



INAOE

# Memorias Asociativas Reticulares Minimax para Recuperación de Imágenes con Ruido

por

**MC. José Angel Nieves Vázquez**

Tesis sometida como requisito parcial para obtener el grado de Doctor en Ciencias en la especialidad de Óptica en el Instituto Nacional de Astrofísica, Óptica y Electrónica

Supervisada por:

**Dr. Gonzalo Urcid Serrano**

©INAOE – Mayo 2009

El autor otorga al INAOE el permiso de reproducir y distribuir copias en su totalidad o en partes de esta tesis



# Índice general

<b>Prólogo</b>	<b>IV</b>
<b>Resumen</b>	<b>VI</b>
<b>Abstract</b>	<b>VII</b>
<b>Dedicatorias</b>	<b>VIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Problema propuesto . . . . .	1
1.2. Antecedentes . . . . .	3
1.2.1. Antecedentes históricos . . . . .	3
1.2.2. Antecedentes recientes . . . . .	4
1.3. Organización de la tesis . . . . .	5
<b>2. Memorias Asociativas</b>	<b>7</b>
2.1. Introducción . . . . .	7
2.2. Memorias asociativas lineales . . . . .	9
2.2.1. Estructura básica . . . . .	9
2.2.2. Almacenamiento y recuperación . . . . .	10
2.2.3. Recuperación de imágenes binarias . . . . .	12
2.3. Memoria recurrente de Hopfield . . . . .	13
2.3.1. Red neuronal artificial de Hopfield . . . . .	14
2.3.2. Almacenamiento y recuperación . . . . .	14
2.3.3. Recuperación de imágenes binarias . . . . .	17
2.4. Memoria de Hamming clasificadora . . . . .	19
2.4.1. Red neuronal artificial de Hamming . . . . .	19
2.4.2. Almacenamiento y recuperación . . . . .	20

2.4.3.	Recuperación de imágenes binarias . . . . .	22
<b>3.</b>	<b>Memorias Asociativas Minimax</b>	<b>25</b>
3.1.	Preliminares matemáticos . . . . .	25
3.1.1.	Álgebra reticular matricial . . . . .	25
3.1.2.	Sumas mín-máx . . . . .	26
3.2.	Memorias asociativas reticulares minimax . . . . .	27
3.2.1.	Almacenamiento y recuperación . . . . .	28
3.2.2.	Recuperación de imágenes . . . . .	30
3.3.	Método de núcleos . . . . .	33
3.3.1.	Fundamento matemático . . . . .	33
3.3.2.	Recuperación de imágenes con ruido . . . . .	36
<b>4.</b>	<b>Memorias Asociativas Difusas</b>	<b>39</b>
4.1.	Lógica difusa . . . . .	39
4.1.1.	Conjuntos difusos y su representación . . . . .	40
4.1.2.	Operaciones y propiedades básicas . . . . .	42
4.2.	Memorias morfológicas difusas . . . . .	44
4.2.1.	Fundamento teórico . . . . .	45
4.2.2.	Recuperación de imágenes binarias . . . . .	53
4.3.	Modelo híbrido MAMD-Hamming . . . . .	56
4.3.1.	Recuperación de imágenes binarias . . . . .	59
<b>5.</b>	<b>Enmascaramiento del ruido</b>	<b>63</b>
5.1.	Preámbulo . . . . .	63
5.2.	Imágenes binarias . . . . .	63
5.2.1.	Máscara positiva . . . . .	63
5.2.2.	Máscara negativa . . . . .	64
5.2.3.	Recuperación de imágenes con ruido . . . . .	67
5.3.	Imágenes en tonos de gris . . . . .	69
5.3.1.	Medidas del error . . . . .	69
5.3.2.	Recuperación de imágenes con ruido . . . . .	71
5.4.	Imágenes a color . . . . .	75
5.4.1.	Modelo de color RGB . . . . .	75
5.4.2.	Investigaciones previas . . . . .	77

5.4.3. Ruido aditivo en imágenes RGB . . . . .	78
5.4.4. Recuperación de imágenes a color . . . . .	82
<b>6. Conclusiones</b>	<b>86</b>
6.1. Conclusiones generales . . . . .	86
6.2. Conclusiones específicas . . . . .	87
6.3. Aportaciones y trabajos relacionados . . . . .	90

# Prólogo

En el presente trabajo se trata el tema del almacenamiento y recuperación de imágenes digitales que han sido distorsionadas con ruido aditivo aleatorio mediante el uso de las redes neuronales artificiales reticulares. Este tipo de redes neuronales basan su operación en el álgebra minimax y no en el álgebra lineal como lo hacen las redes neuronales artificiales tradicionales. En las redes neuronales morfológicas existe una subclase denominada memorias asociativas reticulares minimax, las cuales están basadas en dos memorias fundamentales, la memoria mín y la memoria máx. En el presente trabajo estas memorias se han empleado como memorias autoasociativas para tratar imágenes digitales que pueden ser binarias, en tonos de gris y a color. Se utilizan también algunas técnicas complementarias del reconocimiento de patrones y la lógica difusa. Al construir sistemas híbridos empleando estas técnicas se logra incrementar la robustez de las memorias autoasociativas reticulares minimax. Como parte central del trabajo de tesis se hace uso de la técnica del enmascaramiento del ruido, la cual se aplica tanto en imágenes binarias como en tonos de gris y mediante una extensión se tratan imágenes a color en el sistema basado en los colores primarios rojo, verde y azul, canales que son procesados de manera individual. Además, se establecen comparaciones con diversos modelos de memorias asociativas según el tipo de imagen, haciendo énfasis en aquellos que tratan el caso de las imágenes a color con ruido aleatorio tipo impulsivo.

Quiero señalar mi más sincero agradecimiento al CONACYT que mediante la Beca No. 181663 hizo posible que cursara el Doctorado en el INAOE, institución a la que agradezco todas las facilidades que me brindaron para terminar mis estudios de postgrado. A mis sinodales, Dr. Alejandro Cornejo Rodríguez, Dr. José Javier Báez Rojas, Dr. Fermín Granados Agustín, Dr. Carlos Alberto Reyes García y a la Dra. Rocío Morales Salgado por el tiempo, el interés y dedicación que emplearon en la revisión de este trabajo. Al Dr. Gonzalo Urcid Serrano mi más sincera gratitud por su amistad, confianza, tiempo, dedicación, consejos y paciencia en la revisión y corrección de la tesis ya que sus comentarios y sugerencias sirvieron para enriquecer y mejorar este documento.

**M.C. José Angel Nieves Vázquez**

**Mayo 8, 2009**

# Agradecimientos

Agradezco también a todos los amigos con los cuales compartí clases y momentos de recreación mediante el deporte (que por cierto me lastimé la rodilla). A mis amigos de la Biblioteca quienes siempre están dispuestos a ayudar y encontrar la información que requerimos para nuestros trabajos.

A todos mis maestros de postgrado los cuales han contribuido en mi formación profesional e individual y que mediante sus consejos han dejado huella en mi vida.

Quiero agradecer también al grupo de trabajo de la Coordinación de Óptica por su amistad, confianza, apoyo y orientación en cada una de las dudas que tenía cuando realizaba algún trámite. Así como también al personal del Departamento de Control Escolar quienes siempre me brindaron su amistad y me ayudaron cada vez que lo requería.

No puedo dejar de mencionar a mis amigos más cercanos con los cuales compartí no solo el cubículo o la casa sino también momentos alegres y tristes. Gracias amigos por soportarme.

Desde luego también a todas aquellas personas que me brindaron su apoyo de manera desinteresada para culminar esta importante etapa en mi vida.

Escudo es la ciencia, y escudo es el dinero; pero la sabiduría excede en que da vida a sus poseedores.  
*Eclesiastés 7:12*

# Resumen

En este trabajo se desarrollan diversos modelos de memorias asociativas reticulares minimax capaces de resolver el problema de almacenamiento y recuperación de imágenes digitales, ya sean binarias, en escala de grises o a color, las cuales pueden estar o no distorsionadas con ruido aleatorio aditivo. Las imágenes originales o “ejemplares” son almacenadas y recuperadas por la memoria como vectores. El procesamiento en esta clase de memorias asociativas se basa en operaciones vectoriales y matriciales del álgebra minimax. Se describen diversas técnicas que, dependiendo del tipo de imagen, logran mejorar el desempeño respecto de otros modelos de memorias asociativas ya establecidos. Para imágenes binarias se propone una memoria asociativa morfológica difusa mejorada, que emplea la función indicatriz de Kosko, un procedimiento de umbralización y la distancia de Hamming. El ruido aleatorio se considera como ruido mixto en el contexto de ésta álgebra ya que es una mezcla de ruido erosivo y dilatativo. En general presenta un problema serio en la etapa de recuperación ya que las memorias autoasociativas reticulares son robustas ante un tipo de ruido solamente (erosivo o dilatativo). Como parte central de la tesis se describe el enmascaramiento del ruido que consiste en cambiar una imagen binaria o en tonos de gris con ruido aleatorio en una imagen que contiene un solo tipo de ruido, erosivo o dilatativo pero no ambos; además, permite utilizar las memorias mínima o máxima de manera indistinta. Adicionalmente, se extiende la técnica del enmascaramiento del ruido para tratar imágenes a color RGB ya que éstas pueden tratarse como imágenes en escala de grises en tres canales los cuales pueden procesarse independientemente. En cada uno de los modelos de memorias asociativas se presentan ejemplos representativos de su desempeño ante ruido aleatorio (tipo impulsivo) y se proporcionan los algoritmos correspondientes en pseudocódigo matemático. Finalmente, se establece una comparación con modelos recientes de memorias asociativas empleadas para almacenar y recuperar imágenes a color RGB y el modelo aquí propuesto.

# Abstract

In this work various models of lattice associative memories are developed that solve the problem of storage and recall of digital images, be they binary, gray scale, or color images, which may or may not be distorted with additive random noise. The original images or “exemplars” are stored and recalled by the memory as vectors. In these kind of associative memories processing is based primarily on minimax algebra vector and matrix operations. Several techniques are described that, depending on the kind of image, achieve a better performance with respect to other associative memory models already established. For binary images, an enhanced fuzzy autoassociative morphological memory is proposed, that employs Kosko’s indicator function, a thresholding procedure, and the Hamming distance. Random noise is considered as mixed noise in the context of minimax algebra, since it is a mixture of erosive and dilative noise. Generally, it presents a serious problem during the recall stage because the lattice auto-associative memories are robust to only one type of noise (erosive or dilative). The central part of this thesis describes noise masking consisting in changing a binary or gray scale image with random noise into an image with only just one kind of noise, erosive or dilative but not both; also, it allows the use of the minimum or maximum memories indistinctively. Additionally, an extension of the noise masking technique is given to deal with RGB color images since these images can be considered as gray scale images in three channels that can be processed independently. In each one of the associative memories models, representative examples of their performance are provided for inputs with random noise (impulse type) together with the corresponding algorithms written in mathematical pseudocode. Finally, a comparison is established between recent associative memory models used to store and retrieve RGB color images and the model proposed here.



# Dedicatorias

A Dios creador del universo, quien me ha dado a mis padres, esposa, familiares, amigos y todo lo que poseo. Gracias Señor por que sanas y restauras nuestras vidas y en los momentos de soledad y tristeza siempre estás con nosotros para darnos consuelo y fortaleza. Señor “enseñanos a contar nuestros días, de tal modo que traigamos al corazón sabiduría.” *Salmo 90:12*.

A mis padres, Genaro Nieves Santos y María del Refugio Vázquez Guzmán por que me enseñaron, mediante el ejemplo, el valor del trabajo y la responsabilidad. Gracias Papá, gracias Mamá, sus sacrificios y privaciones han dado frutos, no los defraudaré y les amo.

Desde luego a mi oración contestada, mi esposita bonita, Paty Nieves, quien es mi ayuda idónea, mi complemento. Gracias amor por tus oraciones, paciencia y comprensión, se que no ha sido fácil y quiero que sepas que todo lo hago por nuestro hogar, te amo y te amaré toda la vida. Amor, contigo me siento completo.

A mis hermanos, hermanas, familiares y amigos quienes a través de los momentos que convivimos han dejado huella en mí y siguen siendo parte fundamental en el desarrollo y crecimiento de mi vida.

No podrían faltar los hermanos de San Bernardino Tlaxcalancingo, quienes hicieron que mi estancia en Puebla fuera una gran experiencia, aprecio mucho todo lo que hicieron por mí.

# Capítulo 1

## Introducción

### 1.1. Problema propuesto

El *procesamiento digital de imágenes* (PDI) es una área del conocimiento cuyo objetivo es mejorar la información contenida en imágenes binarias, en tonos de gris o a color, con la finalidad de que el ser humano sea capaz de analizar, extraer e interpretar la información de cualquier imagen que se encuentra bajo estudio. Uno de los problemas en esta área tiene que ver con el tratamiento de imágenes ruidosas, entendiéndose como imagen ruidosa aquella donde sus píxeles originales han sido distorsionados o cambiados. En PDI existen una gran cantidad de filtros lineales y no lineales que tratan este problema y muchos de los cuales se han vuelto clásicos (promedio, mediana y Wiener [15, 24, 48]). A pesar de la utilidad que tienen los filtros clásicos lineales y no lineales, el problema planteado no es fácil de resolver, tal es el caso de imágenes que contienen un alto porcentaje de ruido, por lo que se han incorporado nuevas herramientas que han ayudado en el tratamiento de imágenes ruidosas. Una de estas herramientas que se han utilizado en PDI son las *redes neuronales artificiales* (RNA) que junto con el *reconocimiento de patrones* (RP) y la *lógica difusa* (LD) han dado resultados interesantes en la solución parcial al tratar con imágenes binarias; algunos de estos ejemplos son presentados por Bow [7], Lippmann [33], Ritter & Wilson [53]. Otros autores que presentan modelos que usan imágenes binarias son Kasabov [26], Haykin [20] y Dayhoff [13]. Ritter [52] al igual que Sussner [55, 57] y Urcid [63, 64], tratan las imágenes binarias utilizando álgebra reticular minimax en lugar del álgebra lineal. Otro modelo de memoria asociativa basado en estructuras de orden parcial ha sido desarrollado y aplicado a imágenes binarias por Yañez [70]. El caso de las imágenes en tonos de gris utilizando álgebra reticular minimax ha sido estudiado por Ritter [54], Urcid [62, 63, 65, 66] y Sussner [58, 59]. Existen varios modelos de redes neuronales

artificiales [20] y cada uno de ellos opera de una forma distinta, la operación depende en gran medida de las características de la red como son; la estructura (topología), el tipo de conexión (pesos sinápticos), el modo de aprendizaje (supervisado y no supervisado), el tipo de patrones (binarios o en tonos de gris). El funcionamiento de cada uno de los modelos se basa en la idea de la neurona artificial presentada por McCulloch y Pitts, así como del perceptrón propuesto por Rosenblatt [46]. Los perceptrones y las memorias asociativas forman dos grandes grupos de redes neuronales. Un perceptrón comúnmente se utiliza como clasificador y las memorias asociativas se emplean por lo general para almacenar y recuperar información. Se han alcanzado resultados aceptables al aplicar las memorias asociativas al problema de la recuperación de imágenes afectadas con ruido ya que existen limitaciones relativas a su capacidad de almacenamiento y recuperación [33, 52, 70]. El objetivo del presente trabajo es *proponer variantes novedosas al modelo de las memorias asociativas reticulares para aplicarlas al problema del almacenamiento y recuperación de imágenes digitales que pueden estar o no distorsionadas con ruido aleatorio*. Las imágenes originales, las cuales pueden ser binarias, en escala de grises o a color, que también denominamos como imágenes ejemplares son almacenadas y recuperadas por la memoria en forma de vectores. El modo de operación de estas memorias asociativas se basa en operaciones vectoriales y matriciales del álgebra minimax. Se describen diversas técnicas que dependiendo del tipo de imagen logran mejorar el desempeño respecto de los modelos de memorias asociativas reticulares ya establecidos. Para imágenes binarias se propone una memoria asociativa reticular difusa mejorada, que emplea la función indicatriz de Kosko, un procedimiento de umbralización y la distancia de Hamming. El ruido aleatorio se considera como ruido mixto en el contexto del álgebra minimax puesto que está formado por una mezcla de ruido erosivo y dilatativo, y en general es un problema serio ya que las memorias asociativas reticulares canónicas son robustas ante un tipo de ruido solamente, ya sea erosivo o dilatativo. Como parte central de la tesis se propone el enmascaramiento del ruido que consiste en cambiar una imagen binaria o en tonos de gris que contenga ruido aleatorio en una imagen con un solo tipo de ruido que resulta ser erosivo o dilatativo pero no ambos, y permite utilizar las memorias mínima o máxima de manera indistinta. Adicionalmente, se extiende la técnica del enmascaramiento del ruido para tratar imágenes a color RGB ya que pueden tratarse como imágenes en tonos de gris que pueden ser procesadas de forma independiente. En cada uno de los modelos propuestos se presentan ejemplos representativos de su desempeño ante ruido aleatorio de tipo impulsivo y se proporcionan los algoritmos correspondientes en pseudocódigo matemático.

## 1.2. Antecedentes

### 1.2.1. Antecedentes históricos

El PDI y el RP son áreas donde las RNA han tenido buenos resultados al tratar problemas en el procesamiento de imágenes con ruido o distorsión [25]. El modelo matemático de una RNA surge con la idea de imitar algunos mecanismos del cerebro humano como, por ejemplo, almacenar y recuperar información mediante asociaciones y poder trasladar estos mecanismos en máquinas artificiales. Otro de los descubrimientos que le dió mayor importancia a las RNA fue el hecho por Hebb quien en 1949 propone un esquema de aprendizaje conocido como regla de aprendizaje de correlación. El postulado de aprendizaje de Hebb [43] establece que la activación repetida de una neurona por otra a través de una sinapsis aumenta la probabilidad de ser activada posteriormente. Otro de los avances en esta área fue dado por Widrow y Hoff quienes propusieron un algoritmo de aprendizaje basado en la minimización de la suma del error a la salida de la red. En 1969, Minsky y Papert [37] publicaron un libro donde demostraron matemáticamente las limitaciones del perceptrón de Rosenblatt y como resultado llegaron a la conclusión de que los perceptrones multicapa sufrirían del mismo problema. Sin embargo, se mantenían las investigaciones tratando de encontrar una solución al problema planteado por Minsky y Papert, a medida que se involucraban los investigadores se iban encontrando soluciones parciales al problema. Algunos de los cuales fueron: Werbos [46], quien propuso un nuevo esquema de entrenamiento para el perceptrón multicapa, Amari [1] desarrollo un modelo adaptativo de umbralización de la neurona y utilizó dicho modelo para estudiar el comportamiento dinámico de las redes aleatorias. Nakano [46] propuso un modelo de memoria asociativa a la cual llamo asociatrón; por otra parte, Kohonen [28, 29] y Anderson [3] definieron el concepto de asociación como un reforzamiento entre neuronas. No fue si no hasta la década de los ochenta que hay un resurgimiento en el área de las redes neuronales como resultado de la publicación de algunos trabajos, principalmente los de Hopfield [21, 22], sobre redes neuronales recurrentes equivalentes a memorias asociativas también conocidas como memorias direccionables por contenido.

Los elementos que componen una neurona artificial, modelados a partir del funcionamiento de una neurona biológica se encuentran representados en la Fig. 1.1:

- la neurona recibe en *paralelo* un conjunto finito de estímulos o señales  $x_i$  para  $i = 1, \dots, n$ , provenientes de otras neuronas,
- la intensidad sináptica correspondiente a la señal  $x_i$  se designa como el *peso*  $w_i$  que puede ser positivo (excitación) o negativo (inhibición),

- la neurona realiza una *suma ponderada* considerando todos los estímulos de entrada así como sus pesos correspondientes y responde mediante una *función de activación* que decide en base a un umbral  $u$  si la neurona se dispara o no, y
- las funciones de activación pueden introducir *no-linealidad* en el cómputo neuronal y las más usuales son el limitador discontinuo de umbral (función escalón de Heaviside) y la función sigmoideal.

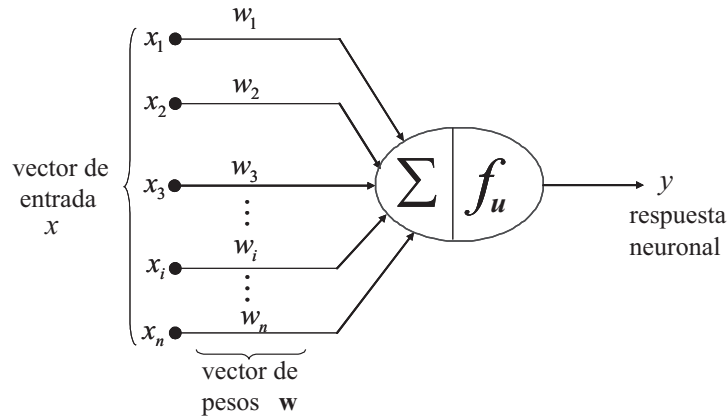


Figura 1.1: Modelo clásico de una neurona artificial.

El modelo que dió inicio a la incursión de las redes neuronales tanto en PDI como en RP fue el perceptrón multicapa. Como ya se mencionó, Hopfield fue uno de los precursores en el uso de un modelo de memoria asociativa para la reconstrucción de estados estables. Sin embargo, este modelo de memoria asociativa resultó estar limitado en el almacenamiento y recuperación de información.

### 1.2.2. Antecedentes recientes

Posteriormente se han propuesto las memorias asociativas morfológicas o reticulares minimax (MARM), subclase de RNA, que fue introducida por Ritter, Sussner y Díaz de León [52] hace un poco más de una década. Las MARM han encontrado aplicaciones en PDI y RP y resuelven, de manera parcial, el problema del almacenamiento y recuperación de imágenes digitales. Aplicaciones adicionales de las MARM han sido presentadas por: Graña [16], Guzmán [19], Raducanu [49] y Villaverde [68]. Las MARM forman una subclase de las redes neuronales reticulares y sus operaciones se basan en dos memorias fundamentales, la memoria mín  $W$  y la memoria máx  $M$ .

La memoria mín  $W$  es capaz de recuperar imágenes con ruido erosivo y la memoria máx  $M$  es capaz de recuperar imágenes con ruido dilatativo, sin embargo, para el caso de ruido mezclado existe error de confusión en la recuperación de una imagen como lo demuestran Ritter [52, 54], Sussner [55] y Urcid [62]. Con la finalidad de dar solución al problema, se han realizado nuevas investigaciones utilizando algunas variantes en los modelos de las memorias asociativas reticulares; una de las primeras ideas en el tratamiento de imágenes con ruido mezclado fue el empleo de la técnica de núcleos para vectores binarios propuesta por Ritter, Sussner y Díaz de León [52]. Más tarde, Sussner [55] propuso un algoritmo para encontrar los núcleos en un conjunto de imágenes binarias, así como un modelo híbrido basado en la técnica de núcleos y la lógica difusa [57]. Asimismo, el trabajo presentado por Wang y Lu [69] muestra una comparación entre las memorias asociativas morfológicas y una versión difusa de éstas (distinta a la propuesta por Sussner). En [64] presentamos un nuevo modelo híbrido basado en el trabajo de Sussner [57], que utiliza la distancia de Hamming para la fase de recuperación mejorando así los resultados obtenidos anteriormente. El caso de imágenes en tonos de gris fue expuesto por primera vez por Ritter, Urcid y Iancu en [54]. En [43, 62] se introduce y se desarrolla una nueva técnica de recuperación basada en la propiedad de independencia fuerte morfológica. Una idea reciente fue presentada por Urcid [63] la cual propone la técnica del enmascaramiento del ruido para utilizar cualquiera de las dos memorias asociativas reticulares. Consiste en separar (desmezclar) el ruido aleatorio por medio de un enmascaramiento del vector con ruido con cada uno de los vectores ejemplares almacenados en la memoria. Los resultados alcanzados en el tratamiento de las imágenes binarias y en tonos de gris son satisfactorios ya que se ha incrementado la robustez de este tipo de memorias en presencia de ruido aleatorio. En esta tesis se propone una variante al enmascaramiento del ruido para mejorar el desempeño y se extiende el modelo al almacenamiento y recuperación de imágenes a color.

### 1.3. Organización de la tesis

El contenido del trabajo de tesis está organizado como sigue: el Capítulo 1 da una introducción al tema central a tratar así como una descripción de la naturaleza del trabajo a realizar; en el Capítulo 2 se detallan tres modelos de redes neuronales basadas en las memorias asociativas, la memoria asociativa lineal, las memorias asociativas recurrentes y finalmente las memorias asociativas clasificadoras; el Capítulo 3 proporciona los fundamentos del álgebra reticular minimax, la cual forma la base matemática de las redes neuronales reticulares. Se describen las memorias asociativas reticulares, algunas de sus propiedades así como la técnica de núcleos utilizando la independencia fuerte morfológica inducida; el Capítulo 4 proporciona los fundamentos de la teoría

de conjuntos difusos que son la base conceptual para las memorias reticulares difusas; en este apartado se describe un nuevo modelo de memorias híbridas difusas y se dan algunos ejemplos para mostrar su desempeño; el trabajo de investigación se desarrolla en el Capítulo 5 donde se expone la descripción matemática de la técnica del enmascaramiento del ruido realizándose algunas variantes a la idea original y presentando ejemplos con la finalidad de evaluar el desempeño de ésta nueva técnica. En este mismo capítulo se presenta una extensión para tratar el caso de la recuperación de imágenes a color RGB contaminadas con ruido aleatorio tipo impulsivo y se compara con algunos modelos que tratan el mismo problema. Finalmente, en el Capítulo 6 se presentan las conclusiones del trabajo donde se expone de manera resumida el desempeño de las diferentes redes neuronales que se utilizaron como memorias asociativas.

# Capítulo 2

## Memorias Asociativas

### 2.1. Introducción

Las *memorias asociativas* [20] (MA) aparecieron por primera vez en los años 50. El primero en realizar investigaciones al respecto fue Taylor en 1956, posteriormente Steinbuch en 1961 introduce la matriz de aprendizaje y en 1969 aparece un artículo donde se presenta un modelo de memoria asociativa no holográfica propuesto por Willshaw, Buneman y Longuet-Higgins.

¿Qué entendemos por memoria asociativa? Para responder a esta pregunta consideremos lo siguiente: el ser humano diariamente realiza procesos de asociación, por ejemplo asociamos nombres con rostros de personas, olores con sabores, es decir, formamos asociaciones entre personas, eventos y lugares, entre formas, objetos y conceptos. Las señales que entran a nuestro cerebro mediante los sentidos nos permiten percibir el mundo a nuestro alrededor, cada señal la podemos ver como información almacenada mediante un proceso de aprendizaje. Es evidente que nuestra memoria trabaja en un modo asociativo por lo que se puede describir a la memoria humana como un sistema que almacena y recupera información mediante asociaciones. Desde el punto de vista de los modelos computacionales la información, en una memoria asociativa, es vista de acuerdo a las propiedades que guardan sus valores más que por su dirección [50]. No debe confundirse la memoria asociativa con la memoria de las computadoras digitales, la diferencia radica en que mientras la primera accesa la información de manera directa pero distribuida, la segunda lo hace de manera indirecta a través del cálculo de una dirección [38], es decir, las memorias de las computadoras digitales recuperan la información utilizando índices, mientras que en las memorias asociativas el proceso de recuperación se realiza utilizando información parcial o total de su contenido. Supóngase que se tienen  $n$  datos con una cierta estructura y cada dato lo representamos por  $P_i$ , donde  $i = 1, \dots, n$ . La



memoria de la computadora digital accede al dato  $P_i$  mediante el subíndice  $i$  (dirección calculada de memoria), mientras que la memoria asociativa lo hace utilizando información parcial o total del propio dato  $P_i$ . Las operaciones que emplean las memorias asociativas se desarrollan mediante matrices y vectores, por lo que es necesario transformar las imágenes de su forma matricial a vectores. Cada imagen es vista entonces como un *vector patrón*, *patrón ejemplar* o simplemente como *vector*, y la conversión de estructura se logra ya sea apilando renglones o columnas. A continuación, se describe esta simple transformación por renglones. Supongamos que tenemos una matriz  $A$  de  $m \times n$  que representa una imagen digital (binaria o en tonos de gris)

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

la cual transformamos en un vector  $\mathbf{x}$  colocando cada uno de sus renglones uno tras otro mediante la relación

$$x_{n(k-1)+l} = a_{kl} ; k = 1, \dots, m \text{ y } l = 1, \dots, n. \quad (2.1)$$

Por lo que la matriz  $A$  puede ser vista como un vector  $\mathbf{x}$  de dimensión  $mn$ . Las memorias asociativas se clasifican en dos grupos, memorias *autoasociativas* (MAA) y *heteroasociativas* (MHA). La Fig. 2.1 muestra de manera esquemática el funcionamiento de una memoria asociativa que almacena pares de vectores  $(\mathbf{x}^\xi, \mathbf{y}^\xi)$ , con  $\mathbf{x}^\xi = (x_1^\xi, \dots, x_n^\xi)^T$  e  $\mathbf{y}^\xi = (y_1^\xi, \dots, y_n^\xi)^T$  para  $\xi = 1, \dots, k$ , siendo  $k$  el número de vectores que se almacenan en la memoria, y que llamaremos *vectores asociados ejemplares*.

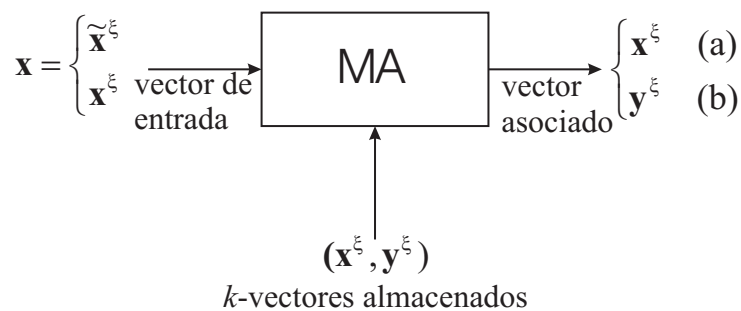


Figura 2.1: Esquema de una memoria asociativa: (a) memoria autoasociativa  $(\mathbf{x}, \mathbf{x}^\xi)$ , (b) memoria heteroasociativa  $(\mathbf{x}, \mathbf{y}^\xi)$ .

En la Fig. 2.1 el vector de entrada  $\mathbf{x}$  puede ser un ejemplar  $\mathbf{x}^\xi$ , o una versión con ruido de éste  $\tilde{\mathbf{x}}^\xi$ . En el primer caso se denomina *entrada perfecta* de lo contrario hablamos de *entrada imperfecta*.

## 2.2. Memorias asociativas lineales

La *memoria asociativa lineal* (MAL) [9, 28, 29] es uno de los modelos derivado de los primeros estudios sobre memorias asociativas. En este trabajo de tesis se emplean las MAL para el reconocimiento de imágenes binarias.

### 2.2.1. Estructura básica

La memoria asociativa lineal es una red que contiene una capa de nodos de entrada y una capa de neuronas de salida. Cada elemento del vector de entrada es conectado a todas las neuronas en la capa de salida.

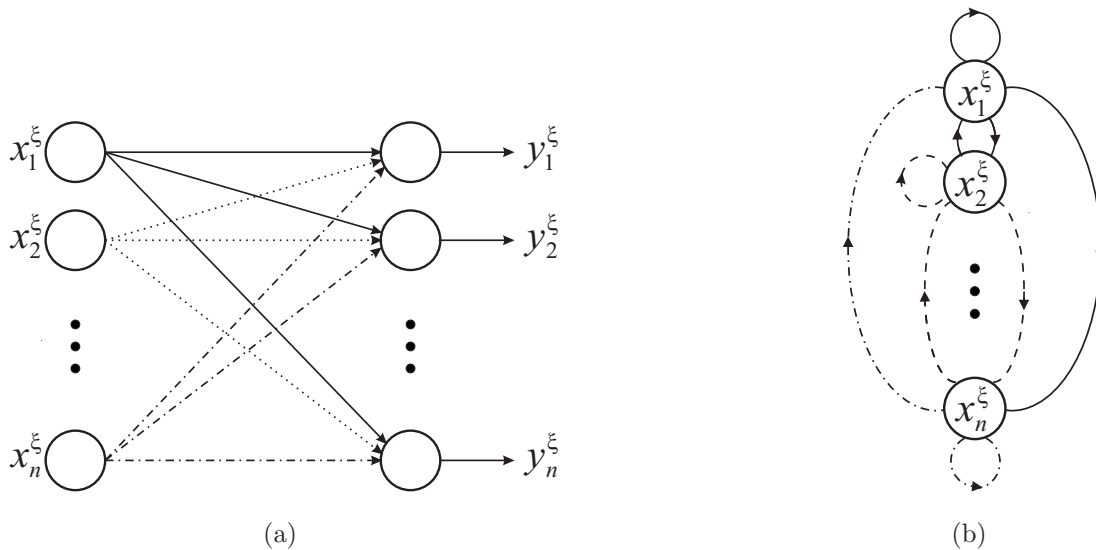


Figura 2.2: Red neuronal de una memoria asociativa lineal: (a) memoria heteroasociativa MHA, (b) memoria autoasociativa MAA.

La Fig.2.2 muestra la memoria asociativa lineal (memoria heteroasociativa MHA y memoria autoasociativa MAA), donde la entrada y salida están representadas por los vectores  $\mathbf{x}^\xi = (x_1^\xi, x_2^\xi, \dots, x_n^\xi)^T$  e  $\mathbf{y}^\xi = (y_1^\xi, y_2^\xi, \dots, y_n^\xi)^T$ . Dado que aquí se utilizan las MAL como MAA, los vectores de entrada y de salida tendrán el mismo espacio dimensional, por tanto decimos que la red asocia vectores de entrada con vectores ejemplares que han sido almacenados previamente en la memoria. La codificación de los vectores de entrada se realiza en forma bipolar  $\{-1, 1\}$ .

### 2.2.2. Almacenamiento y recuperación

En la etapa de almacenamiento, también conocida como fase de aprendizaje o entrenamiento, los vectores de entrada son almacenados por medio del postulado de Hebb. La matriz de pesos sinápticos es el resultado de la operación matricial entre los vectores  $\mathbf{y}^\xi$  y  $\mathbf{x}^{\xi T}$  conocida también como el *producto externo*. La matriz de pesos  $\mathbf{W}^\xi$  en forma matricial está dada por la Ec. (2.2).

$$\mathbf{W}^\xi = \mathbf{y}^\xi \mathbf{x}^{\xi T}, \quad (2.2)$$

donde  $\mathbf{W}^\xi$  es una matriz de tamaño  $n \times n$  y  $(\mathbf{x}^\xi, \mathbf{y}^\xi)$  son los pares de vectores  $n$ -dimensionales, i.e.,  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . La matriz  $\mathbf{W} = \sum_{\xi=1}^k \mathbf{W}^\xi$  puede definirse componente a componente como

$$w_{ij}^\xi = \sum_{\xi=1}^k y_i^\xi x_j^{\xi T} \quad ; \quad i, j = 1, \dots, n \quad (2.3)$$

siendo  $k$  el número de vectores ejemplares,  $ij$  representa la  $ij$ -ésima componente del  $\xi$ -ésimo vector del par de vectores  $(\mathbf{x}^\xi, \mathbf{y}^\xi)$ . El número de neuronas que contiene la red se encuentra determinado por el número de elementos que componen cada vector. La capacidad de almacenamiento en una memoria asociativa lineal lo determina el rango de la matriz, es decir, dado que la matriz de pesos  $\mathbf{W}^\xi$  es una matriz de tamaño  $n \times n$  su capacidad de almacenamiento se encuentra limitada por  $n$ . Una vez que se ha construido la matriz de pesos es posible recuperar uno de los vectores ejemplares contenidos en la memoria de la red. Si se tiene un conjunto de pares de vectores almacenados en la red  $(\mathbf{x}^\xi, \mathbf{y}^\xi)$ , con  $\xi = 1, \dots, k$  y se presenta un vector de entrada  $\mathbf{x}^\xi$ , la respuesta a la salida es el vector asociado  $\mathbf{y}^\xi$ . La siguiente expresión matricial muestra la forma de realizar la operación

$$\mathbf{y}^\xi = \mathbf{W}^\xi \mathbf{x}^\xi. \quad (2.4)$$

La expresión componente a componente puede representarse como

$$y_i^\xi = \sum_{j=1}^n w_{ij}^\xi x_j^\xi, \quad (2.5)$$

donde  $w_{ij}$  con  $j = 1, 2, \dots, n$ , son los pesos sinápticos de la neurona  $i$  correspondientes al  $\xi$ -ésimo par de vectores asociados. En la Fig. 2.3 se muestra en detalle la matriz de pesos sinápticos y el arreglo para la neurona  $i$  en la capa de salida. La *recuperación perfecta* en la red se logra cuando los vectores de entrada son ortonormales. Sea  $\mathbf{W}$  la matriz de pesos de la memoria asociativa, la cual ha almacenado  $k$  vectores ortonormales, i.e.

$$\mathbf{x}^{\xi T} \mathbf{x}^k = \begin{cases} 1 \Leftrightarrow \xi = k \\ 0 \Leftrightarrow \xi \neq k, \end{cases} \quad (2.6)$$

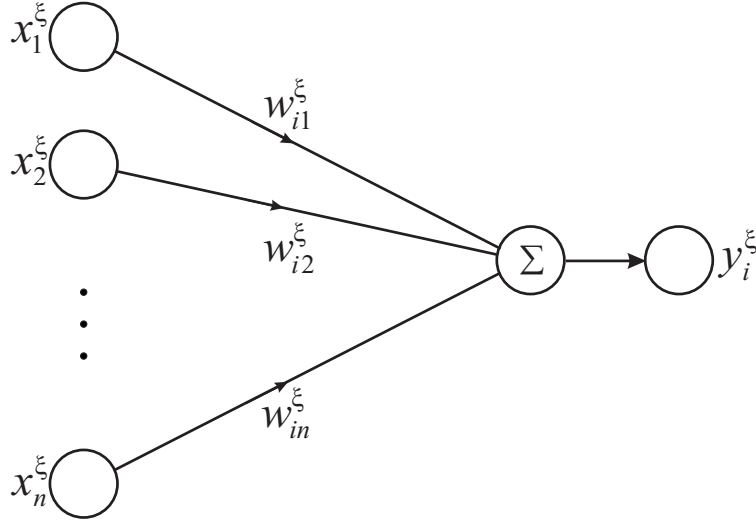


Figura 2.3: Modelo de una memoria asociativa lineal para la neurona  $i$ .

entonces

$$\mathbf{W}\mathbf{x}^k = \left( \sum_{\xi=1}^k \mathbf{y}^\xi \mathbf{x}^{\xi T} \right) \mathbf{x}^k = \sum_{\xi \neq k} \mathbf{y}^\xi (\mathbf{x}^{\xi T} \mathbf{x}^k) + \mathbf{y}^k (\mathbf{x}^{k T} \mathbf{x}^k) = \mathbf{y}^k. \quad (2.7)$$

Si  $\mathbf{x}^1, \dots, \mathbf{x}^k$  no es un conjunto de vectores ortonormales, lo que es más común en un conjunto arbitrario, entonces el término

$$\mathbf{r} = \sum_{\xi \neq k} \mathbf{y}^\xi (\mathbf{x}^{\xi T} \mathbf{x}^k) \neq \mathbf{0} \quad (2.8)$$

representa el *vector de ruido*. Observamos que el vector de ruido impide la recuperación perfecta sin importar si la entrada es perfecta ( $\mathbf{x}^\xi$ ) o imperfecta ( $\tilde{\mathbf{x}}^\xi$ ), hecho que disminuye el desempeño de la memoria asociativa en la etapa de recuperación. El empleo de la memoria asociativa lineal está descrito en pseudocódigo matemático en el Algoritmo 1 (implementado en **Mathcad**).

El conjunto de ejemplares  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \{-1, 1\}^n$  consta de vectores binarios codificados polarmente, el cual es una matriz de tamaño  $n \times k$ , donde  $n$  representa la dimensión del vector y  $k$  es el número de vectores. La matriz de pesos  $W$  se calcula como  $W = \sum_{\xi=1}^k \mathbf{x}^\xi \cdot \mathbf{x}^{\xi T}$ , donde la  $k$ -ésima columna la denotamos  $W^k$  y la  $l$ -ésima componente de este vector lo denotamos por  $w_l^k = w_{lk}$ .

**Input:**  $\tilde{\mathbf{x}} \in \{-1, 1\}^n$

[vector con ruido aleatorio]

$\mathbf{y} = W\tilde{\mathbf{x}}$

[etapa de recuperación]

**Output:**  $\mathbf{y} \in \{-1, 1\}^n$

[vector asociado]

**Algoritmo 1:** MAL - Memoria Asociativa Lineal

### 2.2.3. Recuperación de imágenes binarias

Las imágenes binarias se pueden codificar en dos formas: en la primera el objeto es blanco sobre fondo negro de modo que un pixel de fondo (negro) tiene valor 0 y un pixel objeto (blanco) tiene valor 255; en la segunda, empleada en la presente tesis, el objeto es negro sobre fondo blanco. Dado que la imagen puede tomar solo dos valores, el conjunto que representa a dicha imagen es  $\{0,1\}$ . En el siguiente par de ejemplos se muestran las capacidades y limitaciones del modelo asociativo lineal.

**Ejemplo 2.1.** El modelo almacena 5 imágenes binarias las cuales se muestran en el primer renglón de la Fig. 2.4. Cada imagen es de tamaño  $32 \times 32$  pixeles de modo que la dimensión del vector es  $n = 1024$ . El número de elementos corresponde al número de neuronas en la red. Todos los vectores son ortonormales (a propósito). El vector con ruido que ingresa a la red es  $\tilde{\mathbf{x}}^2$ . El tercer renglón de la Fig. 2.4 muestra los resultados del modelo después de procesar este vector. En la 4ta y 5ta columnas de este renglón se observa que la red se confunde presentando a la salida el vector  $\mathbf{x}^1 + \mathbf{x}^2$  para 45 % de ruido y el vector  $\mathbf{x}^3 + \mathbf{x}^5$  para 50 % de ruido, en lugar del vector esperado  $\mathbf{x}^2$ .

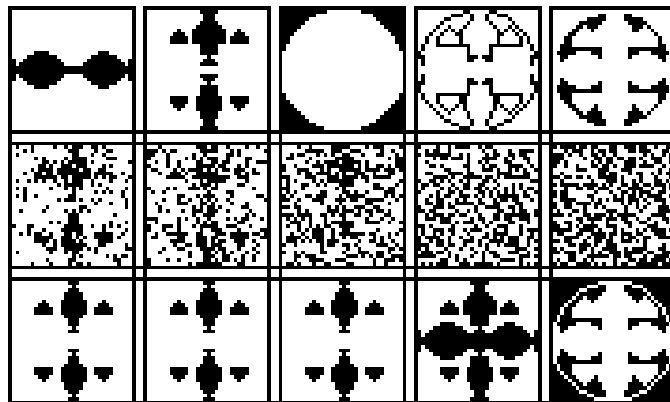


Figura 2.4: 1er. renglón, vectores ejemplares  $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5$ ; 2do. renglón, versiones de  $\mathbf{x}^2$  con 15 %, 25 %, 35 %, 45 % y 50 % de ruido aleatorio; 3er. renglón, respuesta de la MAL.

**Ejemplo 2.2.** El conjunto de vectores corresponde al Font E13B utilizado por la Asociación de Banqueros Americanos (ABA). La memoria almacena 14 imágenes binarias (ver Fig. 2.5) de  $18 \times 14$  pixeles de modo que la dimensión del vector es  $n = 252$  (número de neuronas). El vector

de entrada con ruido corresponde al número '5'. La Fig. 2.6 muestra los resultados de la MAL después de procesar este vector con diferentes niveles de ruido.

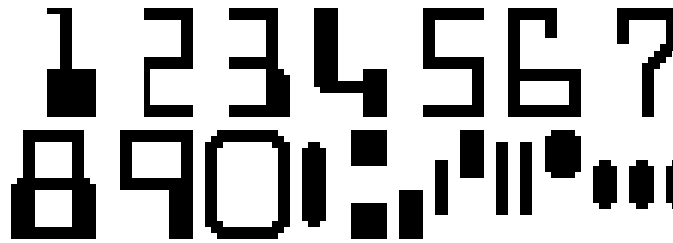


Figura 2.5: Imágenes originales del Font E13B.

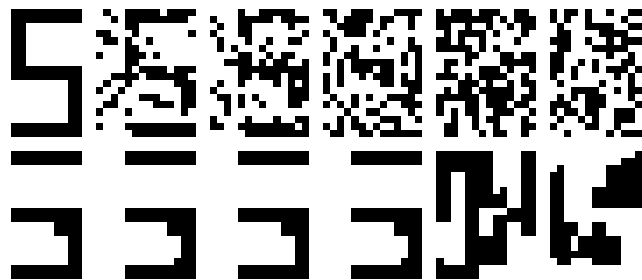


Figura 2.6: 1er. renglón, vectores de entrada a la red: imagen ejemplar del número '5', versiones contaminadas con 15 %, 25 %, 35 %, 45 % y 50 % de ruido aleatorio; 2do. renglón, respuesta de la MAL.

Del par de ejemplos anteriores se puede apreciar la capacidad de recuperación de la MAL para procesar imágenes binarias ortonormales y las limitaciones de ésta para imágenes que no son ortonormales. En el Ejemplo 2.1 se obtiene un alto desempeño de la red para porcentajes de ruido de 0 a 35 %; sin embargo, para porcentajes de ruido mayores al 35 % la memoria se confunde recuperando un vector diferente que es la superposición de vectores ejemplares. En el Ejemplo 2.2, la memoria no logra establecer la asociación correcta del vector deseado aun cuando el vector de entrada no tiene ruido, debiéndose a que los vectores ejemplares al compartir varios pixeles entre sí, no cumplen la condición dada en la Ec. (2.6).

### 2.3. Memoria recurrente de Hopfield

Una *memoria recurrente de Hopfield* (MRH) es una red de una sola capa cuyas neuronas son de entrada y salida a la vez. Una MRH consiste de  $n$  neuronas totalmente interconectadas entre sí,

donde la retroalimentación entre neuronas induce la no linealidad de la red. Uno de los modelos más importantes de red neuronal recurrente es el de Hopfield [21, 22], el cual ha sido estudiado desde el punto de vista teórico [2, 8, 34] y práctico [4, 31, 33].

### 2.3.1. Red neuronal artificial de Hopfield

La red de Hopfield se comporta como una memoria autoasociativa no lineal también conocida como *memoria direccionable por contenido*. La red puede almacenar un conjunto finito de vectores binarios de modo que si se ingresa un vector ejemplar  $\mathbf{x}^\xi$  o una versión ruidosa de éste  $\tilde{\mathbf{x}}^\xi$ , se espera que la red pueda asociarlo correctamente. La estructura básica de la red neuronal de Hopfield para 4 neuronas se muestra en la Fig. 2.7. En esta tesis consideramos únicamente el modelo discreto

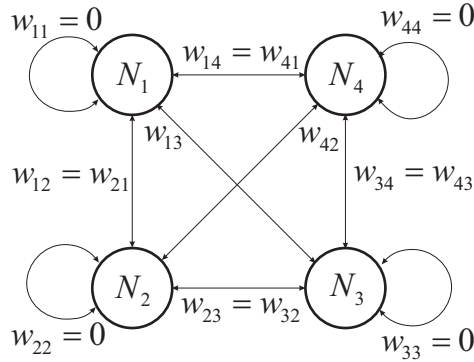


Figura 2.7: Red de Hopfield con 4 neuronas y los  $w_{ij}$  son los pesos sinápticos.

de la red de Hopfield en el cual los vectores binarios pueden codificarse de forma normal  $\{0,1\}$  o en forma bipolar  $\{-1, +1\}$ .

### 2.3.2. Almacenamiento y recuperación

El almacenamiento en la memoria se realiza por medio de la regla de aprendizaje postulada por Hebb. Supongamos que tenemos un conjunto de  $k$  vectores ejemplares  $\mathbf{x}^\xi = (x_1^\xi, x_2^\xi, \dots, x_n^\xi)^T$  que la red almacena para  $\xi = 1, 2, \dots, k$ . La construcción de la matriz de pesos sinápticos se muestra en la Ec. (2.9)

$$\mathbf{W} = \sum_{\xi=1}^k \mathbf{W}^\xi - I = \sum_{\xi=1}^k \mathbf{x}^\xi \mathbf{x}^{\xi T} - I, \quad (2.9)$$

donde  $\mathbf{W}^\xi$  es una matriz de tamaño  $n \times n$  que almacena el  $\xi$ -ésimo vector ejemplar, calculada mediante el *producto externo* dado por  $\mathbf{x}^\xi \mathbf{x}^{\xi T}$  e  $I$  es la matriz identidad del mismo tamaño. De la

Ec. (2.9) se obtiene que la salida de cada neurona en la red es retroalimentada a las otras neuronas, no hay autoconexiones ( $w_{ii} = 0$ ) y que los pesos son simétricos ( $w_{ij} = w_{ji}$ ). Estas condiciones son muy importantes ya que garantizan la convergencia<sup>1</sup> de la red [8, 34, 70]; asimismo, el número de neuronas en la red está determinado por la dimensión del vector. En la etapa de recuperación se presenta en  $t = 0$  un vector de entrada  $\mathbf{x}^s$  (vector inicial). Cada componente de entrada puede tener un valor de  $+1$  o  $-1$ . La red inicia un proceso de iteraciones en búsqueda de un estado estable y cuando el proceso termina se dice que la red converge al vector deseado. Sin embargo, debido a que existen estados estables que no guardan relación con los vectores ejemplares almacenados, la salida de la red puede converger a un vector diferente (*estado espurio*). El modelo matemático del proceso de iteración se encuentra de manera resumida en la Ec. (2.10).

$$\mathbf{x}_{k+1} = f(W\mathbf{x}_k - \mathbf{u}, \mathbf{x}_k), \quad (2.10)$$

donde la función de activación es el limitador discontinuo (abrupto)  $f(x, y)$  con  $x, y \in \mathbb{R}$  definido por la Ec. (2.11)

$$f(x, y) = \begin{cases} -1 \Leftrightarrow x < 0 \\ y \Leftrightarrow x = 0 \\ +1 \Leftrightarrow x > 0. \end{cases} \quad (2.11)$$

El subíndice  $k + 1$  en la Ec. (2.10) indica el *siguiente estado* de las neuronas y  $\mathbf{u}$  es el vector de valores de umbral cuyas componentes pueden tener el mismo valor (mismo umbral para todas las neuronas) y comúnmente se toma  $\mathbf{u} = \mathbf{0}$ . En la red de Hopfield los estados estables guardan relación directa con la *función de energía* la cual asocia a cada estado de la red una energía potencial según la siguiente expresión:

$$E(\mathbf{x}) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j. \quad (2.12)$$

El cambio de energía  $\Delta E$  debido a un cambio  $\Delta x_j$  en el estado de la neurona  $j$  está dado por

$$\Delta E(\mathbf{x}) = -\Delta x_j \sum_{i \neq j} w_{ij} x_i. \quad (2.13)$$

La respuesta de la red ante un cambio durante la etapa de recuperación causa que la función de energía disminuya hasta alcanzar un *mínimo local* correspondiente a un estado estable. Según la

---

<sup>1</sup>En el Apéndice B de la tesis doctoral de C. Yañez (p. 175) se dá la demostración detallada de como estas condiciones establecen la monotonicidad decreciente (codificación bipolar) o no creciente (codificación binaria) de la función de energía dada en la Ec. (2.12).



función de energía, cada vector ejemplar se localiza en un mínimo local; de igual forma, todos los vectores de entrada parecidos a un vector ejemplar serán atraídos hacia el mismo mínimo local. Geométricamente, la función de energía se representa por una hipersuperficie dividida en diferentes zonas de atracción asociadas a cada mínimo local. Cualquier punto situado en la hipersuperficie (entrada de la red) se desplazará posiblemente hacia un mínimo local (salida de la red). En la Fig. 2.8 se muestra gráficamente la idea para el caso de vectores en dos dimensiones. El empleo de la memoria asociativa de Hopfield está descrito en pseudocódigo matemático en el Algoritmo 2 (implementado en Mathcad).

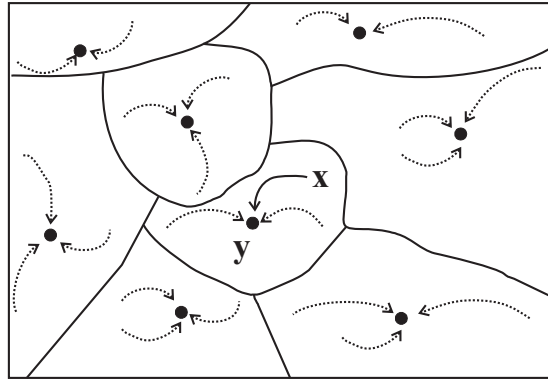


Figura 2.8: Proyección de una superficie correspondiente a una función de energía  $E(\mathbf{x})$  donde  $\mathbf{x} \in \mathbb{R}^2$ ;  $\mathbf{y}$  es el vector ejemplar y  $\mathbf{x}$  es el vector de entrada a la red.

El conjunto de ejemplares  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \{-1, 1\}^n$  consta de vectores binarios codificados polarmente, el cual es una matriz de tamaño  $n \times k$ , donde  $n$  representa la dimensión del vector y  $k$  es el número de vectores. La matriz de pesos  $W$  se calcula como  $W = \sum_{\xi=1}^k \mathbf{x}^\xi \cdot \mathbf{x}^{\xi T} - I$ , donde la  $k$ -ésima columna la denotamos  $W^k$  y la  $l$ -ésima componente de este vector la denotamos por  $w_l^k = w_{lk}$ . La función **if** (condición,  $a, b$ ) evalúa numéricamente una condición lógica verdadera con  $a$  o falsa con  $b$ . Este algoritmo supone la definición de dos funciones auxiliares. La primera corresponde a la función de activación la cual está dada por  $f(x, y) = \mathbf{if}(x = 0, y, \mathbf{if}(x > 0, 1, -1))$  y la segunda es una función que llamamos  $\text{elige}(\mathbf{v})$  que regresa una permutación del vector de enteros positivos  $\mathbf{v} = (1, \dots, n)$ .

```

Input:  $\tilde{\mathbf{x}} \in \{-1, 1\}^n$  [vector ejemplar con ruido aleatorio]
 $\mathbf{y} = \mathbf{0}$  [inicializa vector asociado]
while  $\mathbf{y} \neq \tilde{\mathbf{x}}$  do [itera hasta que converge]
   $\mathbf{y} = \tilde{\mathbf{x}}$ 
   $\mathbf{z} = \text{elige}(\mathbf{v})$  [permuta índices de neuronas]
  for  $i = 1$  to  $n$  do [recorre todas las neuronas]
     $m = z_i$  [elige una neurona al azar]
     $\tilde{x}_m = f(\tilde{x}W^m, \tilde{x}_m)$  [aplica la función de activación]
  end
end
Output:  $\mathbf{y} \in \{-1, 1\}^n$  [vector asociado]

```

**Algoritmo 2:** MRH - Memoria Recurrente de Hopfield

### 2.3.3. Recuperación de imágenes binarias

En el siguiente par de ejemplos se muestran las capacidades y limitaciones del modelo asociativo de Hopfield.

**Ejemplo 2.3.** El conjunto de imágenes utilizadas se muestran en el primer renglón de la Fig. 2.9. Cada imagen binaria de  $32 \times 32$  píxeles es un vector de dimensión  $n = 1024$  (número de neuronas). El conjunto de vectores distorsionados que son utilizados para ingresar a la red se muestran en el segundo renglón y corresponden al vector  $\tilde{\mathbf{x}}^2$ . Los resultados del modelo después del proceso de recuperación se encuentran en el tercer renglón de la Fig. 2.9. En la 4ta y 5ta columnas de éste renglón se observa que la red se confunde presentando a la salida el vector  $\mathbf{x}^3 + \mathbf{x}^5$  invertido para 45 % de ruido y el vector  $\mathbf{x}^3 + \mathbf{x}^5$  para 50 % de ruido, en lugar del vector esperado  $\mathbf{x}^2$ . Sin embargo, para los vectores que tienen ruido aleatorio menores al 35 % la red tiene un buen desempeño.

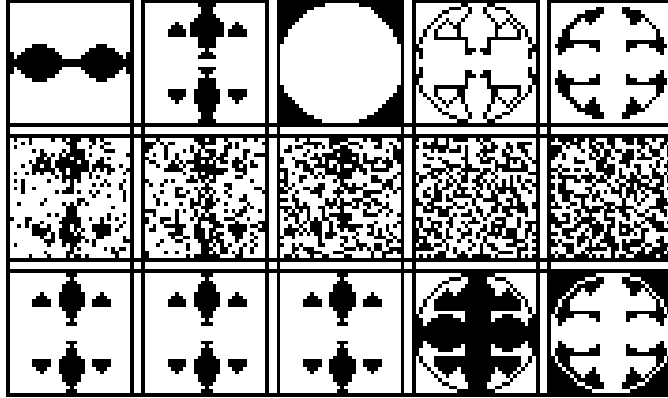


Figura 2.9: 1er. renglón, vectores ejemplares  $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5$ ; 2do. renglón, versiones de  $\mathbf{x}^2$  con 15 %, 25 %, 35 %, 45 % y 50 % de ruido aleatorio; 3er. renglón, respuesta de la red de Hopfield.

**Ejemplo 2.4.** El conjunto de imágenes utilizadas corresponden al Font E13B (ver Fig. 2.5). Cada imagen binaria es de  $18 \times 14$  pixeles de modo que la dimensión del vector es  $n = 252$  (número de neuronas). Nuevamente, la imagen distorsionada que es utilizada para ingresar a la red es el número ‘5’. La Fig.2.10 muestra los resultados del modelo después de procesar la imagen ruidosa.



Figura 2.10: 1er. renglón, vectores de entrada a la red: imagen ejemplar del número ‘5’, versiones contaminadas con 15 %, 25 %, 35 %, 45 % y 50 % de ruido aleatorio; 2do. renglón, respuesta de la red de Hopfield.

Del par de ejemplos anteriores se puede ver la capacidad de recuperación de la memoria de Hopfield para procesar imágenes binarias ortonormales (con pocos pixeles en común) y las limitaciones de ésta para imágenes que no son ortonormales (varios o muchos pixeles en común). Hay que observar que la condición de ortonormalidad de los vectores ejemplares garantizan la recuperación perfecta en las memorias asociativas lineales pero no puede garantizarse que haya re-

cuperación en el caso de la memoria asociativa de Hopfield ya que operan de forma muy diferente. En el Ejemplo 2.3 se obtiene un alto desempeño de la red para porcentajes de ruido de 0 a 35%; sin embargo, para porcentajes de ruido mayores al 35% la memoria se confunde recuperando un vector diferente que en general puede ser la superposición de vectores ejemplares y sus versiones invertidas. En el Ejemplo 2.4, la memoria no logra establecer la asociación correcta del vector deseado aún cuando el vector de entrada no tiene ruido, debiéndose a que los vectores ejemplares comparten varios o muchos pixeles entre sí. El modelo de Hopfield fue creado para tratar con imágenes binarias, sin embargo, en [43] se desarrolla una técnica donde es posible utilizar dicho modelo en el procesamiento de imágenes en tonos de gris a partir del *principio de descomposición por umbral* (PDU). La desventaja de la técnica es el tiempo de cómputo que se requiere para procesar el número de cortes que se deben hacer por cada nivel de gris que contenga la imagen; por ejemplo, si la imagen tiene 255 niveles de gris se requiere procesar 255 cortes binarios. En otra variante de la red de Hopfield [35], se mapean los valores de pixeles de cada ejemplar en un intervalo de  $[-s, s]$ , donde  $0 < s \leq 2$  con la intención de hacer simétrico el rango y emular la codificación bipolar continua, para la cual, la función de activación que se emplea es  $\tanh(x)$ .

## 2.4. Memoria de Hamming clasificadora

La *memoria de Hamming clasificadora* (MHC) se deriva de las redes neuronales utilizadas para la clasificación de vectores [7, 33]. Cada vector posee características propias que lo identifica de los demás y a dichas características se les conoce como descriptores. Los vectores de entrada son de tamaño  $n \times 1$  siendo  $n$  el número de descriptores. Al tratar imágenes digitales como patrones, cada pixel se considera un descriptor.

### 2.4.1. Red neuronal artificial de Hamming

Como Lippman [33] utilizó la distancia de Hamming para medir la similaridad o diferencia de un vector de entrada con un vector ejemplar, la red neuronal correspondiente lleva este nombre. Así, la red neuronal de Hamming es un clasificador de distancia mínima que opera con vectores binarios. La distancia de Hamming entre dos vectores es el número de componentes en los cuales ambos vectores difieren de valor [33, 53]. En la red de Hamming el número de neuronas es igual al número de vectores ejemplares almacenados en la memoria y consta de dos capas o subredes. La primera subred almacena los vectores ejemplares cuyos pesos  $w_{ij}$  no son necesariamente simétricos y  $w_{ii}$  no necesariamente es cero; la segunda subred es una red recurrente cuyas neuronas están

totalmente interconectadas entre si donde  $t_{ij}$  son simétricos y  $t_{ii} = 1 \forall i$ . La topología de la red de Hamming para 3 neuronas se muestra en la Fig. 2.11.

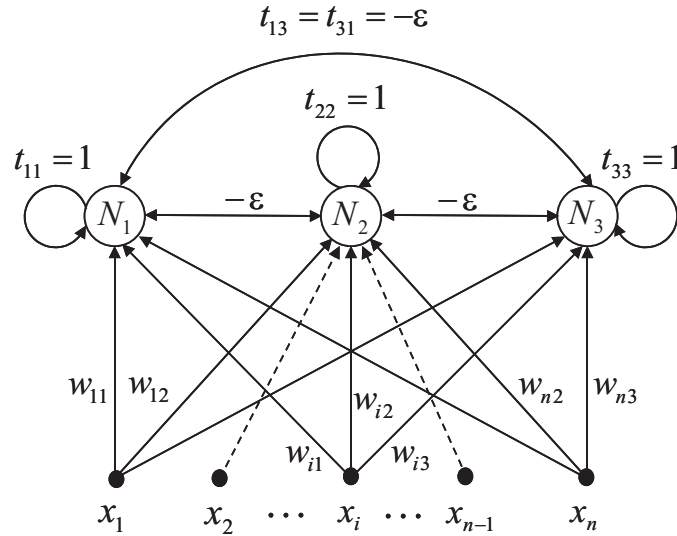


Figura 2.11: Red neuronal heteroasociativa de Hamming que contiene 3 neuronas y acepta vectores de entrada de dimensión  $n$ .

### 2.4.2. Almacenamiento y recuperación

La etapa de almacenamiento se lleva a cabo en la primera subred la cual contiene las conexiones del vector de entrada con cada neurona de la red. Supongamos que tenemos un conjunto de vectores  $\mathbf{x}^j = (x_1^j, \dots, x_n^j) \in \{-1, 1\}$  para almacenar en la red, donde  $j = 1, \dots, k$ . La matriz de pesos (primera subred) que almacena el conjunto de vectores ejemplares se calcula como:

$$w_{ij} = \frac{x_i^j}{2} ; 1 \leq i \leq n , 1 \leq j \leq k , \quad (2.14)$$

donde,  $x_i^j$  representa el  $i$ -ésimo elemento del  $j$ -ésimo vector,  $n$  es la dimensión del vector y  $k$  es el número de vectores ejemplares o clases. Los pesos en la segunda subred necesarios para realizar el proceso de competencia y determinar el máximo de componentes iguales se definen por

$$t_{ij} = \begin{cases} 1 \Leftrightarrow i = j \\ -\epsilon \Leftrightarrow i \neq j \end{cases} ; 0 < \epsilon < 1/k ; 1 \leq i, j \leq k , \quad (2.15)$$

donde  $t_{ij}$  denota el valor sináptico entre las neuronas de clase  $i$  y  $j$ . La etapa de recuperación inicia en la primera subred donde se determina la relación que guardan los vectores ejemplares y el vector

de entrada utilizando la distancia de Hamming, que para dos vectores  $\mathbf{x}$  e  $\mathbf{y}$   $n$ -dimensionales está dada por

$$h(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i| = n - \sum_{i=1}^n x_i y_i \quad \mathbf{x}, \mathbf{y} \in \{0, 1\}^n \quad \text{cod. normal}, \quad (2.16)$$

$$h(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^n |x_i - y_i| = \frac{n}{2} - \frac{1}{2} \sum_{i=1}^n x_i y_i \quad \mathbf{x}, \mathbf{y} \in \{-1, 1\}^n \quad \text{cod. bipolar}. \quad (2.17)$$

Si se inicializa la red con un vector  $\mathbf{y}$  y se calcula el valor de las neuronas en  $t = 0$ , se tiene

$$N_j(0) = \frac{n}{2} + \sum_{i=1}^n w_{ij} x_i^\xi = \frac{n}{2} + \frac{1}{2} \sum_{i=1}^n x_i^j x_i^\xi; \quad 1 \leq j \leq k, \quad (2.18)$$

donde  $N_j(0)$  es el estado de la  $j$ -ésima neurona ( $j$ -ésima clase). La segunda subred se inicializa con los valores determinados por la primera subred e inicia un proceso de competencia entre neuronas para encontrar una “ganadora”. Si todas las neuronas están en cero excepto la ganadora se dice que la red ha alcanzado la convergencia y corresponde al vector ejemplar que guarda mayor relación con la entrada y equivale a la mínima distancia de Hamming. El proceso de iteración está dado por

$$N_j(t+1) = f\left(\sum_{i=1}^M t_{ij} N_i(t)\right) = f\left(N_j(t) - \varepsilon \sum_{i \neq j} N_i(t)\right); \quad 1 \leq i, j \leq k, \quad (2.19)$$

donde  $N_j(t+1)$  es el siguiente estado de la  $j$ -ésima clase,  $N_j(t)$  es su valor actual y  $f(\cdot)$  es la función de activación definida por  $f(x) = x$  si  $x > 0$  y  $f(x) = 0$  en otro caso. El empleo de la memoria de Hamming clasificadora está descrito en pseudocódigo matemático en el Algoritmo 3 (implementado en **Mathcad**).

El conjunto de ejemplares  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \{-1, 1\}^n$  consta de vectores binarios codificados polarmente, el cual es una matriz de tamaño  $n \times k$ , donde  $n$  representa la dimensión del vector y  $k$  es el número de vectores. La matriz de pesos  $W$  se calcula como  $w_{lk} = 0.5x_l^k$ . La función **if** (condición, 1, 0) evalúa una condición lógica verdadera como 1 o falsa como 0.

```

Input:  $\tilde{\mathbf{x}} \in \{-1, 1\}^n$  [vector ejemplar con ruido aleatorio]
for  $\xi = 1$  to  $k$  do [escudriña los vectores ejemplares]
     $x_\xi = 0.5n + \sum_{j=1}^n w_{\xi j} \tilde{x}_j$  [calcula los componentes relacionados]
while  $s \neq 1$  do
    for  $\xi = 1$  to  $k$  do [recorre todas las neuronas]
         $s = 0$ 
        for  $\gamma = 1$  to  $k$  do [recorre todas las neuronas]
             $s = s + x_\gamma$  if  $\xi \neq \gamma$  [asigna el valor de la neurona]
        end
         $x_\xi = f(x_\xi - \epsilon s)$  [aplica la función de activación]
    end
     $s = 0$ 
    for  $\xi = 1$  to  $k$  do [escudriña los elementos del vector]
         $s = s + \text{if}(x_\xi > 0, 1, 0)$  [determina la neurona ganadora]
    end
Output:  $\mathbf{y} \in \{-1, 1\}^n$  [vector asociado]

```

**Algoritmo 3:** MHC - Memoria de Hamming Clasificadora

### 2.4.3. Recuperación de imágenes binarias

El siguiente ejemplo ilustra en forma detallada la operación de la red de Hamming. Según este ejemplo, puede verse que para porcentajes menores al 35% la red converge rápidamente a la clase del ejemplar correcto; sin embargo, para valores mayores a este porcentaje la red requiere de un mayor número de iteraciones y dependiendo del conjunto de ejemplares almacenados en la primera subred, su respuesta puede asociar correctamente o no la clase del vector ejemplar.

**Ejemplo 2.5.** El conjunto de imágenes utilizadas corresponde una vez más al Font E13B (ver Fig. 2.5), por tanto el tamaño de los vectores es  $n = 252$ . El número de neuronas en la red es de 14 y el vector distorsionado que ingresa a la red es el número '5'. La Fig. 2.12 muestra los resultados del modelo después de procesar el vector ruidoso. Los resultados numéricos de la asociación de los vectores se dan en la Tabla 2.1. La 1era columna indica el vector de entrada y el porcentaje de ruido, la 2da columna es la distancia de Hamming entre el vector de entrada y el vector ejemplar

del número '5', la 3era columna es el número de coincidencias entre el vector de entrada y el vector ejemplar, la 4ta columna muestra el número de iteraciones en la cual la red converge, la 5ta columna indica el estado final de la neurona correspondiente al número '5', y la 6ta columna corresponde al vector asociado por la red.



Figura 2.12: 1er. renglón, imágenes de entrada a la red: imagen ejemplar y versiones contaminadas del número '5' con 15 %, 25 %, 35 %, 45 % y 50 % de ruido aleatorio; 2do. renglón, respuesta de la MHC.

Tabla 2.1: Datos numéricos del Ejemplo 2.5;  $\mathbf{x}$  es el vector de entrada,  $h(\mathbf{x}^5, \mathbf{x})$  es la distancia de Hamming,  $n - h(\mathbf{x}^5, \mathbf{x})$  es el no. de coincidencias,  $\tau$  indica el número de iteraciones y  $N_5(\tau)$  es el estado final de la neurona del número '5'.

$\mathbf{x}$	$h(\mathbf{x}^5, \mathbf{x})$	$n - h(\mathbf{x}^5, \mathbf{x})$	$\tau$	$N_5(\tau)$	Asociado
$\mathbf{x}^5$ ( 0 %)	0	252	29	90.731	(si) $\mathbf{x}^5$
$\tilde{\mathbf{x}}^5$ (15 %)	42	210	27	62.151	(si) $\mathbf{x}^5$
$\tilde{\mathbf{x}}^5$ (25 %)	58	194	37	40.811	(si) $\mathbf{x}^5$
$\tilde{\mathbf{x}}^5$ (35 %)	92	160	419	25.376	(si) $\mathbf{x}^5$
$\tilde{\mathbf{x}}^5$ (45 %)	125	127	781	0	(no) $\mathbf{x}^7$
$\tilde{\mathbf{x}}^5$ (50 %)	140	112	851	0	(no) $\mathbf{x}^{11}$

En la Tabla 2.2 se dá el detalle de los valores determinados por la MHC, donde 0 %, 15 %, 25 %, 35 %, 45 % y 50 % son los porcentajes de ruido agregados a la imagen del número '5'. Cada porcentaje de ruido consta de 3 renglones y 14 columnas (vectores ejemplares); el primer renglón es la distancia de Hamming entre el vector de entrada y cada vector ejemplar; el segundo renglón son los aciertos entre el vector de entrada y los vectores almacenados, y el tercer renglón indica el estado final de las neuronas correspondientes a los vectores ejemplares. De acuerdo a estas Tablas la red converge al vector correcto después de 29, 27, 37 y 419 iteraciones. Sin embargo, la red converge al vector incorrecto después de 781 y 851 iteraciones para 45 % y 50 % de ruido. El



Tabla 2.2: Respuesta de la red de Hamming al procesar el número ‘5’.

	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{x}^4$	$\mathbf{x}^5$	$\mathbf{x}^6$	$\mathbf{x}^7$	$\mathbf{x}^8$	$\mathbf{x}^9$	$\mathbf{x}^{10}$	$\mathbf{x}^{11}$	$\mathbf{x}^{12}$	$\mathbf{x}^{13}$	$\mathbf{x}^{14}$
0 %	94	54	45	99	0	98	72	96	68	88	144	142	132	120
	158	198	207	153	252	154	180	156	184	164	108	110	120	132
	0	0	0	0	90.731	0	0	0	0	0	0	0	0	0
15 %	104	78	75	101	42	108	92	104	90	98	134	140	134	120
	148	174	177	151	210	144	160	148	162	154	118	112	118	132
	0	0	0	0	62.151	0	0	0	0	0	0	0	0	0
25 %	112	86	75	117	58	120	104	102	98	106	142	132	126	124
	140	166	177	135	194	132	148	150	154	146	110	120	126	128
	0	0	0	0	46.811	0	0	0	0	0	0	0	0	0
35 %	116	112	109	115	92	122	108	122	104	112	134	136	136	118
	136	140	143	137	160	130	144	130	148	140	118	116	116	134
	0	0	0	0	25.376	0	0	0	0	0	0	0	0	0
45 %	131	131	134	124	125	131	119	133	125	121	137	135	129	127
	121	121	118	128	127	121	133	119	127	131	115	117	123	125
	0	0	0	0	0	0	6.248	0	0	0	0	0	0	0
50 %	120	122	127	133	140	136	128	136	132	126	110	116	112	122
	132	130	125	119	112	116	124	116	120	126	142	136	140	130
	0	0	0	0	0	0	0	0	0	0	8.297	0	0	0

parámetro de inhibición utilizado para los primeros 3 vectores es  $\varepsilon = 0.02$ , mientras que para los últimos 3, el parámetro de inhibición utilizado es de  $\varepsilon = 0.002$ . En ambos casos se cumple que  $\varepsilon < 1/k = 1/14 \approx 0.0714$ .

# Capítulo 3

## Memorias Asociativas Minimax

### 3.1. Preliminares matemáticos

Las *memorias asociativas reticulares minimax* (MARM), conocidas también como *memorias asociativas morfológicas* (MAM), están basadas en el álgebra reticular minimax. El concepto de retículo algebraico fue creado con el objetivo de generalizar y unificar las relaciones de orden existentes entre los subgrupos de un grupo. El precursor de estas ideas fue R. Dedekind en el siglo XIX y posteriormente a partir de 1930, G. Birkhoff desarrolló la teoría de los retículos algebraicos como una rama propia de las matemáticas abstractas [51]. La estructura de retículo se apoya en el concepto de *conjunto parcialmente ordenado* (CPO), cuya teoría puede consultarse en las referencias [14, 17, 18, 43].

#### 3.1.1. Álgebra reticular matricial

Un retículo puede ser visto desde dos puntos de vista, el primero como un sistema algebraico y el segundo desde el punto de vista de la teoría de conjuntos [14]. Razón por la cual, los retículos han encontrado aplicación en diversas áreas tales como las redes de comunicación, calendarización y programación de maquinaria industrial, rutas mínimas en teoría de grafos y procesamiento digital de señales e imágenes. Aplicaciones recientes pueden consultarse en [68]. Merece particular atención el trabajo de Cuninghame-Green quien desarrolló gran parte del álgebra minimax como ahora la conocemos [11, 12]. En el presente trabajo adoptamos el punto de vista algebraico. Un ejemplo que podemos citar es la *morfología matemática* (MM), la cual trata imágenes que tienen una estructura geométrica y se funda en la teoría de retículos algebraicos. El álgebra minimax está basada en los grupos- $l$  acotados cuyo conjunto subyacente son los números reales extendidos, definido por

$\mathbb{R}_{\pm\infty} = \mathbb{R} \cup \{-\infty, \infty\}$  (cf. [43]). El conjunto de matrices utilizado se denota como  $\mathcal{M}_{mn}(\mathbb{R}_{\pm\infty})$  que contiene matrices de tamaño  $m \times n$  cuyas componentes son números reales extendidos; si  $m = n$ , entonces  $\mathcal{M}_n(\mathbb{R}_{\pm\infty})$  contiene matrices cuadradas de tamaño  $n \times n$ . Sobre este conjunto se definen los siguientes grupos reticulares que comúnmente se les denomina como “álgebras”:

1. El *álgebra mín-plus* denotada por  $(\mathcal{M}_{mn}(\mathbb{R}_{\pm\infty}), \text{mín}, +) = (\mathcal{M}_{mn}(\mathbb{R}_{\pm\infty}), \wedge, +)$  cuyo elemento neutro bajo la operación “ $\wedge = \text{mín}$ ” es  $\infty$ ; en esta álgebra, la indeterminación  $\infty + (-\infty) = (-\infty) + \infty$  se resuelve dándole el valor de  $-\infty$ .
2. El *álgebra máx-plus* denotada por  $(\mathcal{M}_{mn}(\mathbb{R}_{\pm\infty}), \text{máx}, +) = (\mathcal{M}_{mn}(\mathbb{R}_{\pm\infty}), \vee, +)$  cuyo elemento neutro bajo la operación “ $\vee = \text{máx}$ ” es  $-\infty$ ; en esta álgebra, la indeterminación  $(-\infty) + \infty = \infty + (-\infty)$  se resuelve dándole el valor de  $\infty$ .
3. El *álgebra reticular* denotada por  $(\mathcal{M}_{mn}(\mathbb{R}_{\pm\infty}), \text{mín}, \text{máx}, +) = (\mathcal{M}_{mn}(\mathbb{R}_{\pm\infty}), \wedge, \vee, +)$  con los elementos neutros indicados antes. En esta álgebra, la indeterminación  $\infty + (-\infty) = (-\infty) + \infty$  se resuelve dándole, respectivamente, el valor de  $-\infty$  o  $\infty$  según la suma esté ligada con la operación mín o máx.

Dada una matriz  $X \in \mathcal{M}_{mn}(\mathbb{R}_{\pm\infty})$  se definen el mínimo y el máximo de  $X$  como los números reales (extendidos)

$$\wedge(X) = \text{mín}(X) = \bigwedge_{i=1}^m \bigwedge_{j=1}^n x_{ij} \quad ; \quad \vee(X) = \text{máx}(X) = \bigvee_{i=1}^m \bigvee_{j=1}^n x_{ij}. \quad (3.1)$$

### 3.1.2. Sumas mín-máx

Dadas dos matrices  $X, Y \in \mathcal{M}_{mn}(\mathbb{R}_{\pm\infty})$  se definen la suma, el *mínimo* y el *máximo* de  $X$  e  $Y$ , componente a componente, es decir, para  $i = 1, \dots, m$  y  $j = 1, \dots, n$ , se tiene

$$(X + Y)_{ij} = x_{ij} + y_{ij} \quad , \quad (X \wedge Y)_{ij} = x_{ij} \wedge y_{ij} \quad , \quad (X \vee Y)_{ij} = x_{ij} \vee y_{ij}. \quad (3.2)$$

Las desigualdades numéricas entre dos matrices  $X$  e  $Y$  para toda  $i, j$  se definen similarmente, i.e.,

$$X \leq Y \Leftrightarrow x_{ij} \leq y_{ij} \quad , \quad X \geq Y \Leftrightarrow x_{ij} \geq y_{ij}, \quad (3.3)$$

y de igual forma para el caso de las desigualdades estrictas. Basta que un par de elementos  $i, j$  no cumpla la desigualdad para decir que tampoco la cumplen las matrices  $X$  e  $Y$ . La suma es *distributiva* sobre las operaciones de mínimo y máximo; así, dadas tres matrices  $X, Y, Z \in \mathcal{M}_{mn}(\mathbb{R}_{\pm\infty})$  se verifica que

$$X + (Y \wedge Z) = (X + Y) \wedge (X + Z) \quad ; \quad X + (Y \vee Z) = (X + Y) \vee (X + Z). \quad (3.4)$$

Dadas dos matrices,  $X \in \mathcal{M}_{mp}(\mathbb{R}_{\pm\infty})$  e  $Y \in \mathcal{M}_{pn}(\mathbb{R}_{\pm\infty})$ , la *suma-mín* y la *suma-máx* están dadas por

$$(X \boxtimes Y)_{ij} = \bigwedge_{k=1}^p (x_{ik} + y_{kj}) \quad ; \quad (X \boxdot Y)_{ij} = \bigvee_{k=1}^p (x_{ik} + y_{kj}). \quad (3.5)$$

Ya que una matriz puede ser representada por un vector supóngase dado un vector columna de dimensión  $m$ , i.e.,  $\mathbf{x} \in \mathcal{M}_{m1}(\mathbb{R}_{\pm\infty})$  y un vector renglón de dimensión  $n$ , i.e.,  $\mathbf{y} \in \mathcal{M}_{1n}(\mathbb{R}_{\pm\infty})$ . Entonces, la suma-mín o la suma-máx de ambos vectores da como resultado la misma matriz de tamaño  $m \times n$  denominada *suma externa* de  $\mathbf{x}, \mathbf{y}$ . Así, para  $i = 1, \dots, m$  y  $j = 1, \dots, n$  se tiene

$$(\mathbf{x} \boxplus \mathbf{y})_{ij} = \left\{ \begin{array}{l} (\mathbf{x} \boxtimes \mathbf{y})_{ij} \\ (\mathbf{x} \boxdot \mathbf{y})_{ij} \end{array} \right\} = x_{i1} + y_{1j} = x_i + y_j, \quad (3.6)$$

en forma desarrollada queda

$$\mathbf{x} \boxplus \mathbf{y} = \begin{pmatrix} x_1 + y_1 & \cdots & x_1 + y_n \\ \vdots & \ddots & \vdots \\ x_m + y_1 & \cdots & x_m + y_n \end{pmatrix}. \quad (3.7)$$

Dada una matriz  $X \in \mathcal{M}_{mn}(\mathbb{R}_{\pm\infty})$  se define la *matriz conjugada*  $X^*$  por  $-X^T$  con el superíndice  $T$  indicando la *transpuesta* de  $X$ . En particular, la operación de conjugación posee las siguientes propiedades:

**MC 1.** Involución,  $(X^*)^* = X$ ,

**MC 2.** Linearidad,  $(X + Y)^* = X^* + Y^*$ ,

**MC 3.** De Morgan mín-máx,  $(X \wedge Y)^* = X^* \vee Y^*$  ;  $(X \vee Y)^* = X^* \wedge Y^*$ ,

**MC 3.** De Morgan suma mín-máx,  $(X \boxtimes Y)^* = Y^* \boxdot X^*$  ;  $(X \boxdot Y)^* = Y^* \boxtimes X^*$ .

## 3.2. Memorias asociativas reticulares minimax

Análogamente al caso de la memoria asociativa lineal, la función de una memoria asociativa reticular consiste en almacenar un conjunto de  $k$  pares de vectores  $(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^k, \mathbf{y}^k)$  donde  $\mathbf{x}^\xi = (x_1^\xi, \dots, x_n^\xi)^T \in \mathbb{R}^n$  e  $\mathbf{y}^\xi = (y_1^\xi, \dots, y_m^\xi)^T \in \mathbb{R}^m$ , para  $\xi = 1, \dots, k$ . El conjunto de *vectores asociados* lo denotamos por  $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) : \xi = 1, \dots, k\}$  o por el par matricial  $(X, Y)$ , siendo  $X = (\mathbf{x}^1, \dots, \mathbf{x}^k)$  e  $Y = (\mathbf{y}^1, \dots, \mathbf{y}^k)$ . En particular,  $X$  es una matriz de tamaño  $n \times k$  con entradas  $x_{i\xi}$  las cuales escribimos también como  $x_i^\xi$  para hacer énfasis en la  $i$ -ésima componente del  $\xi$ -ésimo vector.

### 3.2.1. Almacenamiento y recuperación

El  $\xi$ -ésimo par de vectores asociados  $(\mathbf{x}^\xi, \mathbf{y}^\xi)$  de  $(X, Y)$  queda almacenado en una memoria asociativa reticular mediante la suma externa dada por la Ec. (3.6), i.e.,  $\mathbf{y}^\xi \boxplus (-\mathbf{x}^\xi)$  para  $\xi = 1, \dots, k$ . Partiendo de la construcción anterior se definen dos memorias asociativas reticulares que son duales entre sí.

**Definición 3.1.** La *memoria-mín* asociativa y la *memoria-máx* asociativa, denotadas respectivamente por  $W_{XY}$  y  $M_{XY}$  que almacenan  $k$  vectores, están dadas por las siguientes expresiones

$$W_{XY} = \bigwedge_{\xi=1}^{\xi} [\mathbf{y}^\xi \boxplus (-\mathbf{x}^\xi)] \quad ; \quad w_{ij} = \bigwedge_{\xi=1}^k (y_i^\xi - x_j^\xi), \quad (3.8)$$

$$M_{XY} = \bigvee_{\xi=1}^k [\mathbf{y}^\xi \boxplus (-\mathbf{x}^\xi)] \quad ; \quad m_{ij} = \bigvee_{\xi=1}^k (y_i^\xi - x_j^\xi), \quad (3.9)$$

Nótese que el mínimo o el máximo de las diferencias  $y_i^\xi - x_j^\xi$  representan la magnitud de la interconexión entre las neurona  $i$  y  $j$  para  $i = 1, 2, \dots, n$  y  $j = 1, \dots, m$ , y son los pesos de la red neuronal; además ambas memorias son matrices de tamaño  $m \times n$ . Según esta definición, las memorias  $W_{XY}$  y  $M_{XY}$  son respectivamente una cota inferior y superior de cada una de las matrices individuales que almacenan a cada pareja asociada, simbólicamente

$$W_{XY} \leq \mathbf{y}^\xi \boxplus (-\mathbf{x}^\xi) \leq M_{XY} \quad \forall \xi = 1, \dots, k. \quad (3.10)$$

Si  $X \neq Y$  hablamos de una *MARM heteroasociativa* y si  $X = Y$  se trata de una *MARM autoasociativa*. En este último caso escribimos  $W_{XX}$  y  $M_{XX}$  y las expresiones que definen los pesos son análogas a las Ecs. (3.8) y (3.9), es decir,

$$(W_{XX})_{ij} = w_{ij} = \bigwedge_{\xi=1}^k (x_i^\xi - x_j^\xi) \quad ; \quad (M_{XX})_{ij} = m_{ij} = \bigvee_{\xi=1}^k (x_i^\xi - x_j^\xi). \quad (3.11)$$

Una vez construídas las memorias asociativas fundamentales  $W$  y  $M$ , éstas se utilizan para recuperar alguno de los vectores ejemplares dado un vector de entrada  $\mathbf{x}$ . Por tanto, si  $\mathbf{y}$  denota el vector de salida, entonces

$$\mathbf{y} = W \boxplus \mathbf{x} \quad \text{o} \quad \mathbf{y} = M \boxminus \mathbf{x}. \quad (3.12)$$

Es importante mencionar que en la Ec. (3.12), la recuperación empleando la memoria  $W$  se logra efectuando la suma máxima con el vector de entrada, en cambio la recuperación mediante la memoria  $M$  se realiza con la suma mínima. Como un ejemplo trivial, para una sola pareja de

vectores  $(\mathbf{x}^\xi, \mathbf{y}^\xi)$  donde  $\mathbf{x}^\xi = (x_1^\xi, \dots, x_n^\xi) \in \mathbb{R}^n$  e  $\mathbf{y}^\xi = (y_1^\xi, \dots, y_m^\xi) \in \mathbb{R}^m$ , la memoria asociativa reticular  $W^\xi$  es la siguiente

$$W^\xi = \mathbf{y}^\xi \boxtimes (-\mathbf{x}^\xi) = \begin{pmatrix} y_1^\xi - x_1^\xi & \cdots & y_1^\xi - x_n^\xi \\ \vdots & \ddots & \vdots \\ y_m^\xi - x_1^\xi & \cdots & y_m^\xi - x_n^\xi \end{pmatrix}, \quad (3.13)$$

y puede comprobarse que  $W^\xi$  satisface la ecuación  $W^\xi \boxtimes \mathbf{x}^\xi = \mathbf{y}^\xi$ . Explícitamente,

$$W^\xi \boxtimes \mathbf{x}^\xi = \begin{pmatrix} \bigvee_{i=1}^n (y_1^\xi - x_i^\xi + x_i^\xi) \\ \vdots \\ \bigvee_{i=1}^n (y_m^\xi - x_i^\xi + x_i^\xi) \end{pmatrix} = \mathbf{y}^\xi. \quad (3.14)$$

Recuérdese que el vector de entrada puede ser un vector ejemplar o un vector similar al ejemplar. Este último caso se presenta si alguno de los vectores ejemplares ha sido distorsionado o degradado con ruido, que hemos denotado por  $\tilde{\mathbf{x}}$ . Por lo tanto, el vector de salida asociado  $\tilde{\mathbf{y}}$  no necesariamente corresponde al vector ejemplar asociado  $\mathbf{y}$  que originalmente correspondería a  $\mathbf{x}$ .

Efectuando la suma máxima de  $W$  y la suma mínima de  $M$  con  $\mathbf{x}^\xi$  a izquierda y derecha respectivamente en la desigualdad (3.10) se obtiene para  $\xi = 1, \dots, k$ ,

$$W_{XY} \boxtimes \mathbf{x}^\xi \leq [\mathbf{y}^\xi \boxplus (-\mathbf{x}^\xi)] \boxtimes \mathbf{x}^\xi = \mathbf{y}^\xi = [\mathbf{y}^\xi \boxplus (-\mathbf{x}^\xi)] \boxtimes \mathbf{x}^\xi \leq M_{XY} \boxtimes \mathbf{x}^\xi, \quad (3.15)$$

equivalentemente

$$W_{XY} \boxtimes X \leq Y \leq M_{XY} \boxtimes X. \quad (3.16)$$

En el caso autoasociativo las desigualdades dadas en la Ec. (3.16) son igualdades, resultado demostrado por Ritter [52] et al. Específicamente,

$$W_{XX} \boxtimes X = X = M_{XX} \boxtimes X, \quad (3.17)$$

que se interpreta diciendo que las memorias  $W$  y  $M$  autoasociativas son de recuperación perfecta para entrada perfecta siendo esta última cualquiera de los vectores ejemplares, sin importar su dimensión o el número de ellos. Razón por la cual, se dice que ambas memorias son de capacidad infinita. Podemos observar que en las memorias asociativas reticulares no se requiere que los vectores ejemplares sean ortonormales para lograr recuperación perfecta. La situación es muy diferente si el vector de entrada contiene ruido pues entonces el vector de salida no corresponde necesariamente al vector ejemplar asociado almacenado originalmente en la memoria. Consecuentemente, las memorias asociativas reticulares no son de recuperación perfecta en presencia de ruido.

### 3.2.2. Recuperación de imágenes

Antes de dar los ejemplos relativos al desempeño de las memorias autoasociativas reticulares en la recuperación de imágenes binarias o en tonos de gris que han sido degradadas por ruido aleatorio, introducimos el concepto de ruido desde el punto de vista del álgebra reticular.

**Definición 3.2.** Sea  $I = \{1, \dots, n\}$ , se dice que el vector  $\tilde{\mathbf{x}}$  tiene *ruido erosivo* si sus componentes son menores o iguales a las componentes originales, es decir, si  $\forall i \in I, \tilde{x}_i \leq x_i$ , equivalentemente, si  $\tilde{\mathbf{x}} \leq \mathbf{x}$  donde  $\tilde{\mathbf{x}}$  representa el vector ruidoso y  $\mathbf{x}$  representa el vector ejemplar. Análogamente, se dice que el vector  $\tilde{\mathbf{x}}$  tiene *ruido dilatativo* si sus componentes son mayores o iguales a las componentes originales, es decir, si  $\forall i \in I, \tilde{x}_i \geq x_i$ , equivalentemente, si  $\tilde{\mathbf{x}} \geq \mathbf{x}$ . El vector  $\tilde{\mathbf{x}}$  tiene *ruido mixto* (aleatorio o determinista) si dados dos subconjuntos disjuntos  $L, G \subset I$ , se cumple que  $\forall i \in L, \tilde{x}_i < x_i$  y  $\forall i \in G, \tilde{x}_i > x_i$ .

De forma intuitiva, el “ruido” corresponde a una distorsión o alteración de los valores de píxeles en una imagen modelo u original. El *ruido impulsivo* puede tomar solamente dos valores extremos, en referencia a una cierta escala de grises, el valor máximo  $L - 1$  representa la “sal” mientras que el valor mínimo 0 representa la “pimienta”. De forma más general, estos valores extremos pueden seleccionarse de una vecindad izquierda de  $L - 1$  y una vecindad derecha de 0 siempre que la diferencia de estos valores produzca la sensación visual de “saturación local” alta o baja en determinados píxeles de la imagen. El único tipo de ruido que puede contaminar una imagen binaria corresponde al ruido impulsivo con los valores máximo (255 o 1) y mínimo (0) lo que equivale a *invertir* el valor de los píxeles de acuerdo a cierta probabilidad de ocurrencia. En general, si  $p \in [0, 1]$  representa la probabilidad de ocurrencia del ruido,  $p = 0$  significa que ningún píxel será modificado y si  $p = 1$  todos los píxeles serán modificados; es común expresar esta probabilidad como el *porcentaje de ruido* dado por  $100p\%$ .

**Ejemplo 3.1.** El conjunto de imágenes binarias utilizadas corresponde a los caracteres del Font E13B. Recordemos brevemente que esta fuente tipográfica se utiliza comúnmente para la numeración de cheques bancarios. Las memorias almacenan 14 imágenes binarias previamente convertidas en vectores (ver Fig. 2.5). Cada imagen es de  $18 \times 14$  píxeles de modo que la dimensión de cada vector ejemplar es  $n = 252$  y corresponde al número de neuronas en la red. La imagen distorsionada utilizada como ejemplo de entrada a la red es el número ‘5’. La Fig. 3.1 muestra los resultados obtenidos al recuperar las imágenes ruidosas empleadas como entrada a la red. Se observa que las memorias  $W$  y  $M$  logran una buena recuperación, respectivamente ante ruido erosivo y dilatativo. Sin embargo, presentan error en la recuperación al tratarse de ruido mixto (aleatorio).

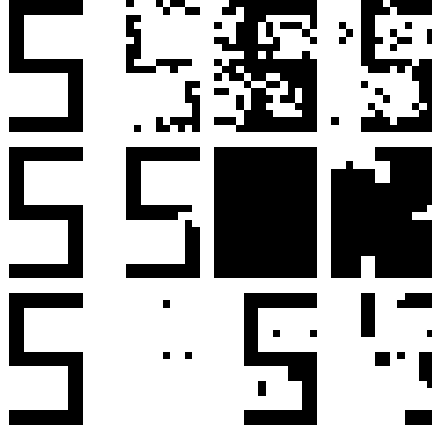


Figura 3.1: 1er renglón, imagen ejemplar, imagen con ruido erosivo (45 %), dilatativo (45 %) y aleatorio (10 %); 2do renglón, respuesta de la memoria  $W$  y 3er renglón, respuesta de la memoria  $M$ .

**Ejemplo 3.2.** Para este caso empleamos 5 imágenes de tamaño  $40 \times 40$  píxeles mostradas en la Fig. 3.2, las cuales son almacenadas como vectores de dimensión  $n = 1600$  y con entradas en la escala de grises  $[0, 255]$ . En la Fig. 3.3, el 3er. renglón muestra la capacidad de recuperación de la memoria mín  $W$  en presencia de diversos niveles de ruido erosivo; similarmente, el 5to. renglón muestra la capacidad de la memoria máx  $M$  ante los mismos niveles de ruido. Aunque a la vista parecería que las imágenes recuperadas por  $W$  y  $M$ , respectivamente ante ruido erosivo y dilatativo son las mismas, numéricamente el valor de error relativo <sup>1</sup> entre la imagen ejemplar y la imagen recuperada en cada caso no es cero. No obstante para estos niveles de ruido puede decirse que ambas memorias son de recuperación *casi perfecta*.

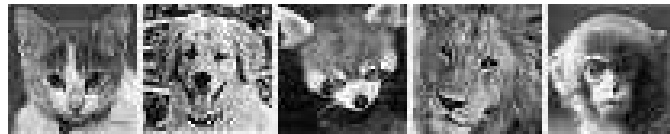


Figura 3.2: Imágenes ejemplares almacenadas en  $W$  y  $M$ .

Sin embargo, para ruido mixto aleatorio como puede verse en la Fig. 3.4, cualquiera de estas memorias no logra la recuperación aun cuando el porcentaje de ruido sea bajo.

---

<sup>1</sup>El error relativo entre los vectores  $\mathbf{x}$  y  $\tilde{\mathbf{x}}$  se define por  $\varepsilon_r(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{i=1}^n (x_i - \tilde{x}_i)^2 / \sum_{i=1}^n x_i^2$



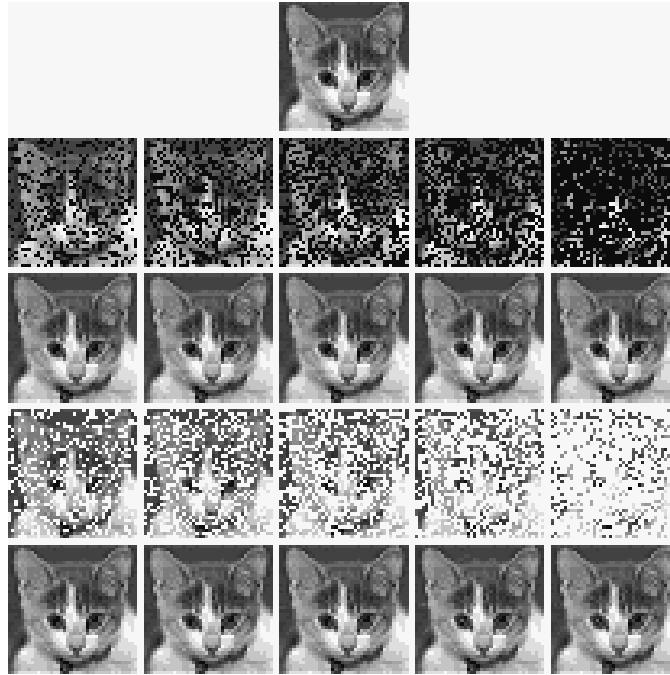


Figura 3.3: 1er renglón, imagen ejemplar; 2do renglón versiones con ruido erosivo (25 %, 35 %, 45 %, 55 % y 75 %); 3er renglón imágenes recuperadas con la memoria  $W$  donde  $\varepsilon_r \leq 10^{-4}$ ; 4to renglón, versiones con ruido dilatativo (25 %, 35 %, 45 %, 55 % y 75 %); 5to renglón imágenes recuperadas con la memoria  $M$  donde  $\varepsilon_r \leq 10^{-4}$ .



Figura 3.4: 1era columna, imagen con ruido mixto aleatorio (5 %); 2da columna, imagen confusa recuperada con la memoria  $W$ ; 3era columna, imagen confusa recuperada con la memoria  $M$ .

### 3.3. Método de núcleos

El método de núcleos es una propuesta para mejorar el desempeño de las MARM en presencia de ruido aleatorio. Como se ilustró antes, las MARM presentan error de confusión o traslapamiento de imágenes ante ruido aleatorio. Los antecedentes de este método contemplan desarrollos relacionados con la caracterización de núcleos en imágenes binarias [52, 55, 56], determinación de núcleos para imágenes en escala de grises basada en el concepto de *independencia fuerte morfológica* (IFM) [54] entre pares de vectores ejemplares de la cual expondremos en la siguiente sección un modelo novedoso conocido como IFM inducida [43, 62].

#### 3.3.1. Fundamento matemático

Dado que la memoria  $W$  es robusta al ruido erosivo, mientras que la memoria  $M$  lo es para el ruido dilatativo, podemos suponer que de alguna manera empleando estas memorias de manera conjunta es posible incrementar el desempeño en la recuperación. La idea básica consiste en utilizar la memoria  $M$  seguida de la memoria  $W$ , donde la memoria  $M$  contendrá los vectores núcleos y cada núcleo es una versión erosionada de los vectores ejemplares almacenados en la memoria  $W$ . Esquemáticamente, la operación de esta nueva combinación de memorias tiene la forma [43, 54, 62]:

$$\tilde{\mathbf{x}} \longrightarrow M_{ZZ} \longrightarrow W_{XX} \longrightarrow \mathbf{y}(\tilde{\mathbf{x}}) \approx \mathbf{x}, \quad (3.18)$$

donde  $\tilde{\mathbf{x}}$  representa el vector con ruido aleatorio e  $\mathbf{y}$  es el vector recuperado cuya dependencia de  $\tilde{\mathbf{x}}$  se ha hecho explícita. Se espera que el vector recuperado sea lo más parecido al vector ejemplar almacenado en  $W$ .

**Definición 3.3.** Un conjunto de vectores ejemplares  $X$  tiene la propiedad de *independencia fuerte morfológica* si las siguientes dos condiciones se satisfacen para toda  $\xi \neq l$

1.  $\forall l \in \{1, \dots, k\}, \mathbf{x}^l \not\subseteq \mathbf{x}^\xi$ .
2.  $\forall l \in \{1, \dots, k\}$  hay un índice  $j \in \{1, \dots, n\}$  tal que  $x_i^l - x_i^\xi \leq x_j^l - x_j^\xi, \forall i = 1, \dots, n$ .

Se hace notar que el subíndice  $j$  utilizado en la segunda condición depende del índice  $l$  seleccionado del conjunto de ejemplares  $\{1, \dots, k\}$ .

**Teorema 3.1.** Si  $X$  tiene la propiedad de IFM, entonces existe un conjunto de vectores  $Z \subseteq X$  con las siguientes propiedades. Para  $l = 1, \dots, k$ : 1)  $\mathbf{z}^l \wedge \mathbf{z}^\xi = 0 \forall \xi \neq l$ , 2)  $\mathbf{z}^l$  contiene una entrada distinta de cero y 3)  $W_{XX} \boxtimes \mathbf{z}^l = \mathbf{x}^l$ .

**Corolario 3.1.** Si  $X$  y  $Z$  satisfacen las hipótesis del Teorema 3.1 y  $W_{XY}$  es una memoria heteroasociativa, entonces  $Z$  es un núcleo para  $(X, Y)$  con  $W = W_{XY} \boxtimes W_{XX}$ .

Dado que solo se considera el caso particular de memorias autoasociativas, i.e.  $Y = X$ , por lo tanto,  $W = W_{XX} \boxtimes W_{XX} = W_{XX}$ , que corresponde al esquema de recuperación mostrado en la Ec. (3.18). Es fundamental que la condición,  $W_{XX} \boxtimes \mathbf{z}^l = \mathbf{x}^l$ , se cumpla para lograr la recuperación a partir de una entrada con ruido aleatorio. Dado un núcleo  $Z$  de  $X$  y una versión ruidosa  $\tilde{\mathbf{x}}^l$  del vector  $\mathbf{x}^l$  con la propiedad de que  $\mathbf{z}^l \leq \tilde{\mathbf{x}}^l$  y  $M_{ZZ} \boxtimes \tilde{\mathbf{x}}^l \leq \mathbf{x}^l$  entonces se obtiene

$$W_{XX} \boxtimes (M_{ZZ} \boxtimes \tilde{\mathbf{x}}^l) = \mathbf{x}^l. \quad (3.19)$$

En la operación de la suma mínima en la Ec. (3.19) se aplica la propiedad de la memoria  $M$ , que es robusta ante el ruido dilatativo ya que el vector con ruido  $\tilde{\mathbf{x}}^l$  es una versión *dilatada* del vector núcleo correspondiente  $\mathbf{z}^l$ . En la segunda operación se realiza la suma-máx entre el resultado obtenido de la suma-mín de  $M_{ZZ}$  y el vector contaminado  $\tilde{\mathbf{x}}^l$  con la memoria  $W_{XX}$ . En esta segunda operación se aprovecha la propiedad de la memoria  $W$  ya que el resultado de la suma-mín entre la matriz de núcleos y el vector de entrada ruidoso es una versión erosionada de los vectores ejemplares almacenados en la red. La recuperación perfecta de un vector de entrada con ruido  $\tilde{\mathbf{x}}^l$  de  $\mathbf{x}^l$  puede fallar en la Ec. (3.19) si existe un índice  $i = j$  dependiente de  $l$  tal que  $z_i^l > x_i^l$  o si hay varios índices  $i \in \{1, \dots, n\}$  para los cuales  $(M_{ZZ} \boxtimes \tilde{\mathbf{x}}^l)_i > x_i^l$ . Lo que nos indica que aún cuando un conjunto de vectores ejemplares satisfaga la propiedad IFM y por tanto haya un núcleo  $Z$  para  $X$ , es evidente que si el núcleo es afectado por el ruido, la recuperación de este esquema de memorias no siempre será exitosa. Aún cuando el método de núcleos basado en independencia fuerte morfológica puede dar buenos resultados en la recuperación de imágenes degradadas por ruido aleatorio, la condición de IFM no siempre se cumple dado un conjunto arbitrario de vectores. No obstante, se ha desarrollado un procedimiento que denominamos IFM *inducida* [43, 62], mediante el cual se logra que un conjunto  $X$  de vectores ejemplares tenga dicha propiedad. Los pasos fundamentales que describen a este procedimiento se dan a continuación.

1. Se calcula el mínimo global  $L$  y el máximo global  $U$  del conjunto de entrada  $X$  que consta de  $k$  vectores ejemplares de dimensión  $p$ , i. e.,

$$L = \min(X) = \bigwedge_{i=1}^p \bigwedge_{\xi=1}^k x_i^\xi ; \quad U = \max(X) = \bigvee_{i=1}^p \bigvee_{\xi=1}^k x_i^\xi . \quad (3.20)$$

2. Sea  $I = \{1, \dots, p\}$  un conjunto finito de índices, entonces para  $\xi = 1, \dots, k$  se determina un

índice  $i_\xi \in I$  donde se encuentre el *primer máximo disponible*, i. e.,

$$x_{i_\xi}^\xi = \bigvee_{i \in I} x_i^\xi, \quad (3.21)$$

se actualiza  $I = I - \{i_\xi\}$ , se pone  $\xi = \xi + 1$  y se recalcula la Ec. (3.21) para el nuevo vector  $\mathbf{x}^\xi$ ; esto asegura que  $\forall l \neq \xi, i_l \neq i_\xi$ .

3. Se cambia el conjunto de vectores ejemplares  $X$ , en todas las posiciones  $i_\xi$  para  $\xi = 1, \dots, k$  con los valores de  $L$  y  $U$  determinados en 1., así

$$x_{i_l}^\xi = \begin{cases} U \Leftrightarrow l = \xi, \\ L \Leftrightarrow l \neq \xi. \end{cases} \quad (3.22)$$

Resulta que el conjunto de vectores modificados, denotado por  $\bar{X}$ , es un conjunto que satisface la propiedad de IFM.

4. Finalmente, se aplica a  $\bar{X}$  el método de núcleos y el esquema de memorias asociativas morfológicas descrito anteriormente. El núcleo  $Z$  de  $\bar{X}$  se obtiene de 3., para  $i = 1, \dots, p$  y  $\xi = 1, \dots, k$

$$z_i^\xi = \begin{cases} U \Leftrightarrow i = i_\xi \\ 0 \Leftrightarrow i \neq i_\xi \end{cases} \quad (3.23)$$

Una vez encontrado el núcleo  $Z$  correspondiente al conjunto  $\bar{X}$  de vectores ejemplares modificados por IFM inducida, se construyen las memorias autoasociativas  $M_{ZZ}$  y  $W_{\bar{X}\bar{X}}$ .

Para completar esta descripción es necesario probar que el conjunto  $\bar{X}$  cumple ambas condiciones de independencia fuerte morfológica.

- $\forall l \in \{1, \dots, k\}$ ,  $x^l \not\leq x^\xi$  significa que para cada  $\xi \neq l$ , existe un índice  $j \in \{1, \dots, p\}$  tal que  $x_j^l > x_j^\xi$  para  $\xi \neq l$ . Esto se cumple cuando se cambia  $j$  por  $i$  en 3.
- $\forall l \in \{1, \dots, k\}$ , existe un  $j \in \{1, \dots, n\}$  tal que  $\forall i \in \{1, \dots, p\}$ ,  $x_j^l - x_j^\xi \geq x_i^l - x_i^\xi$ ; considerando nuevamente que  $j = i$ , se obtiene

$$U - L = \bigvee_{i=1}^p (x_i^l - x_i^\xi) \geq x_i^l - x_i^\xi.$$

El *nuevo esquema* de dos memorias (autoasociativas) ahora se presenta así

$$X \rightarrow \bar{X} ; \quad \tilde{\mathbf{x}} \longrightarrow M_{ZZ} \longrightarrow W_{\bar{X}\bar{X}} \longrightarrow \mathbf{y}(\tilde{\mathbf{x}}) \approx \bar{\mathbf{x}}, \quad (3.24)$$

donde la memoria reticular  $W$  se calcula en base a  $\bar{X}$ .

### 3.3.2. Recuperación de imágenes con ruido

En seguida se presentan dos ejemplos de recuperación de imágenes mediante las memorias  $W$  y  $M$ . En el caso de imágenes binarias utilizamos directamente el Teorema 3.1 para determinar los núcleos binarios correspondientes a los vectores ejemplares. Para el caso de imágenes en tonos de gris empleamos el concepto de IFM inducida siguiendo el procedimiento descrito por las Ecs. (3.20) a (3.23). En los siguientes ejemplos se ha implementado una versión mejorada del algoritmo IFM basado en el empleo de la distancia de Hamming (vectores binarios) definidas en las Ecs. (2.16,2.17) y el error cuadrático medio normalizado (vectores reales) para reactivar el núcleo correspondiente a un ejemplar afectado por ruido y así mejorar la capacidad de recuperación perfecta promedio.

**Ejemplo 3.3. Imágenes binarias** - las imágenes ejemplares utilizadas así como sus correspondientes núcleos se muestran en la Fig. 3.5, cada una de ellas es de  $18 \times 14$  pixeles y son almacenadas como vectores de dimensión  $n = 252$ . Para efectos ilustrativos, intencionalmente se seleccionó el subconjunto de los caracteres ‘0,1,3,5,6’ del Font E13B con la finalidad de obtener fácilmente un conjunto de vectores (núcleo) que cumplen las condiciones listadas en el Teorema 3.1. Específicamente, si  $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$  y  $\mathbf{x}^5$  corresponden respectivamente a ‘0’, ‘1’, ‘3’, ‘5’ y ‘6’ entonces los vectores núcleo,  $\mathbf{z}^1$  a  $\mathbf{z}^5$  cumplen con: 1)  $\mathbf{z}^l \wedge \mathbf{z}^\xi = 0 \forall \xi \neq l$ , 2)  $\mathbf{z}^l$  contiene un solo pixel distinto de cero (negro) y 3)  $W_{XX} \boxtimes \mathbf{z}^l = \mathbf{x}^l$ . Es importante mencionar que el conjunto de vectores núcleos para ejemplares de alta dimensionalidad no es único y por la tanto hay cierta flexibilidad en determinarlos de forma que cumplan el Teorema 3.1.

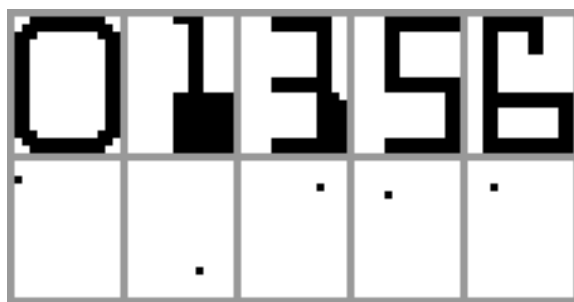


Figura 3.5: 1er renglón, vectores ejemplares almacenados en la memoria  $W_{XX}$ ; 2do renglón vectores núcleo almacenados en la memoria  $M_{ZZ}$ .

En la Fig. 3.6 observamos el desempeño de la memoria para 15 %, 30 %, 45 % y 60 % de ruido aleatorio. En el 2do. renglón, 4ta. y 5ta. columnas la respuesta de la memoria presenta confusión

debido al alto porcentaje de ruido. En particular, para 100 instancias usando los porcentajes de ruido mostrados, los porcentajes de *recuperación perfecta* resultaron ser, respectivamente de 100 %, 100 %, 56.9 % y 0.2 %. Para porcentajes de ruido aleatorio mayores al 45 % el porcentaje de recuperación perfecta está por debajo del 60 %.

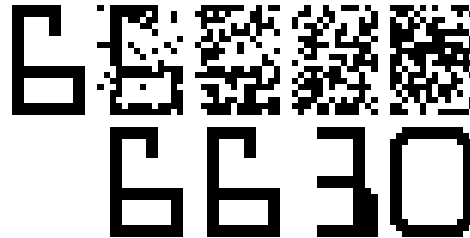


Figura 3.6: 1er renglón, imagen ejemplar y versiones ruidosas con 15 %, 30 % , 45 % y 60 %; 2do renglón, recuperación mediante el método de núcleos mejorado.

**Ejemplo 3.4. Imágenes en tonos de gris** - de acuerdo al procedimiento de IFM inducida, se modifica el conjunto de imágenes ejemplares para que sean morfológicamente independientes fuertemente, resultando en el conjunto  $\bar{X}$  que contiene imágenes ligeramente distintas a las originales. La Fig. 3.7 muestra el conjunto  $X$  de imágenes ejemplares, el conjunto modificado  $\bar{X}$ , así como el conjunto  $Z$  de imágenes núcleo. Dado que los núcleos son imágenes que contienen un solo pixel objeto, se han representado en cruz (5 px) para hacerlos más notorios. Para este conjunto de imágenes ejemplares, el método de núcleos de IFM inducida, considerando niveles de ruido aleatorio de tipo impulsivo del 50 %, 75 % y 100 % con constante aditiva  $\pm 128$  dió respectivamente los porcentajes de *recuperación perfecta* de 100 %, 95.8 % y 60 %. La Fig. 3.8 muestra las imágenes recuperadas para una instancia de los diferentes niveles de ruido antes mencionados. Este esquema de recuperación usando las memorias autoasociativas  $W$  y  $M$  ofrece buen desempeño para porcentajes de ruido menores al 75 % y a partir del 80 % la recuperación decrece ya que aparecen errores de confusión en las imágenes.



Figura 3.7: 1er renglón, imágenes ejemplares ( $X$ ); 2do renglón, imágenes modificadas ( $\bar{X}$ ) almacenadas en la memoria  $W_{\bar{X}\bar{X}}$ ; 3er renglón, conjunto de núcleos respecto de  $\bar{X}$  almacenados en la memoria  $M_{ZZ}$ .

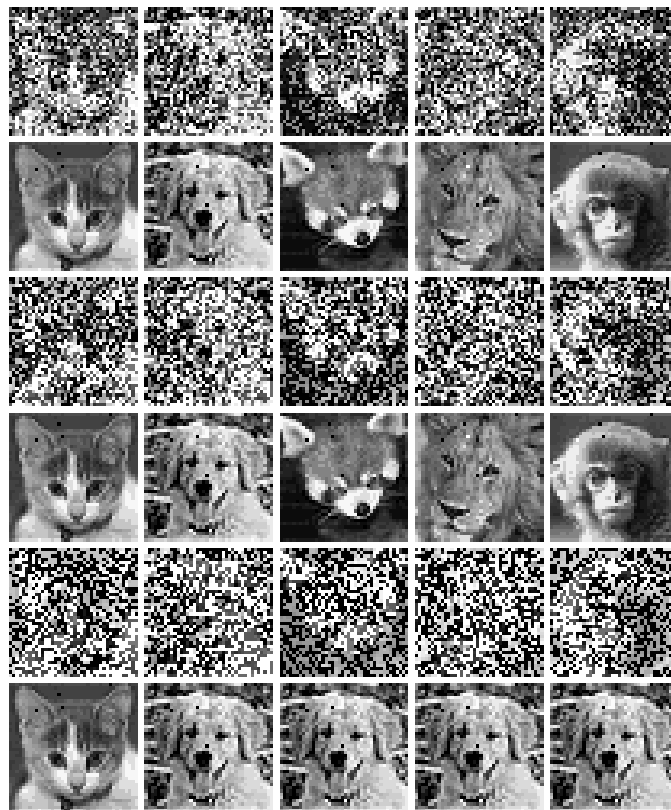


Figura 3.8: Renglones impares, imágenes con ruido aleatorio tipo impulsivo del 50 %, 75 % y 100 %; renglones pares, imágenes recuperadas usando el esquema de la Ec. (3.24) mejorado.

# Capítulo 4

## Memorias Asociativas Difusas

### 4.1. Lógica difusa

La lógica difusa ha sido desarrollada con la finalidad de aproximar o emular en cierto grado varios de los procesos cognitivos del cerebro humano relacionados principalmente con la capacidad de razonamiento. Los conceptos o ideas que se forman en el cerebro para percibir, reconocer y categorizar los fenómenos naturales a menudo no están totalmente precisados o completamente definidos y en ese sentido podemos considerarlos como conceptos o ideas difusas por darles un nombre [32]. Esta característica es lo que ha hecho tan especial y atractivo el uso de la lógica difusa en diversas áreas de la ingeniería tales como el control [10, 42, 72]. La información requerida en la lógica difusa para resolver un problema es de carácter *subjetivo*, *ambiguo* ó *vago*, que se representa lingüísticamente mediante reglas, conocimiento experto o requerimientos de diseño. En general, información de este tipo es imposible de cuantificar utilizando modelos matemáticos convencionales donde la información es de carácter *objetivo* [39]. Un ejemplo de conocimiento “difuso” o “impreciso” corresponde al caso del razonamiento que las personas comúnmente aplican a los problemas cotidianos de la vida. Si alguien desea cruzar una calle y ve que un auto se aproxima, la persona no calcula la velocidad del auto ni la velocidad que él requiere para ganarle al auto y pasar primero, simplemente sus procesos cognitivos le permiten estimar la velocidad que necesita alcanzar para cruzar la calle. Otros ejemplos que tratan con información difusa son: a) el estado de “salud” de una persona, b) el “largo” de ciertos objetos, c) la “edad” de una persona, d) calificar la variable “presión” en un proceso industrial, y e) la “temperatura” del clima.

Los datos inferidos de las variables difusas arriba ejemplificadas pueden considerarse como conjuntos difusos. Por ejemplo, supóngase que se habla de la variable difusa “temperatura”, en-



tonces la forma de pasarla a un conjunto difuso es de la siguiente manera: *muy frío, frío, tibio o templado, caliente o muy caliente*. Si la temperatura fuera de 26°C, para una persona que viva en una zona desértica, dirá que es una temperatura fría, mientras que para alguna otra persona que viva en una zona montañosa dirá que es una temperatura caliente. Sin embargo, alguna persona que viva en la costa dirá que la temperatura es templada o tibia. En base a este tipo de percepciones relativamente subjetivas se desarrolla o establece un conocimiento experto, necesario para determinar reglas que se utilizan en el diseño de sistemas difusos.

La teoría de conjuntos difusos (borrosos) fue introducida en 1965 por L. Zadeh como una manera de representar la vaguedad en la vida cotidiana [41]. Básicamente la teoría de los conjuntos difusos son una generalización de la teoría clásica de conjuntos (“duros” o “nítidos”), en la cual se añade una función de pertenencia o membresía que da un número real en el intervalo  $[0, 1]$ . En particular, un conjunto duro es un conjunto difuso cuya función de pertenencia da únicamente los valores extremos de 0 y 1, donde 0 indica que un elemento no pertenece al conjunto y 1 indica que el elemento pertenece al conjunto. El grado de pertenencia o membresía de un elemento a un conjunto difuso es gradual, siendo estos grados de pertenencia o valores de membresía los que indican la relación que guarda el elemento con el conjunto o subconjunto dado.

En la frase “Juan es joven”, no siempre es posible decir si es una proposición verdadera o falsa. Si la “edad” de Juan la representamos por la variable  $x$ , entonces la veracidad de  $x$  relativa a la cláusula “es joven”, depende del grado de pertenencia, que a su vez depende lingüísticamente del concepto “joven”. Si “Juan es menor a 22 años” y conocemos la edad de Juan, podemos decidir si la proposición es verdadera o falsa, lo que puede formalizarse considerando que las edades están definidas en el intervalo de  $[0, \infty)$ . Así, suponiendo que  $A = \{x : x \in [0, \infty) \text{ y } x < 22\}$ , podemos determinar si la edad de Juan pertenece o no a  $A$ . Sin embargo, el concepto “joven” no puede definirse como un subconjunto ordinario de  $[0, \infty)$ . Zadeh fué el primero en darse cuenta de esto. Dos personas, una de 18 y la otra de 20 años de edad son jóvenes, pero en diferente grado ya que la primera es “más joven” que la segunda. Esto sugiere inmediatamente que la membresía de un elemento a un subconjunto difuso debe graduarse en el intervalo  $[0,1]$ .

#### 4.1.1. Conjuntos difusos y su representación

Comenzamos esta sección dando algunas definiciones fundamentales con el objetivo de establecer la diferencia conceptual entre un conjunto clásico y un conjunto difuso.

**Definición 4.1.** Un *conjunto* (clásico, duro o nítido), denotado por  $A$ , es una colección de distintos elementos que están descritos por una propiedad  $p$  que tienen en común todos ellos. Simbólica-

mente,  $A = \{x|x \text{ tiene la propiedad } p\}$  [32].

Aunque esta definición claramente es de carácter intuitivo, el principio de comprensión de la lógica formal nos permite identificar, respecto de una colección arbitraria suficientemente amplia o universo del discurso, aquellos objetos que cumplan la propiedad  $p$  con los elementos que pertenecen al conjunto  $A$ . Esta relación se escribe formalmente como  $(\forall x)(x \in A \Leftrightarrow x \text{ tiene la propiedad } p)$ , donde el símbolo  $\in$  representa la relación de pertenencia de un elemento  $x$  al conjunto  $A$ . Esta relación queda también descrita por la *función característica*  $f_A$  [23] cuyo dominio es un conjunto universal  $U$  que contiene a  $A$  como subconjunto, i.e.,  $A \subseteq U$ . Nótese, que al conjunto  $U$  lo identificamos con el universo del discurso. Si  $x \in U$  pertenece al subconjunto  $A$ ,  $f_A(x) = 1$ ; en cambio, si  $x \in U$  no pertenece al subconjunto  $A$ , entonces  $f_A(x) = 0$ . De hecho, podemos identificar a un conjunto dado con su función característica referida a un conjunto universal apropiado. Así,

$$f_A : U \rightarrow \{0, 1\} \quad , \quad f_A(x) = \begin{cases} 1 \Leftrightarrow x \in A \\ 0 \Leftrightarrow x \notin A \end{cases} \quad , \quad A \subseteq U . \quad (4.1)$$

Como  $A$  es parte de  $U$  resulta claro que si  $x \notin A$  es equivalente a decir que  $x \in A^c$  donde  $A^c$  es el complemento de  $A$  respecto de  $U$  y  $A \cup A^c = U$  siendo  $A \cap A^c = \emptyset$ . Es importante mencionar que en el contexto de modelos basados en conjuntos, a  $f_A$  se le conoce también como función discriminante o indicatriz [39], y al codominio que es el par de números  $\{0, 1\}$  se le identifica con el grado de membresía que puede tener un elemento, “total” (1) o “nula” (0).

**Definición 4.2.** Un conjunto *difuso* es una colección de elementos cuya pertenencia está descrita por una *función de membresía*  $\mu_A$ , dada por

$$\mu_A : U \rightarrow I = [0, 1] \quad , \quad \mu_A(x) \in I \quad , \quad A \subseteq U \quad , \quad (4.2)$$

donde  $\mu_A(x)$  indica el grado de pertenencia del elemento  $x$  al conjunto  $A$  y,  $U$  e  $I$  son conjuntos (clásicos).

Hay que notar que esta definición es una generalización de un conjunto clásico donde la función de membresía [61] da valores en el intervalo continuo  $[0, 1]$ , a diferencia de la función característica que da valores discretos 0 y 1. El siguiente ejemplo muestra tres formas empleadas para representar un conjunto difuso.

**Ejemplo 4.1.** Sea  $U = \{1, 2, \dots, 9\}$  y sea  $A$  el conjunto difuso cuyos elementos constan de los “números cercanos a 5” y para el cual, la función de membresía está definida por [32]:

$x$	1	2	3	4	5	6	7	8	9
$\mu_A(x)$	0	0.2	0.6	0.9	1.0	0.9	0.6	0.2	0

1. *Conjunto de pares ordenados*: la primera entrada del par es un elemento del conjunto universal  $U$  y la segunda entrada del par es el grado de membresía  $\mu_A(x)$ . Así,  $A = \{(1, 0), (2, 0.2), (3, 0.6), (4, 0.9), (5, 0.1), (6, 0.9), (7, 0.6), (8, 0.2), (9, 0)\}$ .
2. *Vector difuso*: solo se anotan secuencialmente los valores de  $\mu_A(x)$ . Así  $A = (0, 0.2, 0.6, 0.9, 0.1, 0.9, 0.6, 0.2, 0)$ .
3. *Notación de Zadeh*:  $A$  se representa como una suma de fracciones donde el numerador de cada fracción es el grado de membresía del elemento  $x \in U$  anotado en el denominador; el signo de suma no tiene el significado aritmético usual. Si el grado de membresía para algunos elementos es cero, dicho elemento se omite de la expresión. Entonces,

$$A = \frac{0.2}{2} + \frac{0.6}{3} + \frac{0.9}{4} + \frac{1.0}{5} + \frac{0.9}{6} + \frac{0.6}{7} + \frac{0.2}{8} = \sum_{\mu_A(x) \neq 0} \frac{\mu_A(x)}{x}. \quad (4.3)$$

#### 4.1.2. Operaciones y propiedades básicas

Las operaciones entre conjuntos difusos tal como la unión y la intersección se definen mediante los operadores mín ( $\wedge$ ) y máx ( $\vee$ ) aplicados a sus valores de membresía. Si  $\mu_1, \mu_2, \dots, \mu_n$  denotan los valores de membresía respectivos a  $n$  conjuntos difusos  $A_1, A_2, \dots, A_n$ , el mínimo y máximo generalizados son los números dados por

$$\mu = \bigwedge_{i=1}^n \mu_i = \min_{1 \leq i \leq n} \{\mu_i\} \quad ; \quad \mu = \bigvee_{i=1}^n \mu_i = \max_{1 \leq i \leq n} \{\mu_i\}. \quad (4.4)$$

Un conjunto difuso  $A$  es vacío si su función de membresía es cero sobre el conjunto universal  $U$ , i.e.,  $A = \emptyset$  si  $\mu_A(x) = 0, \forall x \in U$ . Un conjunto difuso  $A$  es normal si existe al menos un elemento  $x \in U$  cuya membresía sea igual a uno, esto es,  $\exists x \in U : \mu_A(x) = 1$ . La Tabla 4.1 resume las operaciones básicas de inclusión, igualdad, unión, intersección y complementación para dos conjuntos difusos  $A$  y  $B$ . En la Fig. 4.1 se ilustran las operaciones fundamentales para uno o dos conjuntos difusos las cuales están representadas gráficamente mediante sus funciones de membresía calculadas según las expresiones dadas en la segunda columna de la Tabla 4.1. Prácticamente todas las propiedades algebraicas de los conjuntos clásicos se verifican para los conjuntos difusos traduciendo estas en términos de sus funciones de membresía. Así, por ejemplo, las propiedades de idempotencia, conmutatividad y asociatividad, por nombrar algunas, se cumplen para ambos

Tabla 4.1: Operaciones entre dos conjuntos difusos.

Descripción	Operación	función de membresía
inclusión	$A \subseteq B$	$\mu_A(x) \leq \mu_B(x)$
igualdad	$A = B$	$\mu_A(x) = \mu_B(x)$
unión	$A \cup B$	$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$
intersección	$A \cap B$	$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$
complemento	$A^c$	$\mu_{A^c}(x) = 1 - \mu_A(x)$

tipos de conjuntos. Para ilustrar este procedimiento, consideremos como caso particular una de las leyes de De Morgan entre dos conjuntos  $A$  y  $B$  dada por  $(A \cup B)^c = A^c \cap B^c$ . Si  $A$  y  $B$  fueran conjuntos difusos, la ley anterior se expresa empleando sus respectivas funciones de membresía, de modo que

$$\begin{aligned} (A \cup B)^c &= (\mu_A(x) \vee \mu_B(x))^c = 1 - (\mu_A(x) \vee \mu_B(x)) \\ &= (1 - \mu_A(x)) \wedge (1 - \mu_B(x)) = \mu_{A^c}(x) \wedge \mu_{B^c}(x) = A^c \cap B^c \end{aligned} \quad (4.5)$$

La justificación entre las expresiones  $1 - (\mu_A(x) \vee \mu_B(x))$  y  $(1 - \mu_A(x)) \wedge (1 - \mu_B(x))$  se realiza por casos suponiendo que 1)  $\mu_A > \mu_B$ , 2)  $\mu_A = \mu_B$  y 3)  $\mu_A < \mu_B$ . Es importante mencionar que hay propiedades que son válidas para los conjuntos difusos, pero no se cumplen para los conjuntos clásicos, e.g., dado el conjunto difuso  $A$  entonces

$$A \cup A^c \neq U \quad \text{y} \quad A \cap A^c \neq \emptyset, \quad (4.6)$$

ya que si  $x \in U$  y se verifica que  $0 < \mu_A(x) < 1$  se deduce que  $1 - \mu_A(x) < 1$  y  $1 - \mu_A(x) > 0$ , de donde

$$A \cup A^c = \mu_A(x) \vee (1 - \mu_A(x)) < 1 \quad \text{y} \quad A \cap A^c = \mu_A(x) \wedge (1 - \mu_A(x)) > 0, \quad (4.7)$$

por tanto

$$\mu_{A \cup A^c}(x) \neq 1 \quad \text{y} \quad \mu_{A \cap A^c}(x) \neq 0. \quad (4.8)$$

Finalizamos esta sección resumiendo en la Tabla 4.2 las funciones de membresía correspondientes a tres operaciones aritméticas entre dos conjuntos difusos. Para la suma se utiliza el mínimo con 1 para acotar la suma de  $\mu_A(x)$  y  $\mu_B(x)$  debido a que el resultado de ella puede ser mayor a 1. De manera similar, debido a que la diferencia entre  $\mu_A(x)$  y  $\mu_B(x)$  puede ser menor a cero, se emplea el máximo con cero para evitar valores negativos.

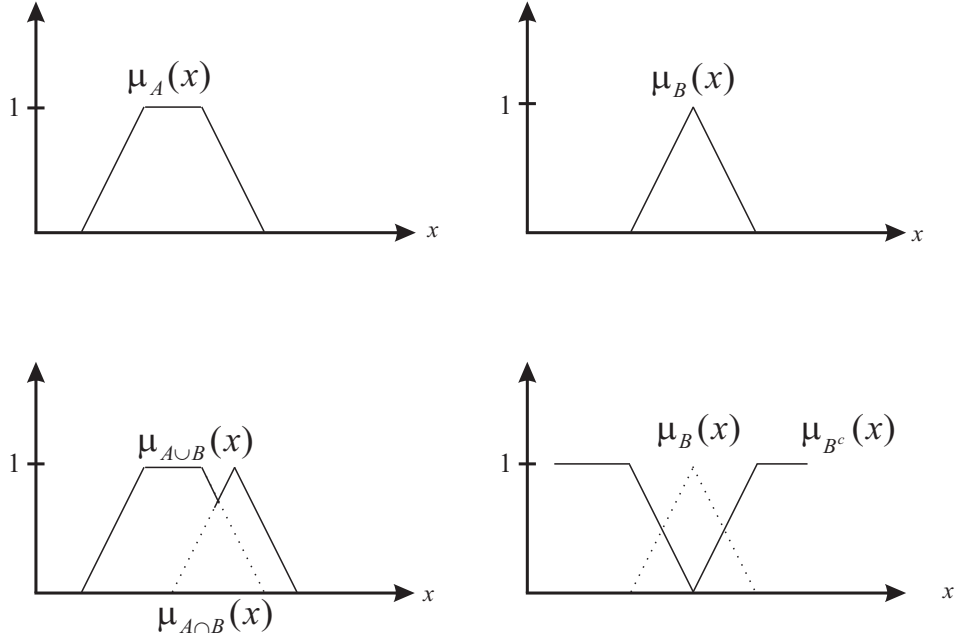


Figura 4.1: 1era hilera, conjuntos difusos  $A, B$  y sus funciones de membresía  $\mu_A(x)$  y  $\mu_B(x)$ ; 2da hilera, izq., unión (línea continua) e intersección (línea punteada); der., complemento (línea continua) del conjunto difuso  $B$ .

Tabla 4.2: Operaciones aritméticas entre dos conjuntos difusos.

Descripción	Operación	Función de membresía
suma	$A + B$	$\mu_{A+B}(x) = (\mu_A(x) + \mu_B(x)) \wedge 1$
producto	$AB$	$\mu_{AB}(x) = \mu_A(x)\mu_B(x)$
diferencia	$A \setminus B$	$\mu_{A \setminus B}(x) = (\mu_A(x) - \mu_B(x)) \vee 0$

## 4.2. Memorias morfológicas difusas

Las memorias morfológicas difusas fueron propuestas por Sussner al presentar un modelo basado en la memoria mín y máx difusa, el modelo hace uso de un algoritmo iterativo basado en el método de núcleos junto con la función limitadora dura, con el fin de encontrar un nivel de umbral apropiado para obtener una mejor recuperación de vectores binarios. Sussner [57, 60] comprobó, para el ejemplo de las vocales (ver Ejemplo 4.4 más adelante), un mejor desempeño de su modelo comparado con otras memorias asociativas tales como la red neuronal discreta de Hopfield, la red dinámica de registro por proyección y una red bidireccional de correlación. Es importante mencionar que debido al potencial que tienen los conjuntos difusos para representar imágenes

digitales [23] existen otros modelos que emplean las redes neuronales y los conjuntos difusos para el reconocimiento de patrones y el procesamiento de imágenes [6, 26, 41].

### 4.2.1. Fundamento teórico

Las *memorias asociativas morfológicas difusas* (MAMD), introducidas por P. Sussner [57, 60], dan una solución al problema de la recuperación de imágenes binarias contaminadas con ruido aleatorio. Con esta nueva clase de memorias asociativas se logra un mejor desempeño al combinar las sumas minimax con algunos elementos de la teoría de conjuntos difusos. El modelo de Sussner muestra un buen desempeño en la fracción promedio de recuperaciones perfectas para el caso de vectores binarios afectados por porcentajes bajos de ruido impulsivo. Este modelo utiliza la función indicatriz de B. Kosko [27, 30] para pasar de la imagen binaria (imagen dura) a una imagen en tonos de gris (imagen difusa).

**Definición 4.3.** La *función indicatriz de inclusión nítida* ( $'\subseteq'$ ),  $\mathcal{I}(A, B)$ , entre dos conjuntos  $A$  y  $B$  está dada por

$$\mathcal{I}(A, B) = \begin{cases} 1 & \Leftrightarrow A \subseteq B, \\ 0 & \Leftrightarrow A \not\subseteq B, \end{cases} \quad ; \quad \mathcal{I}'(A, B) = 1 - \mathcal{I}(A, B), \quad (4.9)$$

donde  $\mathcal{I}'(A, B)$  es la *función indicatriz de contención nítida* ( $'\supseteq'$ ) o función complementaria de  $\mathcal{I}(A, B)$ . La contraparte difusa de estas funciones indicatrices viene a ser el grado de membresía  $\mu_A(x)$  de un elemento  $x$  relativa a un conjunto finito  $A$ , recordando que  $\mu_A(x) \in [0, 1] \subseteq \mathbb{R}$ .

**Definición 4.4.** Sea  $\mathcal{U} \subset \mathbb{R}^n$  un subconjunto acotado,  $\mu_A, \mu_B$  funciones de membresía de  $A, B \subseteq \mathcal{U}$ , y  $\sigma(A) = \int_{\mathcal{U}} \mu_A(\mathbf{x}) d\mathbf{x}$ , que representa el tamaño (hipervolumen  $n+1$ -dimensional) del conjunto  $A$ , entonces la *función indicatriz de contención difusa* de Kosko [27] denotada por  $\mathcal{I}'_K(A, B)$  para  $A \neq \emptyset$  y  $B$ , está dada por

$$\mathcal{I}'_K(A, B) = \frac{1}{\sigma(A)} \int_{\mathcal{U}} [(\mu_A(\mathbf{x}) - \mu_B(\mathbf{x})) \vee \mathbf{0}] d\mathbf{x}. \quad (4.10)$$

**Ejemplo 4.2.** Sean  $A$  y  $B$  dos conjuntos compactos en el plano Euclidiano, entonces partiendo de la función indicatriz complementaria dada en la Ec. (4.9) tenemos que  $1 - \mathcal{I}(A, B) = \mathcal{I}'_K(A, B)$ , y sustituyendo en la Ec. (4.10) determinamos la relación,

$$\mathcal{I}_K(A, B) = 1 - \frac{1}{\sigma(A)} \int_{\mathcal{U}} [(\mu_A(\mathbf{x}) - \mu_B(\mathbf{x})) \vee \mathbf{0}] d\mathbf{x}. \quad (4.11)$$

Por ser  $A, B$  conjuntos nítidos con  $n = 2$ , la función de membresía  $\mu_A(x, y)$  se reduce a la función característica dada en la Ec. (4.1); así,  $\sigma(A) = \int_{\mathcal{U}} \mu_A(x, y) dx dy = \int_{\mathcal{U}} f_A(x, y) dx dy = \iint_A dx dy$  (área de  $A$ ). Por otro lado,  $\mu_{A \setminus B}(x, y) = (\mu_A(x, y) - \mu_B(x, y)) \vee (0, 0)$  se reduce a  $f_{A \setminus B}(x, y)$ , por tanto,  $\int_{\mathcal{U}} \mu_{A \setminus B}(x, y) dx dy = \int_{\mathcal{U}} f_{A \setminus B}(x, y) dx dy = \iint_{A \setminus B} dx dy$  (área de  $A \setminus B$ ), de donde finalmente encontramos que la Ec. (4.11) se simplifica a

$$\mathcal{I}_K(A, B) = 1 - \frac{\sigma(A \setminus B)}{\sigma(A)}. \quad (4.12)$$

Esta definición se particulariza al caso discreto considerando que  $\mathcal{U}$  es un subconjunto finito de  $\mathbb{Z}^n$ , de modo que la función indicatriz de contención difusa correspondiente queda expresada como

$$\mathcal{I}'_K(A, B) = \frac{1}{\sigma(A)} \sum_{\mathbf{x} \in \mathcal{U}} [(\mu_A(\mathbf{x}) - \mu_B(\mathbf{x})) \vee \mathbf{0}]. \quad (4.13)$$

donde  $\sigma(A) = \sum_{\mathbf{x} \in \mathcal{U}} \mu_A(\mathbf{x})$  y  $A, B$  son dos conjuntos finitos cualesquiera. Por lo tanto, la función indicatriz de inclusión difusa  $\mathcal{I}_K(A, B)$  se calcula como  $1 - \mathcal{I}'_K(A, B)$ . Tanto en el caso continuo como en el discreto, si  $A = \emptyset$ , se definen los valores particulares de las funciones indicatrices difusas como:  $\mathcal{I}'_K(A, B) = 0$  y  $\mathcal{I}(A, B) = 1$ .

En el caso de los conjuntos subyacentes de índices  $I$  y  $J$  de dos vectores binarios  $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$ , se utiliza la notación  $\mathcal{I}_K(\mathbf{x}, \mathbf{y})$  en lugar de la notación  $\mathcal{I}_K(I, J)$ . La función de membresía en esta situación especial se interpreta del siguiente modo,

$$\mu_I(i) = \begin{cases} 1 \Leftrightarrow x_i = 1, & i \in I \\ 0 \Leftrightarrow x_i = 0, & i \notin I \end{cases} \quad (4.14)$$

equivalentemente,  $\mu_I(i) = x_i$ . Consecuentemente, si  $\mathbf{x} \neq \mathbf{0}$  y  $\mathbf{y} \in \{0, 1\}^n$  entonces la Ec. (4.13) se escribe del siguiente modo

$$\mathcal{I}'_K(\mathbf{x}, \mathbf{y}) = \frac{1}{\sigma(\mathbf{x})} \sum_{i=1}^n [(x_i - y_i) \vee 0] = \frac{1}{\sum_{i=1}^n x_i} \sum_{i=1}^n [(x_i - y_i) \vee 0]. \quad (4.15)$$

La idea es utilizar el indicador de Kosko para pasar de una imagen binaria a una imagen en tonos de gris. En el Teorema 4.1, demostrado en [57] se dan las ecuaciones correspondientes a la etapa de recuperación de las MARM comunes y las MAM difusas en términos de las funciones indicatrices definidas anteriormente.

**Teorema 4.1.** Sea  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^\xi\}$  una matriz de tamaño  $n \times k$  cuyas columnas representan los vectores binarios. Si la memoria mín- $W$  es una matriz cuadrada de  $n \times n$  con entradas en  $\{-1, 0\}$ ,

i.e.,  $W \in \{-1, 0\}^{n \times n}$  y la memoria máx- $M$  es también una matriz cuadrada de  $n \times n$  con entradas en  $\{0, 1\}$  ( $M \in \{0, 1\}^{n \times n}$ ), entonces para cada  $i$ , la suma dura mín-máx y la *suma difusa mín-máx* se obtienen así

$$(W_{XX} \boxtimes \mathbf{x})_i = \mathcal{I}'(\mathbf{x}, -\mathbf{w}_i^T) \quad , \quad (M_{XX} \boxtimes \mathbf{x})_i = \mathcal{I}(\overline{\mathbf{m}}_i^T, \mathbf{x}) \quad \text{vectores nítidos,} \quad (4.16)$$

$$(W_{XX} \tilde{\boxtimes} \mathbf{x})_i = \mathcal{I}'_K(\mathbf{x}, -\mathbf{w}_i^T) \quad , \quad (M_{XX} \tilde{\boxtimes} \mathbf{x})_i = \mathcal{I}_K(\overline{\mathbf{m}}_i^T, \mathbf{x}) \quad \text{vectores difusos.} \quad (4.17)$$

Los operadores tilde  $\tilde{\boxtimes}$ ,  $\tilde{\boxtimes}$  en la Ec. (4.17) son las versiones difusas de los operadores duros  $\boxtimes$ ,  $\boxtimes$  en la Ec. (4.16), las letras  $\mathbf{w}_i$  y  $\mathbf{m}_i$  indican el  $i$ -ésimo vector renglón de  $W_{XX}$  y  $M_{XX}$ ,  $\overline{\mathbf{m}}$  es el complemento del vector  $\mathbf{m}$  dado por  $1 - \mathbf{m}$ . Además,  $\mathcal{I}_K(\overline{\mathbf{m}}_i^T, \mathbf{x}) = 1 - \mathcal{I}'_K(\overline{\mathbf{m}}_i^T, \mathbf{x})$ . Las salidas difusas se representan como  $\tilde{\mathbf{u}} = W_{XX} \tilde{\boxtimes} \mathbf{x}$  y  $\tilde{\mathbf{v}} = M_{XX} \tilde{\boxtimes} \mathbf{x}$ . Definiendo las funciones limitadoras (duras)  $\alpha, \beta, : \mathbb{R}^n \rightarrow \{0, 1\}^n$  para toda  $i \in \{1, \dots, n\}$  con  $\alpha(x_i) = 1$  si  $x_i > 0$ , en otro caso,  $\alpha(x_i) = 0$ , similarmente,  $\beta(x_i) = 0$  si  $x_i < 1$ , en otro caso  $\beta(x_i) = 1$  entonces,

$$\alpha(\tilde{\mathbf{u}}) = W_{XX} \boxtimes \mathbf{x} \quad \text{y} \quad \beta(\tilde{\mathbf{v}}) = M_{XX} \boxtimes \mathbf{x}. \quad (4.18)$$

En la Ec. (4.18) tanto  $\tilde{\mathbf{u}}$  como  $\tilde{\mathbf{v}}$  contienen información del vector de salida deseado y utilizando un nivel de umbral apropiado es posible recuperar la información del vector ejemplar.

**Ejemplo 4.3.** Se consideran 5 matrices binarias de tamaño  $3 \times 3$ , cada una de ellas representando de forma primitiva las letras C,H,L,U,O, que se consideran como vectores ejemplares. Específicamente:

$$C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, H = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}, L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, U = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, O = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Si consideramos a cada matriz como un conjunto cuyos elementos son las componentes y como el valor de cada componente indica pertenencia o no al conjunto podemos ver que las matrices corresponden a conjuntos duros. Para convertir el conjunto de matrices duras a matrices difusas aplicamos la función indicadora de Kosko dada en la Ec. (4.15). En la evaluación de las matrices, el valor 1 significa “*pertenencia total*” y el valor 0 significa “*pertenencia nula*”. Primero, se convierten las matrices en vectores utilizando la expresión dada en la Ec. 2.1, donde  $m = 3$  es el número de renglones y  $n = 3$  el número de columnas. Así que los vectores que



representan a cada letra tienen 9 componentes, en particular

$$\mathbf{x}^1(\text{C}) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{x}^2(\text{H}) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{x}^3(\text{L}) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{x}^4(\text{U}) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{x}^5(\text{O}) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Las memorias  $W_{XX}$  y  $M_{XX}$  se construyen según las expresiones dadas en la Ec. (3.11) y son (observar que  $W_{XX} \in \{-1, 0\}^{9 \times 9}$  y  $M_{XX} \in \{0, 1\}^{9 \times 9}$ )

$$W_{XX} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, M_{XX} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Al aplicar la función indicatriz de contención difusa dada en la Ec. (4.15) a los vectores correspondientes a las letras C,H,L,U,O y convertirlos de vectores a matrices difusas quedan ahora como

- Memoria mín- $W$  difusa ( $\xi = 1, \dots, 5$  e  $i = 1, \dots, 9$ ):

$$(W_{XX} \tilde{\square} \mathbf{x}^\xi)_i = \mathcal{I}'_K(\mathbf{x}^\xi, -\mathbf{w}_i^T) = \frac{1}{\sum_{j=1}^9 x_j^\xi} \sum_{j=1}^9 [(x_j^\xi + w_{ji}) \vee 0],$$

$$\mathcal{I}'_K(\mathbf{x}^1, -\mathbf{w}_i^T) = \begin{pmatrix} 1.00 & 0.14 & 0.29 \\ 1.00 & 0.00 & 0.00 \\ 1.00 & 0.28 & 1.00 \end{pmatrix}, \mathcal{I}'_K(\mathbf{x}^2, -\mathbf{w}_i^T) = \begin{pmatrix} 1.00 & 0.00 & 0.43 \\ 1.00 & 0.14 & 0.29 \\ 1.00 & 0.00 & 1.00 \end{pmatrix},$$

$$\mathcal{I}'_K(\mathbf{x}^3, -\mathbf{w}_i^T) = \begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 1.00 & 0.00 & 0.00 \\ 1.00 & 0.20 & 1.00 \end{pmatrix}, \mathcal{I}'_K(\mathbf{x}^4, -\mathbf{w}_i^T) = \begin{pmatrix} 1.00 & 0.00 & 0.29 \\ 1.00 & 0.00 & 0.14 \\ 1.00 & 0.14 & 1.00 \end{pmatrix},$$

$$\mathcal{I}'_K(\mathbf{x}^5, -\mathbf{w}_i^T) = \begin{pmatrix} 1.00 & 0.13 & 0.38 \\ 1.00 & 0.00 & 0.16 \\ 1.00 & 0.25 & 1.00 \end{pmatrix}.$$

- Memoria máx- $M$  difusa ( $\bar{\mathbf{m}} = 1 - \mathbf{m}$ ,  $\xi = 1, \dots, 5$  e  $i = 1, \dots, 9$ ):

$$(M_{XX} \tilde{\square} \mathbf{x}^\xi)_i = \mathcal{I}_K(\bar{\mathbf{m}}_i^T, \mathbf{x}^\xi) = 1 - \mathcal{I}'_K(\bar{\mathbf{m}}_i^T, \mathbf{x}^\xi) = 1 - \frac{1}{\sum_{j=1}^9 x_j^\xi} \sum_{j=1}^9 [(1 - m_{ji} - x_j^\xi) \vee 0],$$

$$\mathcal{I}_K(\bar{\mathbf{m}}_i^T, \mathbf{x}^1) = \begin{pmatrix} 1.00 & 1.00 & 1.00 \\ 1.00 & 0.71 & 0.86 \\ 1.00 & 1.00 & 1.00 \end{pmatrix}, \quad \mathcal{I}_K(\bar{\mathbf{m}}_i^T, \mathbf{x}^2) = \begin{pmatrix} 1.00 & 0.71 & 1.00 \\ 1.00 & 1.00 & 1.00 \\ 1.00 & 0.86 & 1.00 \end{pmatrix},$$

$$\mathcal{I}_K(\bar{\mathbf{m}}_i^T, \mathbf{x}^3) = \begin{pmatrix} 1.00 & 0.60 & 0.80 \\ 1.00 & 0.40 & 0.60 \\ 1.00 & 1.00 & 1.00 \end{pmatrix}, \quad \mathcal{I}_K(\bar{\mathbf{m}}_i^T, \mathbf{x}^4) = \begin{pmatrix} 1.00 & 0.86 & 1.00 \\ 1.00 & 0.86 & 1.00 \\ 1.00 & 1.00 & 1.00 \end{pmatrix},$$

$$\mathcal{I}_K(\bar{\mathbf{m}}_i^T, \mathbf{x}^5) = \begin{pmatrix} 1.00 & 1.00 & 1.00 \\ 1.00 & 0.86 & 1.00 \\ 1.00 & 1.00 & 1.00 \end{pmatrix}.$$

La Fig. 4.2 muestra el resultado de la fusificación de un conjunto de imágenes binarias codificadas como objeto negro sobre fondo blanco. Las imágenes resultantes están en tonos de gris y resultan de aplicar escalamiento e inversión al grado de membresía correspondiente a cada pixel.

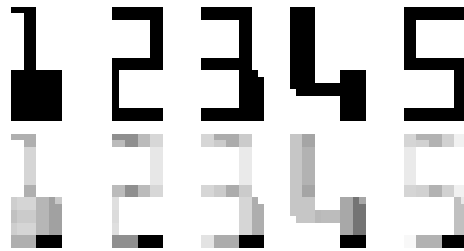


Figura 4.2: 1er renglón, vectores binarios; 2do renglón, vectores en tonos de gris.

En particular, la matriz difusa de la imagen binaria del dígito ‘4’ correspondiente al vector binario  $\mathbf{x}^4$  se muestra en la Fig. 4.3, donde claramente se observa que las entradas respectivas

están en el intervalo  $[0, 1]$ . Para obtener la imagen difusa correspondiente se realiza primero un escalamiento lineal del intervalo  $[0, 1]$  al intervalo  $[0, 255]$  y en segundo lugar una inversión de los valores de pixel donde el tono de gris 0 indica pertenencia total y el tono de gris 255 representa pertenencia nula y los tonos de gris intermedios representan el grado de pertenencia determinados por los vectores renglón de  $W$  y  $M$  (ver Fig. 4.4). Así, una imagen binaria que ha sido fusificada y representada como una imagen en tonos de gris, entre más blanco o cercano a 255 sea el valor del pixel indicará menor grado de pertenencia y entre más oscuro o cercano se encuentre a 0, mayor será el grado de pertenencia.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	0.224	0.224	0.353	0.353	0	0	0	0	0	0	0	0
1	0	0	0.224	0.224	0.353	0.353	0	0	0	0	0	0	0	0
2	0	0	0.224	0.224	0.301	0.301	0	0	0	0	0	0	0	0
3	0	0	0.224	0.224	0.301	0.301	0	0	0	0	0	0	0	0
4	0	0	0.224	0.224	0.301	0.301	0	0	0	0	0	0	0	0
5	0	0	0.224	0.224	0.301	0.301	0	0	0	0	0	0	0	0
6	0	0	0.224	0.224	0.301	0.301	0	0	0	0	0	0	0	0
7	0	0	0.224	0.224	0.301	0.301	0	0	0	0	0	0	0	0
8	0	0	0.224	0.224	0.353	0.353	0	0	0	0	0	0	0	0
9	0	0	0.224	0.224	0.353	0.353	0	0	0	0	0	0	0	0
10	0	0	0.224	0.224	0.224	0.224	0	0	0	0	0.34	0.34	0.545	0.346
11	0	0	0.224	0.224	0.224	0.224	0	0	0	0	0.34	0.34	0.545	0.545
12	0	0	0.224	0.224	0.224	0.224	0.276	0.263	0.263	0.263	0.34	0.34	0.545	0.545
13	0	0	0	0.224	0.224	0.224	0.276	0.263	0.263	0.263	0.34	0.34	0.545	0.545
14	0	0	0	0	0	0	0	0	0	0	0.34	0.34	0.545	0.545
15	0	0	0	0	0	0	0	0	0	0	0.34	0.34	0.545	0.545
16	0	0	0	0	0	0	0	0	0	0	1	1	1	1
17	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Figura 4.3: Matriz difusa correspondiente al dígito ‘4’.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	255	255	197	197	165	165	255	255	255	255	255	255	255	255
1	255	255	197	197	165	165	255	255	255	255	255	255	255	255
2	255	255	197	197	178	178	255	255	255	255	255	255	255	255
3	255	255	197	197	178	178	255	255	255	255	255	255	255	255
4	255	255	197	197	178	178	255	255	255	255	255	255	255	255
5	255	255	197	197	178	178	255	255	255	255	255	255	255	255
6	255	255	197	197	178	178	255	255	255	255	255	255	255	255
7	255	255	197	197	178	178	255	255	255	255	255	255	255	255
8	255	255	197	197	165	165	255	255	255	255	255	255	255	255
9	255	255	197	197	165	165	255	255	255	255	255	255	255	255
10	255	255	197	197	197	197	255	255	255	255	168	168	116	166
11	255	255	197	197	197	197	255	255	255	255	168	168	116	116
12	255	255	197	197	197	197	184	187	187	187	168	168	116	116
13	255	255	255	197	197	197	184	187	187	187	168	168	116	116
14	255	255	255	255	255	255	255	255	255	255	168	168	116	116
15	255	255	255	255	255	255	255	255	255	255	168	168	116	116
16	255	255	255	255	255	255	255	255	255	255	0	0	0	0
17	255	255	255	255	255	255	255	255	255	255	0	0	0	0

Figura 4.4: Imagen fusificada en tonos de gris del dígito '4'.

El empleo de la memoria asociativa difusa está descrito en pseudocódigo matemático en el Algoritmo 4 (implementado en Mathcad).

El conjunto de ejemplares  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \{0, 1\}^n$  consta de vectores binarios, el cual es una matriz de tamaño  $n \times k$ , donde  $n$  representa la dimensión del vector y  $k$  el número de vectores ejemplares. Las matrices de pesos  $W$  y  $M$  para  $X$  se calculan como  $w_{ij} = \bigwedge_{\xi=1}^k (x_i^\xi - x_j^\xi)$  y  $m_{ij} = \bigvee_{\xi=1}^k (x_i^\xi - x_j^\xi)$ . La  $k$ -ésima columna de  $W$  (resp.  $M$ ) la denotamos  $W^k$  (resp.  $M^k$ ) y la  $l$ -ésima componente de este vector lo denotamos por  $w_l^k = w_{lk}$  (resp.  $m_l^k = m_{lk}$ ). Hacemos empleo del algoritmo  $\mathcal{T}$  y  $\mathcal{T}'$  que aparece en [57].

```

Input:  $\tilde{\mathbf{x}} \in \{0, 1\}^n$ ,  $b \in \{0, 1\}$       [vector ejemplar ruidoso, bandera que elige  $W$  o  $M$ ]
if  $b = 0$  then                                [usa la memoria mín  $W$ ]
  for  $i = 1$  to  $n$  do                            [escudriña los elementos del vector ruidoso]
     $d = d + \tilde{x}_i$                                 [suma los elementos del vector ruidoso]
  for  $i = 1$  to  $n$  do                                [escudriña los renglones]
     $u_i = 0$ 
    for  $k = 1$  to  $n$  do                            [escudriña las columnas]
       $u_i = u_i + [(x_k + w_{i,k}) \vee 0]$           [acumula máximos usando la memoria  $W$ ]
    end
   $\mathbf{x} = \mathbf{u}/d$                                 [convierte el vector binario en difuso, memoria  $W$ ]
  let  $\mathbf{y} = \mathcal{T}(\mathbf{x})$                             [vector difuso procesado por el algoritmo  $\mathcal{T}$ ]
else                                            [usa la memoria máx  $M$ ]
  for  $i = 1$  to  $n$  do                            [escudriña los elementos del vector ruidoso]
     $d = d + \tilde{x}_i$                                 [suma los elementos del vector ruidoso]
  for  $i = 1$  to  $n$  do                                [escudriña los renglones]
     $u_i = 0$ 
    for  $k = 1$  to  $n$  do                            [escudriña las columnas]
       $u_i = u_i + [(1 - m_{i,k} - x_k) \vee 0]$       [acumula máximos usando la memoria  $M$ ]
    end
   $\mathbf{x} = 1 - (\mathbf{u}/d)$                             [convierte el vector binario en difuso, memoria  $M$ ]
  let  $\mathbf{y} = \mathcal{T}'(\mathbf{x})$                             [vector difuso procesado por el algoritmo  $\mathcal{T}'$ ]
end
Output:  $\mathbf{y} \in \{0, 1\}$                                 [vector asociado]

```

**Algoritmo 4:** MAMD - Memoria Asociativa Morfológica Difusa

### 4.2.2. Recuperación de imágenes binarias

Los siguientes dos ejemplos muestran la capacidad de recuperación de las memorias morfológicas autoasociativas difusas mín  $W$  y máx  $M$  al presentarles versiones ruidosas de las imágenes binarias ejemplares.

**Ejemplo 4.4.** El conjunto de imágenes está formado por las 5 vocales “A,E,I,O,U”. El tamaño de cada imagen binaria es de  $7 \times 7$  pixeles, por lo que la dimensión de cada vector ejemplar es  $n = 49$ . En la Fig. 4.5 se muestran en el 1er renglón las imágenes originales. El 2do renglón muestra las imágenes de las vocales con ruido aleatorio del 20%. El 3er renglón despliega las imágenes resultantes de haber aplicado la suma máx difusa entre la memoria  $W$  y el vector de entrada con ruido  $\tilde{x}$ , respectivamente, la suma mín difusa entre la memoria  $M$  y el mismo vector de entrada. En el 4to o último renglón se presentan las imágenes defusificadas al aplicar un umbral apropiado  $\tau \in (0, 1)$ . La gráfica de la Fig. 4.6 ilustra el desempeño de la capacidad de recuperación para la MAMD propuesta por Sussner [57].

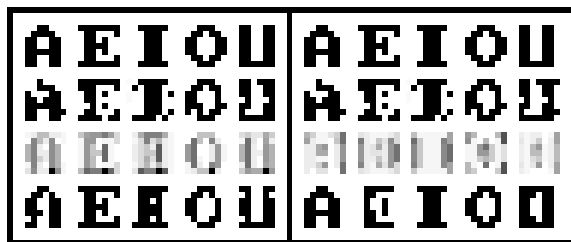


Figura 4.5: Bloque izq.: 1er renglón, imágenes ejemplares; 2do renglón, imágenes afectadas por ruido aleatorio (20%); 3er renglón, salida de la memoria-min difusa; 4to renglón, imágenes recuperadas con un nivel de umbral  $\tau = 0.1$ . Bloque der.: 1er renglón, imágenes ejemplares; 2do renglón, imágenes afectadas por ruido aleatorio (20%); 3er renglón, salida de la memoria-max difusa; 4to renglón, imágenes recuperadas con un nivel de umbral  $\tau = 0.9$ .

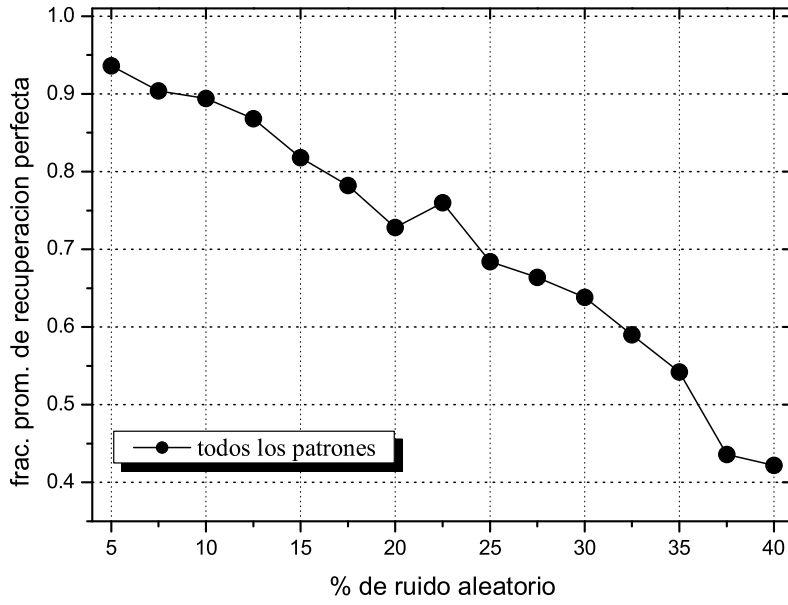


Figura 4.6: Curva de respuesta de la MAMD (Sussner). La gráfica muestra el promedio del porcentaje promedio de recuperación perfecta para niveles de ruido hasta 40 %. La curva se ha trazado considerando 100 versiones con ruido.

**Ejemplo 4.5.** Ahora se considera el conjunto de 14 imágenes binarias de tamaño  $14 \times 18$  pixeles correspondientes a los caracteres del Font E13B, donde la dimensión de cada imagen como vector es  $n = 252$ . Por conveniencia, en la Fig. 4.7, en los primeros dos renglones se vuelven a mostrar las imágenes de estos caracteres. El tercer renglón muestra la imagen ejemplar del dígito ‘6’, dos versiones con ruido del 10 % (2da columna) y 30 % (5ta columna). Las imágenes asociadas por la red aparecen en las columnas 4 y 7 del mismo renglón. La gráfica de la Fig. 4.8 ilustra el desempeño de la memoria propuesta por Sussner [57] para el conjunto de vectores del Font E13B.



Figura 4.7: 1er y 2do renglón, imágenes ejemplares. 3er renglón; 2da y 5ta cols. imagen con ruido (10% y 30%); 3era y 6ta cols., imágenes difusas obtenidas con mín  $W$ ; 4ta y 7ma cols., imágenes defusificadas.

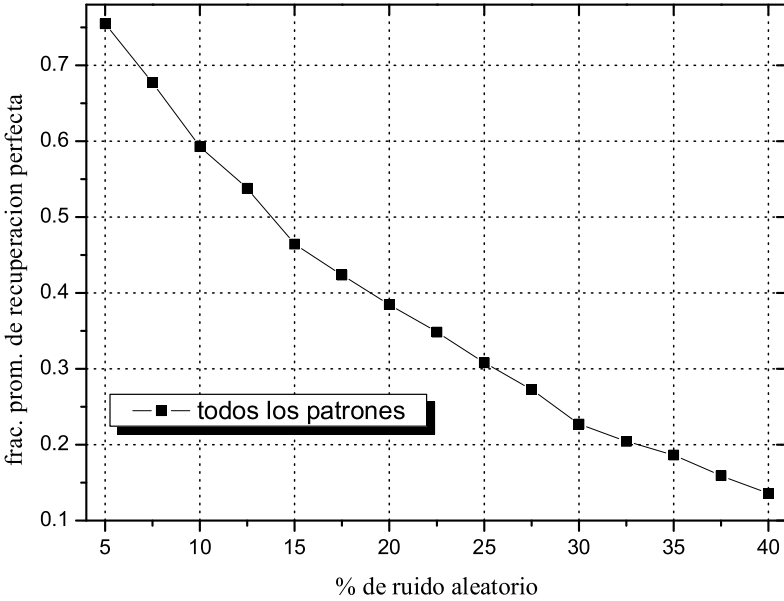


Figura 4.8: Curva de respuesta de la MAMD (Sussner). La gráfica muestra el promedio del porcentaje promedio de recuperación perfecta para niveles de ruido hasta 40%. La curva se ha trazado considerando 100 versiones con ruido.



### 4.3. Modelo híbrido MAMD-Hamming

Un *modelo híbrido* [26] es aquel que emplea de manera conjunta varias técnicas para alcanzar una mejor solución a un problema particular en comparación con una sola técnica aplicada al mismo problema. El objetivo de un modelo híbrido es obtener una estructura más robusta y lograr un mejor desempeño. Ejemplos de sistemas híbridos son aquellos que combinan técnicas basadas en conjuntos difusos con técnicas basadas en redes neuronales artificiales que según la disposición de éstas pueden dar lugar a sistemas *neuro-difusos* [45] o *difuso-neuronales* [6]. En esta sección presentamos un sistema híbrido neuro-difuso que combina una MAMD con una red neuronal de Hamming con la finalidad de mejorar el modelo neuro-difuso propuesto por Sussner [57]. Para obtener una mejora en el desempeño del modelo de Sussner se ha diseñado un procedimiento para alcanzar niveles de umbral óptimos, el cual incrementa gradualmente el nivel de umbral obteniendo vectores nítidos que sirven como entradas a una red de Hamming discreta [33, 40, 53]. Esta segunda red trabaja como un clasificador de distancia mínima, la cual acoplada a una MAMD se comporta como una memoria heteroasociativa. Recordemos que la distancia de Hamming entre dos vectores binarios se calcula como:

$$h(\mathbf{x}, \mathbf{y}) = \begin{cases} \sum_{i=1}^n |x_i - y_i| ; & \mathbf{x}, \mathbf{y} \in \{0, 1\}^n \quad \text{cod. normal} \\ \frac{1}{2} \sum_{i=1}^n |x_i - y_i| ; & \mathbf{x}, \mathbf{y} \in \{-1, 1\}^n \quad \text{cod. bipolar} \end{cases} \quad (4.19)$$

Conduciendo simulaciones similares a las descritas por Sussner [57], el sistema neuro-difuso aquí propuesto alcanza un mejor desempeño que la MAMD basada en el método de núcleos. Al modelo propuesto lo denominamos *MAMD-Hamming* o simplemente *MAMD* mejorada cuyo esquema de recuperación es el siguiente:

$$\mathbf{x} \rightarrow f(W_{XX} \tilde{\boxtimes} \mathbf{x} \mid M_{XX} \tilde{\boxtimes} \mathbf{x}) \rightarrow H \rightarrow \gamma \rightarrow \mathbf{x}^\gamma, \quad (4.20)$$

donde  $\mathbf{x}$  puede ser un vector ejemplar  $\mathbf{x}^\xi$  o una versión ruidosa  $\tilde{\mathbf{x}}^\xi$  de éste y la barra vertical significa que la operación de suma máx o mín difusa se realiza con  $W$  o  $M$  respectivamente. Además, la función umbral está definida por

$$f(x) = \begin{cases} -1 \Leftrightarrow x > \tau \\ 1 \Leftrightarrow x < \tau \end{cases}, \quad (4.21)$$

en la cual el valor óptimo de umbral  $\tau$  depende del conjunto de vectores ejemplares que se almacenan en la red. Este valor se determina después de repetir el esquema de recuperación empleando para cada ejemplar versiones con diversos niveles de ruido aleatorio e incrementando el valor de  $\tau$  a partir de un valor inicial, de modo que,  $\tau$  óptimo corresponda al mayor porcentaje promedio en

la recuperación. Una vez establecido el valor de  $\tau$  para la red ésta se reutiliza para otras instancias de entradas con ruido. Nótese que la aplicación de  $f(\cdot)$  convierte el vector difuso de salida de cualquiera de las memorias a un vector bipolar que es la entrada a la red de Hamming, denotada por  $H$ , la cual determina el índice  $\gamma$  que indica el vector ejemplar que asocia la red. El empleo de la memoria asociativa difusa está descrito en pseudocódigo matemático en el Algoritmo 5 (implementado en **Mathcad**).

El conjunto de ejemplares consta de vectores binarios  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \{0, 1\}^n$  el cual es una matriz de tamaño  $n \times k$ , donde  $n$  representa la dimensión del vector y  $k$  el número de vectores ejemplares. Las matrices de pesos  $W$  y  $M$  para  $X$  se calculan como  $w_{ij} = \bigwedge_{\xi=1}^k (x_i^\xi - x_j^\xi)$  y  $m_{ij} = \bigvee_{\xi=1}^k (x_i^\xi - x_j^\xi)$ . La  $k$ -ésima columna de  $W$  (resp.  $M$ ) la denotamos  $W^k$  (resp.  $M^k$ ) y la  $l$ -ésima componente de este vector columna lo denotamos por  $w_l^k = w_{lk}$  (resp.  $m_l^k = m_{lk}$ ). La función de umbralización definida por  $f(x, \tau) = \mathbf{if}(x > \tau, -1, 1)$  convierte un vector difuso en bipolar.

**Input:**  $\tilde{\mathbf{x}} \in \{0, 1\}^n$ ,  $b \in \{0, 1\}$  [vector ejemplar ruidoso, bandera que elige  $W$  o  $M$ ]

```

if  $b = 0$  then [usa la memoria mín  $W$ ]
  for  $i = 1$  to  $n$  do [escudriña los elementos del vector ruidoso]
     $d = d + \tilde{x}_i$  [suma los elementos del vector ruidoso]
  for  $i = 1$  to  $n$  do [escudriña los renglones]
     $u_i = 0$ 
    for  $k = 1$  to  $n$  do [escudriña las columnas]
       $u_i = u_i + [(x_k + w_{i,k}) \vee 0]$  [acumula máximos usando la memoria  $W$ ]
    end
   $\mathbf{x} = \mathbf{u}/d$  [convierte el vector binario en difuso, memoria  $W$ ]
else [usa la memoria máx  $M$ ]
  for  $i = 1$  to  $n$  do [escudriña los elementos del vector ruidoso]
     $d = d + \tilde{x}_i$  [suma los elementos del vector ruidoso]
  for  $i = 1$  to  $n$  do [escudriña los renglones]
     $u_i = 0$ 
    for  $k = 1$  to  $n$  do [escudriña las columnas]
       $u_i = u_i + [(1 - m_{i,k} - x_k) \vee 0]$  [acumula máximos usando la memoria  $M$ ]
    end
   $\mathbf{x} = 1 - (\mathbf{u}/d)$  [convierte el vector binario en difuso, memoria  $M$ ]
end
 $\mu = f(\mathbf{x}, \tau)$  [convierte el vector difuso en bipolar dado un umbral  $\tau$ ]
find  $\gamma \in \{1, \dots, k\}$  such that [busca el conjuntos de índices para]
   $g_\gamma = \bigwedge_{\xi=1}^k h(\mu, \mathbf{x}^\xi)$  [aplicar la red de Hamming]
let  $\mathbf{y} = \mathbf{if}(b = 0, W_{XX} \boxtimes \mathbf{x}^\gamma, M_{XX} \boxtimes \mathbf{x}^\gamma)$  [calcula el vector de salida]
Output:  $\mathbf{y} \in \{0, 1\}$  [vector asociado]

```

**Algoritmo 5:** MAMD-Hamming - Memoria Asociativa Morfológica Difusa Hamming

### 4.3.1. Recuperación de imágenes binarias

Los resultados alcanzados han mejorado a los que se obtuvieron con el modelo de Sussner [57], por tanto, el nuevo modelo híbrido alcanza mejor desempeño que el anterior. En seguida se ilustra el nuevo modelo mediante un par de ejemplos.

**Ejemplo 4.6.** El conjunto de vectores binarios ejemplares corresponde a las vocales A,E,I,O,U. Se realiza una prueba computacional similar a la provista en [57]. Se agrega ruido aleatorio con probabilidad  $p$  a cada vocal  $\mathbf{x}^\xi$ , donde  $\xi = 1, \dots, 5$ . La Fig. 4.9 muestra las curvas de respuesta de

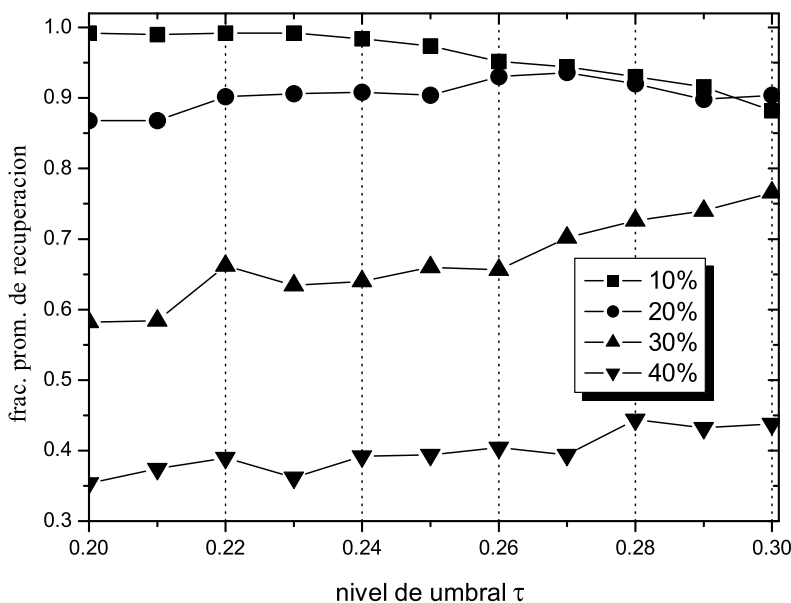


Figura 4.9: Curvas de respuesta de la MAMD-Hamming para 4 niveles representativos de ruido aleatorio; cada curva da el porcentaje promedio de recuperación perfecta en función del valor de umbral. Cada curva se ha trazado considerando el promedio sobre el conjunto de imágenes ejemplares (vocales) considerando 100 versiones con ruido de cada una de ellas.

la memoria mín- $W$  para diferentes niveles de umbral y valores específicos de  $p$ . Cada curva indica el porcentaje promedio de recuperación perfecta obtenido con 100 instancias de versiones con ruido para cada vector ejemplar  $\mathbf{x}^\xi$ . Del conjunto de curvas se obtienen valores de umbral óptimos para que la red alcance un buen desempeño en la recuperación. Para este conjunto de ejemplares los valores de umbral óptimos son;  $\tau = 0.23, 0.24, 0.25$ . La Fig. 4.10 muestra la comparación entre la MAMD de Sussner y la MAMD-Hamming propuesta.

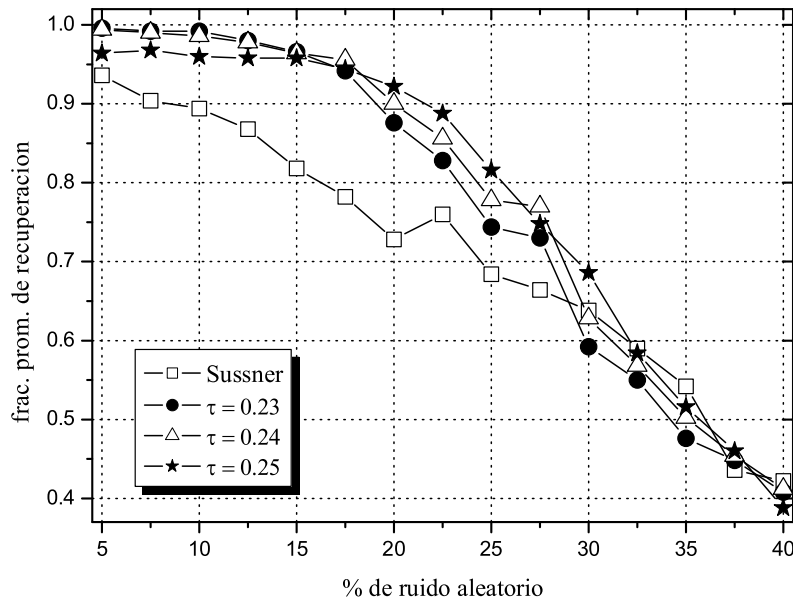


Figura 4.10: Comparación entre las curvas de respuesta del modelo de Sussner vs. MAMD-Hamming para 3 valores de umbral; cada curva muestra el porcentaje promedio de recuperación perfecta para niveles de ruido hasta 40%. Cada curva se ha trazado considerando el promedio sobre el conjunto de imágenes ejemplares (vocales) considerando 100 versiones con ruido de cada una de ellas.

**Ejemplo 4.7.** El conjunto de vectores ejemplares corresponde ahora a los caracteres del Font E13B, a cada uno de los cuales se agrega ruido aleatorio con probabilidad  $p$ . La Fig. 4.11 muestra las curvas de respuesta obtenidas usando la memoria mín- $W$  para diferentes niveles de umbral. Cada curva se ha trazado considerando el promedio sobre el conjunto de imágenes ejemplares (caracteres) considerando 100 versiones con ruido de cada una de ellas. Para este conjunto de vectores ejemplares los valores de umbral óptimo son:  $\tau = 0.06, 0.07, 0.08$ . La Fig. 4.12 muestra la comparación entre la MAMD de Sussner y la MAMD-Hamming propuesta.

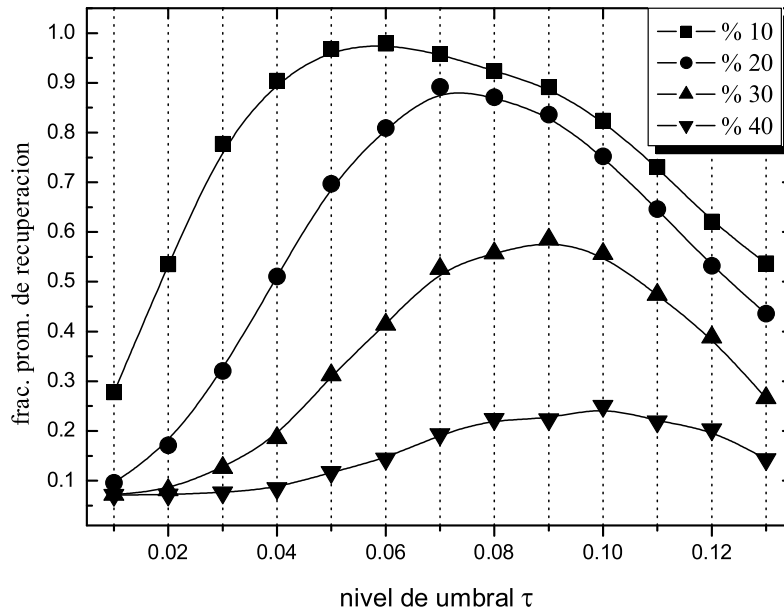


Figura 4.11: Curvas de respuesta de la MAMD-Hamming para 4 niveles representativos de ruido aleatorio; cada curva da el porcentaje promedio de recuperación perfecta en función del valor de umbral.

Hacemos notar que el intervalo de umbrales es dependiente del conjunto de vectores ejemplares bajo prueba. En particular, se consideró un intervalo donde  $\tau \in [0, 0.5]$  subdividido en cada centésimo, del cual se obtiene un subintervalo de umbrales apropiados donde las curvas de respuesta no se traslapen. Podemos concluir que la combinación entre las MARM y la lógica difusa es una alternativa que resuelve *parcialmente* el problema de recuperación de imágenes binarias afectadas por ruido aleatorio. De los dos ejemplos anteriores, el modelo MAMD-Hamming presenta un mejor desempeño que el modelo MAMD introducido por Sussner.

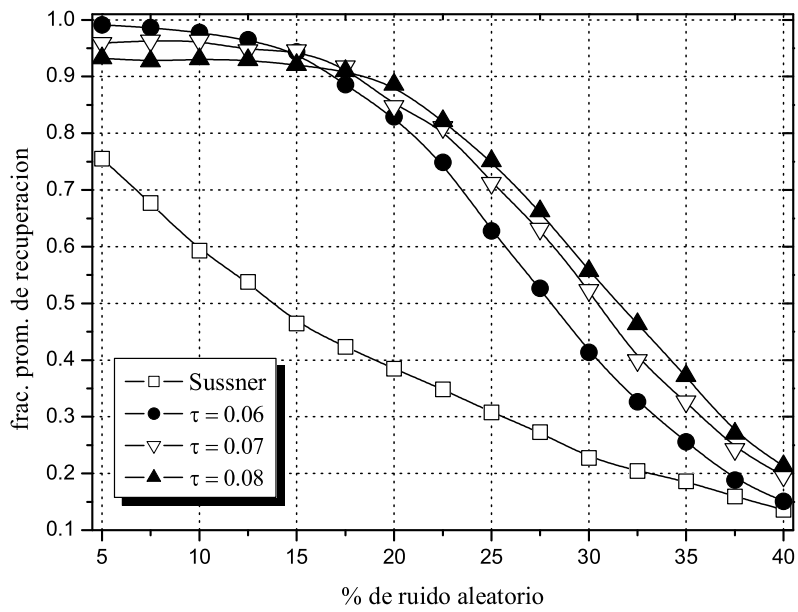


Figura 4.12: Comparación entre las curvas de respuesta del modelo de Sussner vs. MAMD-Hamming para 3 valores de umbral; cada curva muestra el porcentaje promedio de recuperación perfecta para niveles de ruido hasta 40 %.

# Capítulo 5

## Enmascaramiento del ruido

### 5.1. Preámbulo

La técnica del enmascaramiento del ruido consiste en sobreponer el vector de entrada con ruido aleatorio con cada uno de los vectores ejemplares con el propósito de obtener un vector máscara adecuado que separe el ruido mezclado en ruido erosivo o dilatativo dependiendo de si el enmascaramiento utiliza el operador mínimo o máximo. Esta técnica se propuso en [63] para resolver el problema del ruido aleatorio en las imágenes binarias y en tonos de gris utilizando las MARM. De este modo puede utilizarse de manera directa cualquiera de las memorias autoasociativas  $W$  o  $M$ . Esta misma técnica se extiende para tratar el ruido en imágenes a color RGB.

### 5.2. Imágenes binarias

El enmascaramiento del ruido para el caso binario se basa en dos máscaras, la máscara positiva y negativa. La primera es utilizada para procesar vectores con niveles de ruido aleatorio por debajo del 50% y la segunda para vectores con ruido aleatorio mayores al 50%. Recordemos que en los vectores binarios el ruido aleatorio es ruido de tipo impulsivo ya que solo es posible que la imagen tome uno de dos valores 0 o 1.

#### 5.2.1. Máscara positiva

La máscara positiva se obtiene al tomar el mínimo o máximo entre el vector con ruido y cada vector ejemplar. La máscara obtenida mediante el operador mín se utiliza con la memoria  $W_{XX}$  ya que ésta es una versión erosionada del vector de entrada. Análogamente, la máscara obtenida



mediante el operador  $\text{máx}$  se emplea con la memoria  $M_{XX}$  ya que ésta es una versión dilatada del vector de entrada.

**Definición 5.1.** Sea  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \subset \{0, 1\}^n$  el conjunto de vectores binarios ejemplares almacenados en la memoria  $W_{XX}$  o  $M_{XX}$ . Sea  $\tilde{\mathbf{x}}^\xi$  la versión con ruido del vector  $\mathbf{x}^\xi$  cuya probabilidad de ocurrencia es  $p \in [0, 0.5)$ , entonces las *máscaras mín positiva* y *máx positiva* de  $\mathbf{x}^\xi$ , están dadas respectivamente por

$$\tilde{\mathbf{x}}_{+e}^\xi = \tilde{\mathbf{x}}^\xi \wedge \mathbf{x}^\xi \quad ; \quad \tilde{\mathbf{x}}_{+d}^\xi = \tilde{\mathbf{x}}^\xi \vee \mathbf{x}^\xi. \quad (5.1)$$

**Ejemplo 5.1.** Se emplea el número ‘5’ del conjunto de vectores del Font-E13B [15]. La Fig. 5.1 muestra cinco versiones ruidosas del vector ejemplar así como el resultado de haber aplicado la técnica del enmascaramiento del ruido. Obsérvese que para porcentajes de ruido mayores al 50 % el enmascaramiento positivo ya no preserva la forma del dígito.



Figura 5.1: 1er renglón, vector ejemplar número ‘5’; 2do renglón, vectores afectados con ruido aleatorio al 25 %, 45 %, 55 %, 75 % y 95 %; 3er renglón, máscaras  $\tilde{\mathbf{x}}_{+e}^\xi$ ; 4to renglón, máscaras  $\tilde{\mathbf{x}}_{+d}^\xi$ .

### 5.2.2. Máscara negativa

La máscara negativa se obtiene al tomar el mínimo o máximo entre el vector con ruido y la diferencia del vector constante unidad con cada vector ejemplar. La máscara obtenida mediante el operador mín se utiliza con la memoria  $W_{XX}$  ya que ésta es una versión erosionada del vector de entrada. Análogamente, la máscara obtenida mediante el operador máx se emplea con la memoria  $M_{XX}$  ya que ésta es una versión dilatada del vector de entrada.

**Definición 5.2.** Sea  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \subset \{0, 1\}^n$  el conjunto de vectores binarios ejemplares almacenados en la memoria  $W_{XX}$  o  $M_{XX}$ . Sea  $\tilde{\mathbf{x}}^\xi$  la versión con ruido del vector  $\mathbf{x}^\xi$  cuya probabilidad de ocurrencia es  $p \in [0.5, 1)$ , entonces las máscaras *mín negativa* y *máx negativa* de  $\mathbf{x}^\xi$ , están dadas respectivamente por

$$\tilde{\mathbf{x}}_{-e}^\xi = \mathbf{x}^\xi \wedge (\mathbf{1} - \tilde{\mathbf{x}}^\xi) \quad ; \quad \tilde{\mathbf{x}}_{-d}^\xi = \mathbf{x}^\xi \vee (\mathbf{1} - \tilde{\mathbf{x}}^\xi), \quad (5.2)$$

donde  $\mathbf{1} = (1, 1, \dots, 1) \in \{0, 1\}^n$ .

**Ejemplo 5.2.** Se emplea el número ‘5’ del conjunto de vectores del Font-E13B [15]. La Fig. 5.2 muestra cinco versiones ruidosas del vector ejemplar así como el resultado de haber aplicado la técnica del enmascaramiento del ruido. Obsérvese que para porcentajes de ruido mayores al 50% el enmascaramiento negativo preserva la forma del dígito.

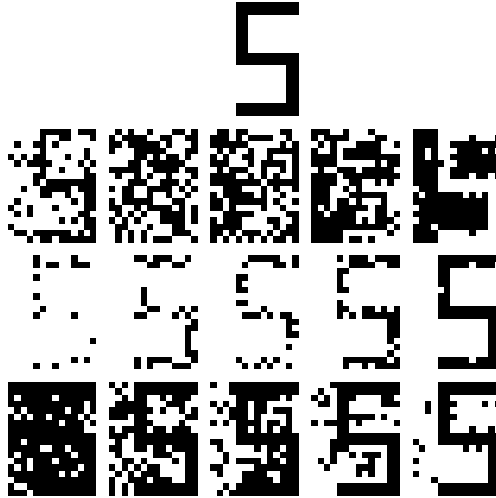


Figura 5.2: 1er renglón, vector ejemplar número ‘5’; 2do renglón, vectores afectados con ruido aleatorio al 25 %, 45 %, 55 %, 75 % y 95 %; 3er renglón, máscaras  $\tilde{\mathbf{x}}_{-e}^\xi$ ; 4to renglón, máscaras  $\tilde{\mathbf{x}}_{-d}^\xi$

En la Fig. 5.1 se muestra la ventaja de utilizar el enmascaramiento positivo usando la Ec. (5.1) y la Fig. 5.2 la ventaja de usar el enmascaramiento negativo mediante la Ec. (5.2). Se puede notar que si el ruido es mayor al 50% entonces, el vector de entrada ruidoso *invierte* su codificación, es decir, si el vector representa inicialmente un objeto negro sobre fondo blanco, al incrementar el ruido se modificará ahora a un objeto blanco sobre fondo negro (véase el caso de 95% de ruido en las Figs. 5.1 y 5.2). Es por esta razón que se usa el enmascaramiento negativo. De la Ec. (2.16) se deriva la *distancia de Hamming normalizada* (DHN) entre un par de vectores como una medida

de su proximidad y está dada por

$$h_n(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^n |x_i - \tilde{x}_i|. \quad (5.3)$$

Se observa que  $0 \leq h_n(\mathbf{x}, \tilde{\mathbf{x}}) \leq 1$ . En particular, si los vectores que se comparan están relacionados en todas sus entradas, entonces  $h_n(\mathbf{x}, \tilde{\mathbf{x}}) = 0$  (recuperación perfecta), si existe una simple diferencia entre ellos, entonces  $h_n(\mathbf{x}, \tilde{\mathbf{x}}) = 1/n$ , y si todas las entradas son diferentes entonces  $h_n(\mathbf{x}, \tilde{\mathbf{x}}) = 1$ . La memoria autoasociativa reticular matricial  $W_{XX}$  tiene *recuperación casi perfecta* de un vector binario  $\mathbf{x}$  de una versión ruidosa  $\tilde{\mathbf{x}}$  si y solo si  $h_n(W_{XX} \boxtimes \tilde{\mathbf{x}}, \mathbf{x}) \leq \varepsilon$ , donde  $0 < \varepsilon \leq 0.1$ . El Algoritmo 6 detalla la técnica del enmáscaramiento del ruido en imágenes binarias.

```

El conjunto de ejemplares  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\}$  es una matriz de tamaño  $n \times k$ , donde  $n$  es el número de píxeles en cada imagen y  $k$  es el número de vectores almacenados en la matriz de pesos  $W$ .
Input:  $\tilde{\mathbf{x}} \in \{0, 1\}^n$ ,  $b \in \{0, 1\}$  [vector ejemplar con ruido, bandera que elige  $W$  o  $M$ ]
if  $b = 0$  then [usa la memoria mín- $W$ ]
    for  $\xi = 1$  to  $k$  do [escudriña todos los vectores en  $X$ ]
         $\tilde{\mathbf{x}}_{+e}^\xi = \mathbf{x}^\xi \wedge \tilde{\mathbf{x}}$ ;  $\tilde{\mathbf{x}}_{-e}^\xi = \mathbf{x}^\xi \wedge (\mathbf{1} - \tilde{\mathbf{x}})$  [máscaras-pos/neg de  $k$  vectores]
         $v_\xi = D_{+e}^\xi \wedge D_{-e}^\xi$  [máscara-e, vector discriminante]
    end
    find  $\gamma \in \{1, \dots, k\}$  such that [busca índice correspondiente al]
         $v_\gamma = \bigwedge_{\xi=1}^k v_\xi$  [vector candidato enmasc.-min]
    let  $\mathbf{x}_+^\gamma = W_{XX} \boxtimes \tilde{\mathbf{x}}_{+e}^\gamma$ ;  $\mathbf{x}_-^\gamma = W_{XX} \boxtimes \tilde{\mathbf{x}}_{-e}^\gamma$  [vector de salida de enmasc.-min]
else [usa la memoria máx  $M$ ]
     $\tilde{\mathbf{x}} \in \{0, 1\}^n$ , [ejemplar no conocido con ruido aleatorio]
    for  $\xi = 1$  to  $k$  do [escanea todos los vectores en  $X$ ]
         $\tilde{\mathbf{x}}_{+d}^\xi = \mathbf{x}^\xi \vee \tilde{\mathbf{x}}$ ;  $\tilde{\mathbf{x}}_{-d}^\xi = \mathbf{x}^\xi \vee (\mathbf{1} - \tilde{\mathbf{x}})$  [máscaras-pos/neg de  $k$  vectores]
         $v_\xi = D_{+d}^\xi \wedge D_{-d}^\xi$  [máscara-d vector discriminante]
    end
    find  $\gamma \in \{1, \dots, k\}$  such that [busca índice correspondiente al]
         $v_\gamma = \bigwedge_{\xi=1}^k v_\xi$  [vector candidato enmasc.-max]
    let  $\mathbf{x}_+^\gamma = M_{XX} \boxtimes \tilde{\mathbf{x}}_{+d}^\gamma$ ;  $\mathbf{x}_-^\gamma = M_{XX} \boxtimes \tilde{\mathbf{x}}_{-d}^\gamma$  [vector de salida enmasc.-max]
end
y  $\mathbf{y} = \text{if}(h(\mathbf{x}_+^\gamma, \mathbf{x}^\gamma) < h(\mathbf{x}_-^\gamma, \mathbf{x}^\gamma), \mathbf{x}_+^\gamma, \mathbf{x}_-^\gamma)$  [calcular la distancia de Hamming]
Output:  $\mathbf{y} \in \{0, 1\}$  [vector asociado]

```

**Algoritmo 6:** Enmascaramiento de imágenes binarias

En el Algoritmo 6, los valores de distancia para las máscaras mín positivas y negativas y las máscaras máx positivas y negativas, denotadas respectivamente por  $D_{+e}^\xi, D_{-e}^\xi$  y  $D_{+d}^\xi, D_{-d}^\xi$  se calculan como los siguientes promedios:

$$D_{+e}^\xi = 0.5[h(\tilde{\mathbf{x}}_{+e}^\xi, \mathbf{x}^\xi) + h(\tilde{\mathbf{x}}_{+e}^\xi, \tilde{\mathbf{x}})] , \quad D_{-e}^\xi = 0.5[h(\tilde{\mathbf{x}}_{-e}^\xi, \mathbf{x}^\xi) + h(\tilde{\mathbf{x}}_{-e}^\xi, 1 - \tilde{\mathbf{x}})] , \quad (5.4)$$

$$D_{+d}^\xi = 0.5[h(\tilde{\mathbf{x}}_{+d}^\xi, \mathbf{x}^\xi) + h(\tilde{\mathbf{x}}_{+d}^\xi, \tilde{\mathbf{x}})] , \quad D_{-d}^\xi = 0.5[h(\tilde{\mathbf{x}}_{-d}^\xi, \mathbf{x}^\xi) + h(\tilde{\mathbf{x}}_{-d}^\xi, 1 - \tilde{\mathbf{x}})] . \quad (5.5)$$

### 5.2.3. Recuperación de imágenes con ruido

En el siguiente par de ejemplos se muestra en particular la capacidad de recuperación de la memoria  $W$  al utilizar la técnica del enmascaramiento del ruido en vectores binarios. Se realizaron un total de 100 pruebas siendo cada una de éstas una instancia en donde la imagen seleccionada de un conjunto de imágenes ejemplares, fue afectada con ruido aleatorio.



Figura 5.3: Las 5 vocales como conjunto de vectores ejemplares.

**Ejemplo 5.3.** El conjunto de vectores ejemplares consta de 5 imágenes binarias correspondientes a las vocales ‘A,E,I,O,U’ [57] (ver Fig. 5.3) cuyo tamaño es de  $7 \times 7$  pixeles de modo que cada vector tiene una dimensión  $n = 49$ . En la gráfica de la Fig. 5.4 se muestra el desempeño de la memoria  $W$  para niveles de ruido del 10 % al 90 %.

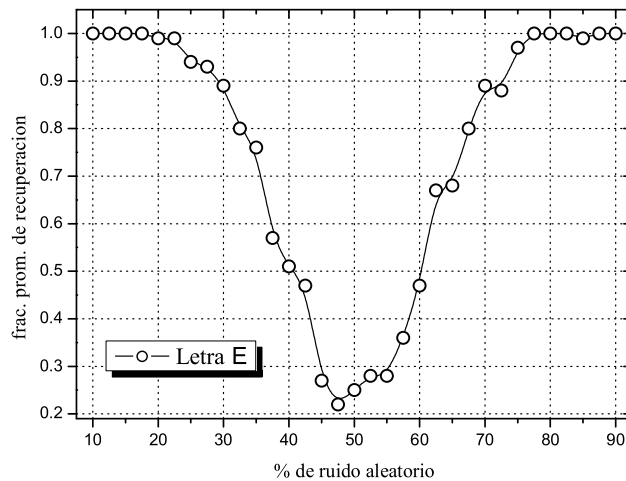


Figura 5.4: Desempeño de la memoria  $W$  en la recuperación de la letra E.

**Ejemplo 5.4.** El conjunto de vectores ejemplares consta de las 14 imágenes binarias correspondientes al Font E13B (ver Fig. 2.5) de tamaño  $14 \times 18$  píxeles de modo que cada vector tiene una dimensión  $n = 252$ . La gráfica de la Fig. 5.5 muestra el desempeño de la memoria  $W$  para niveles de ruido del 10 % al 90 %.

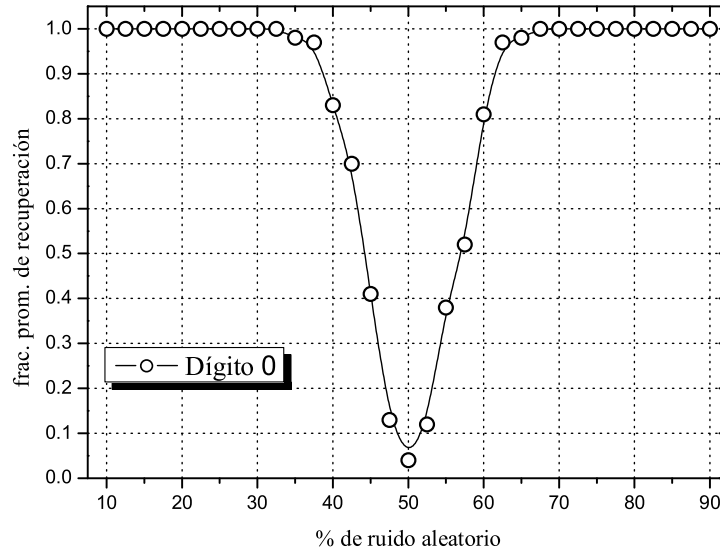


Figura 5.5: Desempeño de la memoria  $W$  en la recuperación del dígito 0.

En ambas gráficas se observa que los porcentajes de recuperación promedio para ruido entre 45 % y 55 % decaen considerablemente y evidentemente el desempeño de la memoria  $W$  en la recuperación disminuye drásticamente. Esto se debe a que en este intervalo de porcentajes de niveles de ruido, los objetos y el fondo de la imagen se confunden y en consecuencia son indistinguibles en relación a cualquiera de las imágenes ejemplares. Sin embargo, para porcentajes de ruido fuera de este rango (bajos y altos niveles de ruido) la memoria tiene un desempeño robusto. En el caso de la letra ‘E’, para porcentajes de ruido menores a 30 % y mayores al 70 % la recuperación es casi perfecta pues  $fpr. \approx 0.9$ . Similarmente para el dígito ‘0’, para porcentajes menores al 40 % y mayores al 60 % la recuperación es casi perfecta pues  $fpr. \approx 0.9$ . A demás, aunque no se muestra en las gráficas, para porcentajes menores al 10 % y mayores al 90 % la recuperación es perfecta.

### 5.3. Imágenes en tonos de gris

La técnica del enmascaramiento del ruido para procesar imágenes en tonos de gris es más sencilla y su tiempo de cómputo es menor que en el caso binario ya que no es necesario utilizar la máscara negativa. La Fig. 5.6 muestra una imagen ejemplar (gato), varias versiones con ruido y las correspondientes máscaras mín y máx.

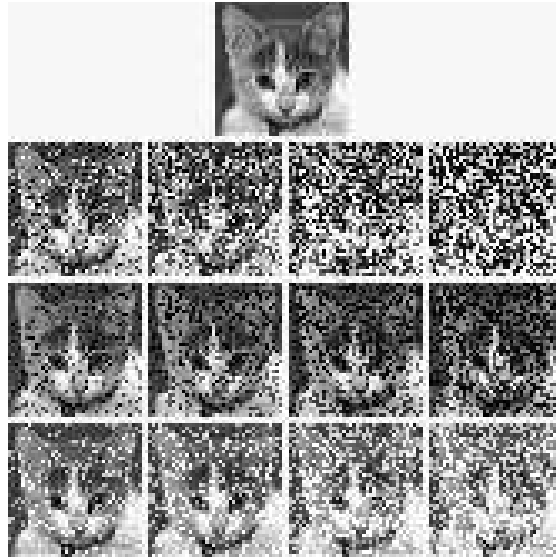


Figura 5.6: 1er renglón, imagen ejemplar; 2do renglón, imágenes afectadas con ruido aleatorio al 25 %, 45 %, 65 % y 85 %; 3er renglón, máscaras mín ( $\tilde{\mathbf{x}}_e^\xi$ ); 4to renglón, máscaras máx ( $\tilde{\mathbf{x}}_d^\xi$ ).

#### 5.3.1. Medidas del error

Para cuantificar la diferencia entre dos vectores, por ejemplo, entre el vector recuperado por la memoria  $W$  (resp.  $M$ ) con un vector ejemplar o entre una versión ruidosa de un ejemplar con el mismo, hacemos uso de las medidas dadas en la siguiente definición.

**Definición 5.3.** Si  $\tilde{\mathbf{x}}^\xi$  es la versión con ruido aleatorio del vector  $\mathbf{x}^\xi$ , el *error cuadrático medio normalizado* (ECMN) y la *razón señal a ruido* (RSR) están dados por

$$\varepsilon_r(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{i=1}^n (x_i - \tilde{x}_i)^2 / \sum_{i=1}^n x_i^2 \quad (5.6)$$

$$r_s(\mathbf{x}, \tilde{\mathbf{x}}) = 10 \log \left( \frac{\sum_{i=1}^n x_i^2}{\sum_{i=1}^n (x_i - \tilde{x}_i)^2} \right) \quad (5.7)$$

El Algoritmo 7 detalla la técnica del enmáscaramiento del ruido para imágenes en tonos de gris.

El conjunto de entrada  $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\}$  es una matriz de tamaño  $n \times k$ , donde  $n$  es el número de píxeles en cada imagen y  $k$  es el número de vectores ejemplares almacenados en la matriz de pesos  $W$  o  $M$ . El mecanismo provee un vector de entrada para utilizar la memoria  $W$  o  $M$ .

**Input:**  $\tilde{\mathbf{x}} \in \mathbb{R}^n$ ,  $b \in \{0, 1\}$  [vector ejemplar ruidoso, bandera que elige  $W$  o  $M$ ]

**if**  $b = 0$  **then** [usa la memoria mín  $W$ ]

**for**  $\xi = 1$  **to**  $k$  **do** [escudriña todos los vectores en  $X$ ]

$\tilde{\mathbf{x}}_e^\xi = \mathbf{x}^\xi \wedge \tilde{\mathbf{x}}$  [máscara-min de  $k$  vectores]

$v_\xi = 0.5[\mu(\tilde{\mathbf{x}}_e^\xi, \mathbf{x}^\xi) + \mu(\tilde{\mathbf{x}}_e^\xi, \tilde{\mathbf{x}})]$  [máscara-e, vector discriminante]

**end**

**find**  $\gamma \in \{1, \dots, k\}$  **such that** [busca índice correspondiente al]

$v_\gamma = \bigwedge_{\xi=1}^k v_\xi$  [vector candidato enmasc.-min]

**let**  $\mathbf{y} = W_{XX} \boxtimes \tilde{\mathbf{x}}_e^\gamma$  [vector de salida de enmasc.-min]

**else** [usa la memoria máx  $M$ ]

$\tilde{\mathbf{x}} \in \mathbb{R}^n$  [ejemplar con ruido aleatorio]

**for**  $\xi = 1$  **do** [escudriña todos los vectores en  $X$ ]

$\tilde{\mathbf{x}}_d^\xi = \mathbf{x}^\xi \vee \tilde{\mathbf{x}}$  [máscara-máx de  $k$  vectores]

$v_\xi = 0.5[\mu(\tilde{\mathbf{x}}_d^\xi, \mathbf{x}^\xi) + \mu(\tilde{\mathbf{x}}_d^\xi, \tilde{\mathbf{x}})]$  [máscara-d vector discriminante]

**end**

**find**  $\gamma \in \{1, \dots, k\}$  **such that** [busca índice correspondiente al]

$v_\gamma = \bigwedge_{\xi=1}^k v_\xi$  [vector candidato enmasc.-máx]

**let**  $\mathbf{y} = M_{XX} \boxtimes \tilde{\mathbf{x}}_d^\gamma$  [vector de salida de enmasc.-máx]

**end**

**Output:**  $\mathbf{y} \in \mathbb{R}^n$  [mejor asociación para el ejemplar]

**Algoritmo 7:** Enmascaramiento de imágenes en tonos de gris

La variante realizada al algoritmo reduce el tiempo de cómputo al efectuar la siguiente modificación en el cálculo del vector discriminante

- si el ECMN es elegido como el criterio discriminante,

$$v_\xi = \mu(\tilde{\mathbf{x}}_e^\xi, \mathbf{x}^\xi), \quad v_\gamma = \bigwedge_{\xi=1}^k v_\xi \quad \text{si } b = 0,$$

$$v_\xi = \mu(\tilde{\mathbf{x}}_d^\xi - \tilde{\mathbf{x}}, \mathbf{x}^\xi), \quad v_\gamma = \bigvee_{\xi=1}^k v_\xi \quad \text{si } b = 1.$$

- en cambio, si se selecciona la medida RSR,

$$v_\xi = \mu(\tilde{\mathbf{x}}_e^\xi, \mathbf{x}^\xi), \quad v_\gamma = \bigvee_{\xi=1}^k v_\xi \quad \text{si } b = 0,$$

$$v_\xi = \mu(\tilde{\mathbf{x}}_d^\xi - \tilde{\mathbf{x}}, \mathbf{x}^\xi), \quad v_\gamma = \bigwedge_{\xi=1}^k v_\xi \quad \text{si } b = 1.$$

### 5.3.2. Recuperación de imágenes con ruido

Se presentan un par de ejemplos y sus gráficas para ver el desempeño de la memoria  $W$  después de efectuar 100 pruebas contaminando cada vector ejemplar. Todas las imágenes han sido degradadas con ruido impulsivo aleatorio utilizando la función  $\text{IN}(\cdot)$  (Impulse Noise, ver Algoritmo 8 para el pseudocódigo de esta función). Esta función cambia los valores de pixel en la imagen original según un porcentaje de nivel de ruido por un 0 o 255, los cuales ocurren con la misma probabilidad.

<b>Input:</b> $\mathbf{A} \in \mathbb{R}^{n \times n}, p$	[imagen ejemplar, porcentaje de ruido]
$q = (1 - \frac{p}{100})$	
<b>for</b> $i = 1$ <b>to</b> $n$ <b>do</b>	[escudriña todos los renglones en $A$ ]
<b>for</b> $j = 1$ <b>to</b> $n$ <b>do</b>	[escudriña todos las columnas en $A$ ]
$\eta_{ij} = \text{if}(\text{rnd}(1) > q, \text{if}(\text{rnd}(1) > 0.5, 255, 0), A_{ij})$	
<b>end</b>	
<b>Output:</b> $\eta$	

**Algoritmo 8:** Función para generar ruido impulsivo

**Ejemplo 5.5.** El conjunto de vectores ejemplares consta de 5 imágenes en tonos de gris gato, perro, zorro, león y mono (ver Fig. 5.7), de tamaño  $40 \times 40$  pixeles de modo que cada vector tiene una dimensión  $n = 1600$ .

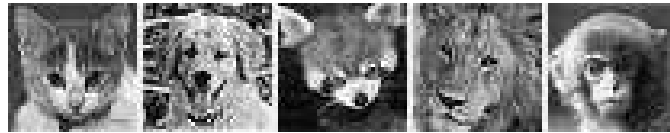


Figura 5.7: 1er conjunto de imágenes ejemplares en tonos de gris (animales).

Las figuras 5.8 y 5.9 muestran la capacidad de recuperación de la memoria  $W$  para niveles de ruido del 50% al 100% utilizando las medidas del error  $\varepsilon_r$  y  $r_s$ . En estas mismas gráficas



se observa, respectivamente que, los porcentajes de recuperación promedio para ruido del 85 % al 100 % y del 75 % al 100 % respectivamente, decaen considerablemente y evidentemente el desempeño de la memoria  $W$  en la recuperación disminuye como es de esperarse. Naturalmente, esto se debe a que para altos porcentajes de niveles de ruido la memoria confunde la imagen ruidosa y en consecuencia no asocia correctamente su imagen ejemplar. Este comportamiento es típico en cualquier modelo de memoria asociativa. No obstante, para porcentajes de ruido menores a 85 % y 75 % respectivamente, puede afirmarse que la capacidad de recuperación de la memoria  $W$  es *robusta* en presencia de cantidades considerables de ruido aleatorio. Por ejemplo, para porcentajes menores a 50 % la recuperación es *perfecta*.

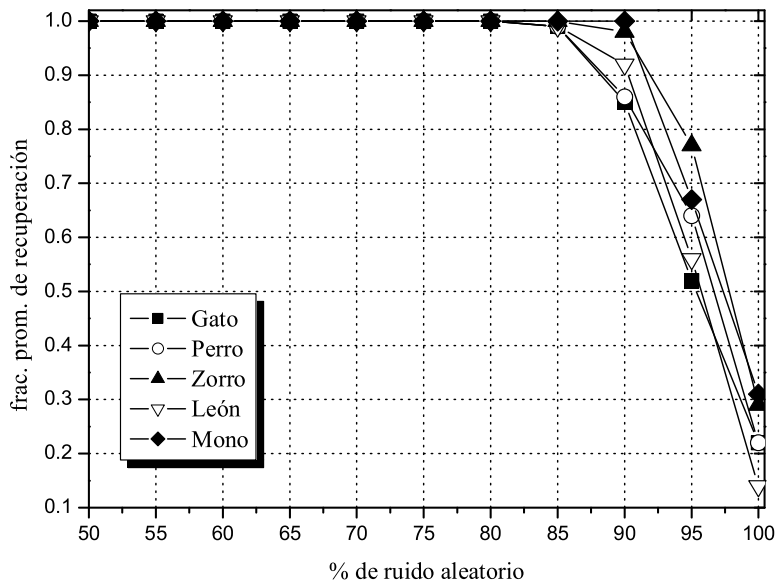


Figura 5.8: Curvas de respuesta de la memoria  $W$  según  $\varepsilon_r$ .

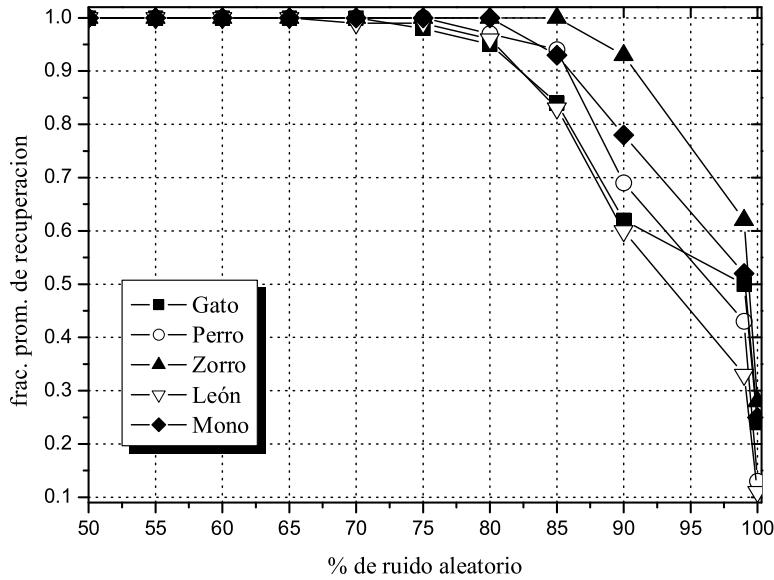


Figura 5.9: Curvas de respuesta de la memoria  $W$  según  $r_s$ .

**Ejemplo 5.6.** El conjunto de vectores ejemplares consta de 5 imágenes en tonos de gris (ver Fig. 5.10) de tamaño  $50 \times 50$  pixeles de modo que cada vector tiene una dimensión  $n = 2500$ .



Figura 5.10: 2do conjunto de imágenes ejemplares en tonos de gris (pericos). De izq. a der.:  $\mathbf{x}^1, \dots, \mathbf{x}^5$ .

Las figuras 5.11 y 5.12 muestran la capacidad de recuperación de la memoria  $W$  para niveles de ruido del 50% al 100% utilizando las medidas del error  $\varepsilon_r$  y  $r_s$ . En estas mismas gráficas se observa que los porcentajes de recuperación promedio para ruido del 80% al 100% y del 70% al 100% respectivamente, decaen considerablemente y evidentemente el desempeño de la memoria  $W$  en la recuperación disminuye considerablemente. Naturalmente, esto se debe a que para altos porcentajes de niveles de ruido la memoria confunde la imagen ruidosa y en consecuencia no asocia correctamente su imagen ejemplar. Este comportamiento es típico en cualquier modelo de memoria asociativa. No obstante, para porcentajes de ruido menores a 80% y 70% respectivamente, puede afirmarse que la capacidad de recuperación de la memoria  $W$  es *robusta* en presencia de cantidades considerables de ruido aleatorio. Por ejemplo, para porcentajes menores a 50% la recuperación es *perfecta*.

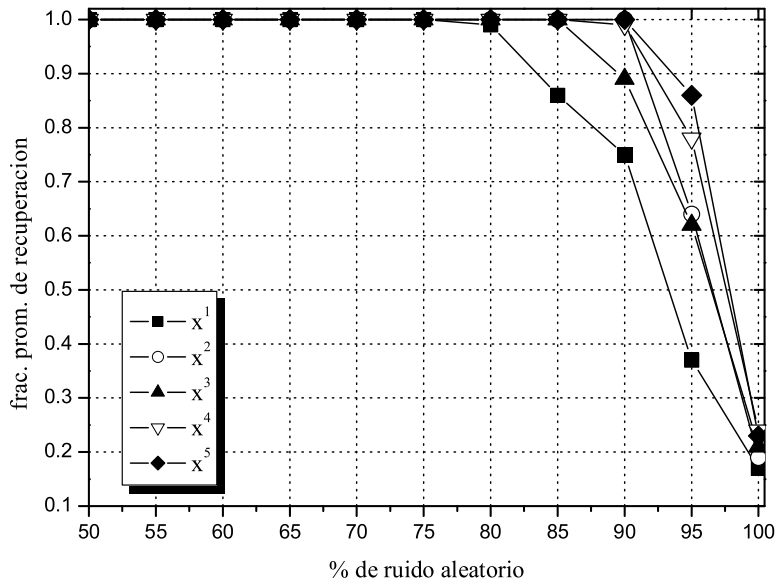


Figura 5.11: Curvas de respuesta de la memoria  $W$  según  $\varepsilon_r$ .

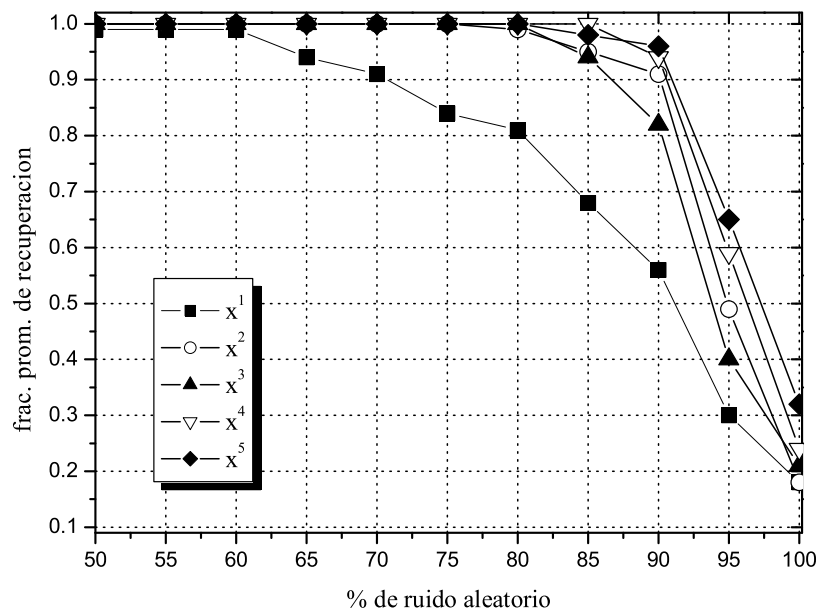


Figura 5.12: Curvas de respuesta de la memoria  $W$  según  $r_s$ .

## 5.4. Imágenes a color

En esta sección se desarrolla una extensión de la técnica de enmáscaramiento del ruido para imágenes digitales codificadas o representadas en el sistema de color RGB (Red, Green, Blue) basado en los colores primarios rojo, verde y azul [15]. Comenzamos con una breve descripción del modelo de color RGB.

### 5.4.1. Modelo de color RGB

El modelo de color RGB es el más común en el procesamiento de imágenes digitales y está basado en el sistema de coordenadas Cartesianas. La Fig. 5.13 muestra un cubo donde los colores pri-

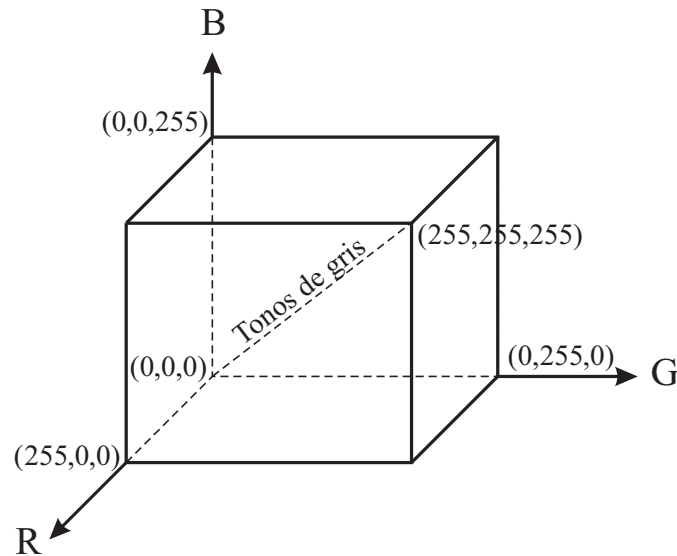


Figura 5.13: Cubo RGB en el primer octante del sistema Cartesiano tridimensional.

marios R,G y B se sitúan en tres esquinas cuyas coordenadas son  $(255, 0, 0)$ ,  $(0, 255, 0)$ ,  $(0, 0, 255)$ . El negro se encuentra en el origen  $(0, 0, 0)$  y el blanco en las coordenadas  $(255, 255, 255)$ . Dado que el número de bits para representar una imagen monocromática con 256 tonos de gris es de 8 bits por pixel, entonces para una imagen a color RGB se requieren 24 bits (8 bits por canal). Así, el número total de colores posibles en una imagen de 24 bits es de  $256^3 = 16,777,216$ . La Fig. 5.14 muestra el cubo de color RGB de 24 bits correspondiente a la Fig. 5.13. El procesamiento de imágenes a color se divide en dos categorías principales [15]. La primera, procesa cada componente de la imagen a color individualmente, es decir, por canal y después del procesamiento se integran los canales para formar la imagen a color. La segunda, procesa los pixeles de color

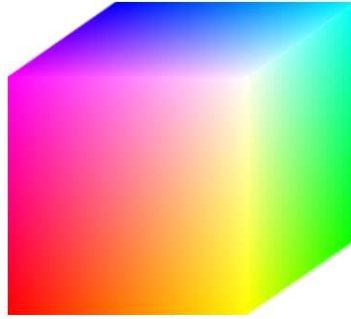


Figura 5.14: Cubo RGB (sólido) que muestra una cierta gama de colores.

directamente, es decir, como vectores de tres dimensiones o también como cadenas binarias de 24 bits. En este trabajo el tratamiento de las imágenes a color se realiza siguiendo la primera categoría. La Fig. 5.15 muestra un ejemplo de una imagen a color RGB de tamaño  $256 \times 256$  píxeles

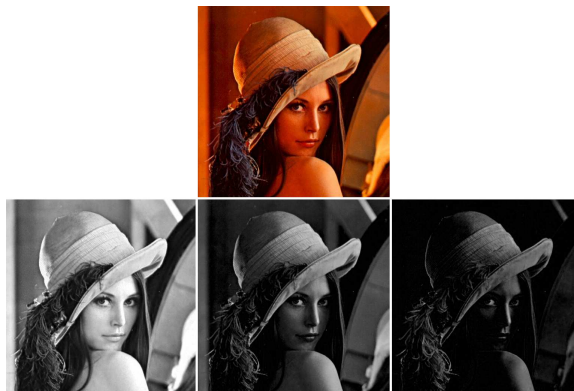


Figura 5.15: 1er renglón, imagen a color RGB; 2do renglón imágenes monocromáticas, canales R, G y B.

y las imágenes monocromáticas correspondientes a los canales rojo, verde y azul, en las cuales se descompone ésta imagen. Obsérvese que estas imágenes monocromáticas tienen el mismo tamaño. Simbólicamente, una imagen a color la representamos como

$$\mathbf{X} = (\mathbf{X}^R, \mathbf{X}^G, \mathbf{X}^B), \quad (5.8)$$

donde  $\mathbf{X}$  es una *matriz empacada*, en general de tamaño  $m \times 3n$ , que representa la imagen a color, mientras que  $\mathbf{X}^R, \mathbf{X}^G, \mathbf{X}^B$  son matrices ordinarias de tamaño  $m \times n$ , cada una de las cuales representa la imagen monocromática de los canales R, G y B. La forma vectorial equivalente de la

representación anterior es la siguiente.

$$\mathbf{x} = (\mathbf{x}^R, \mathbf{x}^G, \mathbf{x}^B), \quad (5.9)$$

donde  $\mathbf{x}$  es un *vector empacado* de tamaño  $3mn$  que representa la imagen a color y  $\mathbf{x}^R, \mathbf{x}^G, \mathbf{x}^B$  son los vectores de tamaño  $mn$  correspondientes a las imágenes monocromáticas de los canales R, G y B. Recuérdese que la conversión de una matriz a un vector se realiza usando la Ec. (2.1).

### 5.4.2. Investigaciones previas

Oh y Žak [44] proponen un modelo de red neuronal denominado “estado cerebral encajonado generalizado” (*gBSB, generalized Brain-State-in-a-Box*). El modelo es una red neuronal recurrente no lineal (tipo Hopfield) que emplea el método de descomposición por traslapamiento. Consiste en descomponer una imagen en  $k$  sub-imágenes del mismo tamaño que son convertidas en vectores binarios utilizados para construir sub-matrices que los almacenan como puntos de equilibrio estables. Dado que cada sub-matriz forma una sub-red habrá  $k$  sub-redes que se construyen independientemente de las demás. Al reducir el tamaño de la imagen ejemplar y procesarla mediante sub-redes, se logra disminuir considerablemente el esfuerzo computacional. Sin embargo, el modelo puede perder una cierta capacidad de recuperación por lo que emplean un algoritmo de corrección del error en el traslapamiento de las sub-imágenes para mejorar el desempeño de la red. También, cuantizan el rango dinámico para reducir el número de bits requeridos en la representación binaria de un pixel. La cuantización uniforme la efectúan dividiendo el rango  $[0, 255]$  usando 4 intervalos de la misma longitud y asignando un número binario de 2 dígitos a cada intervalo. Por tanto cada pixel de una imagen a color RGB lo consideran una cadena de 24 bits que cuantizan a 6 bits (2 bits por canal). Además, proponen una red híbrida *gBSB* combinada con el modelo neuronal de McCulloch-Pitts cuya diferencia con el modelo antes descrito radica en la función de activación usada. Yañez et al. [71] proponen el uso de las memorias autoasociativas alfa-beta para almacenar y recuperar imágenes a color. Dado que estas memorias se diseñaron para tratar imágenes binarias, extienden éstas al caso de imágenes a color codificando cada pixel como la concatenación de vectores binarios de 8 bits correspondientes a los pixeles de las imágenes en tonos de gris de cada canal. Así, forman vectores de 24 bits que son empleados durante las etapas de almacenamiento y recuperación siguiendo la definición de una memoria alfa-beta [70] con operador máximo. Sin embargo, este modelo está restringido a tratar únicamente con ruido aleatorio dilatativo. Un tercer modelo (indirecto), recientemente desarrollado por Vázquez y Sossa [67] propone una memoria heteroasociativa bidireccional (*BAM, Bidirectional Associative Memory*) denominada *BW* para

tratar imágenes a color usando el concepto de pesos sinápticos variables. Emplea dos matrices, la primera almacena pares de vectores hacia delante y la segunda pares de vectores hacia atrás. Previo a la construcción de estas dos matrices, los vectores ejemplares son modificados mediante un valor de desplazamiento (parámetro interactivo) el cual es también empleado en la etapa de recuperación y básicamente determina la capacidad de recuperación. Este modelo requiere de varias transformaciones antes y después de aplicar las memorias.

Puede observarse que los dos primeros modelos descritos antes tratan cada pixel de la imagen a color como una cadena de 24 bits ya que las redes subyacentes operan con vectores binarios y el tercer modelo descrito convierte las cadenas de 24 bits en números decimales. En la subsección 5.4.4 se detalla el desempeño en la recuperación de imágenes a color con ruido de cada uno de estos modelos, así como una comparación con la extensión propuesta aquí. Cualitativamente, nuestro modelo presenta las siguientes ventajas: 1) es directo por que puede trabajar con los valores en tonos de gris sin necesidad de realizar una codificación o conversión previa, 2) no es paramétrico, 3) no requiere cuantizar el rango dinámico, 4) el método computacional no requiere pasos intermedios y es paralelizable ya que el modo de operación es por canal y 5) tiene mayor tolerancia a cantidades de ruido aditivo (tipo impulsivo) mayores al 50 % y menores al 90 %.

### 5.4.3. Ruido aditivo en imágenes RGB

A diferencia de las imágenes binarias y en tonos de gris en las cuales se utilizó ruido impulsivo bipolar (“sal y pimienta”), para las imágenes a color RGB consideramos en este trabajo que pueden ser degradadas aditivamente por canal con ruido aleatorio tipo impulsivo. Por tanto, si  $\mathbf{x}$  es el vector que representa la imagen monocromática de un cierto canal de una imagen a color RGB el modelo aditivo de degradación se define como [36]

$$\tilde{\mathbf{x}}^c = \mathbf{x}^c + \mathbf{a}^c \quad ; \quad c = R, G, B. \quad (5.10)$$

donde  $\tilde{\mathbf{x}}^c$  es el vector correspondiente al canal  $c$  afectado con ruido, mientras que  $\mathbf{x}^c$  es el vector que representa el canal  $c$  de la imagen original y  $\mathbf{a}^c = (\pm a_i^c)$  es el vector correspondiente al ruido agregado a ese canal, donde  $a_i^c \in \{-a, 0, a\}$  y  $a \in (0, 255] \subset \mathbb{Z}^+$ . Dada la naturaleza multicanal de una imagen digital RGB el ruido puede afectar uno o más canales en el proceso de adquisición o captura de la imagen. Si la imagen a color es alterada, entonces según la Ec. (5.10), ésta corresponde a tres imágenes en tonos de gris las cuales pueden procesarse de manera independiente para eliminar el ruido. Para un tratamiento general de diversos modelos de ruido en imágenes a color pueden consultarse [5, 36, 47]. Desde un punto de vista computacional y con el propósito

de realizar pruebas con diversas instancias de ruido, supondremos que los canales en una imagen a color pueden alterarse en forma *no correlacionada* o *correlacionada espacialmente*. De este modo, no es difícil ver que existen 11 eventos de ruido aditivo dados en la Tabla 5.1; de estos posibles eventos, 7 son no correlacionados y 4 son correlacionados. A continuación se describen matemáticamente las expresiones relativas a cada uno de los eventos de ruido listados en esta tabla.

Tabla 5.1: Eventos de ruido de una imagen a color RGB.

Evento de ruido	Tipo	Canales afectados
$E_1$	nc	R
$E_2$	nc	G
$E_3$	nc	B
$E_4$	nc/c	R,G
$E_5$	nc/c	R,B
$E_6$	nc/c	G,B
$E_7$	nc/c	R,G,B

**Ruido no correlacionado:** en este caso el ruido simplemente se agrega a un canal, a una combinación de dos canales, o a los tres canales como se indica en las siguientes expresiones,

$$\begin{aligned}
 E_1 : (\tilde{\mathbf{x}}^R, \mathbf{x}^G, \mathbf{x}^B) &= (\mathbf{x}^R + \mathbf{a}^R, \mathbf{x}^G, \mathbf{x}^B), \\
 E_2 : (\mathbf{x}^R, \tilde{\mathbf{x}}^G, \mathbf{x}^B) &= (\mathbf{x}^R, \mathbf{x}^G + \mathbf{a}^G, \mathbf{x}^B),
 \end{aligned} \tag{5.11}$$

$$\begin{aligned}
 E_3 : (\mathbf{x}^R, \mathbf{x}^G, \tilde{\mathbf{x}}^B) &= (\mathbf{x}^R, \mathbf{x}^G, \mathbf{x}^B + \mathbf{a}^B). \\
 E_4 : (\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G, \mathbf{x}^B) &= (\mathbf{x}^R + \mathbf{a}^R, \mathbf{x}^G + \mathbf{a}^G, \mathbf{x}^B), \\
 E_5 : (\tilde{\mathbf{x}}^R, \mathbf{x}^G, \tilde{\mathbf{x}}^B) &= (\mathbf{x}^R + \mathbf{a}^R, \mathbf{x}^G, \mathbf{x}^B + \mathbf{a}^B),
 \end{aligned} \tag{5.12}$$

$$\begin{aligned}
 E_6 : (\mathbf{x}^R, \tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B) &= (\mathbf{x}^R, \mathbf{x}^G + \mathbf{a}^G, \mathbf{x}^B + \mathbf{a}^B). \\
 E_7 : (\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B) &= (\mathbf{x}^R + \mathbf{a}^R, \mathbf{x}^G + \mathbf{a}^G, \mathbf{x}^B + \mathbf{a}^B).
 \end{aligned} \tag{5.13}$$

**Ruido correlacionado:** los eventos  $E_1, E_2, E_3$  claramente no pueden ser correlacionados porque representan un solo canal, en cambio la correlación espacial (píxeles afectados por ruidos que ocurren en las mismas posiciones) puede ocurrir en una combinación de dos canales o en los tres canales fijando uno de ellos como referencia para alterar los otros. Primero, una vez fijado el canal  $c$ , se marcan las posiciones en las que este canal se ve afectado por ruido y se obtienen calculando el siguiente vector auxiliar

$$\tilde{\mathbf{y}}^c = |\tilde{\mathbf{x}}^c - \mathbf{x}^c|, \tag{5.14}$$



cuyas componentes distintas de cero indican la posición de aquellos pixeles modificados por el ruido. Entonces, los pixeles de otro canal  $d \neq c$  se modifican según el vector  $\mathbf{y}$ ; por tanto, la siguiente operación vectorial

$$\mathbf{z}^d = \tilde{\mathbf{y}}^c \wedge \mathbf{x}^d \quad (5.15)$$

realiza un enmascaramiento cuyas componentes mayores (estrictamente) a cero señalan que pixeles en el canal  $d$  deben ser modificados, de manera que para  $i = 1, \dots, mn$

$$\tilde{x}_i^d = \begin{cases} x_i^d \pm a \Leftrightarrow z_i^d > 0 \\ x_i^d \Leftrightarrow z_i^d = 0, \end{cases} \quad (5.16)$$

expresión que escribiremos en forma simplificada mediante la *función condicional*, es decir,  $\tilde{x}_i^d = \zeta(z_i^d > 0, x_i^d \pm a, x_i^d)$  (recuerdese que esta función la representamos en pseudocódigo como  $\mathbf{if}(z_i^d > 0, x_i^d \pm a, x_i^d)$ ). Específicamente, las ecuaciones que describen los eventos  $E_4, \dots, E_7$  correspondientes a ruido correlacionado se listan a continuación,

$$\begin{aligned} E_4 : (\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G, \mathbf{x}^B) &= (\tilde{\mathbf{x}}^R, \zeta(z_i^G > 0, x_i^G \pm a, x_i^G), \mathbf{x}^B) \quad \text{con} \quad \mathbf{z}^G = |\tilde{\mathbf{x}}^R - \mathbf{x}^R| \wedge \mathbf{x}^G, \\ E_5 : (\tilde{\mathbf{x}}^R, \mathbf{x}^G, \tilde{\mathbf{x}}^B) &= (\tilde{\mathbf{x}}^R, \mathbf{x}^G, \zeta(z_i^B > 0, x_i^B \pm a, x_i^B)) \quad \text{con} \quad \mathbf{z}^B = |\tilde{\mathbf{x}}^R - \mathbf{x}^R| \wedge \mathbf{x}^B, \\ E_6 : (\mathbf{x}^R, \tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B) &= (\mathbf{x}^R, \tilde{\mathbf{x}}^G, \zeta(z_i^B > 0, x_i^B \pm a, x_i^B)) \quad \text{con} \quad \mathbf{z}^B = |\tilde{\mathbf{x}}^G - \mathbf{x}^G| \wedge \mathbf{x}^B. \\ E_7 : (\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B) &= (\tilde{\mathbf{x}}^R, \zeta(z_i^G > 0, x_i^G \pm a, x_i^G), \zeta(z_i^B > 0, x_i^B \pm a, x_i^B)) \quad \text{con} \\ &\mathbf{z}^G = |\tilde{\mathbf{x}}^R - \mathbf{x}^R| \wedge \mathbf{x}^G \quad \text{y} \quad \mathbf{z}^B = |\tilde{\mathbf{x}}^R - \mathbf{x}^R| \wedge \mathbf{x}^B. \end{aligned} \quad (5.18)$$

La Fig. 5.16 ilustra un ejemplo con 8 eventos de ruido seleccionados arbitrariamente. En particular, se eligieron  $E_1, E_2, E_3$  y  $E_7$  no correlacionados y  $E_4, E_5, E_6, E_7$  correlacionados. El tamaño de la imagen es de  $334 \times 334$  pixeles. En cada evento la imagen ha sido degradada con ruido aditivo utilizando una probabilidad del 20% y una constante aditiva (intensidad)  $a = 128$ .

**Ejemplo 5.7.** La Fig 5.17 muestra una imagen a color con sus tres componentes R,G y B, la imagen degradada con el 20% de ruido correlacionado (evento  $E_7$ , Ec. (5.18)) y las máscaras erosivas y dilatativas correspondientes a cada canal.



Figura 5.16: 1er. renglón primeras 3 cols., ruido agregado a los canales R,G y B (no correlacionado); 1er renglón 4ta col., ruido agregado a los canales R,G (correlacionado); 2do. renglón primeras dos cols., ruido agregado a los canales R,B y G,B (correlacionado); 2do. renglón 3ra col. ruido agregado a los tres canales R,G,B (correlacionado); 2do. renglón última col., ruido agregado a los tres canales (no correlacionado).

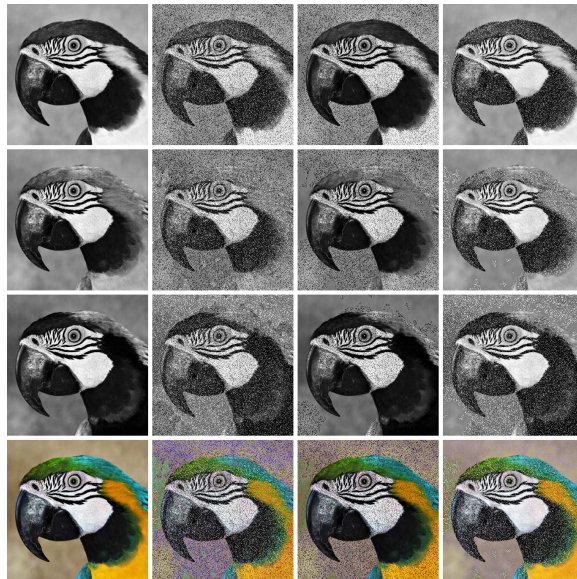


Figura 5.17: 1ra col., imágenes monocromáticas de los canales R,G y B y su correspondiente imagen a color; 2da col., canales afectados con 20% de ruido correlacionado ( $a = 128$ ) e imagen a color compuesta; 3era col., máscaras erosivas por canal y equivalente en color; 4ta col., máscaras dilatativas por canal y equivalente en color.

#### 5.4.4. Recuperación de imágenes a color

La Fig. 5.18 muestra el conjunto de 5 imágenes ejemplares a color utilizadas para determinar la capacidad de recuperación de las memorias autoasociativas reticulares, en particular empleamos la memoria mín- $W$ .



Figura 5.18: Imágenes ejemplares a color almacenadas en la MARM compuesta  $\mathbf{W} = (W^R, W^G, W^B)$ . De izq. a der.:  $\mathbf{x}^1, \dots, \mathbf{x}^5$ .

Cada imagen ejemplar, de  $50 \times 50$  pixeles, se contamina 100 veces con ruido aditivo tipo impulsivo donde  $a = 128$ . Ya que las imágenes a color se separan en sus tres componentes, se requiere construir tres memorias  $W$ , una para cada canal almacenando 5 imágenes monocromáticas. El algoritmo dado para tratar con imágenes en tonos de gris se aplica entonces sucesivamente a los conjuntos  $(X^{1R}, \dots, X^{5R}), (X^{1G}, \dots, X^{5G})$  y  $(X^{1B}, \dots, X^{5B})$  con sus respectivas memorias mínimas  $W^R, W^G$  y  $W^B$  que denotamos por  $\mathbf{W}$ . Las figuras 5.19 y 5.20 muestran las curvas de

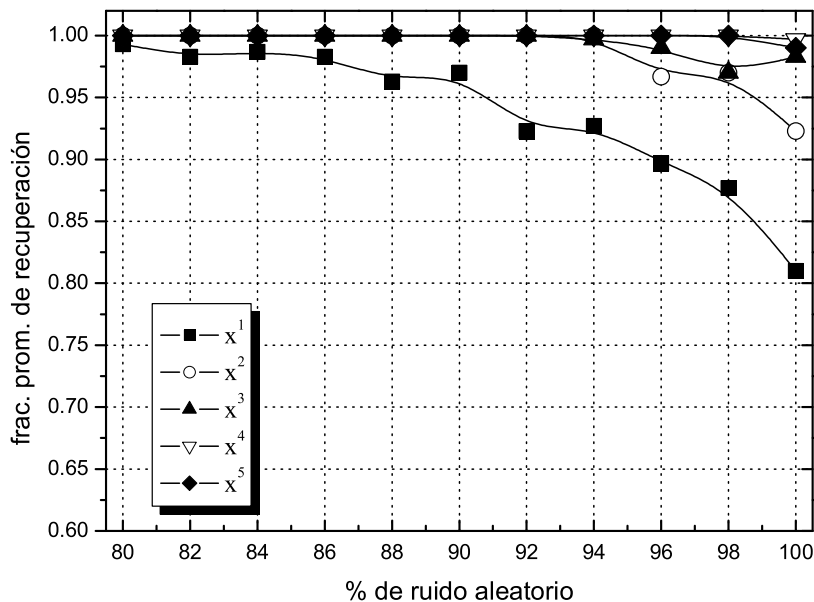


Figura 5.19: Curvas de respuestas de la MARM compuesta  $\mathbf{W} = (W^R, W^G, W^B)$  para  $E_7$  (no correlacionado).

respuesta (promedio de los tres canales) que dan el porcentaje promedio de recuperación perfecta para niveles de ruido en el intervalo de 80% al 100%. Queda entendido que este porcentaje

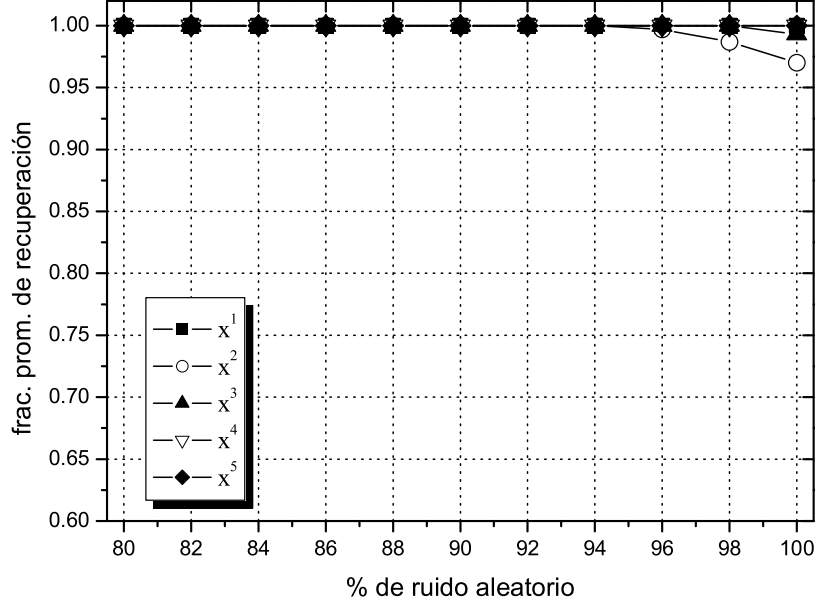


Figura 5.20: Curvas de respuestas de la MARM compuesta  $\mathbf{W} = (W^R, W^G, W^B)$  para  $E_7$  (correlacionado).

promedio se realiza sobre un cierto número de instancias (100) de ruido para cada imagen ejemplar. Específicamente se han considerado los eventos  $E_7$  no correlacionado y correlacionado que consta de afectar con ruido los tres canales. Por otra parte, la Tabla 5.2 lista valores numéricos representativos sobre 100 pruebas de cada imagen ejemplar distorsionada según cada uno de los 11 diferentes eventos de ruido (ver Tabla 5.1). En particular, el valor numérico en cada entrada de esta tabla es el promedio ( $\bar{f}_\xi$ ) de recuperaciones perfectas en 100 pruebas ( $t$ ) en el intervalo de niveles de ruido [80 %, 100 %]. Es decir, supongamos el caso del  $E_7$  no correlacionado donde el valor de la fracción promedio de recuperación perfecta ( $\bar{f}_\xi$ ) fue de 0.938. Para obtener este valor primero se le agrega ruido del 80 % al vector  $\mathbf{x}^1$  100 veces, el resultado se promedia y se divide entre 3 (número de canales), el procedimiento se repite para los porcentajes de ruido del 82 %, 84 %, ..., 100 %, estos datos forman un vector de 11 elementos que contienen el porcentaje promedio de recuperación perfecta para cada instancia de ruido del vector  $\mathbf{x}^1$ . Finalmente, se obtiene el promedio de estos resultados para llegar al valor de 0.938. En general, la Tabla 5.2 muestra que la memoria mín- $W$  tienen un alto desempeño al procesar el conjunto de vectores ejemplares correspondientes a los pericos, a demás, se puede notar que se comporta mejor al procesar los vectores correspondientes a  $\mathbf{x}^4$  y  $\mathbf{x}^5$ .

$$\bar{f}_\xi = \frac{1}{33} \sum_{i=1}^{11} \sum_{t=1}^{100} [(\gamma_{p(i)}^R = \xi) + (\gamma_{p(i)}^G = \xi) + (\gamma_{p(i)}^B = \xi)] \quad \text{con} \quad p(i) \% = 80 + 2(i - 1). \quad (5.19)$$

Tabla 5.2: Fracción promedio de recuperación perfecta en cada evento de ruido por imagen ejemplar.

Evento	Ruido	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{x}^4$	$\mathbf{x}^5$
$E_1 \{\tilde{\mathbf{x}}^R\}$	nc	1	1	1	1	1
$E_2 \{\tilde{\mathbf{x}}^G\}$	nc	0.934	0.992	0.999	1	0.999
$E_3 \{\tilde{\mathbf{x}}^B\}$	nc	0.996	0.999	0.994	1	1
$E_4 \{\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G\}$	nc	0.946	0.987	0.999	0.999	1
$E_5 \{\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^B\}$	nc	0.999	0.998	0.995	1	1
$E_6 \{\tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B\}$	nc	0.943	0.989	0.994	1	1
$E_7 \{\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B\}$	nc	0.938	0.987	0.995	0.999	0.999
$E_4 \{\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G\}$	c	1	1	1	1	1
$E_5 \{\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^B\}$	c	1	0.997	1	1	1
$E_6 \{\tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B\}$	c	0.945	0.989	0.999	1	0.999
$E_7 \{\tilde{\mathbf{x}}^R, \tilde{\mathbf{x}}^G, \tilde{\mathbf{x}}^B\}$	c	1	0.996	0.999	1	1

La Tabla 5.3 muestra las características y el desempeño en la recuperación de imágenes a color RGB con ruido de cada uno de los modelos de memorias asociativas descritos anteriormente en la subsección 5.4.2 (Investigaciones previas) y del modelo aquí desarrollado. Brevemente,

- Modelo gBSB (Oh y Žak [44]): el intervalo de porcentajes de ruido impulsivo (extremo) va del 0% al 40%; los autores dan un ejemplo de recuperación de imagen a color para una sola instancia de ruido no correlacionado contaminando cada canal con el 40%. Como resultado obtienen una diferencia de 150 píxeles erróneos de un total de 60,000 píxeles (14 imágenes ejemplares de  $300 \times 200$  píxeles), equivalente a una fracción de recuperación igual a 0.9975.
- Modelo Alfa-Beta máx (Yáñez et al. [71]): el intervalo de porcentajes de ruido impulsivo dilatativo (extremo) no correlacionado va del 0% al 50%; los autores consideran un conjunto de 10 imágenes a color de tamaño  $100 \times 75$  píxeles y dan como resultado, sin especificar el número de instancias de ruido, una fracción de recuperación unitaria. Es decir, obtienen recuperación perfecta para cualquier porcentaje en el intervalo señalado, desafortunadamente el ruido impulsivo dilatativo es una restricción de su modelo pues no tratan ruido mixto.
- Modelo BAM-BW (Vazquez y Sossa [67]): el intervalo de porcentajes de ruido impulsivo (extremo) no correlacionado va del 0% al 80%; hacemos énfasis en que este modelo propone un procedimiento indirecto, ya que para efectuar una autoasociación, los autores emplean una memoria bidireccional que implica dos conjuntos de imágenes ejemplares (20 pares

heteroasociados de  $255 \times 170$  píxeles). Como resultado obtienen una fracción de recuperación unitaria, equivalente a recuperación perfecta; no obstante, este resultado corresponde a una sola instancia de ruido para cada nivel de ruido y no dan explícitamente para su conjunto de ejemplares el valor del parámetro fundamental con el cual opera su algoritmo.

- **Modelo Enmascaramiento-MARM mín W:** para las imágenes a color se emplea el Algoritmo 7, el cual se utiliza para procesar el conjunto de imágenes monocromáticas correspondientes a los canales **R**, **G** y **B**. El intervalo de porcentajes de ruido tipo impulsivo (extremo y no extremo) correlacionado y no correlacionado va del 80 % al 100 %; se considera un conjunto de 5 imágenes a color de tamaño  $50 \times 50$  píxeles. Se obtiene una recuperación perfecta para niveles de ruido que va de 0 % a 80 % y recuperación casi-perfecta según la medida  $\varepsilon_r(\cdot)$  para niveles de ruido del 82 % al 100 %. La fracción promedio de recuperación casi-perfecta de  $E_7$  (nc) fue de  $\bar{f} = \frac{1}{5} \sum_{\xi}^5 \bar{f}_{\xi} = 0.941$ ; similarmente, para  $E_7$  (c) fue de  $\bar{f} = 0.993$  (ver Tabla 5.3). Nuestro procedimiento basa sus resultados en un promedio de la capacidad de recuperación sobre un número significativo de instancias de ruido; además, el modelo de ruido tipo impulsivo ha sido explicado en detalle.

Tabla 5.3: Comparación de memorias asociativas para almacenamiento y recuperación de imágenes a color RGB. Nomenclatura: I= gBSB, II=Alfa-Beta máx, III=BAM-BW y IV=Enmascaramiento-MARM mín W; FR=Fracción de Recuperación, NP=Número de pruebas, PA=Procesos Adicionales, Par.=Paramétrico, CE=Conjunto de Ejemplares y TD=Tipo de Dato (numérico).

Mod.	% Ruido	Tipo	FR	NP	PA	Par.	CE	TD
I	[0,40] (nc)	imp.	0.998	1	3	no	1	binarios
II	[0,50] (nc)	imp. (dilatativo)	1	1	2	no	1	binarios
III	[0,80] (nc)	imp.	1	?	3	si	2	enteros largos
IV	[0,100] (nc)	imp. ( $a = 128$ )	0.941	100	1	no	1	enteros cortos
IV	[0,100] (c)	imp. ( $a = 128$ )	0.993	100	1	no	1	enteros cortos

Tomando en consideración las ventajas cualitativas listadas en el segundo párrafo de la subsección 5.4.2 (Investigaciones previas) junto con la comparación cuantitativa dada en la Tabla 5.3, puede afirmarse que el modelo aquí propuesto (modelo IV) supera a otros modelos de memorias asociativas en el presente contexto de recuperar imágenes a color cuando éstas son afectadas por ruido impulsivo. Las dos principales ventajas de nuestro modelo son su aplicabilidad directa y su capacidad de recuperación en presencia de niveles altos de ruido.

# Capítulo 6

## Conclusiones

### 6.1. Conclusiones generales

En este trabajo de tesis se han empleado las memorias autoasociativas reticulares minimax MARM para *solucionar* el problema de la recuperación de vectores multidimensionales sujetos a ruido aleatorio, particularmente en la recuperación de imágenes binarias, en tonos de gris y a color RGB que han sido afectadas por ruido. Se introdujo una manera alternativa de solución mediante la *técnica de núcleos mejorada* la cual permite incrementar la capacidad de recuperación de las MARM al utilizar la distancia de Hamming (resp. ECMN) como medida del error y reactivar el núcleo de un vector ejemplar binario (resp. vector ejemplar en grises) posiblemente afectado con ruido. También, se desarrolló con el mayor detalle posible un esquema de memoria asociativa de naturaleza híbrida (MAMD-Hamming) para mejorar el desempeño en la recuperación de imágenes binarias al compararse con la memoria morfológica difusa introducida por Sussner. Posteriormente, y como parte central de esta tesis, se expuso la técnica de enmascaramiento del ruido [63] haciendo énfasis en las diferencias que ocurren entre imágenes binarias y en tonos de gris. Resultando que, el cómputo para imágenes binarias es mayor que para imágenes en tonos de gris debido a que el caso binario requiere de máscaras positivas para porcentajes de error menores al 50 % y de máscaras negativas para porcentajes de error mayores al 50 % (cambio inherente de codificación). En su forma inicial, la técnica de enmascaramiento aplicada a imágenes en tonos de gris, aparte de mantener su simplicidad computacional mostró ciertas limitaciones al tratar diferentes conjuntos de imágenes ejemplares. Por ello, en esta tesis se ha mejorado esta técnica al aplicarla a cualquier conjunto de imágenes sean estas en tonos de gris o a color. En este último caso, se desarrolló detalladamente un modelo de ruido tipo impulsivo que contempla los diversos eventos de ruido posibles en una



imagen a color RGB. Los resultados obtenidos e ilustrados mediante ejemplos cuidadosamente seleccionados o diseñados demuestran que el modelo de memoria autoasociativa aquí propuesto en combinación con la técnica del enmascaramiento del ruido y métricas vectoriales apropiadas ofrece una nueva solución al problema inicialmente planteado. Específicamente, el modelo arroja recuperación perfecta para vectores que han sido corrompidos con ruido en un intervalo de 0% a 80% y da recuperación casi-perfecta para vectores que han sido corrompidos con ruido en un intervalo de 82% a 100%. Además, la técnica del enmascaramiento emplea operaciones reticulares simples y puede utilizar una sola de ambas memorias autoasociativas reticulares.

## 6.2. Conclusiones específicas

Con la finalidad de puntualizar algunas conclusiones específicas del trabajo de tesis, las presentamos por capítulos.

- **Capítulo 2:** la capacidad de recuperación de vectores binarios ortogonales en la memoria asociativa lineal (MAL) y la memoria recurrente de Hopfield (MRH) es alta para porcentajes de ruido de 0% a 35% (ver Figs. 2.1 y 2.3); sin embargo, cuando los vectores binarios son arbitrarios la memoria no logra establecer la asociación correcta del vector deseado aún cuando el vector de entrada no tenga ruido (ver Figs. 2.2 y 2.4). Esto se debe a que los vectores ejemplares comparten varios píxeles entre sí y no cumplen la condición de ortogonalidad dada en la Ec. (2.6). Por otro lado, la memoria de Hamming clasificadora (MHC) propuesta por Lippman tiene un buen desempeño para porcentajes de ruido de 0% a 35% (ver Fig. 2.5), *sin requerir* la condición de ortogonalidad entre vectores ejemplares.
- **Capítulo 3:** las memorias autoasociativas reticulares minimax  $W$  y  $M$  son de recuperación perfecta para entrada perfecta y son de recuperación casi-perfecta para ruido erosivo ( $W$ ) y dilatativo ( $M$ ). En el caso de ruido mezclado se utilizó una variante del método de núcleos basado en la independencia fuerte morfológica inducida obteniendo un buen desempeño. Esta variante incrementa el desempeño de la memoria al reactivar el núcleo binario o en tonos de gris; en el primer caso, utilizando la distancia de Hamming se logra una recuperación perfecta para porcentajes de ruido aleatorio del 0% a 35%, y en el segundo caso, empleando el error cuadrático medio normalizado se obtiene recuperación perfecta para porcentajes de ruido aleatorio del 0% a 75%.



- **Capítulo 4:** se presentó la memoria asociativa morfológica difusa MAMD, introducida por Sussner, cuyo desempeño en la recuperación de imágenes binarias contaminadas con ruido aleatorio menores al 10 % es del 89.4 % para el conjunto de ejemplares de las vocales y 59.9 % para el conjunto de ejemplares del Font E13B, ambos en 100 pruebas. Además, se presentaron algunos ejemplos numéricos e imágenes para explicar el funcionamiento de una MAMD. Se desarrolló una variante para incrementar la capacidad de recuperación acoplando a la MAMD una red de Hamming, mostrando este nuevo modelo un mejor desempeño para porcentajes de ruido aleatorio por debajo del 20 %. En particular, se obtuvo el 92.2 % y el 88.6 % de recuperación perfecta, respectivamente para las vocales y el Font E13B (mismo número de pruebas).
- **Capítulo 5:** se describió la técnica del enmascaramiento del ruido para imágenes binarias y en tonos de gris, la cual emplea operaciones reticulares simples y puede utilizar una sola de ambas memorias MARM. Específicamente, las imágenes binarias hacen uso de la máscara positiva y negativa, siendo la primera utilizada para porcentajes de ruido menores al 50 % y la segunda para porcentajes de ruido mayores al 50 % (inversión de la codificación). Se usó como medida del error entre vectores binarios la distancia de Hamming normalizada (DHN). Se modificó la técnica del enmascaramiento del ruido en la parte correspondiente a la determinación de distancias entre vectores enmascarados y ejemplares. Mediante esta técnica mejorada, se obtuvieron los porcentajes de recuperación perfecta promedio elevados para porcentajes de ruido del 0 % al 80 %, mostrando así un alto desempeño para cantidades apreciables de ruido aleatorio. Este comportamiento se observó para dos conjuntos de imágenes en grises ejemplares, animales y pericos (ver, resp. Figs. 5.8 y 5.9; Figs. 5.11 y 5.12). También se propuso una extensión para tratar imágenes a color RGB realizando las etapas de almacenamiento y recuperación por canal. Esta extensión ofrece recuperación perfecta para niveles de ruido del 0 % a 80 % y recuperación casi-perfecta según el error relativo para niveles de ruido del 82 % al 100 %. Tomando en consideración las ventajas cualitativas listadas en el segundo párrafo de la subsección 5.4.2 (Investigaciones previas) junto con la comparación cuantitativa dada en la Tabla 5.3, puede afirmarse que el modelo aquí propuesto supera a otros modelos de memorias asociativas en el contexto de recuperar imágenes a color cuando éstas son afectadas por ruido impulsivo. Las dos principales ventajas de nuestro modelo son su aplicabilidad directa y su capacidad de recuperación en presencia de niveles altos de ruido. De los resultados obtenidos confirmamos que este esquema combinado, basado en las memorias mín- $W$  o máx- $M$ , resuelve definitivamente el problema de la recuperación de

imágenes binarias, en tonos de gris y a colores contaminadas con ruido aleatorio.

La figura 6.1 muestra un diagrama conceptual de los tipos de memorias asociativas tratadas en esta tesis, su interrelación y las técnicas de recuperación empleadas usando MARM. Contribuciones específicas aparecen señaladas con un asterisco.

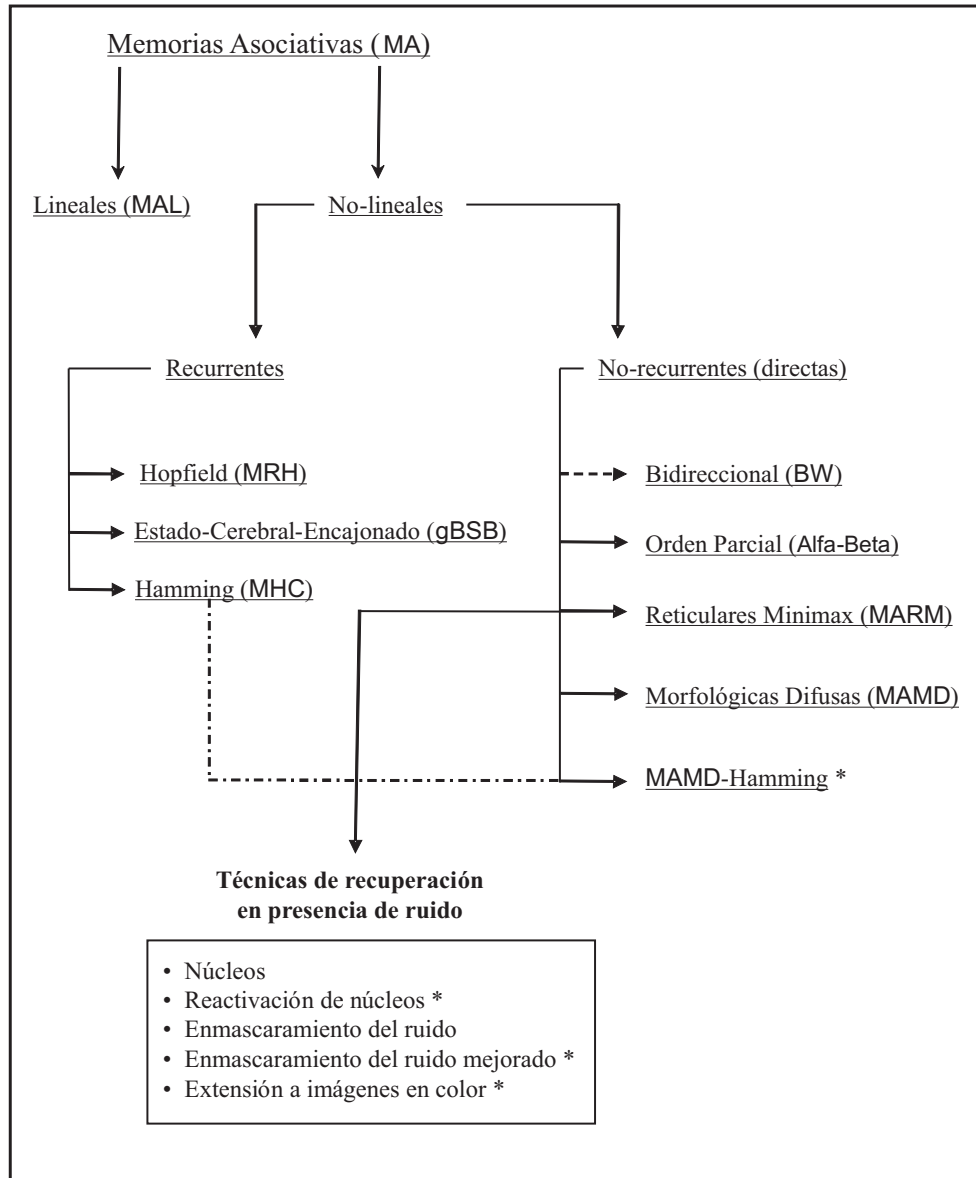


Figura 6.1: Memorias asociativas consideradas en esta tesis.

### 6.3. Aportaciones y trabajos relacionados

Las aportaciones del autor de este trabajo de tesis abarcan el diseño de ejemplos, diagramas, esquemas, nuevas ideas y variantes a diversos algoritmos. Éstas se presentan a continuación por capítulos.

- **Capítulo 2:** se diseñó el conjunto de vectores binarios ejemplares ortogonales utilizados para ejemplificar el funcionamiento de la memoria asociativa lineal y de la memoria recurrente de Hopfield (ver Fig. 2.4).
- **Capítulo 3:** se propuso una modificación al método de núcleos basado en la distancia de Hamming (imágenes binarias) o basado en el error cuadrático medio normalizado (imágenes en tonos de gris) para reactivar el núcleo correspondiente a un ejemplar afectado con ruido y así incrementar la capacidad de recuperación perfecta promedio.
- **Capítulo 4:** se diseñó un conjunto de ejemplares numéricos (ver Ejemplo 4.3) para ver el funcionamiento de las memorias autoasociativas morfológicas difusas, así como un ejemplo de imagen binaria (ver Figs. 4.3 y 4.4) para esclarecer el proceso de fusificación. Se reprodujo el algoritmo propuesto por Sussner y se propuso un nuevo modelo híbrido que acopla una red de Hamming a una MAMD (ver Algoritmo 5). La base de este capítulo se publicó en la memoria en extenso – Urcid, G., \*Nieves, J. A., Reyes, C., “Enhanced Fuzzy Autoassociative Morphological Memory for Binary Pattern Recall,” *Proceedings 4th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, pp. 116–121, 2007.
- **Capítulo 5:** se desarrolló una variante a la técnica del enmascaramiento del ruido [63] en el cálculo de distancias entre vectores enmascarados y ejemplares (imágenes en tonos de gris). La base de este capítulo se publicó en la memoria en extenso – Urcid, G., \*Nieves-V., J. A. et al. “Robust image retrieval from noisy inputs using lattice associative memories,” *Proceedings of SPIE, Image Processing: Algorithms and Systems VII*, Vol. **7245**, pp. 724517:1–12, 2009. Adicionalmente, se realizó una extensión de esta técnica para recuperar imágenes a color RGB distorsionadas con ruido aleatorio.

Otros trabajos relacionados con el desarrollo de esta tesis, presentados en Congresos y Seminarios, se listan a continuación:

■ **Memorias en extenso arbitradas**

- Urcid, G., Ritter, G. X, Nieves-V., J. A., “Redundant encoding of patterns in lattice associative memories,” *Proceedings 6th International Conference on Concept Lattices and Their Applications*, LBM, Palacký University, Olomouc, Chzech Republic, pp. 45–57, 2008.
- Urcid, G., Granados-A., F., Nieves-V., J.A., “H-L-S images of complex special functions” *Memorias XLIX Congreso Nacional de Física y XIX Reunión Anual de la AMO*, PI04:1-9, 2006.

■ **Memorias en extenso no arbitradas**

- Nieves-V. J.A., Urcid-S. G., “Memorias morfológicas autoasociativas y enmascaramiento de ruido en imágenes,” *Memorias del 9no Encuentro de Investigación INAOE*, ed. R.S. Murphy-A. et al, pp. 31-34, Noviembre, 2008.
- Nieves-V. J.A., Urcid-S. G., “Enmascaramiento de ruido aleatorio para recuperación de imágenes binarias,” *Memorias del 8avo Encuentro de Investigación INAOE*, ed. R.S. Murphy-A. et al, pp. 37-40, Noviembre, 2007.
- Nieves-V. J.A., Urcid-S. G., “Mejoramiento de las memorias autoasociativas morfológicas difusas”, *Memorias del 7to Encuentro de Investigación INAOE*, ed. R.S. Murphy-A. et al, pp. 89-92, Noviembre, 2006.

■ **Seminarios**

- Memorias asociativas morfológicas para la recuperación de imágenes binarias, *Seminario del Grupo de Instrumentación*, Noviembre, 2007.
- Redes neuronales minimax y su aplicación al procesamiento digital de imágenes binarias, *Seminario del Grupo de Instrumentación*, Abril, 2007.
- Reconocimiento de patrones e imágenes digitales usando memorias asociativas Morfológicas, *Seminario del Grupo de Instrumentación*, Octubre, 2006.

# Bibliografía

- [1] Amari, S. I., “Learning pattern sequences by self-organizing nets,” *IEEE Trans. Comp.*, **21**, pp. 1197–1206, 1972.
- [2] Amari, S. I., “Mathematical foundations of neurocomputing,” *Proceedings IEEE*, **78**(9), pp. 1443–1463, 1990.
- [3] Anderson, J. A., “A simple neural network generating an interactive memory,” *Mathematical Biosciences*, **14**, pp. 197–220, 1972.
- [4] August, M., Gerald, W., Markus, S., *Applications of Hopfield Networks*, Institute of Computer Science, University of Salzburg, 1999.
- [5] Báez, L. D., Hernández, M. F., Ramírez, J. M., “Noise in color digital images” in *Proc. of IEEE, Midwest Symposium on Circuits and Systems*, Notre Dame, Indiana, pp. 403–406, 1998.
- [6] Bezdek, J., Keller, J., Krisnapuram, R., Pal, N., *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic, Boston, MA, 1999.
- [7] Bow, S. T., *Pattern Recognition and Image Processing 2nd ed.*, Marcel Dekker, New York, 2002.
- [8] Bruck, J., “On the convergence properties of the Hopfield model,” *Proceedings IEEE*, **78**(10), pp. 1579–1585, 1990.
- [9] Carpenter, G. A., “Neural network model for pattern recognition and associative memory,” *Neural Networks*, **2**, pp. 243–257, 1989.
- [10] Cox, E., *The Fuzzy Systems Handbook; A Practitioner’s Guide to Building, Using, and Maintaining Fuzzy Systems 2nd ed.*, AP Professional, Chappaqua, New York, NY, 1998.
- [11] Cuninghame-Green, R., *Minimax Algebra*, Lecture Notes in Economics and Mathematical Systems, **166**, Springer, New York, NY, 1979.
- [12] Cuninghame-Green, R., “Minimax algebra and applications,” in *Advances in Imaging and Electron Physics*, P. Hawkes (Ed.), **90**, pp.1–121, Academic Press, California, 1995.

- [13] Dayhoff, J. E., *Neural Networks Architectures: An Introduction*, Van Nostrand Reinhold, New York, 1990.
- [14] Donnellan, T., *Lattice Theory*, Pergamon Press, Oxford, London, 1968.
- [15] Gonzales, R. C., Woods, R. E., *Digital Image Processing 2nd. ed.*, Prentice Hall, New Jersey, NJ, 2002.
- [16] Graña, M., Sussner, P., Ritter, G. X., “Innovative applications of associative morphological memories for image processing and pattern recognition,” *Mathware and Soft Computing*, **10**, pp. 155–168, 2003.
- [17] Grigori, A., *Conceptos Fundamentales de Álgebra Booleana*, Editorial Trillas, México, 1992.
- [18] Grimaldi, R. P., *Matemáticas Discreta y Combinatoria: Introducción y Aplicaciones*, Addison-Wesley Iberoamericana, México, 1989.
- [19] Guzmán, E., Pogrebnyak, O., Yañez, C., Moreno, J. A., “Image compression algorithm based on morphological associative memories,” *Progress in Pattern Recognition, Image Analysis and Applications, Proceedings Lecture Notes in Computer Science*, Springer, Berlin, Germany, **4225**, pp. 519–528, 2006.
- [20] Haykin, S., *Neural Networks: A Comprehensive Foundation*, Maxwell Macmillan, Ontario, Canada, 1994.
- [21] Hopfield, J. J., “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of the National Academy of Sciences*, **79**, pp. 2554–2558, 1982.
- [22] Hopfield, J. J., “Neurons with graded response have collective computational properties like those of two-state neurons,” *Proceedings of the National Academy of Sciences*, **81**, pp. 625–633, 1984.
- [23] Jähne, B., Haußecker, H., Geißler, P., *Handbook of Computer Vision and Applications: Signal Processing and Pattern Recognition*, **2**, Academic Press, San Diego, CA, 1999.
- [24] Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice Hall, New Jersey, NJ, 1989.
- [25] Jau, J. Y., Fainman, Y., Lee, S. H., “Comparison of artificial neural networks with pattern recognition and image processing,” *Applied Optics*, **28**(2), pp. 302–305, 1989.
- [26] Kasabov, N. K., *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, MIT Press, London, 1998.
- [27] Kerre, E., Nachtgael, M., *Fuzzy Techniques in Image Processing*, Physica Verlag, Springer, 2000.

- [28] Kohonen, T., “Correlation matrix memory,” *IEEE Transactions on Computers*, **C-21**, pp. 353–359, 1972.
- [29] Kohonen, T., “Associative memory, content addressing, and associative recall,” in *Content Addressable Memories*, Springer, Berlin, 1979.
- [30] Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice Hall, Englewood Cliffs, NJ, 1992.
- [31] Lau, C., (Ed.), *Neural Networks, Theoretical Foundations, and Analysis*, IEEE Press, New York, NY, 1992.
- [32] Li, H., Chen, P. C. L., Huang, H. P., *Fuzzy Neural Intelligent Systems: Mathematical Foundation and the Applications in Engineering*, CRC Press, New York, NY, 2001.
- [33] Lippmann, R. P., “An introduction to computing with neural networks,” *IEEE ASSP Magazine*, pp. 4–22, USA, 1987.
- [34] Looney, C., *Pattern Recognition using Neural Networks: Theory and Algorithms for Engineers and Scientists*, Oxford University Press, London, 1997.
- [35] López, C. D., *Memorias Asociativas Reticulares con Capacidad Dendrítica para Restauración de Imágenes Digitales*, INAOE, tesis de maestría (Óptica), XM1216, 2007.
- [36] Lukac, R., Smolka, B., Martin, K., Plataniotis, K. and Venetsanopoulos, A., “Vector filtering for color imaging,” *IEEE Signal Processing Magazine*, No. 1, pp. 74–86, 2005.
- [37] Maren, A. J., Harston, C. T., Pap, R. M., *Handbook of Neural Computer Applications*, University of Salzburg, Academic Press Professional Inc., San Diego, 1990.
- [38] Martín del Brío, B., Sanz, M. A., *Redes Neuronales y Sistemas Difusos 2nd. ed.*, Alfaomega, México, 2002.
- [39] Mendel, M. J., *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, University of Southern California, Prentice Hall PTR, Los Angeles, CA, 2001.
- [40] Meilijson, I., Ruppin, E., Sipper, M., “Fast Computation in Hamming and Hopfield Networks,” in C. T. Leondes (Ed.) *Neural Network Systems Techniques and Applications*, **1**, San Diego, CA, Academic Press, pp. 123–153, 1998.
- [41] Mitra, S., Gupta, M. M., Kraske, W. F., *Neural And Fuzzy Systems: The emerging science of intelligent computing*, SPIE Optical Engineering Press, Washington, 1994.
- [42] Nguyen, H. T., Prasad, N. R., Walker, C. L., Walker, E. A., *A first Course in Fuzzy Logic and Neural Control*, Chapman and Hall/CRC, Boca Raton, Florida, 2000.

- [43] Nieves-V., J. A., Urcid, G., *Reconocimiento de Patrones e Imágenes Digitales usando Memorias Asociativas Morfológicas*. INAOE, tesis de maestría (Óptica), XM1096, 2005.
- [44] Oh, C., Žak S. H., “Image recall using a large scale generalized brain-state-in-a-box neural network,” *Int. J. Appl. Math. Comput. Sci.*, **15**(1), pp. 99–114, 2005.
- [45] Pal, S. K., Mitra, S., *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*, John Wiley and Sons, Calcuta, 1999.
- [46] Pandya, A. S., Macy, R. B., *Pattern Recognition with Neural Networks in C++*, IEEE Press, Boca Raton, Florida, 1996.
- [47] Plataniotis, K. N., Venetsanopoulos, A. N., *Color image processing and applications*, Springer, Heidelberg, 2000.
- [48] Pratt, W. K., *Digital Image Processing 2nd. ed.*, John Wiley & Sons, New York, NY, 1991.
- [49] Raducanu, B., Graña, M., “On the Application of Heteroassociative Morphological Memories to Face Localization,” *Bio-Inspired Applications of Connectionism, Proceedings Lecture Notes in Computer Science*, Springer Berlin, Heidelberg, **II**, pp. 563–570, 2001.
- [50] Ralston, A., Rely, E. D., Hemmendinger, D., *Encyclopedia of Computer Science 4th. ed.*, London, UK, 2000.
- [51] Ritter, G. X., *Image algebra with applications*, Unpublished manuscript, available via anonymous ftp at <ftp://cis.ufl.edu./pub/src/ia/documents>, 1994.
- [52] Ritter, G. X., Sussner, P., Díaz de León, J. L., “Morphological associative memories,” *IEEE Transaction on Neural Networks*, **9**(2), pp. 281–293, 1998.
- [53] Ritter, G. X., Wilson, J. N., *Handbook of Computer Vision Algorithms in Image Algebra 2nd ed.*, CRC Press, Boca Raton, Florida, 2000.
- [54] Ritter, G. X., Urcid, G., Iancu, L., “Reconstruction of patterns from noisy inputs using morphological associative memories,” *Journal of Mathematical Imaging and Vision*, **19**2, pp. 95–111, 2003.
- [55] Sussner, P., “Kernels for morphological associative memories,” in *Proceedings of the International ICSC/IFAC Symposium on Neural Computation*, Vienna, Austria, pp. 79–85, 1998.
- [56] Sussner, P., “Observations on morphological associative memories and the kernel method,” *Neuro-computing*, Vol. **31**, pp. 167–183, 2000.
- [57] Sussner, P., “A fuzzy autoassociative morphological memory,” *Proc. of the Int. Joint Conf. on Neural Networks*, Portland, OR, pp. 326–331, 2003.



- [58] Sussner, P., “Gray-scale morphological associative memories,” *IEEE Transactions on Neural Networks*, **17**(3), pp. 559–570, 2006.
- [59] Sussner, P., “Recall of patterns using morphological and certain fuzzy morphological associative memories with applications in classification and prediction,” *IEEE International Conference on Fuzzy Systems*, Vancouver, Canada, pp. 805–812, 2006.
- [60] Sussner, P., Estevão, E. L., “A brief introduction to a two-layer morphological associative memory based on fuzzy operations,” *Proceedings of the 8th International Symposium on Mathematical Morphology*, **2**, Río de Janeiro, Brazil, pp. 57–58, 2007.
- [61] Tsoukalas, L. H., Uhrig, R. E., *Fuzzy and Neural Approaches in Engineering*, John Wiley and Sons, New York, NY, 1997.
- [62] Urcid, G., Ritter, G. X., “Kernel computation in morphological associative memories for grayscale image recollection,” *Proceedings 5th IASTED International Conference on Signal and Image Processing*, Honolulu, Hawaii, pp. 450–455, 2003.
- [63] Urcid, G., Ritter, G. X., “Noise masking for pattern recall using a single lattice matrix auto-associative memory,” *IEEE Proceedings International Conference on Fuzzy Systems*, Vancouver, Canada, pp. 783–790, 2006.
- [64] Urcid, G., Nieves-V., J. A., Reyes-G., C. A., “Enhanced fuzzy autoassociative morphological memory for binary pattern recall,” *Proceedings 4th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, Innsbruck, Austria, pp. 116–121, 2007.
- [65] Urcid, G., Ritter, G. X., Nieves-V., J. A., “Redundant encoding of patterns in lattice associative memories,” *Proceedings 6th International Conference on Concept Lattices and Their Applications*, LBM, Olomouc, Czech Republic, pp. 45–57, 2008.
- [66] Urcid, G., Nieves-V., J. A., et al., “Robust image retrieval from noisy inputs using lattice associative memories,” *Proceeding of SPIE, Image Processing: Algorithms and Systems VII*, **7245**, San Jose, USA, pp. 724517:1–12, 2009.
- [67] Vazquez, R., Sossa, H., “A bidirectional hetero-associative memory for true-color patterns,” *Neural Processing Letters*, Springer, **28**, pp. 131–153, 2008.
- [68] Villaverde, I., Graña, M., d’Ánjou, A., “Morphological neural networks and vision based simultaneous localization and mapping,” *Integrated Computer-Aided Engineering*, **14**(4), pp. 355–363, 2007.
- [69] Wang, S., Lu, H., “On new fuzzy morphological associative memories,” *IEEE Transaction on Fuzzy Systems*, **12**(3), pp. 316–323, 2004.

- [70] Yáñez, M. C., *Memorias Asociativas Basadas en Relaciones de Orden y Operadores Binarios*, Tesis doctoral, CIC-IPN, México, 2002.
- [71] Yáñez, M. C., Cruz, M. M., Sánchez, G. F., López, Y. I., “Using alpha-beta associative memories to learn and recall RGB images,” *Lecture Notes in Computer Science*, **4493**, Springer, pp. 828–833, 2007.
- [72] Yen, J., Langari, R., *Fuzzy Logic: Intelligence, Control and Information*, Pearson Education, Texas, 2005.