | 15   | 8    |
| Mor.heurística | 100 | 100  | 100  | 100  | 100  | 96   | 27   | 27   | 27   |
| AB heurística  | 100 | 100  | 100  | 100  | 100  | 100  | 100  | 100  | 100  |

tabla 5.23. Porcentaje de efectividad de recuperación de patrones en el conjunto fundamental 4 afectado con ruido aditivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido aditivo



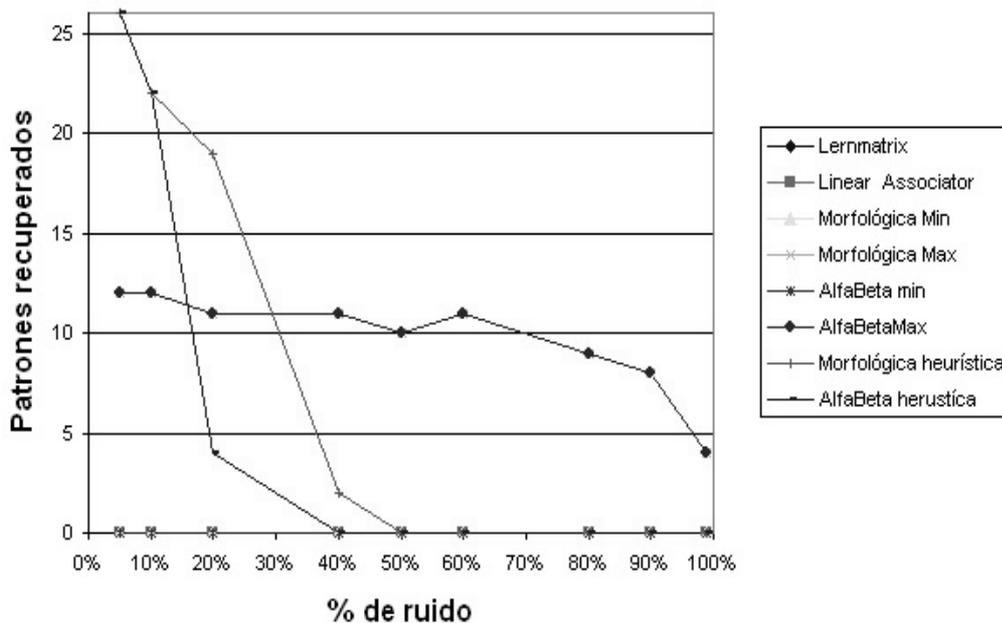
gráfica 5.23. Porcentaje de efectividad de recuperación de patrones en el conjunto fundamental 4 afectado con ruido aditivo.

El conjunto fundamental 4 afectado por ruido sustractivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 12  | 12   | 11   | 11   | 10   | 11   | 9    | 8    | 4    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 12  | 12   | 11   | 11   | 10   | 11   | 9    | 8    | 4    |
| Mor.heurística | 26  | 22   | 19   | 2    | 0    | 0    | 0    | 0    | 0    |
| AB heurística  | 26  | 22   | 4    | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.24 Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido sustractivo.

### Patrones recuperados con ruido sustractivo



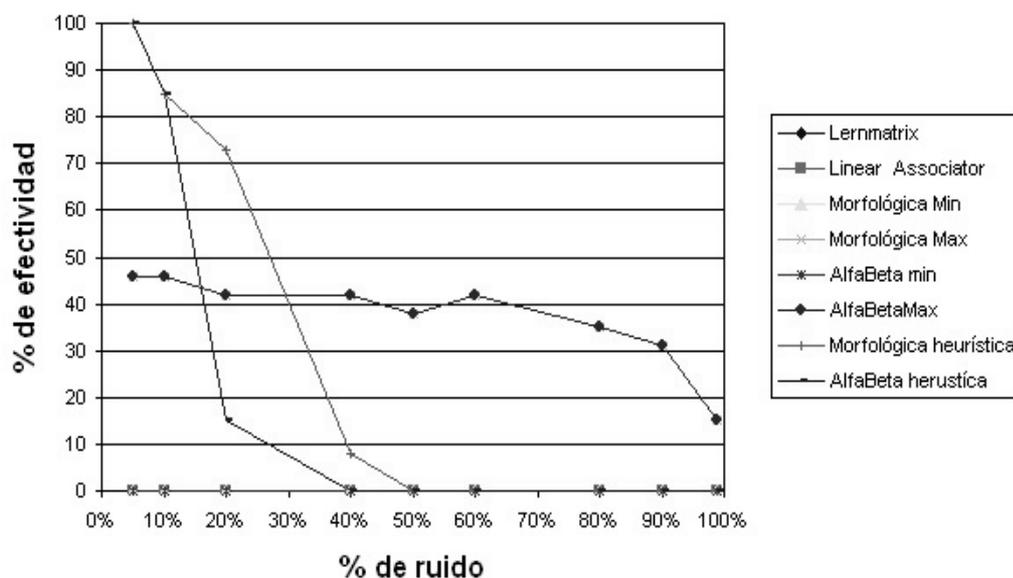
gráfica 5.24 Total de patrones recuperados en el conjunto fundamental 4 alterado con ruido sustractivo.

Se puede observar en la tabla 5.24 y en la gráfica 5.24 que las memorias de tipo min son robustas al ruido sustractivo, aún por encima de las memorias heurísticas.

|                 | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|-----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min       | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max       | 46  | 46   | 42   | 42   | 38   | 42   | 35   | 31   | 15   |
| AlfaBeta min    | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max    | 46  | 46   | 42   | 42   | 38   | 42   | 35   | 31   | 15   |
| Morf.heurística | 100 | 85   | 73   | 8    | 0    | 0    | 0    | 0    | 0    |
| AB heurística   | 100 | 85   | 15   | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.25. Porcentaje de efectividad en la recuperación de patrones del conjunto fundamental 4 afectado con ruido sustractivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido sustractivo

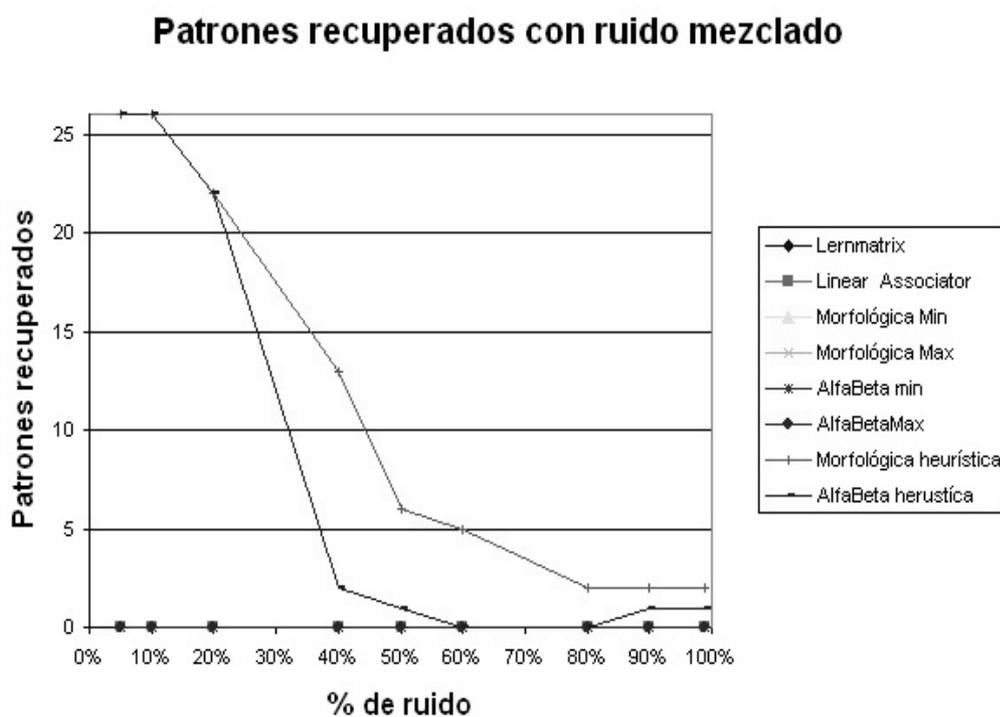


gráfica 5.25. Porcentaje de efectividad en la recuperación de patrones del conjunto fundamental 4 afectado con ruido sustractivo.

El conjunto fundamental 4 afectado por ruido mezclado, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 26   | 22   | 13   | 6    | 5    | 2    | 2    | 2    |
| AB heurística  | 26  | 26   | 22   | 2    | 1    | 0    | 0    | 1    | 1    |

tabla 5.26. Total de patrones recuperados en el conjunto fundamental 4 afectado con ruido mezclado.



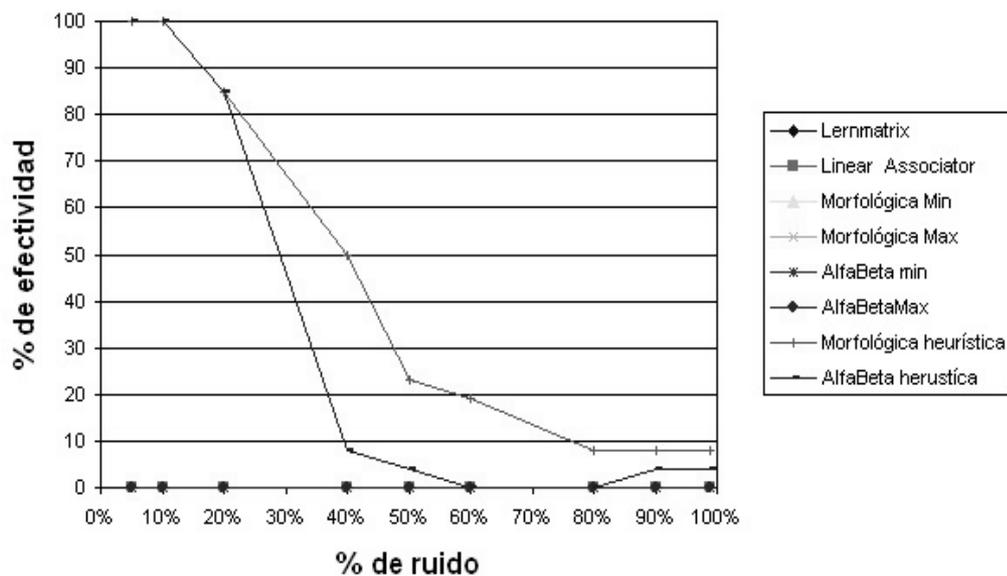
gráfica 5.26. Total de patrones recuperados en el conjunto fundamental 4 afectado con ruido mezclado.

Para el conjunto fundamental 4 se manifiesta lo que se ha venido corroborando, que las memorias heurísticas han tenido recuperaciones perfectas de patrones afectados con un alto porcentaje de ruido mezclado.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 85   | 50   | 23   | 19   | 8    | 8    | 8    |
| AB heurística  | 100 | 100  | 85   | 8    | 4    | 0    | 0    | 4    | 4    |

tabla 5.27. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 4 afectado con ruido mezclado.

### Porcentaje de efectividad de las memorias heteroasociativas mezclado



gráfica 5.27. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 4 afectado con ruido mezclado.

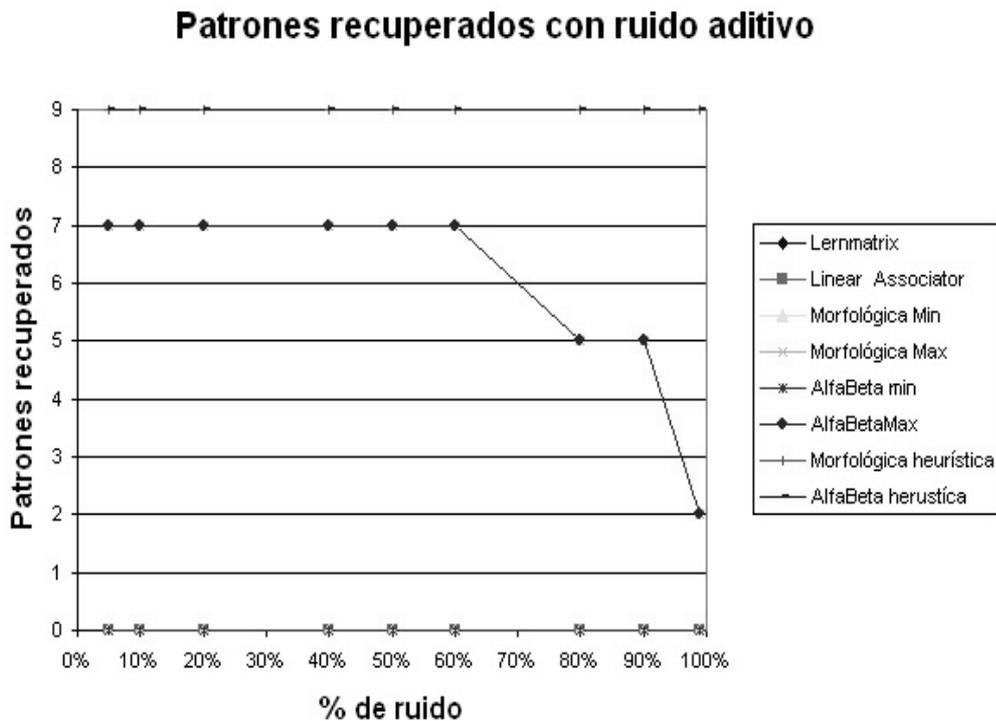
### Conjunto fundamental 5.

Este conjunto fundamental es importante porque se estará tratando con tan solo nueve patrones fundamentales. Las tablas y las gráficas que se mostrarán más adelante, describirán el desempeño de las memorias con pocos patrones.

El conjunto fundamental 5 afectado por ruido aditivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 7   | 7    | 7    | 7    | 7    | 7    | 5    | 5    | 2    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 7   | 7    | 7    | 7    | 7    | 7    | 5    | 5    | 2    |
| Mor.heurística | 9   | 9    | 9    | 9    | 9    | 9    | 9    | 9    | 9    |
| AB heurística  | 9   | 9    | 9    | 9    | 9    | 9    | 9    | 9    | 9    |

tabla 5.28. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido aditivo.

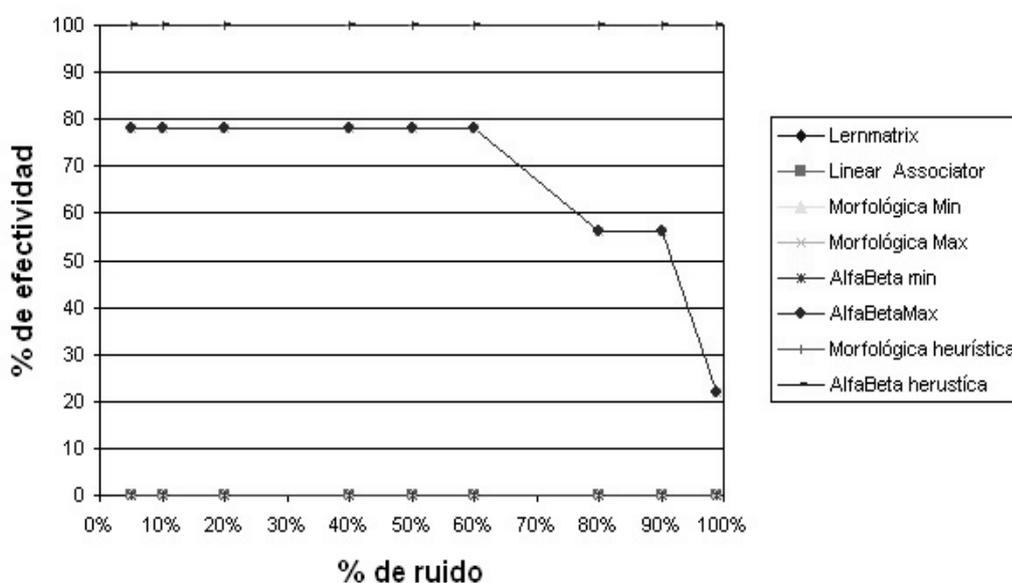


gráfica 5.28. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido aditivo.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 78  | 78   | 78   | 78   | 78   | 78   | 56   | 56   | 22   |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 78  | 78   | 78   | 78   | 78   | 78   | 56   | 56   | 22   |
| Mor.heurística | 100 | 100  | 100  | 100  | 100  | 100  | 100  | 100  | 100  |
| AB heurística  | 100 | 100  | 100  | 100  | 100  | 100  | 100  | 100  | 100  |

tabla 5.29. Porcentaje de efectividad de las memorias heteroasociativas para el conjunto fundamental 5 alterado con ruido aditivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido aditivo

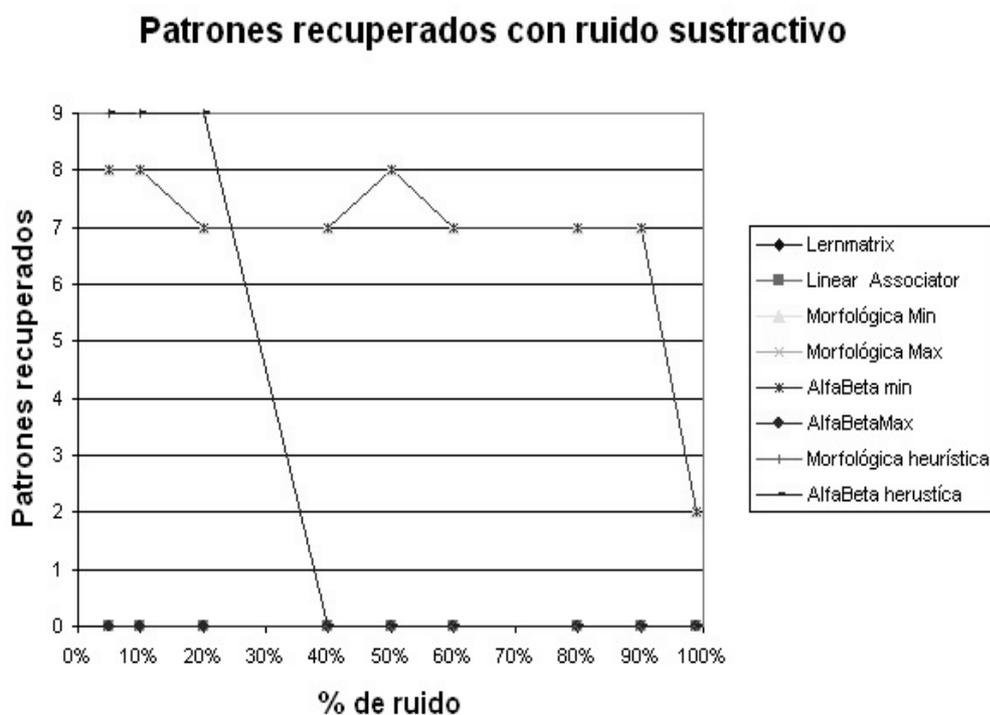


gráfica 5.29. Porcentaje de efectividad de las memorias heteroasociativas para el conjunto fundamental 5 alterado con ruido aditivo.

El conjunto fundamental 5 afectado por ruido sustractivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 8   | 8    | 7    | 7    | 8    | 7    | 7    | 7    | 2    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 8   | 8    | 7    | 7    | 8    | 7    | 7    | 7    | 2    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 9   | 9    | 9    | 0    | 0    | 0    | 0    | 0    | 0    |
| AB heurística  | 9   | 9    | 9    | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.30. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido sustractivo.



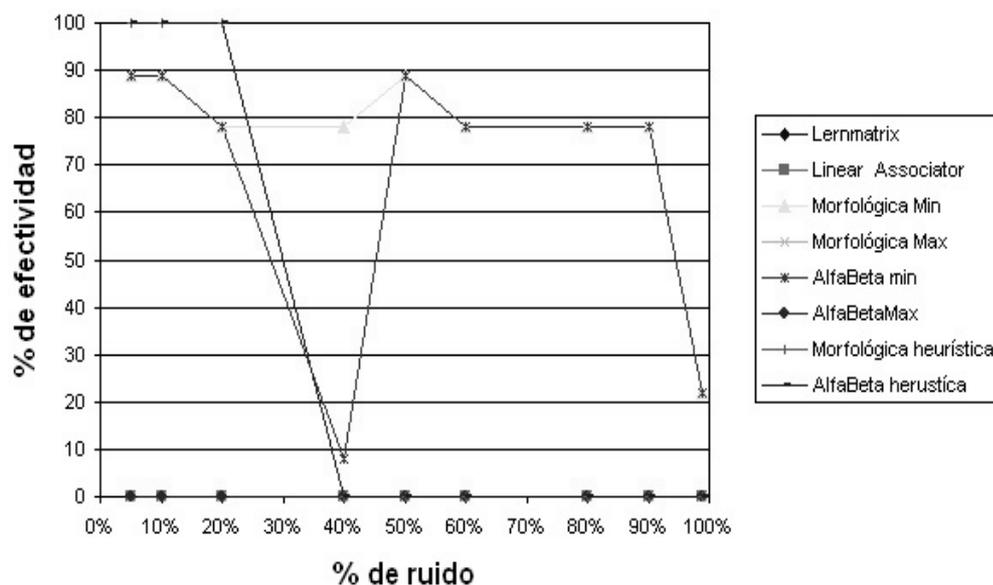
gráfica 5.30. Total de patrones recuperados en el conjunto fundamental 5 afectado por ruido sustractivo.

Las memorias de tipo min han estado mostrando que son muy robustas ante el ruido sustractivo.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 89  | 89   | 78   | 78   | 89   | 78   | 78   | 78   | 22   |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 89  | 89   | 78   | 78   | 89   | 78   | 78   | 78   | 22   |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 100  | 100  | 0    | 0    | 0    | 0    | 0    | 0    |
| AB heurística  | 100 | 100  | 100  | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.31. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 5 afectado por ruido sustractivo.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido sustractivo

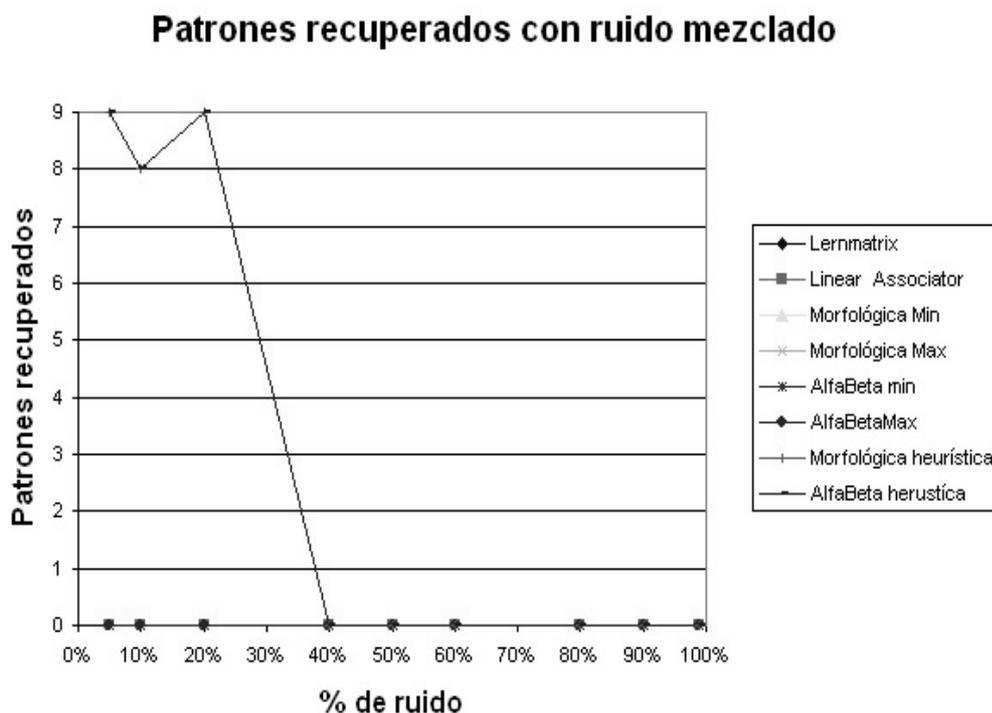


grafica 5.31. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 5 afectado por ruido sustractivo.

El conjunto fundamental 5 afectado por ruido mezclado, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 9   | 8    | 9    | 0    | 0    | 0    | 0    | 0    | 0    |
| AB heurística  | 9   | 8    | 9    | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.32. Total de patrones recuperados en el conjunto fundamental 5 afectado con ruido mezclado.



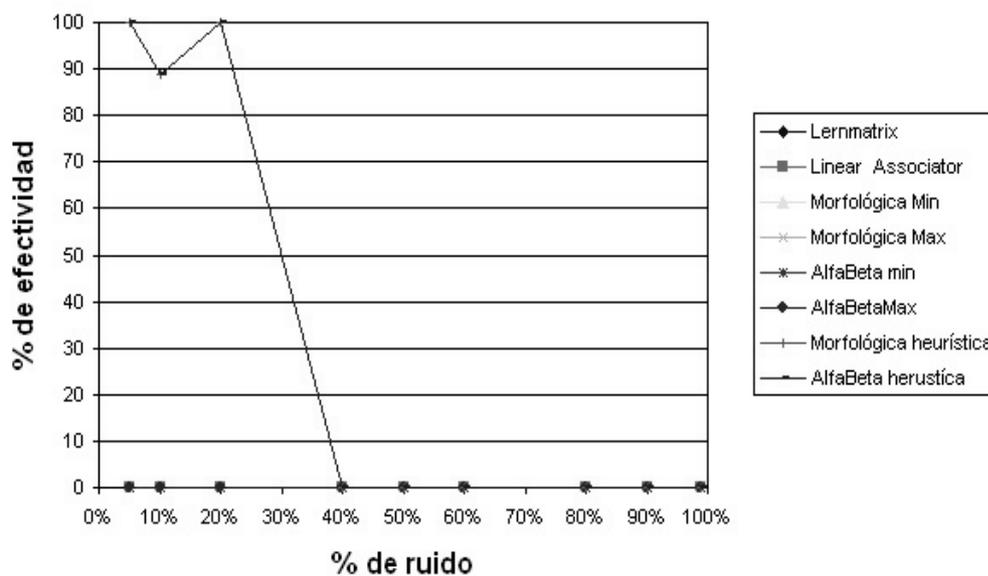
gráfica 5.32. Total de patrones recuperados en el conjunto fundamental 5 afectado con ruido mezclado.

Las memorias heurísticas nuevamente mostraron su efectividad en este conjunto fundamental.

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Lernmatrix     | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 100 | 89   | 100  | 0    | 0    | 0    | 0    | 0    | 0    |
| AB heurística  | 100 | 89   | 100  | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.33. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 5 alterado con ruido mezclado.

### Porcentaje de efectividad de las memorias heteroasociativas con ruido mezclado



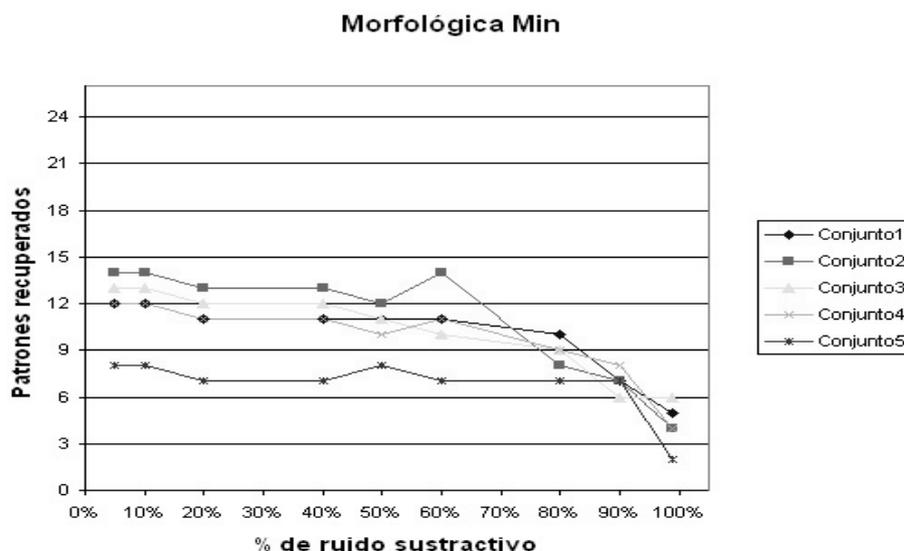
gráfica 5.33. Porcentaje de efectividad en la recuperación de patrones para el conjunto fundamental 5 alterado con ruido mezclado.

#### 5.1.4. Comportamiento de las memorias heteroasociativas por tipo

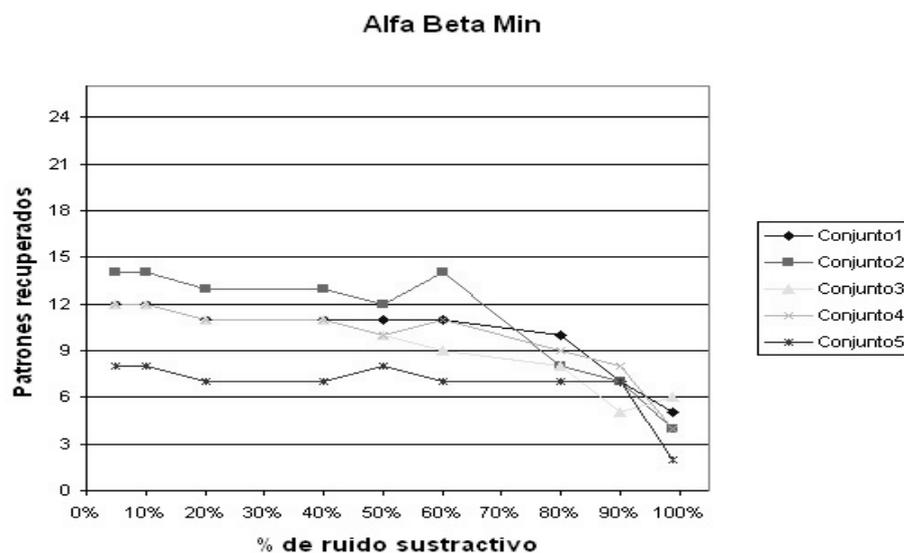
La finalidad de esta sección es mostrar el resumen del comportamiento que tuvieron las memorias heteroasociativas de manera individual, ante los diferentes conjuntos fundamentales que se han usado en las comparaciones experimentales. Las siguientes gráficas ilustran dicho resumen; estas gráficas estarán presentadas de la siguiente manera: memorias de tipo min, memorias de tipo max y memorias heurísticas. Se organizaron de esta manera las gráficas para que el lector pueda observar el comportamiento de las memorias de una manera sencilla.

**Nota:**

Las memorias heurísticas, aunque obtuvieron buenos resultados en la recuperación de patrones ante ruido aditivo y sustractivo no se mostrarán los resultados de éstas en las siguientes gráficas. Si desea ver las gráficas del comportamiento de estas memorias ante estos tipos de ruido por favor busque en el CD anexo en la ruta \Resultados.



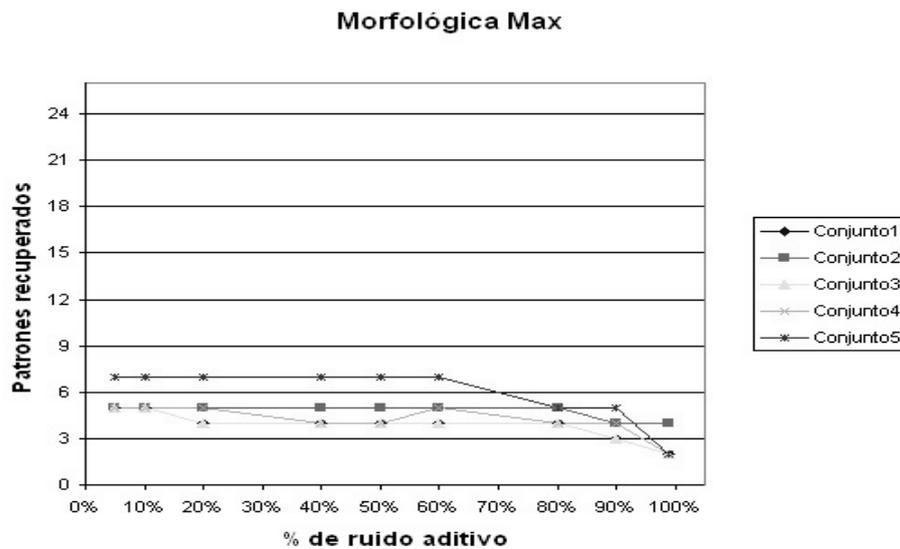
gráfica 5.34. Total de patrones recuperados por la memoria morfológica min en todos los conjuntos fundamentales.



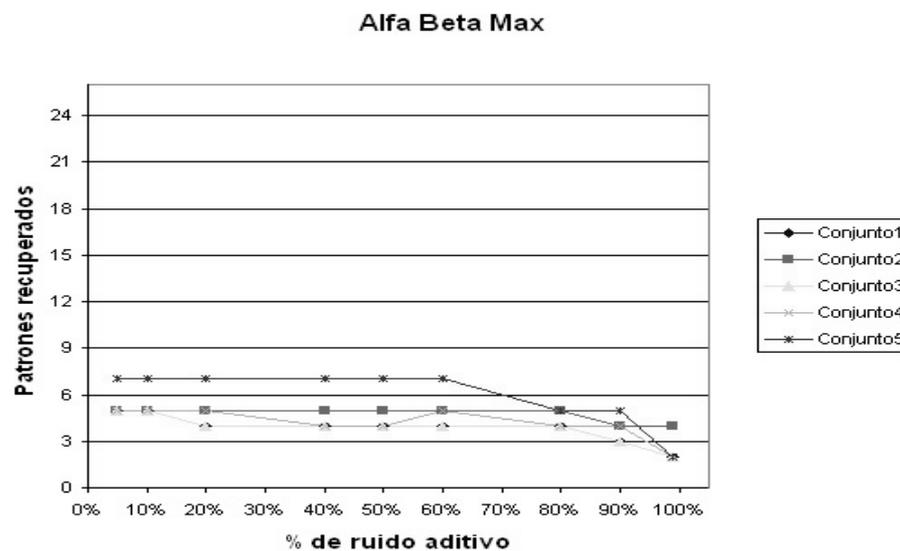
gráfica 5.35. Total de patrones recuperados por la memoria Alfa Beta min en todos los conjuntos fundamentales.

El desempeño de las memorias de tipo  $\bigwedge$  son muy similares; las gráficas ilustran que en el conjunto fundamental 3 la memoria morfológica min tuvo mejor desempeño

que la memoria  $\alpha\beta$  min. Es muy importante destacar este hecho, ya que aunque estas dos memorias se basan en el operador  $\bigwedge$ , esto no indica que deban obtener siempre el mismo resultado; como ya se mencionó en el capítulo 2 y al inicio de este capítulo, estas memorias son diferentes.

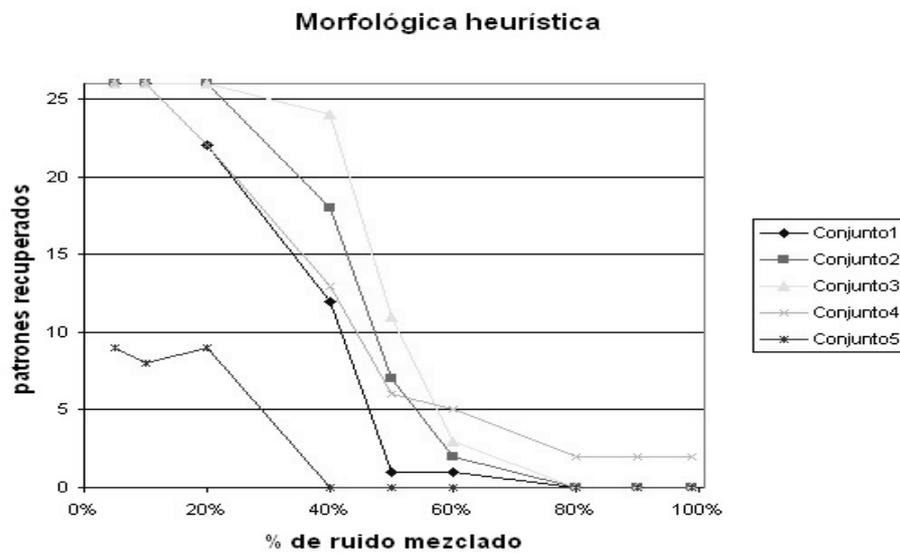


gráfica 5.36. Total de patrones recuperados por la memoria morfológica max en todos los conjuntos fundamentales.

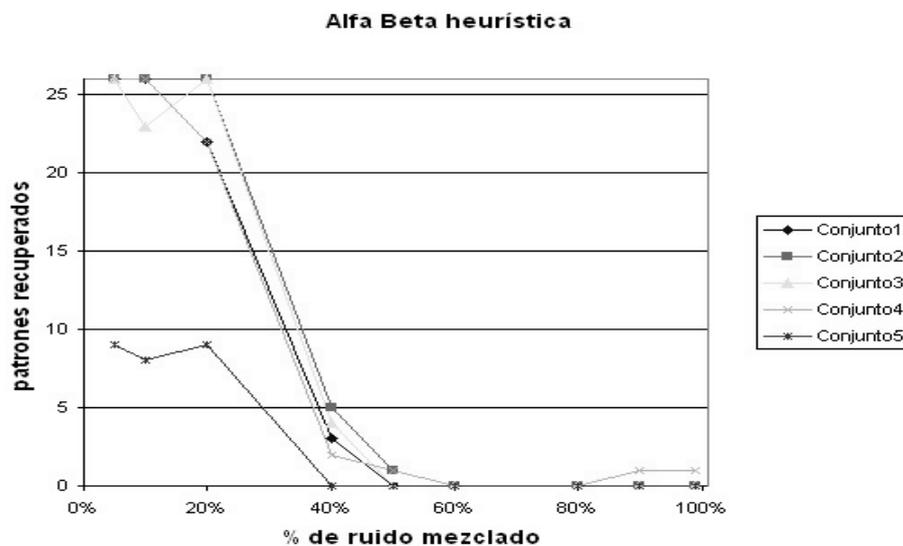


gráfica 5.37. Total de patrones recuperados por la memoria Alfa Beta max en todos los conjuntos fundamentales.

En las memorias de tipo  $\bigvee$  tanto la memoria morfológica como la memoria  $\alpha\beta$  obtuvieron los mismos resultados.



gráfica 5.38. Total de patrones recuperados por la memoria morfológica heurística en todos los conjuntos fundamentales.



gráfica 5.39. Total de patrones recuperados por la memoria Alfa Beta heurística en todos los conjuntos fundamentales.

Las gráficas anteriores, ilustran que las memoria morfológica heurística tuvo mejor desempeño que la memoria Alfa Beta heurística prácticamente en todos los conjuntos fundamentales.

### 5.1.5. Resultados obtenidos con las memorias heteroasociativas cuando $m \neq n$

Hasta el momento las memorias de tipo min, de tipo max y las heurísticas han recuperado patrones para el tipo de ruido en que son especialistas; sin embargo, la

memoria *Lernmatrix* de Steinbuch y la memoria *Linear Associator* de Anderson & Kohonen no han mostrado nada interesante en las comparaciones con las otras memorias. En esta sección se probará a todas las memorias en condiciones en las cuales la memoria *Lernmatrix* y la memoria *Linear Associator* son buenas.

Los conjuntos fundamentales son los mismos cinco que se han venido usando para las disquisiciones experimentales para los patrones  $x^\mu$  de entrada que se han usado y los patrones  $y^\mu$  de salida se conforman de un vector de 26 elementos, a excepción del conjunto fundamental 5 ya que sólo se cuenta con 9 patrones. Los valores de los patrones de salida  $y^\mu$  son "0" con la excepción de la posición cuyo  $y_i^\mu, i = 1, 2, \dots, n$  le corresponda, tendrá el valor "1", por ejemplo:

$$\begin{array}{l} \left[ \begin{array}{c} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array} \right] ; \text{ es el } y^1 \text{ que le corresponde al primer patrón } x^1 \\ \left[ \begin{array}{c} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{array} \right] ; \text{ es el } y^2 \text{ que le corresponde al segundo patrón } x^2 \\ \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{array} \right] ; \text{ es el } y^\mu \text{ que le corresponde al último patrón } x^\mu \end{array}$$

Los resultados obtenidos fueron generados por los conjuntos fundamentales sin ruido. Los resultados son los siguientes:

|                   | Conj.1 | Conj.2 | Conj.3 | Conj.4 | Conj.5 |
|-------------------|--------|--------|--------|--------|--------|
| Lernmatrix        | 25     | 26     | 25     | 26     | 9      |
| Linear Associator | 0      | 0      | 0      | 0      | 0      |
| Morfológica min   | 6      | 6      | 6      | 7      | 9      |
| Morfológica max   | 26     | 26     | 26     | 26     | 9      |
| Alfa Beta min     | 6      | 6      | 6      | 7      | 9      |
| Alfa Beta max     | 26     | 26     | 26     | 26     | 9      |

tabla 5.34 a. Total de patrones recuperados por las memorias heteroasociativas cuando  $m \neq n$ .

|                   | Conj.1 | Conj.2 | Conj.3 | Conj.4 | Conj.5 |
|-------------------|--------|--------|--------|--------|--------|
| Lernmatrix        | 0      | 0      | 0      | 0      | 0      |
| Linear Associator | 26     | 26     | 26     | 26     | 9      |
| Morfológica min   | 26     | 26     | 26     | 26     | 9      |
| Morfológica max   | 2      | 2      | 2      | 2      | 2      |
| Alfa Beta min     | 26     | 26     | 26     | 26     | 9      |
| Alfa Beta max     | 2      | 2      | 2      | 2      | 2      |

tabla 5.34 b. Total de patrones recuperados por las memorias heteroasociativas cuando  $m \neq n$ .

Cuando las condiciones son idóneas para el funcionamiento de todos los modelos de memorias asociativas, como resultado éstas responden con un excelente desempeño. La realidad ilustra que las dos primeras memorias que se encuentran en la tabla 5.34, aunque están en sus condiciones ideales, no superan a las memorias de tipo min y de tipo max; esto indica que éstas últimas también se encuentran en sus condiciones ideales.

## 5.2. Estudio comparativo experimental para los modelos de memorias autoasociativas

En esta sección quedarán plasmados los resultados obtenidos de los diferentes modelos de memorias autoasociativas ante las mismas condiciones de aprendizaje y recuperación.

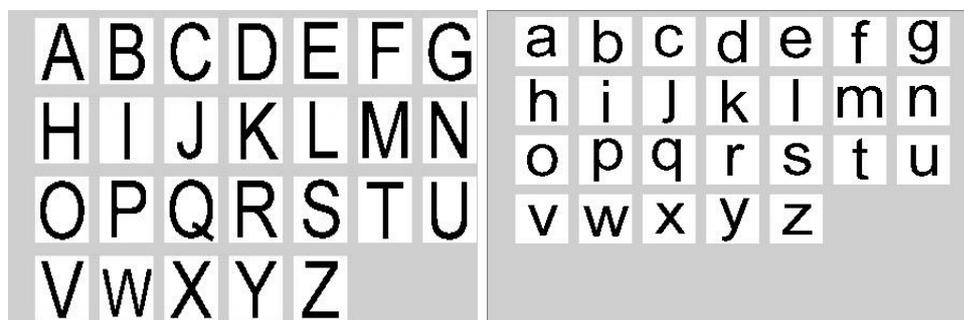
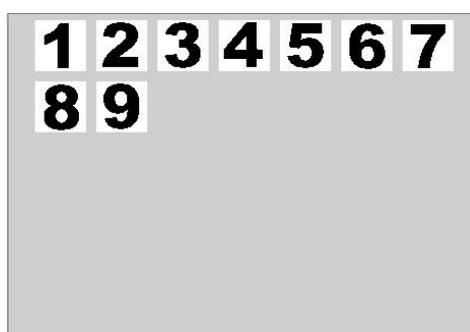
Una memoria es autoasociativa si se cumple que:  $x^\mu = y^\mu \forall \mu \in 1, 2, \dots, p$ . El conjunto fundamental  $(x^\omega, y^\omega) \mid \omega \in \{1, 2, \dots, p\}$ , donde  $p = 26$ , a excepción del conjunto fundamental 5, donde  $p = 9$ , han quedado definidos como:

Los cinco conjuntos fundamentales están definidos de la siguiente manera:



Conjunto fundamental  $x^1$

Conjunto fundamental  $x^2$

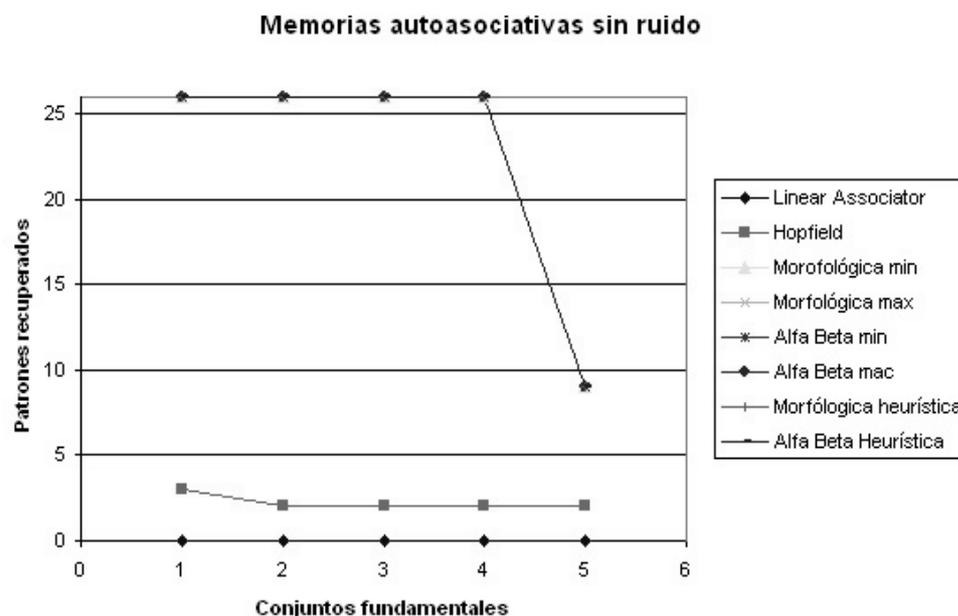
Conjunto fundamental  $x^3$ Conjunto fundamental  $x^4$ Conjunto fundamental  $x^5$ 

### 5.2.1. Resultados obtenidos con las memorias autoasociativas sin ruido

Las tablas y las gráficas que se mostrarán en esta sección, presentarán los resultados obtenidos de las memorias autoasociativas referentes a los patrones recuperados y porcentajes de efectividad que tuvieron las memorias en el proceso de recuperación de patrones.

|                   | Conj.1 | Conj.2 | Conj.3 | Conj.4 | Conj.5 |
|-------------------|--------|--------|--------|--------|--------|
| Linear Associator | 0      | 0      | 0      | 0      | 0      |
| Hopfield          | 3      | 2      | 2      | 2      | 2      |
| Morfológica min   | 26     | 26     | 26     | 26     | 9      |
| Morfológica max   | 26     | 26     | 26     | 26     | 9      |
| Alfa Beta min     | 26     | 26     | 26     | 26     | 9      |
| Alfa Beta max     | 26     | 26     | 26     | 26     | 9      |
| Morf. heurist.    | 26     | 26     | 26     | 26     | 9      |
| AB heurist.       | 26     | 26     | 26     | 26     | 9      |

tabla 5.35. Recuperación de patrones por conjunto fundamental sin ruido.



gráfica 5.40. Total de patrones recuperados por memorias autoasociativa para los conjuntos fundamentales sin ruido.

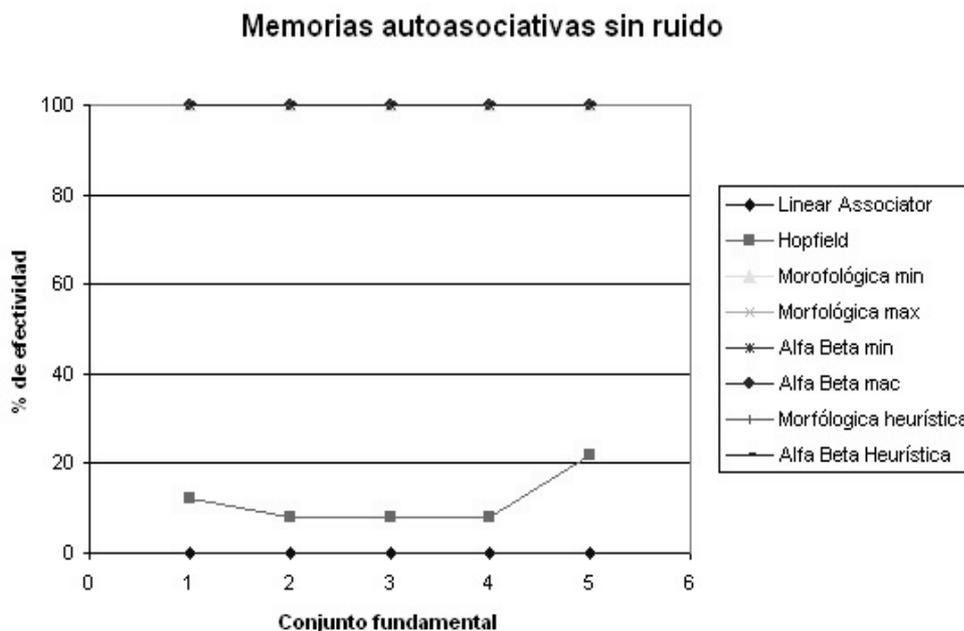
Los resultados que se ilustran en la gráfica 5.40 y la tabla 5.35 ilustran que las memorias morfológicas,  $\alpha\beta$  y heurísticas recuperaron todos los patrones aprendidos. Hopfield, por su lado solo recuperó 3 patrones en el conjunto fundamental 1 y 2 patrones en todos los demás conjuntos, es importante resaltar que la memoria Hopfield solo es capaz de recuperar solo los 2 primeros patrones de manera perfecta, después de éstos tiene muchos problemas para la recuperación. La figura 5.7 ilustra la salida a partir del quinto patrón aprendido e intentados recuperar para el conjunto fundamental 1.



figura 5.7. Apariencia de los 5 patrones recuperados por la memoria Hopfield.

|                | Conj.1 | Conj.2 | Conj.3 | Conj.4 | Conj.5 |
|----------------|--------|--------|--------|--------|--------|
| Linear Assoc.  | 0      | 0      | 0      | 0      | 0      |
| Hopfield       | 12     | 8      | 8      | 8      | 22     |
| Morf. min      | 100    | 100    | 100    | 100    | 100    |
| Morf.max       | 100    | 100    | 100    | 100    | 100    |
| Alfa Beta min  | 100    | 100    | 100    | 100    | 100    |
| Alfa Beta max  | 100    | 100    | 100    | 100    | 100    |
| Morf. heurist. | 100    | 100    | 100    | 100    | 100    |
| AB heurist.    | 100    | 100    | 100    | 100    | 100    |

tabla 5.36. Porcentaje de efectividad de recuperación de patrones por las memorias autoasociativas para todos los conjuntos fundamentales.



gráfica 5.41. Porcentaje de efectividad de recuperación de patrones obtenido por las memorias autoasociativas para todos los conjuntos fundamentales.

Los resultados mostrados en la gráfica 5.41 y la tabla 5.36 son contundentes, las memorias autoasociativas de tipo  $\wedge$ ,  $\vee$  y las heurísticas recuperaron todos los patrones sin ruido.

### 5.2.2. Resultados obtenidos con las memorias autoasociativas con ruido

En esta sección se presentará el comportamiento de las memorias autoasociativas ante ruido aditivo, sustractivo y mezclado. Los porcentajes de ruido con los que se afectarán a los patrones son: 5, 10, 20, 40, 50, 60, 80, 90 y 99% respectivamente. La apariencia de los patrones afectados con algún tipo de ruido para las memorias autoasociativas es la misma que para las heteroasociativas (ver figura 5.4).

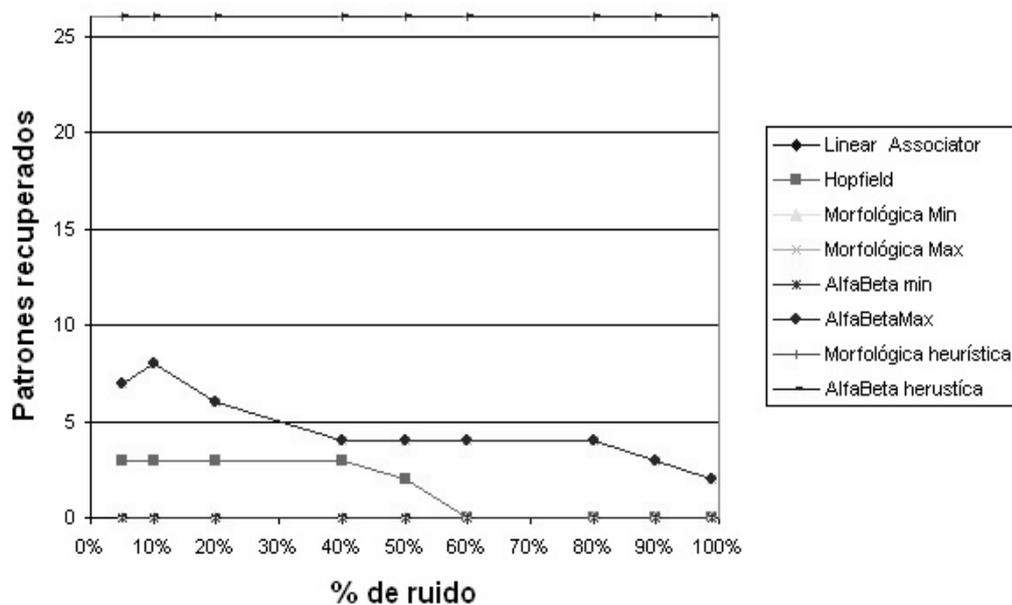
## Conjunto fundamental 1

El conjunto fundamental 1 afectado por ruido aditivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Hopfield       | 3   | 3    | 3    | 3    | 2    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 7   | 8    | 6    | 4    | 4    | 4    | 4    | 3    | 2    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 7   | 8    | 6    | 4    | 4    | 4    | 4    | 3    | 2    |
| Mor.heurística | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |
| AB heurística  | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |

tabla 5.37. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido aditivo.

## Patrones recuperados con ruido aditivo



gráfica 5.42. Total de patrones recuperados por las memorias autoasociativas para el conjunto fundamental 1 afectado con ruido aditivo.

En los resultados obtenidos para el ruido aditivo en las memorias heteroasociativas de tipo max, se comentó que los patrones recuperados de manera no perfecta contenían

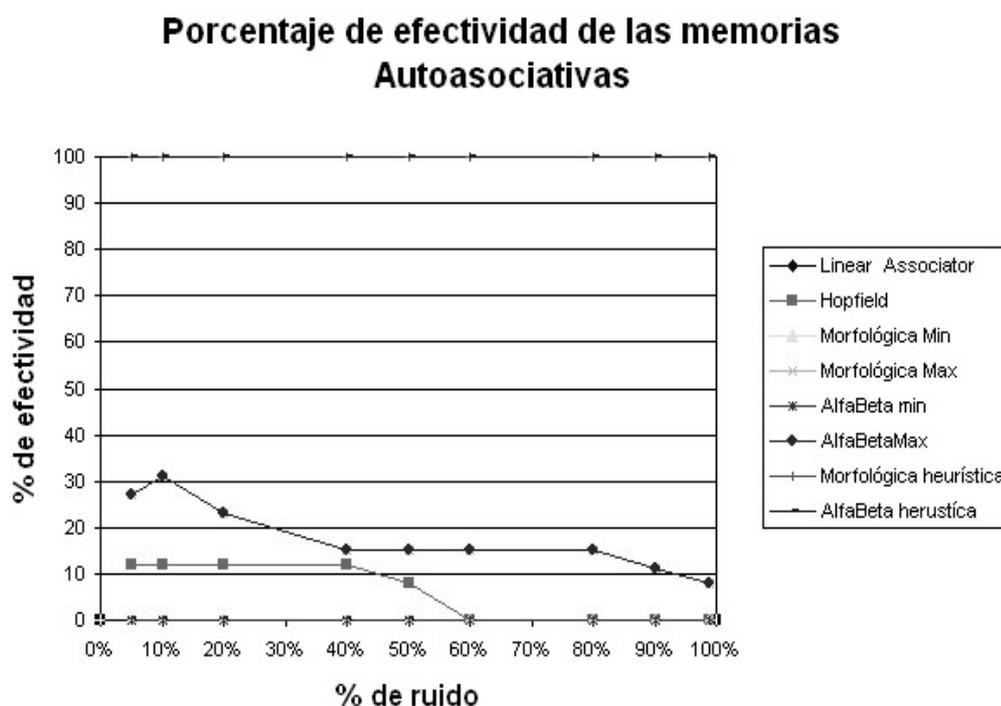
las características del patrón en cuestión intentado recuperar, solo que lo recuperaba con ruido aditivo, en el caso de las memorias autoasociativas de tipo max, éstas están en la misma condición. La figura 5.8 ilustra a 10 patrones recuperados con un total de 50% de ruido aditivo para el conjunto fundamental 1.



figura 5.8. 10 patrones recuperados que estaban afectados con el 50% de ruido aditivo.

La figura 5.8 es muy clara, ya que efectivamente son los patrones aprendidos pero afectados con un bajo porcentaje de ruido aditivo.

Es importante mencionar el porcentaje de efectividad de recuperación de patrones, con base al total de patrones aprendidos por memoria asociativa. La siguiente gráfica 5.43 ilustra este porcentaje.



gráfica 5.43. Porcentaje de efectividad en la recuperación de patrones afectados con ruido aditivo.

El conjunto fundamental 1 afectado por ruido sustractivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Hopfield       | 3   | 3    | 3    | 2    | 2    | 1    | 1    | 1    | 1    |
| Morf. min      | 15  | 17   | 12   | 10   | 10   | 12   | 8    | 8    | 5    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 15  | 17   | 12   | 10   | 10   | 12   | 8    | 8    | 5    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 24  | 24   | 14   | 0    | 0    | 0    | 0    | 0    | 0    |
| AB heurística  | 24  | 24   | 4    | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.38. Total de patrones recuperados para el conjunto fundamental 1 afectado con ruido sustractivo.

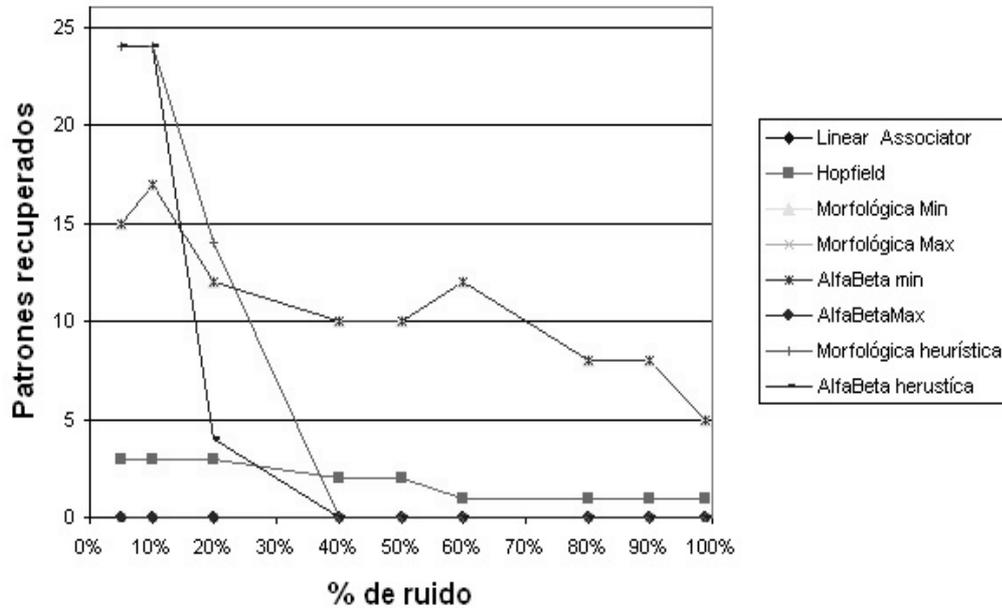
Las memorias autoasociativas de tipo min recuperaron patrones. La misma situación, en cuestión de recuperación, en las que se encontraron las memorias heteroasociativas de tipo min, se encuentran las memorias autoasociativas de tipo min, es decir, éstas últimas también recuperan el patrón en cuestión, solo que lo recuperan afectado con ruido sustractivo. La siguiente figura ilustra a 10 patrones afectados con un 90 % de ruido sustractivo después de recuperarlos.



figura 5.9 Apariencia de 10 patrones recuperados con las memorias de tipo min después de que los patrones estaban afectados con el 90 % de ruido sustractivo.

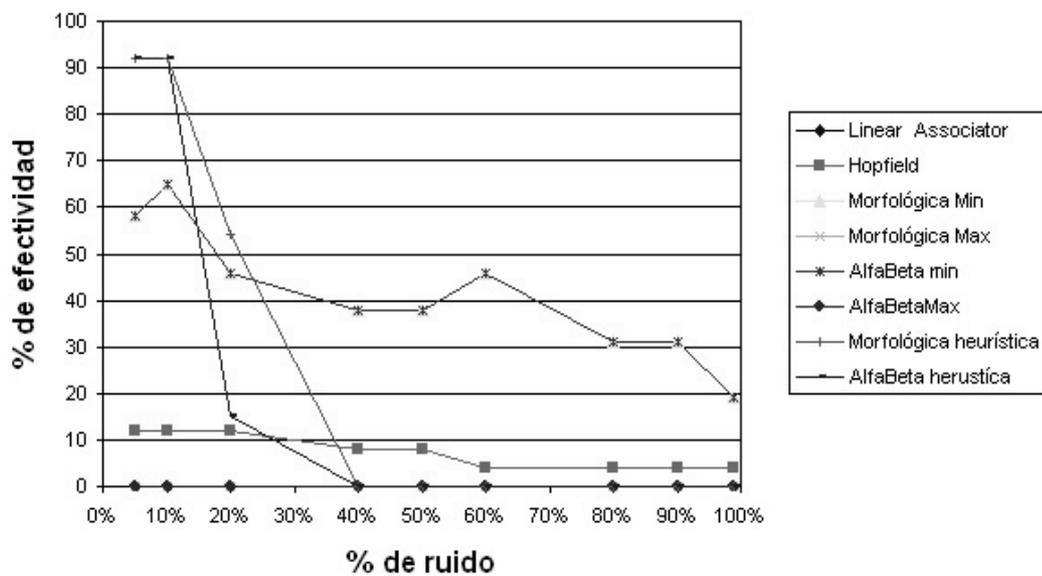
La figura 5.9 resume perfectamente lo comentado. En las memorias autoasociativas de tipo  $\wedge$  cuando la recuperación no es perfecta los patrones que son recuperados contienen un porcentaje de ruido sustractivo muy bajo.

### Patrones recuperados con ruido sustractivo



gráfica 5.44. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido sustractivo.

### Porcentaje de efectividad de las memorias Autoasociativas con ruido sustractivo



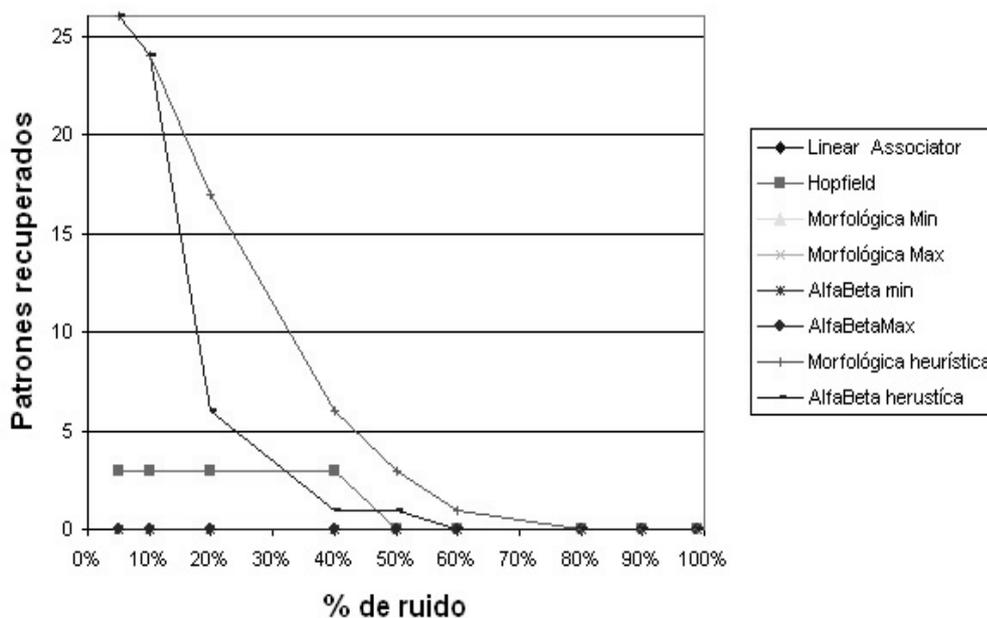
gráfica 5.45. Porcentaje de la efectividad de recuperación de patrones ante ruido sustractivo para el conjunto fundamental 1.

El conjunto fundamental 1 afectado por ruido mezclado, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Hopfield       | 3   | 3    | 3    | 3    | 0    | 0    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Mor.heurística | 26  | 24   | 17   | 6    | 3    | 1    | 0    | 0    | 0    |
| AB heurística  | 26  | 24   | 6    | 1    | 1    | 0    | 0    | 0    | 0    |

tabla 5.39. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido mezclado.

### Patrones recuperados con ruido mezclado



gráfica 5.46. Total de patrones recuperados en el conjunto fundamental 1 afectado con ruido mezclado.

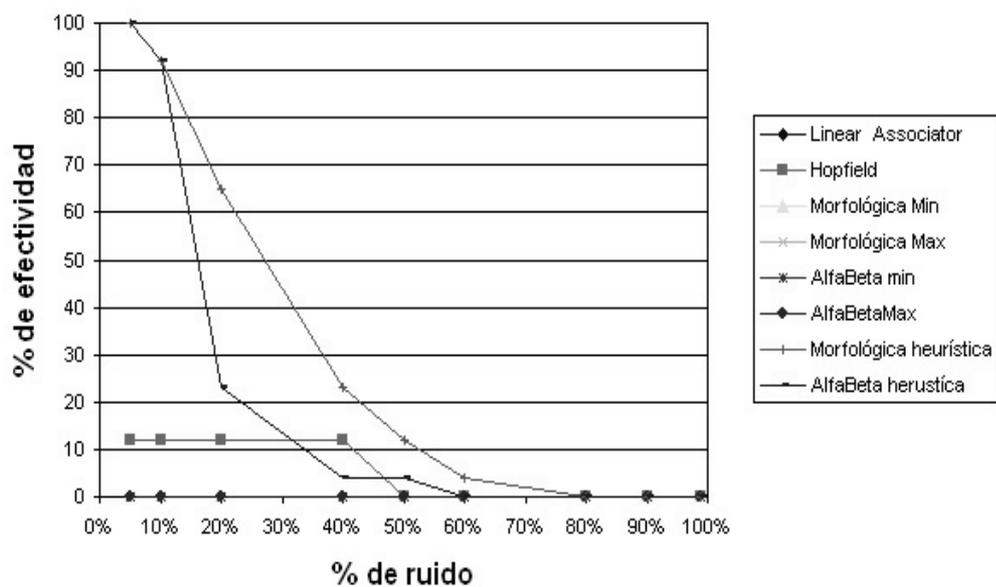
La tabla 5.39 y la gráfica 5.46 ilustran que además de las memorias heurísticas, la memoria Hopfield fue capaz de recuperar patrones alterados con ruido mezclado. La figura 5.10 ilustra la apariencia que tienen los patrones recuperados por la memoria Hopfield después de 4 patrones aprendidos.



figura 5.10. Apariencia después de cuatro patrones recuperados por la memoria Hopfield

Hay que resaltar también el hecho de que la memoria Hopfield a partir del 5 patrón aprendido, en la fase de recuperación obtienen los patrones recuperados la misma apariencia mostrada en la figura 5.7 y al finalizar los intentos de recuperar los 26 patrones, ningún patrón recuperado se parece en nada al patrón aprendido.

### Porcentaje de efectividad de las memorias Autoasociativas con ruido mezclado



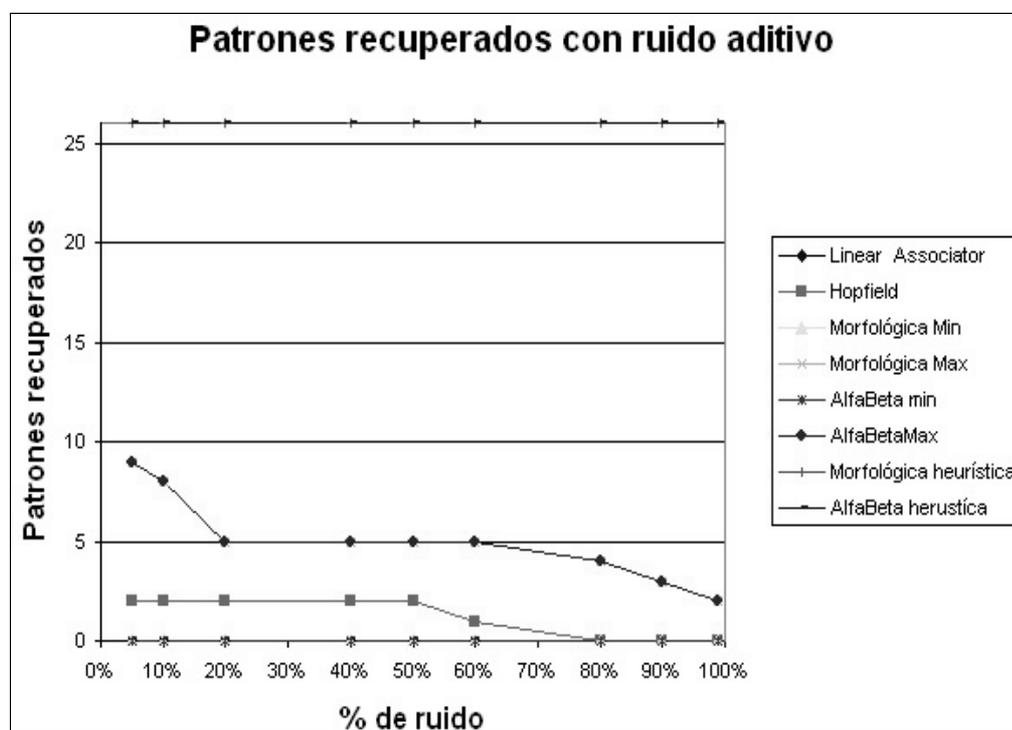
gráfica 5.47. Porcentaje de efectividad en la recuperación de patrones en el conjunto fundamental 1 afectado por ruido mezclado.

## Conjunto fundamental 2.

El conjunto fundamental 2 afectado por ruido aditivo, arrojó los siguientes resultados:

|                | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|----------------|-----|------|------|------|------|------|------|------|------|
| Linear Assoc.  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Hopfield       | 2   | 2    | 2    | 2    | 2    | 1    | 0    | 0    | 0    |
| Morf. min      | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Morf. max      | 9   | 8    | 5    | 5    | 5    | 5    | 4    | 3    | 2    |
| AlfaBeta min   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| AlfaBeta max   | 9   | 8    | 5    | 5    | 5    | 5    | 4    | 3    | 2    |
| Mor.heurística | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |
| AB heurística  | 26  | 26   | 26   | 26   | 26   | 26   | 26   | 26   | 26   |

tabla 5.40. Total de patrones recuperados en el conjunto fundamental 2 afectado con ruido aditivo.



gráfica 5.48. Total de patrones recuperados en el conjunto fundamental 2 afectado con ruido aditivo.

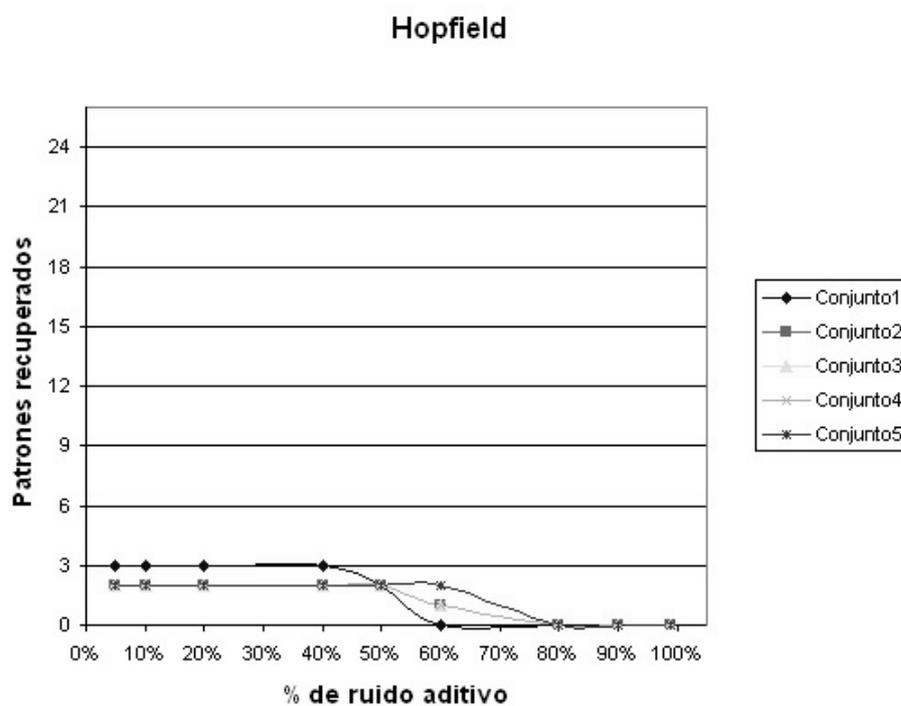
La tabla 5.40 y la gráfica 5.48 ilustran que el comportamiento de las memorias autoasociativas, tuvieron un desempeño muy similar al presentado en el conjunto fundamental 1. En realidad estas memorias autoasociativas tendrán la misma tendencia en todos los conjuntos fundamentales que las mostradas en el conjunto anterior. Hopfield mostrará que son solo 2 patrones que recuperará sin ningún problema; además Hopfield mostrará que es una excelente memoria aunque sea para solo dos patrones, debido a que no tiene ningún problema con el tipo de ruido que se le presente, aunque el porcentaje de ruido sea alto. Las memorias de tipo min seguirán siendo muy robustas al ruido sustractivo, las memorias del tipo max, serán robustas al ruido aditivo y las heurísticas resolverán el problema con el ruido mezclado, debido a estas tendencias se mostrará en las siguientes tablas y gráficas el total de patrones recuperados por memoria autoasociativa en los diferentes conjuntos fundamentales. Si el lector desea observar en forma individual los resultados obtenidos para las memorias autoasociativas por favor busque en el CD anexo a este trabajo en la ruta \Resultados\Consolidados\Autoasociativa.

El orden en que aparecerán los resultados obtenidos por las memorias autoasociativas es: memoria Hopfield, memorias de tipo  $\bigwedge$ , memorias de tipo  $\bigvee$ , y memorias heurísticas.

### Memoria Hopfield con ruido aditivo

|            | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|------------|-----|------|------|------|------|------|------|------|------|
| Conjunto 1 | 3   | 3    | 3    | 3    | 3    | 2    | 0    | 0    | 0    |
| Conjunto 2 | 2   | 2    | 2    | 2    | 2    | 1    | 0    | 0    | 0    |
| Conjunto 3 | 2   | 2    | 2    | 2    | 2    | 1    | 0    | 0    | 0    |
| Conjunto 4 | 2   | 2    | 2    | 2    | 2    | 1    | 0    | 0    | 0    |
| Conjunto 5 | 2   | 2    | 2    | 2    | 2    | 2    | 0    | 0    | 0    |

tabla 5.41. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.

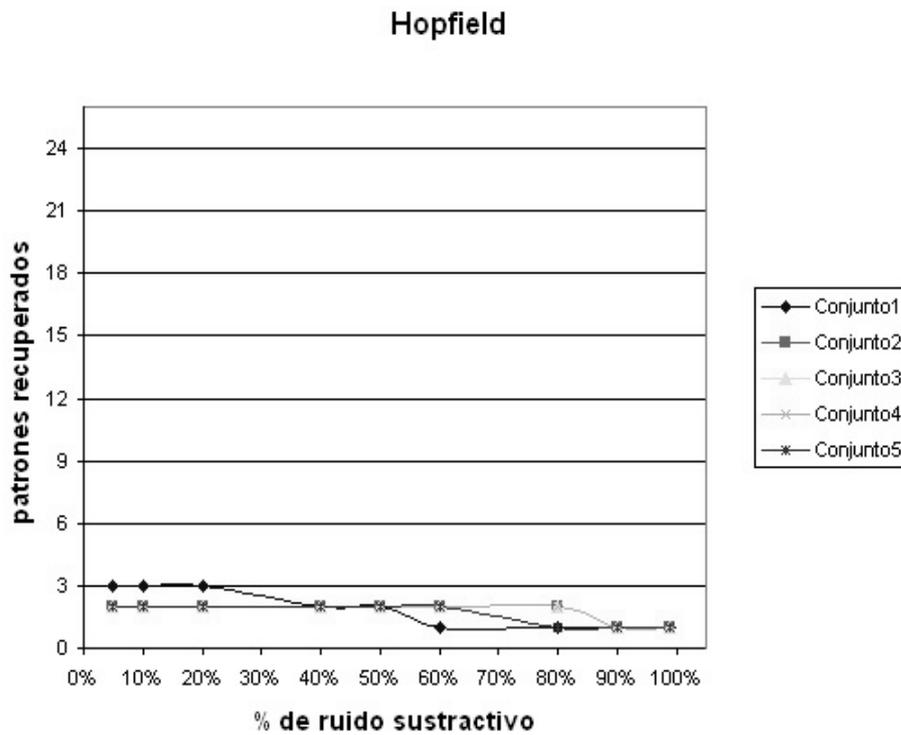


gráfica 5.49. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.

### Memoria Hopfield con ruido sustractivo

|            | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|------------|-----|------|------|------|------|------|------|------|------|
| Conjunto 1 | 3   | 3    | 3    | 2    | 2    | 1    | 1    | 1    | 1    |
| Conjunto 2 | 2   | 2    | 2    | 2    | 2    | 2    | 2    | 1    | 1    |
| Conjunto 3 | 2   | 2    | 2    | 2    | 2    | 2    | 2    | 1    | 1    |
| Conjunto 4 | 2   | 2    | 2    | 2    | 2    | 2    | 2    | 1    | 1    |
| Conjunto 5 | 2   | 2    | 2    | 2    | 2    | 2    | 1    | 1    | 1    |

tabla 5.42. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.

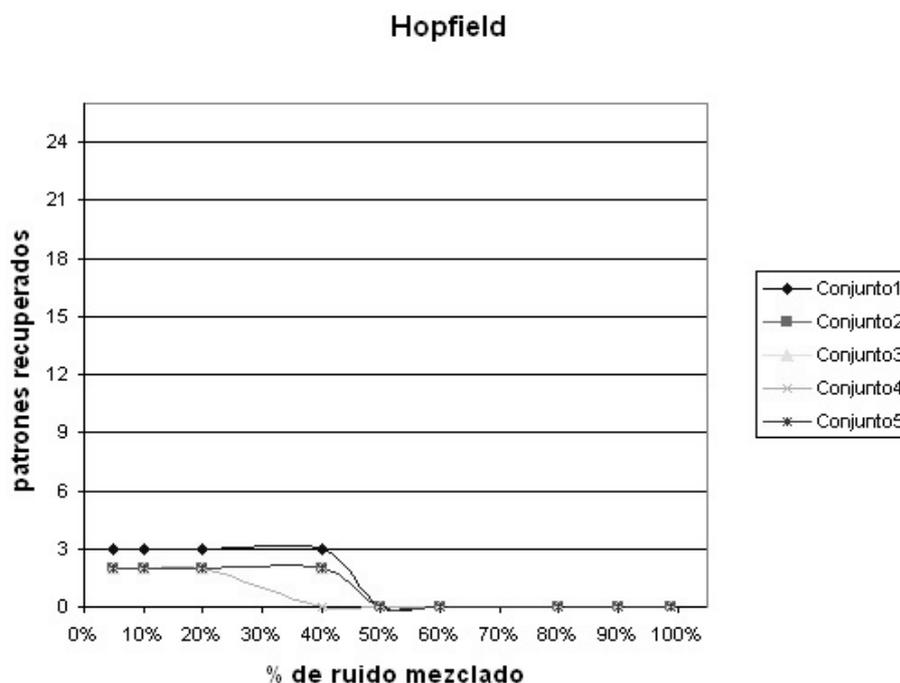


gráfica 5.50. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.

### Memoria Hopfield con ruido mezclado

|            | 5 % | 10 % | 20 % | 40 % | 50 % | 60 % | 80 % | 90 % | 99 % |
|------------|-----|------|------|------|------|------|------|------|------|
| Conjunto 1 | 3   | 3    | 3    | 3    | 0    | 0    | 0    | 0    | 0    |
| Conjunto 2 | 2   | 2    | 2    | 2    | 0    | 0    | 0    | 0    | 0    |
| Conjunto 3 | 2   | 2    | 2    | 2    | 0    | 0    | 0    | 0    | 0    |
| Conjunto 4 | 2   | 2    | 2    | 0    | 0    | 0    | 0    | 0    | 0    |
| Conjunto 5 | 2   | 2    | 2    | 2    | 0    | 0    | 0    | 0    | 0    |

tabla 5.43. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.



gráfica 5.51. Total de patrones recuperados por la memoria Hopfield en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.

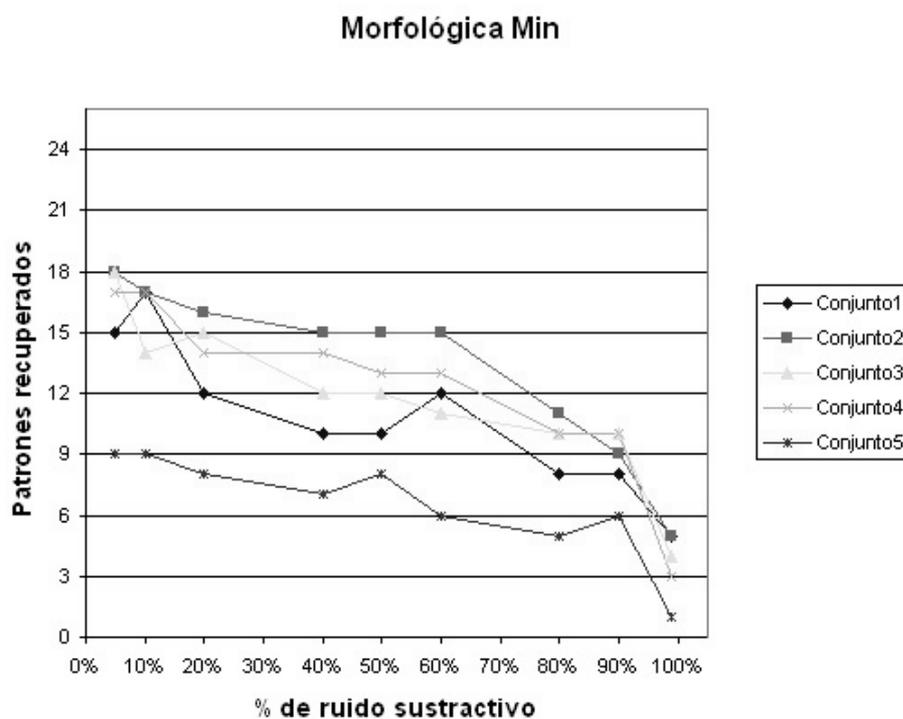
Como se puede observar en las tablas 5.41, 5.42, 5.43 y las gráficas 5.49, 5.50 y 5.51, la tendencia que tiene la memoria Hopfield es muy estable con los primeros 2 patrones afectados con cualquier tipo de ruido hasta con un 40%. Esta memoria ilustra que tiene mejores resultados con patrones afectados con ruido sustractivo, ante el ruido mezclado mostró menor desempeño con porcentajes mayores al 40%, la razón de este menor desempeño, como ya se había comentado anteriormente con relación a las memorias heurísticas, es que el ruido mezclado aunque es probabilístico esta condicionado de tal manera que a mayor porcentaje de ruido mezclado, el patrón se acerca más a su negativo, por lo tanto cuando se habla de patrones con un alto contenido de ruido mezclado se está hablando de un patrón que no se ha aprendido.

### Memorias de tipo $\wedge$

#### Morfológica $\wedge$

|            | 5% | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 99% |
|------------|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Conjunto 1 | 15 | 17  | 12  | 10  | 10  | 12  | 8   | 8   | 5   |
| Conjunto 2 | 18 | 17  | 16  | 15  | 11  | 11  | 11  | 9   | 5   |
| Conjunto 3 | 18 | 14  | 15  | 12  | 12  | 11  | 10  | 10  | 4   |
| Conjunto 4 | 17 | 17  | 14  | 14  | 13  | 13  | 10  | 10  | 3   |
| Conjunto 5 | 9  | 9   | 8   | 7   | 8   | 6   | 5   | 6   | 1   |

tabla 5.44. Total de patrones recuperados por la memoria morfológica min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.

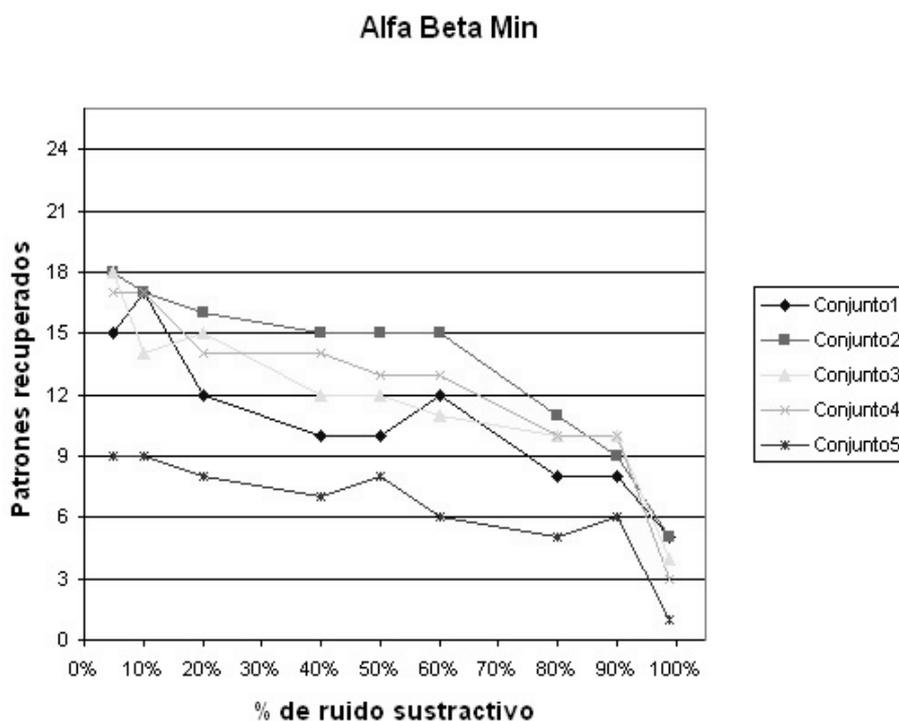


gráfica 5.52. Total de patrones recuperados por la memoria morfológica min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.

*Alfa Beta*  $\wedge$

|            | 5% | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 99% |
|------------|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Conjunto 1 | 15 | 17  | 12  | 10  | 10  | 12  | 8   | 8   | 5   |
| Conjunto 2 | 18 | 17  | 16  | 15  | 11  | 11  | 11  | 9   | 5   |
| Conjunto 3 | 18 | 14  | 15  | 12  | 12  | 11  | 10  | 10  | 4   |
| Conjunto 4 | 17 | 17  | 14  | 14  | 13  | 13  | 10  | 10  | 3   |
| Conjunto 5 | 9  | 9   | 8   | 7   | 8   | 6   | 5   | 6   | 1   |

tabla 5.45. Total de patrones recuperados por la memoria Alfa Beta min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.



gráfica 5.53. Total de patrones recuperados por la memoria Alfa Beta min en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido sustractivo.

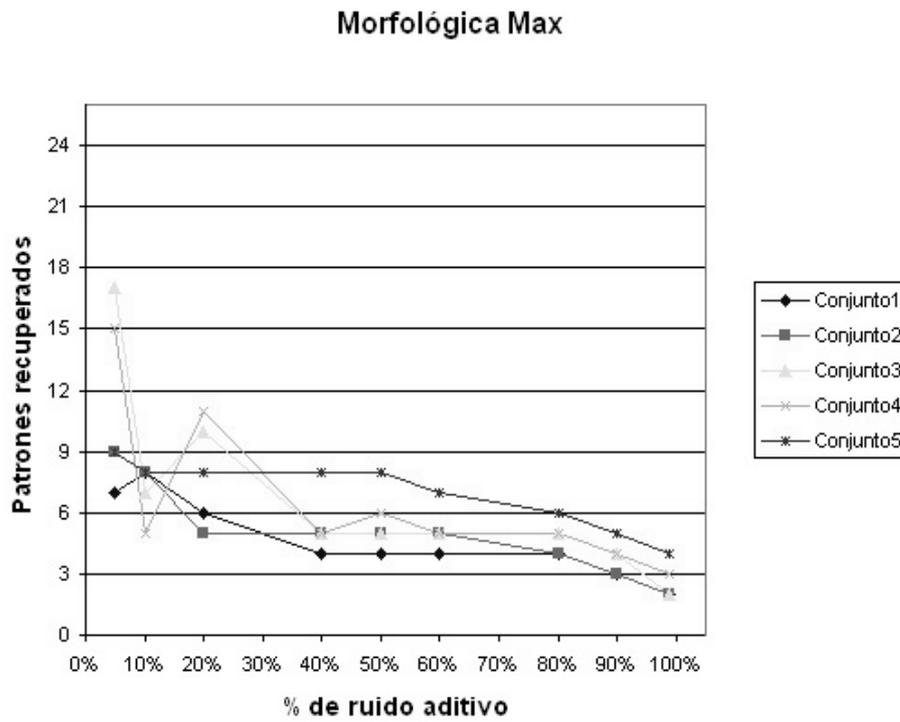
Las memorias de tipo  $\bigwedge$  demuestran su robustez ante el ruido sustractivo. La memoria morfológica min y la memoria Alfa Beta min obtuvieron exactamente los mismo resultados.

**Memorias de tipo  $\bigvee$**

*Morfológica  $\bigvee$*

|            | 5% | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 99% |
|------------|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Conjunto 1 | 7  | 8   | 6   | 4   | 4   | 4   | 4   | 3   | 2   |
| Conjunto 2 | 9  | 8   | 5   | 5   | 5   | 5   | 4   | 3   | 2   |
| Conjunto 3 | 17 | 7   | 10  | 5   | 5   | 5   | 5   | 4   | 2   |
| Conjunto 4 | 15 | 5   | 11  | 5   | 6   | 5   | 5   | 4   | 3   |
| Conjunto 5 | 9  | 8   | 8   | 8   | 8   | 7   | 6   | 5   | 4   |

tabla 5.46. Total de patrones recuperados por la memoria morfológica max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.

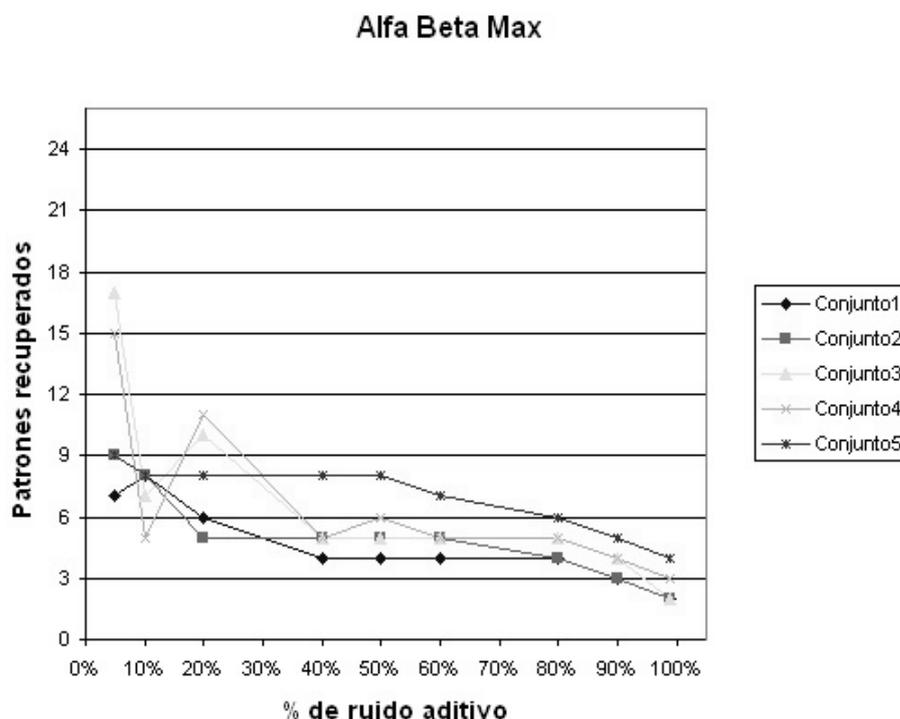


gráfica 5.54. Total de patrones recuperados por la memoria morfológica max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.

*Alfa Beta*  $\vee$

|            | 5% | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 99% |
|------------|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Conjunto 1 | 7  | 8   | 6   | 4   | 4   | 4   | 4   | 3   | 2   |
| Conjunto 2 | 9  | 8   | 5   | 5   | 5   | 5   | 4   | 3   | 2   |
| Conjunto 3 | 17 | 7   | 10  | 5   | 5   | 5   | 5   | 4   | 2   |
| Conjunto 4 | 15 | 5   | 11  | 5   | 6   | 5   | 5   | 4   | 3   |
| Conjunto 5 | 9  | 8   | 8   | 8   | 8   | 7   | 6   | 5   | 4   |

tabla 5.47. Total de patrones recuperados por la memoria Alfa Beta max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.



gráfica 5.55. Total de patrones recuperados por la memoria Alfa Beta max en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido aditivo.

Las memorias de tipo  $\nabla$ , como se ha venido comentando a lo largo de este capítulo, son robustas al ruido aditivo; se sabe ya que los patrones recuperados de manera no perfecta, son los mismos patrones pero afectados con ruido aditivo con un porcentaje menor que con los que se intentaron recuperar. Ambas memorias de tipo  $\nabla$  obtuvieron los mismos resultados.

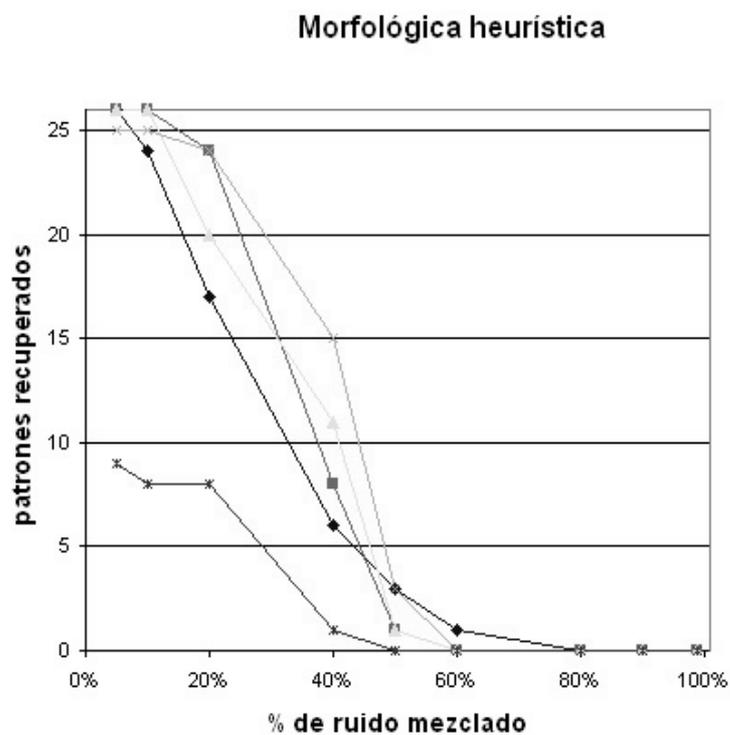
### Memorias heurísticas

Aunque las memorias heurísticas obtuvieron buenos resultados ante el tipo de ruido aditivo y sustractivo no se mostrarán estos resultados en esta sección. Si desea ver las tablas y las gráficas con esta información por favor dirijase al CD anexo en la ruta \Resultados\Autoasociativa\Consolidados.

#### *Morfológica heurística*

|            | 5% | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 99% |
|------------|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Conjunto 1 | 26 | 24  | 17  | 6   | 3   | 1   | 0   | 0   | 0   |
| Conjunto 2 | 26 | 26  | 24  | 8   | 1   | 0   | 0   | 0   | 0   |
| Conjunto 3 | 26 | 26  | 20  | 11  | 1   | 0   | 0   | 0   | 0   |
| Conjunto 4 | 25 | 25  | 24  | 15  | 3   | 0   | 0   | 0   | 0   |
| Conjunto 5 | 9  | 9   | 8   | 1   | 0   | 0   | 0   | 0   | 0   |

tabla 5.48. Total de patrones recuperados por la memoria morfológica heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.

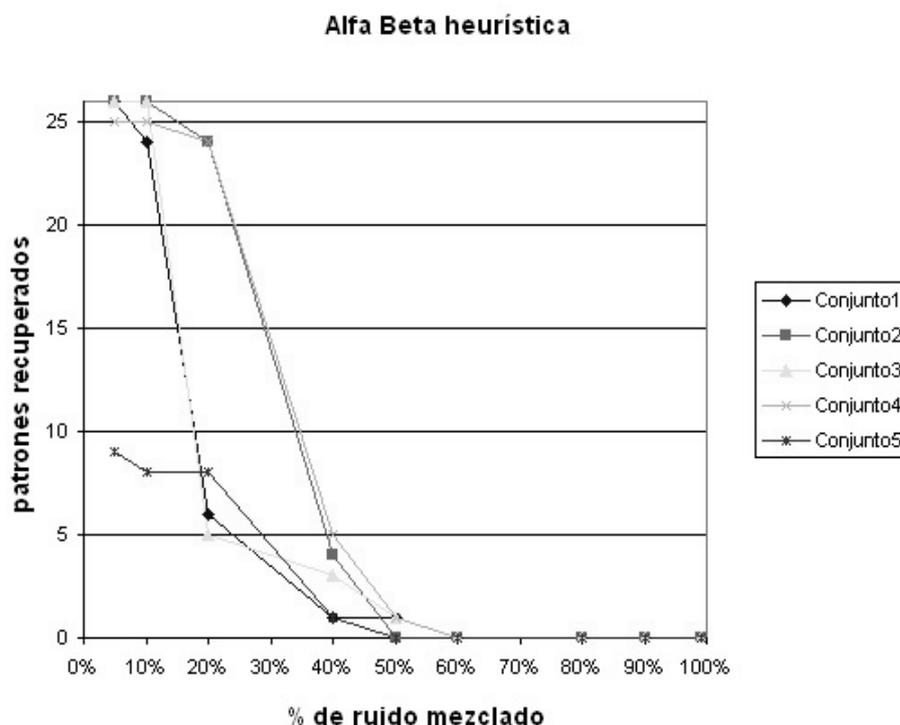


gráfica 5.56. Total de patrones recuperados por la memoria morfológica heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.

*Alfa Beta heurística*

|            | 5% | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 99% |
|------------|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Conjunto 1 | 26 | 24  | 6   | 1   | 1   | 0   | 0   | 0   | 0   |
| Conjunto 2 | 26 | 26  | 24  | 4   | 0   | 0   | 0   | 0   | 0   |
| Conjunto 3 | 26 | 26  | 5   | 3   | 1   | 0   | 0   | 0   | 0   |
| Conjunto 4 | 25 | 25  | 24  | 5   | 1   | 0   | 0   | 0   | 0   |
| Conjunto 5 | 9  | 8   | 8   | 1   | 0   | 0   | 0   | 0   | 0   |

tabla 5.49. Total de patrones recuperados por la memoria Alfa Beta heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.



gráfica 5.57. Total de patrones recuperados por la memoria Alfa Beta heurística en todos los conjuntos fundamentales afectados por todos los porcentajes de ruido mezclado.

Las memorias heurísticas mostraron experimentalmente que son robustas para el ruido mezclado con porcentajes altos. La memoria morfológica heurística obtuvo mejores resultados que la memoria Alfa Beta heurística.

Las memorias autoasociativas y heteroasociativas de tipo  $\bigwedge$  y de tipo  $\bigvee$ , y por lo tanto las memorias heurísticas, son exactamente las mismas para ambos modos, es decir para el modo heteroasociativo y el modo autoasociativo, esto es una gran ventaja de estas memorias ante las memorias clásicas. Se mostró experimentalmente también que estas memorias son mejores que las memorias clásicas en todos los aspectos. La memoria Hopfield es buena, solo para los dos primeros patrones, las memorias tipo  $\bigwedge$  son excelentes para un número considerable de patrones afectados con ruido sustractivo, las memorias de tipo  $\bigvee$  son excelentes para un buen número de patrones alterados con ruido aditivo, y las heurísticas aunque son híbridas y consumen más tiempo computacional han mostrado experimentalmente que son muy robustas para el ruido mezclado.

### 5.3. Casos especiales

En esta sección se mostrarán algunos casos experimentales que se originaron por los resultados obtenidos hasta el momento. Estos casos especiales principalmente están enfocados en las memorias de tipo  $\bigwedge$  y  $\bigvee$ . En la primera parte de esta sección se

presentarán los resultados de iterarán los patrones recuperados por las memorias de tipo min y max; se observó que las memorias de tipo min cuando están los patrones afectados con ruido sustractivo, algunos de los patrones recuperados continúan con ruido sustractivo, y las memorias de tipo max, cuando los patrones están afectados con ruido aditivo algunos patrones conservan este mismo ruido. En la segunda parte se presentarán los resultados del comportamiento de las memorias de tipo min, max y heurísticas, en función a los algoritmos de generación de ruido que se explicaron en el capítulo 4, se especificó en ese capítulo, que el ruido es generado de manera aleatoria, esto ilustra que las memorias quizás pueden recuperar o no un mismo patrón si el ruido aleatorio cambia, se presentarán los resultados obtenidos por estas memorias. En la última parte de esta sección se mostrarán los tiempos computacionales que consumieron las memorias asociativas.

### 5.3.1. Resultados de iterar los patrones recuperados por las memorias min y max

Las memorias de tipo  $\bigwedge$  cuando existen muchos patrones aprendidos, aunque el porcentaje de ruido sustractivo sea muy bajo aún así no recupera todos los patrones, los patrones recuperados si se vuelven a introducir en la memoria, se esperaría que ahora sí recuperase, sino todos los patrones, sí más de los que se recuperaron anteriormente, sin embargo no sucede así, resulta que los patrones recuperados que son iterados, son recuperados exactamente iguales que como se presentaron como patrones de entrada, para ilustrar este hecho se presentarán las siguientes figuras:

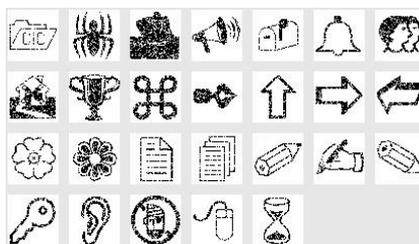


figura 5.11. Patrones afectados con el 20% de ruido sustractivo



figura 5.12. Patrones recuperados por las memorias de tipo min. realizando la primera iteración.



figura 5.13. Patrones recuperados por las memorias de tipo min realizando la segunda iteración.

La figura 5.11 ilustra los patrones que están afectados con el 20% de ruido sustractivo antes de ser intentados recuperar con las memorias de tipo  $\wedge$ . La figura 5.12 ilustra la apariencia de los patrones recuperados por primera ocasión, en ésta se puede ver que la mayor parte del ruido sustractivo fue eliminado. En la figura 5.13 se puede observar que la apariencia de los patrones es idéntica a la apariencia de los patrones que se encuentran en la figura 5.12. El proceso de iteración se realizó en 5 ocasiones obteniendo exactamente los mismos resultados.

Se detectó que al intentar recuperar los patrones de salida afectados con ruido sustractivo generados por las memorias de tipo  $\wedge$ , en las memorias de tipo  $\vee$ , el resultado fue que éstas recuperaron exactamente los patrones como fueron presentados al principio del proceso de recuperación; tuvieron muchos problemas las memorias de tipo  $\vee$  inicialmente, como lo ilustra la figura 5.14, sin embargo al ir aprendiendo e intentando recuperar, se estabilizó la memoria de aprendizaje de tal manera que los patrones de entrada fueron exactamente iguales a los de salida, esto lo ilustra la figura 5.15.



figura 5.14. Apariencia de los 4 primeros patrones recuperados por las memorias de tipo max.



figura 5.15. Apariencia de los patrones recuperados por las memorias de tipo max. Los patrones presentados como entrada son los patrones que recuperó las memorias de tipo min.

El mismo experimento se realizó con las memorias de tipo  $\vee$ , obviamente los experimentos se hicieron con ruido aditivo, y se obtuvo el fenómeno similar en las memorias de tipo  $\wedge$ , es decir que se recuperaron los mismo patrones pero afectados

con ruido aditivo. La figura 5.16 ilustra los patrones de salida para las memorias de tipo min y de tipo max. Esta imagen es el resultado de haber intentado recuperar el conjunto fundamental 1 afectado con el 20 % de ruido aditivo, y es el mismo resultado después de haber iterado en la memoria de tipo max y en la memoria de tipo min. Las iteraciones se realizaron en cinco ocasiones obteniendo el mismo resultado.

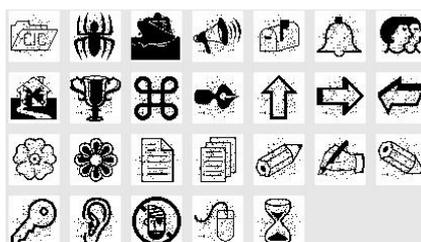


figura 5.16. Apariencia de los patrones de salida, después de haber sido recuperados por las memorias de tipo min con un 20 % de ruido aditivo., y ser nuevamente haber sido iterados con las memorias de tipo min y max.

### 5.3.2. Comportamiento de las memorias de tipo min, de tipo max y heurísticas ante el ruido aleatorio

Para garantizar la capacidad de las memorias asociativas de tipo min y de tipo max se realizó un experimento para determinar qué tan sensibles son estas memorias al porcentaje de ruido asignado en forma aleatoria, queriendo decir con esto, que se detectaron casos en que éste tipo de memorias recuperaron patrones que no recuperaron en una ocasión previa. Para determinar el por qué de esta situación se realizó el experimento donde un mismo patrón fue afectado con algún tipo de ruido y fue recuperado por algún modelo de memoria asociativa, este proceso se realizó por patrón mil veces para las memorias de tipo min, de tipo max, las heurísticas y la memoria Hopfield obteniendo los siguientes resultados:

#### Nota:

El conjunto fundamental con el cual se hicieron estos experimentos fue el conjunto fundamental 5 que está compuesto por los números del 1 al 9.

#### Ruido aditivo 40 %

|               | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8   | 9    |
|---------------|------|------|------|------|------|------|------|-----|------|
| Morf. max     | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 997 | 1000 |
| Alfa Beta max | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 983 | 1000 |
| Hopfield      | 1000 | 1000 | 0    | 0    | 0    | 0    | 0    | 0   | 0    |

tabla 5.50. Tabla que ilustra el total de recuperaciones perfectas de los patrones 1 al 9 con tipo de ruido aditivo aleatorio.

La tabla 5.50 ilustra que salvo en el patrón que es el número 8 todos los intentos por recuperar el patrón fueron exitosos por las memorias de tipo  $\sqrt{\quad}$ ; la memoria Hopfield recuperó los 2 primeros patrones sin ningún problema.

**Ruido sustractivo 40 %**

|               | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    |
|---------------|------|------|------|------|------|------|------|------|------|
| Morf. min     | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| Alfa Beta min | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| Hopfield      | 1000 | 1000 | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

tabla 5.51. Tabla que ilustra el total de recuperaciones perfectas de los patrones 1 al 9 con tipo de ruido sustractivo aleatorio.

La tabla 5.51 ilustra que las memorias de tipo  $\bigwedge$  ante el 40% de ruido sustractivo no tuvieron ningún problema para la recuperación. Al hacer una retrospectiva de los resultados obtenidos, exceptuando las memorias heurísticas, las memorias que tuvieron mejor desempeño en la recuperación perfecta de patrones fueron precisamente las memorias de tipo min. La memoria Hopfield por su parte, demostró que para los dos primeros patrones no tiene absolutamente ningún problema para recuperar de manera perfecta.

**Ruido mezclado 40 %**

|                          | 1    | 2    | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------------------------|------|------|---|---|---|---|---|---|---|
| Morfológica heurística   | 997  | 6    | 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| Alfa Beta min heurística | 1000 | 9    | 3 | 0 | 2 | 0 | 0 | 0 | 0 |
| Hopfield                 | 1000 | 1000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

tabla 5.52. Tabla que ilustra el total de recuperaciones perfectas de los patrones 1 al 9 con tipo de ruido mezclado aleatorio.

La tabla 5.52 ilustra que ante un porcentaje alto de ruido las recuperaciones de patrones pueden variar considerablemente, aunque las memorias heurísticas obtienen buenos resultados, el tipo de ruido aleatorio puede causar que en determinado momento su desempeño pueda bajar. La memoria Hopfield continua siendo muy robusta con los dos primeros patrones, después de esto la memoria Hopfield resulta totalmente ineficiente para realizar recuperaciones perfectas.

No cabe duda que el ruido siempre será un problema latente para el buen desempeño de las memorias, aunque algunos modelos resultan ser eficientes para determinado tipo de ruido, no se puede garantizar al 100% que siempre habrá recuperaciones perfectas ante grandes cantidades de ruido y mucho menos si el ruido es aleatorio.

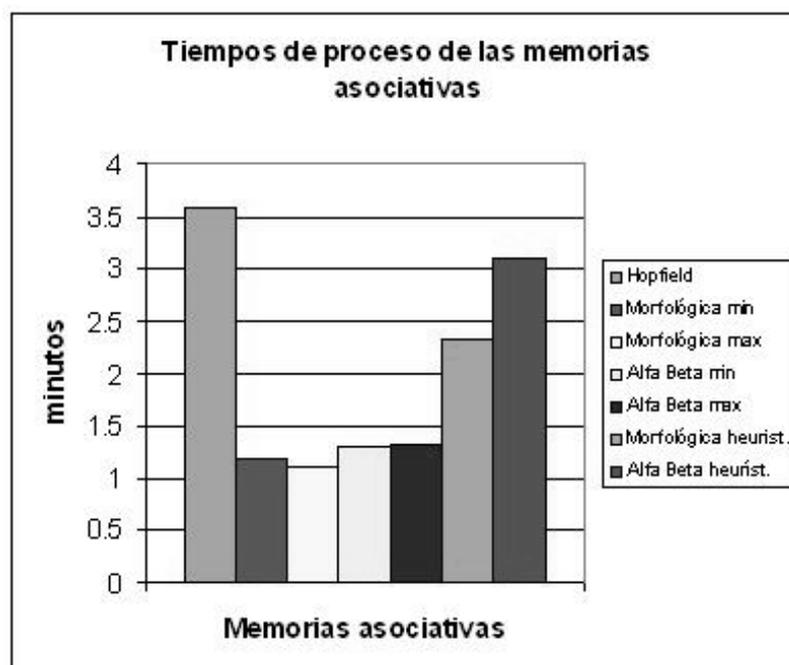
### 5.3.3. Tiempos de ejecución de los procesos de las memorias asociativas

En esta parte del capítulo 5 se mostrará el tiempo computacional que consumió cada memoria asociativa en ejecutar el proceso de aprendizaje y recuperación de patrones. No se contemplará en las estadísticas de tiempo a la memoria Lernmatrix ni la memoria Linear Associator debido a que no tuvieron ninguna recuperación perfecta en las condiciones en las que se probaron todas las memorias asociativas. El tiempo que

tiene cada memoria esta basada en el promedio de la ejecución de los cuatro primeros conjuntos fundamentales. Cabe mencionar que estos tiempos variarán en equipos con características diferentes al equipo donde se realizaron las disquisiciones experimentales.

|                   | minutos y segundos |
|-------------------|--------------------|
| Hopfield          | 3,59               |
| Morfológica min   | 1,17               |
| Morfológica max   | 1,11               |
| Alfa Beta min     | 1,30               |
| Alfa Beta max     | 1,32               |
| Morof. heurística | 2,34               |
| AB heurística     | 3,11               |

tabla 5.53. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación por modelos de memorias asociativas para 26 patrones.



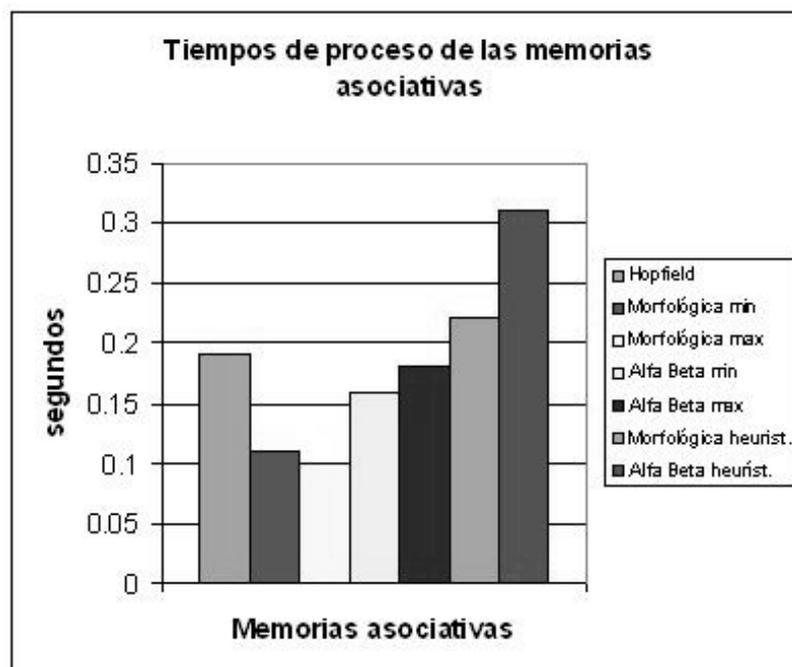
gráfica 5.58. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación por modelos de memorias asociativas para 26 patrones.

La gráfica 5.58 y la tabla 5.53 ilustran que la memoria Hopfield fue la que consumió más tiempo en el proceso de aprendizaje y recuperación de patrones, este resultado tiene mucho sentido ya que la memoria Hopfield itera tantas veces sea necesario hasta que no haya más cambios entre el patrón de entrada con respecto al patrón de salida. La memoria Alfa Beta heurística y la morfológica heurística tienen el segundo y tercer lugar, respectivamente, en tiempo computacional consumido, esto es debido a que estas

memorias realizan doble proceso de aprendizaje y recuperación, además de que tienen que dilatar y erosionar los patrones que se intentan recuperar y al final hacen un AND lógico con los patrones recuperados para obtener el patrón recuperado final, aun que realizan todos estos procesos de más, estas memorias son un poco más rápidas que la memoria Hopfield. Cabe mencionar que las memorias morfológicas son más rápidas que las Alfa Beta, ya que computacionalmente hablando las sumas y restas son más rápidas que las comparaciones, y es cierto ya que las memorias Alfa Beta se basan en comparaciones de valores y las morfológicas se basan en parte de su proceso en sumas.

|                   | segundos |
|-------------------|----------|
| Hopfield          | 19       |
| Morfológica min   | 11       |
| Morfológica max   | 10       |
| Alfa Beta min     | 16       |
| Alfa Beta max     | 18       |
| Morof. heurística | 22       |
| AB heurística     | 31       |

tabla 5.54. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación de memorias asociativas para 9 patrones.



gráfica 5.59. Tiempos promedio en la ejecución de procesos de aprendizaje y recuperación de memorias asociativas para 9 patrones.

En la gráfica 5.59 y la tabla 5.54, las memorias heurísticas son las que consumieron más tiempo computacional, la razón es clara por la manera en que estas memorias

están diseñadas; algo que es relevante es que la memoria Hopfield fue un segundo más tardada que la memoria Alfa Beta max, en este caso también este tiempo es lógico debido a que la memoria Hopfield tan solo esta trabajando con 9 patrones, lo cual indica que realiza menos iteraciones para converger. Por lo regular la memoria Hopfield converge en las tres primeras iteraciones para los dos primeros patrones.

Los tiempos computacionales que consumieron los diferentes modelos de memorias generan un indicio de qué memoria puede usarse en cierta aplicación.

## Capítulo 6

# Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones obtenidas de los resultados del presente trabajo de tesis, una vez desarrolladas las ideas que inicialmente se propusieron a lo largo de los capítulos anteriores, plasmando de manera puntual el cumplimiento de los objetivos propuestos. Conjuntamente con las conclusiones, daremos algunas de las posibles tareas que se podrán desarrollar para el cabal entendimiento de lo propuesto en el presente trabajo, dando la pauta a futuros investigadores sobre los puntos no cubiertos, pero que pudiesen ser abordados de forma inmediata.

### 6.1. Conclusiones

1. Se desarrolló el software requerido para la realización del estudio comparativo experimental de los modelos de memorias asociativas analizados en este trabajo de tesis.
2. Se ha ilustrado experimentalmente el comportamiento de los diferentes modelos de memorias asociativas en los casos ideales y ante circunstancias extremas. Este estudio comparativo ha enfatizado las bondades y deficiencias de cada modelo de memoria asociativa.
3. El estudio comparativo experimental presentado en este trabajo de tesis viene a llenar un hueco notable en la literatura científica actual, en virtud de que, después de una exhaustiva investigación documental, fue evidente para el autor de esta tesis que no existen, salvo en (Yañez-Márquez, 2002), estudios experimentales comparativos para los modelos más importantes de memorias asociativas.
4. En el modo heteroasociativo sin ruido las memorias *min*, tanto morfológicas como  $\alpha\beta$ , recuperaron aproximadamente el 50% de los patrones del conjunto fundamental de manera perfecta (excepto en el conjunto fundamental 5, donde recuperaron el 90%), mientras que las memorias *max*, lo hicieron con aproximadamente un 15% de los patrones (excepto en el conjunto fundamental 5, donde recuperaron el 80%). Por su parte los demás modelos, excluyendo la memoria Hopfield por ser inherentemente autoasociativa, no recuperaron de manera perfecta ni un solo patrón.

**Nota 1:** Es pertinente hacer notar que los patrones recuperados de manera no perfecta por las memorias heteroasociativas *min*, tanto morfológicas como  $\alpha\beta$ , son "muy parecidos" a los patrones del conjunto fundamental. Esto es una consecuencia de las restricciones impuestas por lo expresado en la NOTA de la página 69.

**Nota 2:** No obstante que las memorias heurísticas, tanto morfológicas como  $\alpha\beta$ , recuperaron de manera perfecta la totalidad de los patrones, técnicamente de manera estricta, no se puede afirmar que las recuperaciones fueron "sin ruido", dado que al presentarle a la entrada un patrón a este tipo de memorias, necesariamente se altera.

5. Para estudiar experimentalmente el comportamiento de los diferentes modelos de memorias asociativas ante patrones de entrada alterados con algún tipo de ruido, se usaron valores de 5 %, 10 %, 20 %, 40 %, 50 %, 60 %, 80 %, 90 % y 99 % para cada tipo de ruido: aditivo, sustractivo y mezclado.
6. En el modo heteroasociativo con ruido aditivo las memorias *max*, tanto morfológicas como  $\alpha\beta$ , tuvieron un comportamiento similar, en cuanto a porcentajes de patrones recuperados de manera perfecta, ante los diferentes porcentajes de ruido: con ruido entre 5 % y 60 % recuperaron en promedio el 30 % de los patrones; con 80 % y 90 % de ruido recuperaron poco más del 20 % de patrones, y con 99 % de ruido, recuperaron poco más del 10 % de patrones de manera perfecta. Con excepción de las heurísticas, en todos los demás modelos no hubo recuperaciones perfectas.
7. Las memorias heteroasociativas *min*, tanto morfológicas como  $\alpha\beta$ , son muy sensibles a ruido aditivo, aún en pequeños porcentajes.
8. Las memorias heurísticas  $\alpha\beta$  recuperaron el 100 % de los patrones en todos los niveles de ruido aditivo, incluyendo el 99 % el cual es un valor de ruido muy considerable. Las memorias heurísticas morfológicas recuperaron aproximadamente el 100 % de los patrones, con valores de ruido del 5 % al 60 %, mientras que con 80 % de ruido recuperaron poco menos del 50 % de patrones, y con 90 % y 99 % de ruido aditivo, su rendimiento bajó a un 35 % de patrones recuperados de manera perfecta.
9. En el modo heteroasociativo con ruido sustractivo las memorias *min*, tanto morfológicas como  $\alpha\beta$ , tuvieron un comportamiento similar, en cuanto a porcentajes de patrones recuperados de manera perfecta, ante los diferentes porcentajes de ruido: con valores de ruido de 5 % y 10 % recuperaron en promedio casi el 60 % de los patrones; del 20 % al 60 % de ruido, recuperaron el 50 % de patrones; con 80 % y 90 % de ruido recuperaron aproximadamente 40 % de patrones, y con el 99 % de ruido recuperaron el 20 % de patrones de manera perfecta. Con excepción de las heurísticas, en todos los demás modelos no hubo recuperaciones perfectas.
10. Las memorias heteroasociativas *max*, tanto morfológicas como  $\alpha\beta$ , son muy sensibles a ruido sustractivo, aún en pequeños porcentajes.

11. Con 5 % y 10 % de ruido sustractivo, las memorias heurísticas, tanto morfológicas como  $\alpha\beta$ , recuperaron casi el 100 % de los patrones del conjunto fundamental; con 20 % de ruido sustractivo, las memorias heurísticas morfológicas recuperaron poco más del 80 % de patrones, mientras que las memorias heurísticas  $\alpha\beta$  recuperaron el 60 % de patrones. A partir del 40 % y hasta el 99 % de ruido sustractivo, el rendimiento de las heurísticas baja notoriamente: en las morfológicas, de 30 % a casi 0 % de patrones recuperados, y en las heurísticas  $\alpha\beta$ , del 5 % al 0 % de patrones recuperados de manera perfecta.
12. Las memorias heteroasociativas heurísticas  $\alpha\beta$  son más robustas a ruido aditivo que a sustractivo.
13. En el modo heteroasociativo con ruido mezclado fallan todos los modelos considerados, con excepción de las memorias asociativas heurísticas, tanto morfológicas como  $\alpha\beta$ . Ambos modelos tuvieron un comportamiento similar, en cuanto a porcentajes de patrones recuperados de manera perfecta, ante los diferentes porcentajes de ruido: con valores de ruido de 5 %, 10 % y 20 % de ruido mezclado, ambos modelos recuperaron casi el 100 % de los patrones del conjunto fundamental. A partir del 40 % y hasta el 99 % de ruido mezclado, el rendimiento de las heurísticas baja notoriamente: en las morfológicas, de 50 % a casi 0 % de patrones recuperados, y en las heurísticas  $\alpha\beta$ , del 10 % al 0 % de patrones recuperados de manera perfecta.
14. Las memorias heteroasociativas *max* y *min*, tanto morfológicas como  $\alpha\beta$ , son muy sensibles a ruido mezclado, aún en pequeños porcentajes.
15. En el modo autoasociativo sin ruido, las memorias morfológicas,  $\alpha\beta$  y heurísticas, recuperaron de manera perfecta el 100 % de los patrones del conjunto fundamental. Por su parte la memoria Hopfield recuperó, en promedio, el 15 % de los patrones. El *Linear Associator* no recuperó ningún patrón de manera perfecta.
16. En el modo autoasociativo con ruido aditivo las memorias *max*, tanto morfológicas como  $\alpha\beta$ , tuvieron un comportamiento similar, en cuanto a porcentajes de patrones recuperados de manera perfecta, ante los diferentes porcentajes de ruido, desde 60 % de patrones recuperados con 5 % de ruido, hasta 10 % de patrones recuperados con 99 % de ruido aditivo. Las memorias heurísticas, tanto morfológicas como  $\alpha\beta$ , recuperaron el 100 % de los patrones en todos los niveles de ruido aditivo, incluyendo el 99 % el cual es un valor de ruido muy considerable. Por su parte la memoria Hopfield recuperó, en promedio, el 12 % de patrones con ruidos del 5 % al 50 %, y de ahí en adelante su rendimiento bajó hasta 0 %. El *Linear Associator* no recuperó ningún patrón de manera perfecta.
17. En el modo autoasociativo con ruido sustractivo las memorias *min*, tanto morfológicas como  $\alpha\beta$ , tuvieron un comportamiento similar, en cuanto a porcentajes de patrones recuperados de manera perfecta, ante los diferentes porcentajes de ruido: desde poco más de 70 % de patrones recuperados con 5 % de ruido, hasta 17 % de patrones recuperados con 99 % de ruido sustractivo. Las memorias heurísticas, tanto morfológicas como  $\alpha\beta$ , recuperaron casi el 100 % de los patrones en

los niveles de ruido sustractivo de 5% y 10%; con 20% de ruido, recuperaron el 65% de patrones y de ahí en adelante, su rendimiento bajó hasta el 0% de patrones recuperados. Por su parte la memoria Hopfield recuperó, en promedio, el 10% de patrones. El *Linear Associator* no recuperó ningún patrón de manera perfecta.

18. Las memorias autoasociativas *min*, tanto morfológicas como  $\alpha\beta$ , son más robustas a ruido sustractivo de lo robustas que son las memorias autoasociativas *max*, tanto morfológicas como  $\alpha\beta$ , ante ruido aditivo.
19. En el modo autoasociativo con ruido mezclado fallan todos los modelos considerados, con excepción de las memorias asociativas heurísticas, tanto morfológicas como  $\alpha\beta$ , y la memoria Hopfield. Ambos modelos de memorias heurísticas tuvieron un comportamiento similar, en cuanto a porcentajes de patrones recuperados de manera perfecta, ante los diferentes porcentajes de ruido: con valores de ruido de 5% y 10% de ruido mezclado, ambos modelos recuperaron casi el 100% de los patrones del conjunto fundamental. A partir del 20% y hasta el 99% de ruido mezclado, el rendimiento de las heurísticas baja notoriamente: en las morfológicas, de 80% a casi 0% de patrones recuperados, y en las heurísticas  $\alpha\beta$ , del 60% al 0% de patrones recuperados de manera perfecta. Por su parte la memoria Hopfield recuperó, en promedio, el 10% de patrones con ruidos del 5% al 40%; de ahí en adelante no recuperó patrones de manera perfecta.
20. La memoria Hopfield recupera de manera perfecta todo el conjunto fundamental ante cualquier tipo de ruido, cuando la cardinalidad del conjunto fundamental es dos; sin embargo, el rendimiento disminuye considerablemente cuando se consideran más de dos patrones en el conjunto fundamental.
21. Las memorias heurísticas recuperaron correctamente patrones con ruido mezclado; además, respondieron de manera adecuada a patrones de entrada afectados por ruido aditivo, no así con el ruido sustractivo. Sin embargo, el tiempo computacional en las heurísticas es mayor que para las memorias de tipo  $\bigwedge$  y para las memorias de tipo  $\bigvee$ .
22. Al iterar los patrones recuperados por las memorias morfológicas y las memorias  $\alpha\beta$ , ante éstas mismas, obtienen exactamente las mismas características; es decir, que si estas memorias recuperaron ciertos patrones con cierto tipo de ruido, al iterarse, se recuperarán los mismos patrones con la misma cantidad de ruido. Dicho de otra forma, no tiene sentido iterar los patrones recuperados porque no habrá mejoría. Estas memorias no requieren convergencia.

## 6.2. Trabajo futuro

En esta sección se presentan ideas tentativas que pueden ser atacadas de manera directa a fin de contribuir al campo de las memorias asociativas y, por ende, al estudio comparativo estadístico como tal.

1. Profundizar en la base teórica para atacar el problema del ruido mezclado en las memorias morfológicas.
2. Encontrar la base teórica para atacar el problema del ruido mezclado en las memorias  $\alpha\beta$ .
3. Respecto a lo afirmado en la Nota 1 de la conclusión 4, dado que los patrones recuperados de manera no perfecta por las memorias heteroasociativas *min*, tanto morfológicas como  $\alpha\beta$ , son "muy parecidos." a los patrones del conjunto fundamental, se propone utilizar un criterio para medir qué tanto "se parecen" los patrones recuperados a los correspondientes del conjunto fundamental, y realizar un estudio experimental en este tema. Por ejemplo, se podría utilizar la distancia de Hamming, y flexibilizar la definición de recuperación perfecta, considerando como tal una recuperación cuyo valor en la distancia de Hamming sea inferior a cierto valor de umbral propuesto.
4. Encontrar algoritmos que permitan una ejecución computacional mucho más rápida de las memorias asociativas .
5. Realizar algoritmos que permitan la ejecución de manera paralela de las memorias asociativas.
6. Investigar propiedades matriciales para intentar mejorar la capacidad de las memorias clásicas e inclusive las memorias morfológicas y las memorias  $\alpha\beta$ .
7. Investigar propiedades matriciales para poder particionar la matriz de aprendizaje y poder ejecutar los procesos tanto de aprendizaje como de recuperación en forma paralela.

## Apéndice A

# Simbología

|  |   |
|--|---|
| memorias $\alpha\beta$   | memorias asociativas $\alpha\beta$                                |
| $\alpha, \beta$  | operadores originales en que se basan las memorias $\alpha\beta$  |
| <b>M</b>   | memoria asociativa, memoria asociativa morfológica <i>max</i>     |
| <b>W</b>   | memoria asociativa morfológica <i>min</i>                         |
| $m_{ij}$   | $ij$ -ésima componente de la memoria asociativa <b>M</b>          |
| $B$  | conjunto al que pertenecen las componentes $m_{ij}$               |
| $\in$  | pertenencia de un elemento a un conjunto                          |
| <b>x</b>   | vector columna que corresponde a un patrón de entrada             |
| <b>y</b>   | vector columna que corresponde a un patrón de salida              |
| $A$  | conjunto al que pertenecen las componentes de <b>x</b> y <b>y</b> |
| $(\mathbf{x}, \mathbf{y})$                                       | asociación de un patrón de entrada con uno de salida              |
| $(\mathbf{x}^k, \mathbf{y}^k)$                                   | asociación de la $k$ -ésima pareja de patrones                    |
| $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ | conjunto fundamental  |
| $\tilde{\mathbf{x}}^k$   | versión alterada del patrón fundamental $\mathbf{x}^k$            |
| $x_j^\mu$  | $j$ -ésima componente de un vector columna $\mathbf{x}^\mu$       |
| $(\mathbf{x}^\mu)^t$   | transpuesto del vector $\mathbf{x}^\mu$                           |
| $n$  | dimensión de los patrones de entrada                              |
| $m$  | dimensión de los patrones de salida                               |
| $p$  | número de parejas del conjunto fundamental                        |
| $\Delta m_{ij}$  | incremento en $m_{ij}$  |
| $\bigvee$  | operador máximo   |
| $\bigwedge$  | operador mínimo   |
| $\cdot$  | producto usual entre vectores o matrices                          |
| $\times$   | producto cruz (entre conjuntos)                                   |
| $\delta_{ij}$  | delta de Kronecker (afecta a los índices $i$ y $j$ )              |
| $\nabla$   | producto máximo (entre matrices)                                  |
| $\triangle$  | producto mínimo (entre matrices)                                  |

---

|                             |  |
|-----------------------------|--|
| $\alpha(x, y)$              | operación binaria $\alpha$ con argumentos $x$ y $y$                                |
| $\beta(x, y)$               | operación binaria $\beta$ con argumentos $x$ y $y$                                 |
| $P = [p_{ij}]_{m \times r}$ | matriz $P$ de dimensiones $m \times r$ y componente $ij$ -ésima $p_{ij}$           |
| $\forall$                   | cuantificador universal  |
| $\exists$                   | cuantificador existencial  |
| $\Psi_\alpha$               | operador $\alpha$ <b>max</b>   |
| $\Psi_\beta$                | operador $\beta$ <b>max</b>  |
| $\mathfrak{M}_\alpha$       | operador $\alpha$ <b>min</b>   |
| $\mathfrak{M}_\beta$        | operador $\beta$ <b>min</b>  |
| $\boxtimes$                 | símbolo que representa a las dos operaciones $\Psi_\alpha$ y $\mathfrak{M}_\alpha$ |
| $\mathbf{V}$                | memorias $\alpha\beta$ tipo $\mathbf{V}$   |
| $\mathbf{\Lambda}$          | memorias $\alpha\beta$ tipo $\mathbf{\Lambda}$                                     |
| $\otimes$                   | operación para calcular la distancia de Hamming                                    |
| $\oplus$                    | operación lógica <b>or</b> exclusiva   |

# Bibliografía

- [1] Abe, S. (2001). *Pattern classification, Neuro-Fuzzy Methods and their Comparison*, UK: Springer-Verlag.
- [2] Abu-Mostafa, Y. & St. Jacques, J. (1985). Information capacity of the Hopfield model, *IEEE Transactions on Information Theory*, *IT-31*, 4, 461-464.
- [3] Acharya, U. R. & et. al. (2003). Classification of Heart rate data using artificial neural network and fuzzy equivalence relation, *Pattern Recognition*, *36* (2003), 61-81.
- [4] Adeodato P. J. L. & Taylor J. G. (1996). Autoassociative memory with high storage capacity, In C. von der Malsburg, W. von Seelen, J. C. Vorbrueggen & B. Sendhoff (Eds.), *Lecture Notes in Computer Science*, 1112, (pp. 29-34),. Bochum, Germany: ICANN'96.
- [5] Aleksander, I. & Morton, H. B. (1997). Weightless and other memory-based networks, In Emile Fiesler (Ed.), *Handbook of Neural Computation*, (pp. C1.5:1-C1.5:15). New York: Oxford.
- [6] Ammeraal, L. (1997). *STL for C++ programmers*, USA: John Wiley & Sons.
- [7] Amari, S. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements, *IEEE Transactions on Computers*, *C-21*, 11, 1197-1206.
- [8] Amari, S. (1977). Neural theory of association and concept-formation, *Biological Cybernetics*, *26*, 175-185.
- [9] Anderson, J. A. (1972). A simple neural network generating an interactive memory, *Mathematical Biosciences*, *14*, 197-220.
- [10] Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). *Neurocomputing: Foundations of Research*, Cambridge: MIT Press.
- [11] Anderson, J. A., Silverstein, J., Ritz, S. & Jones, R. (1977). Distinctive features, categorical perception, and probability learning: some applications of a neural model, *Psychological Review*, *84*, 413-451.
- [12] Anderson, J. R. & Bower, G. (1977). *Memoria Asociativa*, México: Limusa.

- [13] Austin, J. (1987). ADAM: A Distributed Associative Memory for Scene Analysis, In *Proceedings of First International Conference on Neural Networks*, (pp. 285-295). San Diego: Ed. M Caudhill and C Butler.
- [14] Austin, J., Buckle, S., Kennedy, J., Moulds, A., Pack, R. & Turner, A. (1997). The cellular neural network associative processor, In A. Krikelis & C. C. Weems (Eds.), *Associative Processing and Processors*, (pp. 284-306). Los Alamitos: IEEE Computer Society.
- [15] Ayaquica-Martínez, I. O. (2002). *Algoritmo C-means usando funciones de disimilitud*, Tesis de maestría, CIC-IPN, México.
- [16] Bandyopadhyay, S. & Datta, A. K. (1996). A novel neural hetero-associative memory model for pattern recognition, *Pattern Recognition*, 29, 5, 789-795.
- [17] Bandyopadhyay, S. & Maulik, U. (2002). Efficient prototy reordering in nearest neighbor classification, *Pattern Recognition*, 35 (2002), 2791-2799.
- [18] Bosch, H. & Kurfess, F. J. (1998). Information storage capacity of incompletely connected associative memories, *Neural Networks* (11), 5, 869-876.
- [19] Buhmann, J. (1995). Oscillatory associative memories, In M. Arbib (Ed.), *Handbook of Brain Theory & Neural Networks*, (pp. 691-694). Cambridge: MIT Press.
- [20] Chen, C. & Honavar, V. (1995). A neural architecture for content as well as address-based storage and recall: theory and applications, Iowa State University Technical Report TR95-03.
- [21] Cherkassky, V., Fassett, K. & Vassilas, N. (1991). Linear algebra approach to neural associative memories and noise performance of neural classifiers, *IEEE Transactions on Computers*, Vol. 40, Issue 12, 1429 - 1435.
- [22] Daconta, C. M. (1995). *C++ pointers and Dynamic memory manayement*, USA: John Wiley & Sons.
- [23] Dasarathy, B. V. (1991). *Nearest Neighbor (NN) Norms: NN pattern Classification Techniques*, USA:IEEE Computer Society Pres
- [24] Davey, N. & Hunt, S. P. (2000). A Comparative Analysis of High Performance Associative Memory Models, *2nd International ICSC Symposium on Neural Computation (NC'2000)*, Berlin, Germany.
- [25] Demeniconi, C., Peng J. & Gunopulos, D. (2002). Locally Adaptive Metric Nearest-Neighbor Classification, *IEEE Transations on Pattern Análisis and Machine Intellegence*, vol. 24, no. 9, 1281-1285.
- [26] Díaz-de-León, J. L. & Yáñez, C. (1999). Memorias asociativas con respuesta perfecta y capacidad infinita, *Memoria del TAINA '99, México, D.F.*, 23-38.
- [27] Dietel, H. M. & Dietel, P. J. (1999). *Como programar en C++*, México: Pearson.

- 
- [28] Duda, R. O., Hart, P. E. & Stork, D. G. (2001). *Pattern Classification*, USA: John Wiley & Sons.
- [29] Duin, R. P.W. (1996). A note on comparing classifiers, *Pattern Recognition Letters* 17, 529-536
- [30] Eckel, B: (1995). *Thinking in C++*, USA: Prentice Hall.
- [31] Friedman, M. & Kandel, A. (2000). *Introduction to Pattern Recognition (Statistical, Structural, Neural and Fuzzy logic Approaches)*, Singapore, World Scientific.
- [32] Gill, P. E., Murray, W. & Wright, M. H. (1989). *Numerical Linear Algebra and Optimization, vol. 1*, USA: Addison Wesley.
- [33] González, R. C. & Woods, R. E. (2001). *Digital Image Processing*, USA: Prentice Hall.
- [34] Graham, B. & Willshaw, D. (1995). Improving recall from an associative memory, *Biological Cybernetics*, 72, 337-346.
- [35] Grimaldi, R. P. (1998). *Matemáticas discreta y combinatoria*, México: Addison Wesley.
- [36] Grossman, S. I. (1996). *Algebra lineal*, México: McGraw-Hill.
- [37] Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation*, USA: Prentice Hall.
- [38] Haralick, R. M., Sternberg, S. R. & Zhuang, X. (1987). Image analysis using mathematical morphology, *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9, 4*, 532-550.
- [39] Hassoun, M. H. (Ed.) (1993). *Associative Neural Memories*, New York: Oxford University Press.
- [40] Hassoun, M. H. (1995). *Fundamentals of Artificial Neural Networks*, Cambridge: MIT Press.
- [41] Ho, S. Y., Liu, C. C. & Liu, S. (2002). Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm, *Pattern Recognition Letters*, 23 (2002), 1495-1503.
- [42] Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences*, 79, 2554-2558.
- [43] Hopfield, J.J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons, *Proceedings of the National Academy of Sciences*, 81, 3088-3092.
- [44] Huang, Y. S., & et. al. (2002). Prototype optimization for nearest-neighbor classification, *Pattern Recognition*, 35 (2002), 1237-1245.

- [45] Imada, A. & Araki K. (1995). Genetic algorithm enlarges the capacity of associative memory. In L.J.Eshelman(Ed.), *Proceedings of 6th International Conference on Genetic Algorithms*, (pp 413-420). San Francisco: Morgan Kaufmann.
- [46] Jagota, A., Narasimhan, G. & Regan, K. W. (1998). Information Capacity of Binary Weights Associative Memories, *Neurocomputing*, 19(1-3), 35-38.
- [47] Kanerva, P. (1988). *Sparse Distributed Memory*, Cambridge: MIT Press.
- [48] Kennedy, J. V., Austin, J. & Cass, B. (1995). A hardware implementation of a binary neural image processor, *Proceedings of the IEE Conference on Image Processing and its Applications*, Edinburgh, UK.
- [49] Kinsler, J. M. (1995). Fast analog associative memory, *Pocceedings of the SPIE*, 2568, 290-293.
- [50] Kittler, J. & Alkoot, F.M. (2003). Sum vesus Vote Fusion in Multiple Classifier Systems, *IEEE Transations on Pattern Análisis and Machine Intellegence*, vol. 25, no. 1, 110-115.
- [51] Kishan, M., Chilukuri, K. M. & Sanjay, R. (1997). *Elements of Artificial Neural Networks*, USA: MIT Press.
- [52] Kohonen, T. (1972). Correlation matrix memories, *IEEE Transactions on Computers*, C-21, 4, 353-359.
- [53] Kohonen, T. (1974). An adaptive associative memory principle, *IEEE Transactions on Computers*, C-24, 4, 444-445.
- [54] Kohonen, T. (1987). *Content-Addressable Memories*, Berlin: Springer-Verlag.
- [55] Kohonen, T. (1989). *Self-Organization and Associative Memory*, Berlin: Springer-Verlag.
- [56] Kohonen, T. (1997). *Self-Organizing Maps*, Berlin: Springer.
- [57] Kohonen, T. & Ruohonen, M. (1973). Representation of associated data by matrix operators, *IEEE Transactions on Computers*, C-22, 701-702.
- [58] Kolen, J. F., & Pollack, J. B. (1991). Multiassociative memory, *The Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, 785-789.
- [59] Krikelis, A. & Weems, C. C. (1997). Associative Processing and Processors, In A. Krikelis & C. C. Weems (Eds.), *Associative Processing and Processors*, (pp. 2-9). Los Alamitos: IEEE Computer Society.
- [60] Kuncheva, L. I. (2000). *Fuzzy Classifier Design*, Germany: Physica-Verlag.
- [61] Kuncheva, L. I. (2002). A theoretical Study on Six Classifier Fusion Strategies, *IEEE Transations on Pattern Análisis and Machine Intellegence*, vol. 24, no. 2, 281-286.

- [62] Lang, S. (1976). *Algebra lineal*, México: Fondo Educativo Interamericano.
- [63] Leithold, L. (1994). *Matemáticas previas al cálculo*, México: Harla.
- [64] Lewis, R. H. & Papadimitriou, C. H. (1998). *Element of the theory of computation*, USA: Prentice-Hall.
- [65] Little, W. & Shaw, G. (1975). A statistical theory of short and long term memory, *Behavioral Biology*, 14, 115-133.
- [66] Lu, Y. & Tan, C. L. (2002). Combination of múltiple classifiers using probabilistic dictionary and its application to pastcode recognition, *Pattern Recognition*, 35 (2002), 2823-2832.
- [67] Marqués de Sá, J. P. (2001). *Pattern Recognition, Concepts, Methods and Application*, Germany: Springer.
- [68] McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5, 115-133.
- [69] McEliece, R., Posner, E., Rodemich, E. & Venkatesh, S. (1987). The capacity of the Hopfield associative memory, *IEEE Transactions on Information Theory*, IT-33, 4, 461-482.
- [70] Minsky, M. & Papert, S. (1969). *Perceptrons*, Cambridge: MIT Press.
- [71] Moore, J. (1968). *Elements of linear algebra and matrix theory*, New York: McGraw-Hill.
- [72] Murua, A. (2002). Upper Bounds for Error Rates of Linear Combinations of Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, 591-602.
- [73] Nakano, K. (1972). Associatron-A model of associative memory, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2, 3, 380-388.
- [74] Nicola, G., Masulli, F. & Sperduti, A. (2002). Theoretical and experimental análisis of a two-stage system for classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 24, no. 7, 893-904.
- [75] Pal, S. (1999). *Neuro - Fuzzy, Pattern Recognition: Methods in Soft Computing*, USA: John Wiley & Sons.
- [76] Palm, G., Schwenker, F., Sommer F. T. & Strey, A. (1997). Neural associative memories, In A. Krikelis & C. C. Weems (Eds.), *Associative Processing and Processors*, (pp. 307-326). Los Alamitos: IEEE Computer Society.
- [77] Pandya, A. S. (1996). *Pattern recognition with neural networks in C++*, Great Britain: Springer-Verlag.
- [78] Pitas, I. (2000). *Digital Image Processing Algorithms and Applications*, J. Wiley Sons, Inc.

- [79] Raudys, A. (2003). High speed associative memories for features extraction and visualisation, *Pattern Recognition Letters*, 24, 1317-1329.
- [80] Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, vol. 65, 386-408.
- [81] Ritter, G. X., Sussner, P. & Diaz-de-Leon, J. L. (1998). Morphological associative memories, *IEEE Transactions on Neural Networks*, 9, 281-293.
- [82] Ritter, G. X., Diaz-de-Leon, J. L. & Sussner, P. (1999). Morphological bidirectional associative memories, *Neural Networks*, 12, 851-867.
- [83] Rueda, L. & Oommen, B. J. (2002). On optimal pairwise linear classifiers for normal distributions the two-dimensional case, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, 274-273 .
- [84] Sánchez-Garfias, F. (2004). *Lernmatrix de Steinbuch: Condiciones necesarias y suficientes para recuperación perfecta de patrones*, Tesis de Maestría en Ciencias de la Computación, CIC-IPN, México.
- [85] Santiago-Montero, R. (2003). *Clasificador híbrido de patrones basado en la Lenmarix de Steinbuch y el Linear Associator de Anderson-Kohonen*, Tesis de Maestría en Ciencias de la Computación, CIC-IPN, México.
- [86] Santiago-Montero, R., Yáñez-Márquez, C. & Díaz de León, J. L (2002). Clasificador híbrido de patrones basado en la Lenmarix de Steinbuch y el Linear Associator de Anderson-Kohonen, *Research on computing science, Reconocimiento de patrones, avances y perspectivas*, 449-460. México: CIC-IPN.
- [87] Šarūnas, R. (2001). *Statistical and Neural Classifiers, An integrated Approach to design*, England: MIT Press.
- [88] Schürmann, J. (1996). *Pattern classification, A unified view of statistical and neural approaches*, USA: John Wiley.
- [89] Seed, G. (1996). *An Introduction to object-oriented programming in C++*, Great Britain: Springer-Verlag.
- [90] Serra, J. (1992). *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*, London: Academic Press.
- [91] Shalkoff, R. (1992). *Pattern recognition, Statical, Structural and Neural Approaches*, USA: John Wiley.
- [92] Simpson, P. K. (1990). *Artificial Neural Systems*, New York: Pergamon Press.
- [93] Sonka, M., Vaclav, H. & Boyle, R. (1999). *Imagen Processing, Analysis and Machine Vision*, USA: ITP.
- [94] Steinbuch, K. (1961). Die Lernmatrix, *Kybernetik*, 1, 1, 36-45.

- 
- [95] Steinbuch, K. & Frank, H. (1961). Nichtdigitale Lernmatrizen als Perzeptoren, *Kybernetik*, 1, 3, 117-124.
- [96] Storkey, A. J. (1997). Increasing the capacity of a Hopfield network without sacrificing functionality, *International Conference on Artificial Neural Networks*, 451-456.
- [97] Stright, J. R., Coffield, P. C. & Brooks, G. W. (1998). Analog VLSI implementation of a morphological associative memory, *Proceedings of the SPIE*, 3452-03, 14-22.
- [98] Webb, A. (1999). *Statistical Pattern Recognition*, USA: Oxford University Press.
- [99] Willshaw, D., Buneman, O. & Longuet-Higgins, H. (1969). Non-holographic associative memory, *Nature*, 222, 960-962.
- [100] Wu, Y., Ianekev, K. & Govindaraju, V. (2002). Improved k-nearest neighbor classification, *Pattern Recognition*, 35 (2002), 2311-2318.
- [101] Yáñez-Márquez, C. & Díaz de León, J. L. (2001a). *Lernmatrix de Steinbuch*, IT 48, Serie Verde, CIC-IPN, México.
- [102] Yáñez-Márquez, C. & Díaz de León, J. L. (2001b). *Linear Associator de Anderson-Kohonen*, IT 50, Serie Verde, CIC-IPN, México.
- [103] Yáñez-Márquez, C. & Díaz de León, J. L. (2001c). *Memoria Asociativa Hopfield*, IT 52, Serie Verde, CIC-IPN, México.
- [104] Yáñez-Márquez, C. (2002). *Memorias Asociativas Basadas en Relaciones de Orden y Operadores Binarios*, Tesis doctoral, CIC-IPN, México.