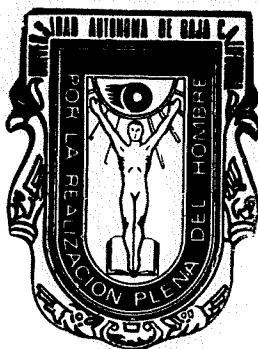


UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE CIENCIAS



**Diseño, implantación y prueba de una red neuronal difusa para
la clasificación de imágenes multiespectrales.**

T E S I S

QUE PARA OBTENER EL TITULO DE LIC. EN CIENCIAS COMPUTACIONALES

PRESENTA:

MARIA VICTORIA MEZA KUBO

ENSENADA, BAJA CALIFORNIA, SEPTIEMBRE DE 1995.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE CIENCIAS

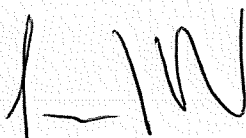
**DISEÑO, IMPLANTACIÓN Y PRUEBA DE UNA RED
NEURONAL DIFUSA PARA LA CLASIFICACIÓN DE
IMÁGENES MULTIESPECTRALES**

T E S I S P R O F E S I O N A L

QUE PRESENTA

MARÍA VICTORIA MEZA KUBO

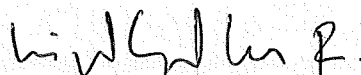
APROBADO POR:



DR. JESÚS FAVELA VARA
Presidente del jurado



**M. C. JORGE TORRES
RODRÍGUEZ**
Secretario



**M. C. MIGUEL ÁNGEL IBARRA
RIVERA**
1er. vocal

Agradecimientos

Quiero dar gracias primeramente a Dios, por la vida y permitirme llegar al término de mi carrera profesional.

Gracias a mis padres, quienes siempre me han apoyado e impulsado a seguir adelante.

Gracias a mi alma mater, donde maestros y compañeros me compartieron sus experiencias y brindaron su amistad.

Gracias de manera especial, a mi asesor de tesis, Dr. Jesús Favela Vara, por su guía y ayuda para la realización de este trabajo.

Gracias también al M. C. Jorge Torres por su ayuda en el procesamiento de imágenes.

Muchas gracias a mi compañera Diana Ruiz, por su amistad y colaboración para el desarrollo del sistema.

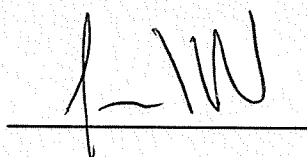
Y por último, gracias a Paco por su gran amor y comprensión.

Vicky

RESUMEN de la tesis de *María Victoria Meza Kubo* presentada como requisito parcial para la obtención de la *Licenciatura en Ciencias Computacionales*. Ensenada, Baja California, México. Septiembre de 1995.

**Diseño, implantación y prueba de una red neuronal difusa para la
clasificación de imágenes multiespectrales**

Resumen aprobado por:



Dr. Jesús Favela Vara.

Un área de especial interés en el procesamiento digital de imágenes es la clasificación de imágenes multiespectrales de satélite. Se han desarrollado una gran variedad de métodos para clasificar estas imágenes basados en técnicas estadísticas. En este trabajo de tesis se presenta una nueva técnica de clasificación para determinar tipos de terrenos en imágenes multiespectrales utilizando redes neuronales. Se realizó un estudio del comportamiento de los parámetros que definen la red neuronal Fuzzy ART y se presentó un sistema que integra técnicas de procesamiento digital para la clasificación y visualización de las imágenes. Las pruebas de clasificación se efectuaron sobre imágenes de satélite de la región deltaica del Río Colorado, ya que en esta zona se han realizado estudios utilizando técnicas tradicionales de clasificación, lo cual nos ha permitido evaluar las técnicas propuestas. Hemos concluido que las redes neuronales representan una buena alternativa de clasificación ya que los resultados obtenidos son muy semejantes a los obtenidos con técnicas tradicionales y muestran ciertas ventajas sobre éstos.

Contenido

	Página
1. Introducción.....	1
1.1 Problemática.....	1
1.2 Objetivo.....	2
1.3 Metodología.....	2
1.4 Organización del trabajo.....	3
2. Redes Neuronales Artificiales.....	4
2.1 De redes neuronales biológicas a redes neuronales artificiales.....	4
2.2 Funciones de una neurona.....	5
2.3 Características de las redes neuronales artificiales.....	6
2.4 Elementos de una red neuronal artificial.....	8
2.4.1 Topología o arquitectura de una red neuronal artificial.....	8
2.4.2 Funciones de transferencia.....	10
2.4.3 Aprendizaje.....	12
2.4.3.1 Modos de aprendizaje.....	13
2.4.3.2 Tasas de aprendizaje.....	14
2.5 Algunas arquitecturas de redes neuronales.....	14
2.5.1 Red de Hebb.....	15
2.5.2 Perceptrón.....	15
2.5.3 ADALINE/MADALINE.....	16
2.5.4 Retropropagación.....	16

Contenido (*continuación*)

	Página
2.5.5 Memoria Asociativa Bidireccional (BAM).....	17
2.5.6 Máquina de Boltzman.....	17
2.5.7 Contrapropagación.....	18
2.5.8 Mapas de auto-organización.....	18
2.5.9 Teoría de Resonancia Adaptativa (ART).....	19
2.5.10 Neocognitron.....	20
 3. Teoría de Resonancia Adaptativa (ART).....	 21
3.1 Dilema de la estabilidad-plasticidad.....	21
3.2 Descripción de la red ART.....	23
3.3 ART1	28
3.3.1 Funcionamiento de ART1	28
3.4 ART2	31
3.4.1 Arquitectura de ART2.....	31
3.4.2 Funcionamiento de ART2	33
3.4.3 Trazos LTM	36
3.4.4 Sistema de desactivación.....	37
3.5 Lógica difusa	38
3.6 FuzzyART	39
3.6.1 Descripción de FuzzyART	39
3.6.2 Operaciones opcionales	40

Contenido (*continuación*)

	Página
4. Clasificación de imágenes de satélite	43
4.1 Métodos tradicionales	43
4.1.1 Técnicas de clasificación supervisadas	44
4.1.2 Técnicas de clasificación no supervisadas	45
4.2 Redes neuronales para la clasificación de imágenes	46
4.2.1 Red neuronal de retropropagación	48
4.2.2 Red neuronal FuzzyART	52
5. Sistema para la clasificación de imágenes multiespectrales	54
5.1 El sistema XWindows	54
5.2 NeuroClas	55
5.2.1 Preprocesamiento	57
5.2.2 Clasificación	61
5.2.3 Postprocesamiento	61
5.3 Descripción de NeuroClas	61
6. Resultados y discusiones	63
6.1 Características de la imagen clasificada	63
6.2 Pruebas de clasificación	64
6.2.1 Opción de normalización de la entrada	65
6.2.2 Opción de compromiso rápido, recodificación lenta	67

Contenido (*continuación*)

	Página
6.3 Resultados	67
7. Conclusiones	72
7.1 Conclusiones	72
7.2 Aportaciones	74
7.3 Recomendaciones	74
8. Referencias citadas	76
Apéndice A. Descripción del sistema NeuroClas	81
Apéndice B. Funcionamiento de NeuroClas	87

Lista de figuras

	Página
2.1 Estructura de una neurona biológica.....	5
2.2 Estructura de una neurona artificial.....	6
2.3 Estructura de una red neuronal con una capa oculta.....	10
2.4 Función sigmoideal logística.....	13
3.1 Arquitectura de un sistema ART.....	25
3.2 Ciclo de comparación de patrones en una red ART	29
3.3 Arquitectura de ART2	33
4.1 Proceso de clasificación con redes neuronales	48
5.1 Declaración de la clase y estructura ART	57
5.2 Módulos principales de NeuroClas	58
5.3 Gráfica de una transformación lineal	60
5.4 Gráfica de la función de transformación exponencial	60
5.5 Gráfica de la función de transformación logarítmica	61
5.6 Ejemplo de una transformación lineal	61
5.7 Ventana principal de NeuroClas	63
6.1 Banda espectral utilizada como entrada a la red	65
6.2 Resultado de una prueba de clasificación con FuzzyART	72
7.1 Prueba de clasificación con métodos tradicionales	74
B-1 Obtención de los datos necesarios para el despliegue de una imagen.....	89
B-2 Manejo de histograma lineal	91
B-3 Características de los puntos de la imagen	93

Lista de figuras (continuación)

	Página
B-4 Diálogo para introducir los parámetros de la red de Retropropagación	95
B-5 Diálogo para pedir puntos de entrenamiento	97
B-6 Obtención del nombre para salvar la red activa	98
B-7 Diálogo para obtención de los parámetros de la red neuronal FuzzyART.....	100

Lista de tablas

	Página
Tabla I. Factores en la ecuación de activación para cada una de las subcapas.....	35
Tabla II. Analogía entre las operaciones de ART1 y Fuzzy ART.....	41
Tabla III. Resultados obtenidos variando el factor de vigilancia	69
Tabla IV. Variación de la tasa de aprendizaje	69
Tabla V. Variación de $\Delta\beta$	70
Tabla VI. Variación del parámetro de elección	70
Tabla VII. Resultados obtenidos con la normalización de la entrada II.....	71

1 Introducción.

1.1 Problemática.

La nueva tecnología satelital está generando grandes cantidades de datos en imágenes digitales diariamente. Esta información es utilizada en diversos campos de aplicación, tales como planeación urbana, exploración geológica, evaluación de cosechas y estudios de impacto ambiental.

La dificultad para el manejo de imágenes digitales de alta resolución ha provocado un incremento en la demanda de algoritmos de clasificación más eficientes para un aprovechamiento óptimo de esta información.

Se han desarrollado una gran variedad de métodos para clasificar este tipo de imágenes. Estas técnicas se dividen en métodos supervisados y no supervisados. No obstante que los primeros tienen más precisión, éstos requieren de la intervención del usuario para determinar las clases a encontrar así como la elección de las regiones que se utilizarán para su entrenamiento. En contraste, los métodos no supervisados determinan las clases automáticamente, aunque muestran una capacidad limitada al no poder establecer una relación directa entre clases espectrales y clases informativas.

Las redes neuronales representan métodos alternativos a los clasificadores probabilísticos tradicionales, ya que éstas al no basarse en un modelo probabilístico, pueden ser utilizadas en un contexto más universal. Las redes neuronales que se han

utilizado para clasificar imágenes multiespectrales son en su mayoría del tipo supervisado [Bendiktsson et al, 1990; Heermann et al, 1992; Ruiz 1995] y han tenido un gran éxito. Se han desarrollado muy pocas del modo no supervisado como es el caso de un modelo basado en la Teoría de Resonancia Adaptativa (ART) [Hara et al, 1994], sin embargo los resultados que se obtuvieron de éste fueron de poca calidad.

1.2 Objetivo.

En este trabajo de tesis se presenta una arquitectura neuronal difusa no supervisada para la clasificación de imágenes multiespectrales. Esta red es llamada FuzzyART ya que se basa en la Teoría de Resonancia Adaptativa (ART) la cual resuelve el problema de la estabilidad-plasticidad [Carpenter et al, 1991b] que afecta a otro tipo de redes neuronales (ver sección 3.1).

1.3 Metodología.

La motivación de implementar una red neuronal basada en la Teoría de Resonancia Adaptativa (ART) fue un trabajo de clasificación de imágenes que se basa en esta teoría [Hara et al, 1994] del cual se obtuvieron muy malos resultados

Se investigaron los modelos desarrollados con ART que pudieran utilizar vectores continuos como entrada. De aquí se eligió la red FuzzyART ya que es sólo una pequeña modificación del modelo de ART1, la cual es una red muy sencilla pero que sólo acepta vectores binarios como entrada. FuzzyART agrega operaciones de lógica difusa para el

reconocimiento de vectores continuos además de binarios como entrada.

1.4 Organización del trabajo.

En el siguiente capítulo se presenta una breve introducción a las redes neuronales, mostrando sus características más importantes. En el capítulo tres se verán con detalle algunas de las redes de la Teoría de Resonancia Adaptativa (ART), en especial las redes ART1 y FuzzyART. En el capítulo cuatro se dará una explicación de las técnicas de clasificación tradicionales, además de los modos supervisados y no supervisados. También se explica el proceso de clasificación usando redes neuronales y se presentan las redes de Retropropagación y Fuzzy ART como prototipos de redes del modo supervisado y no supervisado, respectivamente. En el capítulo cinco se presentará el sistema *NeuroClas* el cual reúne técnicas de procesamiento digital para el preprocesamiento, clasificación y postprocesamiento de imágenes. Este sistema contiene técnicas de clasificación con redes neuronales del modo supervisado y no supervisado. Por último se presentan los resultados obtenidos de las pruebas de clasificación, las conclusiones, las aportaciones más importantes y algunas recomendaciones para trabajo futuro.

2. Redes Neuronales Artificiales.

2.1 De redes neuronales biológicas a redes neuronales artificiales.

Algunas de las características de las neuronas biológicas pueden ayudarnos a clarificar las características más importantes de las redes neuronales artificiales.

Las redes artificiales fueron inspiradas por el sistema biológico neuronal [Nelson et al, 1991], existe una gran analogía entre la estructura de una neurona biológica y un elemento de procesamiento de una red neuronal artificial.

La Figura 2.1 muestra una neurona biológica. Ésta tiene cuatro tipos de componentes que son de especial interés para el entendimiento de una neurona artificial: *las dendritas*, *las sinapsis*, *el núcleo* y *el axón*. Las **dendritas** reciben señales de otras neuronas a partir de impulsos eléctricos que son transmitidos a través de las **sinapsis**.

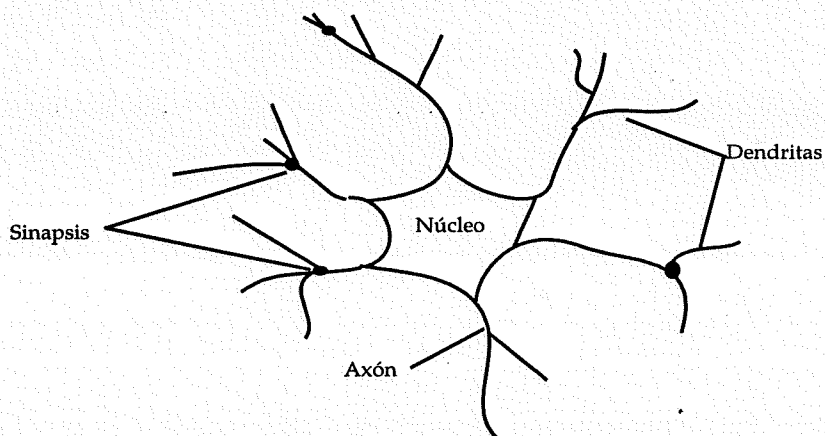


Figura 2.1 Estructura de una neurona biológica.

El núcleo suma todas las señales recibidas. Cuando la entrada recibida es suficiente, entonces la neurona transmite una señal a través del axón a otras neuronas.

De este modo podemos visualizar una neurona artificial o elemento de procesamiento como en la figura 2.2.

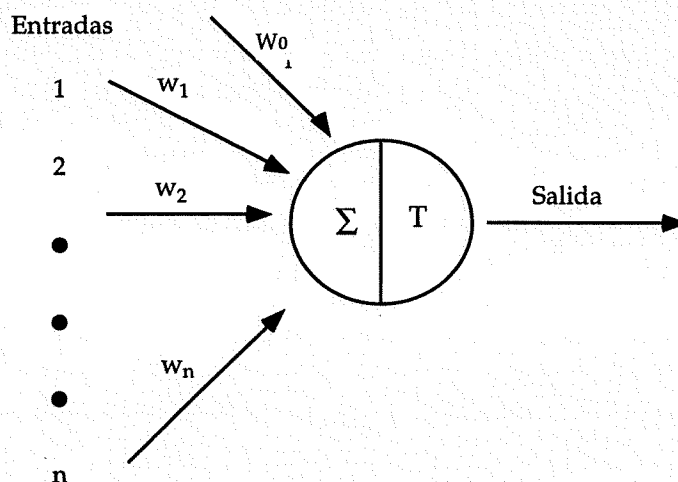


Figura 2.2 Estructura de una neurona artificial.

2.2. Funciones de una neurona.

Una neurona artificial también es conocida como nodo o elemento de procesamiento. En la figura 2.2 una neurona es representada por un círculo. Las funciones básicas que manejan los elementos de procesamiento son : evaluar las señales de entrada, determinando la fuerza (pesos w_i) de cada una. Calcular un total combinando todas las señales de entrada y compararlo con algún nivel de umbral. Y finalmente determinar qué valor de salida se dará.

Algunas redes agregan otro componente (w_0) llamado sesgo (bias). El sesgo actúa

exactamente igual que un peso cuya activación siempre será 1.

2.3 Características de las redes neuronales artificiales.

Varios atributos de los elementos de procesamiento de una red neuronal artificial son sugeridos por las propiedades de las neuronas biológicas, como por ejemplo:

1. Los elementos de procesamiento reciben muchas señales.
2. Las señales pueden ser modificadas por un peso en la sinapsis recibida.
3. Los elementos de procesamiento suman los pesos de entrada.
4. Una entrada suficientemente grande produce que la neurona se active y transmita una salida.
5. La salida de una neurona puede ir a muchas otras neuronas.
6. Las transmisiones de las sinapsis pueden ser excitatorias (positivas) o inhibitorias (negativas).
7. La memoria es distribuida:
 - a. Memorias de corto plazo corresponden a las señales enviadas por las neuronas.
 - b. Memorias de largo plazo residen en las sinapsis de las neuronas o pesos.
8. La fuerza de las sinapsis pueden ser modificadas con la experiencia. Esta capacidad de adaptación o aprendizaje, constituye el atributo más importante de una red neuronal.

Otra característica importante que una red neuronal artificial comparte con el sistema biológico neuronal es que es **tolerante a fallas**. El sistema biológico neuronal es

tolerante a fallas en dos aspectos. Primero, es capaz de reconocer muchas señales de entrada que son diferentes a señales recibidas antes. Por ejemplo la capacidad de reconocer a una persona en una fotografía, o después de no haberla visto en un largo período de tiempo.

El segundo aspecto en el cual una red es tolerante a fallas es la capacidad de tolerar daños al sistema neuronal. El cerebro continuamente está perdiendo neuronas y continúa aprendiendo sin ningún problema. Algunas veces otras neuronas pueden ser entrenadas para tomar la función de neuronas dañadas. En una forma similar, las redes neuronales artificiales pueden soportar pequeños daños en la red, y pueden ser reentrenadas en caso de un daño mayor.

Las redes neuronales tienen la capacidad de recibir patrones parciales como entrada y suponer el resto, a esta característica se le llama **intuición**. Esta capacidad es combinada con una habilidad para reconstruir patrones estadísticamente, donde el patrón calculado es lo más cercano al patrón parcial dado como entrada.

Las redes neuronales tienen la habilidad de auto-ajustarse, a esta característica se le conoce como **adaptabilidad**. Existen cuatro aspectos en el proceso de la adaptabilidad:

- ***El aprendizaje.***

El aprendizaje se da en una red al ajustarse los pesos de las conexiones.

- ***La auto-organización.***

La auto-organización es la modificación de muchos elementos de procesamiento a la vez en respuesta a vectores dados como entrada sin algún orden específico.

- La generalización.

La generalización es la habilidad de la red para responder a entradas que no han aparecido antes. La entrada puede ser parcial o incompleta y mediante la intuición e igualación de patrones estadísticos el sistema será capaz de generar una respuesta.

- El entrenamiento.

El entrenamiento es el proceso que sigue la red para aprender.

2.4 Elementos de una red neuronal.

Las redes neuronales están especificadas por:

- La topología o arquitectura de la red.
- Las posibles funciones de transferencia.
- Los métodos de entrenamiento o reglas de aprendizaje.

2.4.1 Topología o arquitectura de una red neuronal artificial.

En la figura 2.3 se presenta la arquitectura de una red neuronal artificial sencilla. Las neuronas son combinadas con otras para formar una capa. La capa inferior o capa de entrada es la capa que recibe los vectores de entrada, está conectada con una capa central o

capa oculta a través de pesos W_{ij} , y ésta a su vez se encuentra conectada con la capa superior o capa de salida por medio de pesos W_{jk} , la cual genera las salidas de la red.

Se dice que la red está **completamente conectada** ya que cada salida de una capa alimenta a cada una de las neuronas de la siguiente capa. Cuando la salida de una neurona pasa a una neurona de la misma capa o a la capa que precede, se dice que la red es de **alimentación hacia adelante**. Cuando las salidas pueden ser dirigidas de vuelta hacia la capa anterior, la red es una red de **retroalimentación**.

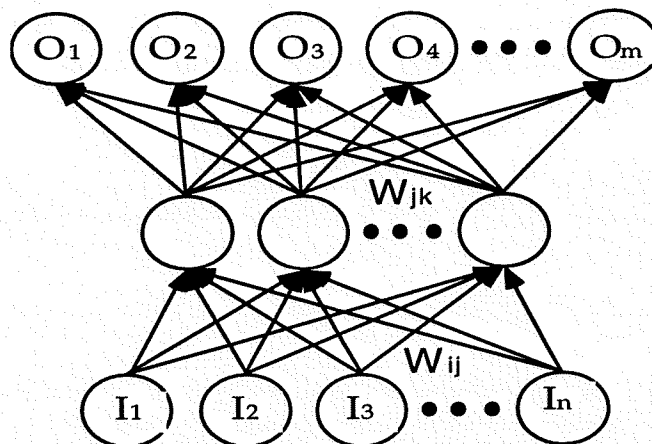


Figura 2.3 Estructura de una red neuronal con una capa oculta.

El número de elementos de procesamiento necesarios para las capas de entrada y salida dependerán del número de entradas y salidas del sistema.

Las redes de **capa sencilla** tienen dos capas que están conectadas por pesos. Por lo general la primera capa son las unidades de entrada, las cuales reciben señales del mundo externo, y la segunda capa son las unidades de salida, de las cuales se obtiene la respuesta

de la red. Las unidades de entrada son completamente conectadas a las unidades de salida.

Para la clasificación de patrones, cada unidad de salida corresponde a una categoría particular a la cual el patrón de entrada puede o no pertenecer.

Las **redes multicapas** son redes con una o más capas (llamadas ocultas) entre las unidades de entrada y las unidades de salida. Las redes multicapa pueden resolver problemas más complicados que las redes de capa sencilla, pero el entrenamiento puede ser más difícil.

2.4.2 Función de transferencia.

Cada una de las entradas tiene un peso relativo, el cual afecta la fuerza de esta entrada. Algunas entradas son más importantes que otras en la forma en que se combinan para producir un impulso. Los pesos son coeficientes adaptativos en la red que determinan la intensidad de la señal de entrada.

Matemáticamente se pueden considerar las entradas y los pesos como vectores, tales como $(I_1, I_2, \dots, I_n)$ y $(W_1, W_2, \dots, W_n)$, respectivamente, para el caso de una unidad alimentada por n neuronas. La señal total recibida es el producto punto de los dos vectores.

La neurona es responsable de realizar la suma (Σ) de todas las entradas y comparar este resultado con un umbral para generar una señal de respuesta.

El resultado de la suma es llamado función de activación y es pasado a través de una función de transferencia, representado por T en la figura 2.2. Típicamente, la misma función de transferencia es usada en todas las neuronas de cada capa de la red.

Para redes de capas sencillas se usa una función para convertir la entrada neta a una unidad de salida que recibe señales binarias (1 o 0) o bipolares (1 o -1). Esta función es conocida como *función de umbral*. Para los casos binarios y bipolares, se tiene respectivamente:

$$f(x) = \begin{cases} 1 & \text{si } x \geq \theta \\ 0 & \text{si } x < \theta \end{cases} \quad f(x) = \begin{cases} 1 & \text{si } x \geq \theta \\ -1 & \text{si } x < \theta \end{cases}$$

donde θ representa el valor de umbral.

Las *funciones sigmoidales* (curvas en forma de S) son funciones de transferencia muy utilizadas ya que estas varían sus salidas en un intervalo conocido. Las funciones más comunes son las funciones logísticas.

La *función logística* es una función sigmoidal con rango de 0 a 1. Es usada como función de transferencia para redes neuronales en la cuales se desea que los valores de salida sean binarios o estén en el intervalo entre 0 y 1 (Figura 2.4). Su función matemática es la siguiente:

$$f(x) = \frac{1}{1 + e^{-x}}$$

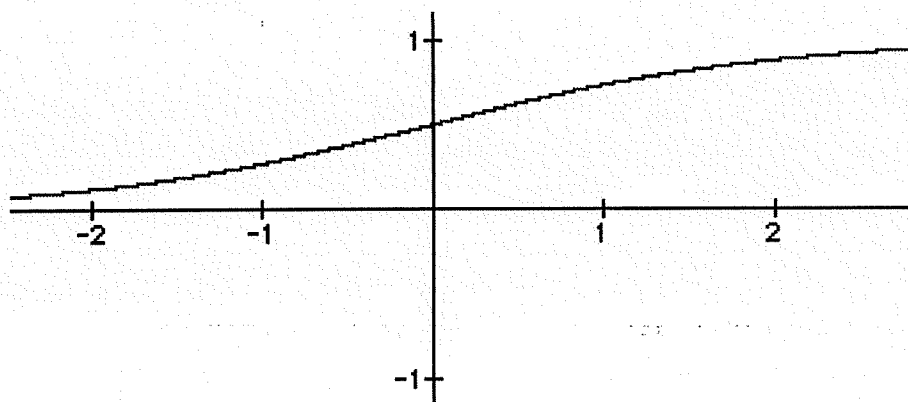


Figura 2.4. Función sigmoideal logística, sus valores varían en el intervalo de (0,1).

2.4.3 Aprendizaje.

El aprendizaje puede ser definido como *“la modificación del comportamiento por la interacción con el ambiente y como un resultado de experiencias que llevan al establecimiento de nuevos patrones de respuesta a estímulos externos”* [Nelson et al, 1991].

En una red neuronal los pesos son variables. Estos pueden ser dinámicamente ajustados para producir la salida deseada. Esta modificación dinámica de los pesos es la esencia del aprendizaje.

2.4.3.1 Modos de aprendizaje.

Existen dos tipos de aprendizaje: supervisado y no supervisado. El aprendizaje supervisado requiere de un maestro. Este maestro puede ser un conjunto de datos de entrenamiento que le indican a la red cual es la salida esperada para cada patrón de entrenamiento. Cuando no existe un maestro externo, el sistema deberá auto-organizarse por criterios internos del diseño de la red.

Aprendizaje supervisado.- En este modo de aprendizaje, la salida real es comparada con la salida deseada. Los pesos son generalmente inicializados aleatoriamente, y son entonces ajustados por la red para que a la siguiente iteración o ciclo, produzca una salida más cercana a la deseada. La meta de todos los procedimientos de aprendizaje es minimizar el error entre la salida deseada y la salida real mediante la modificación continua de los pesos.

Con el aprendizaje supervisado, es necesario “entrenar” a la red antes de que empiece su operación. El entrenamiento consiste en presentar datos de entrada y salida a la red. Estos datos son llamados *conjunto de entrenamiento*. La fase del entrenamiento puede consumir una cantidad considerable de tiempo. El entrenamiento se considera completo cuando la red produce las salidas esperadas para una secuencia de entradas dadas.

Aprendizaje no supervisado.- El aprendizaje no supervisado es también llamado aprendizaje autosupervisado. Las redes neuronales con aprendizaje no supervisado se

auto-organizan para agrupar vectores de entrada en grupos similares sin el uso de datos de entrenamiento que especifiquen los miembros típicos de cada grupo [Fausett, 1994]. Se da como entrada una secuencia de vectores pero no se especifican las salidas deseadas. La red modifica los pesos para que la mayoría de los vectores de entrada que son similares sean asignados a la misma unidad de salida ("cluster" o grupo).

2.4.3.2 Tasas de aprendizaje.

La tasa de aprendizaje determina qué tan rápido aprende la red, es decir, qué tanto cambiarán los pesos al presentarse un patrón de entrada y ajustarse para obtener la salida deseada. La tasa de aprendizaje puede ser variable; una tasa de aprendizaje baja, significa que la red tomará más tiempo para completar el entrenamiento. Con una tasa de aprendizaje alta, el entrenamiento tomará menos tiempo, sin embargo, puede ser que la red no sea capaz de hacer discriminaciones tan finas de los datos, como lo haría una red que aprende más lentamente.

2.5 Algunas arquitecturas de redes neuronales.

En esta sección se presentan algunas de las arquitecturas de redes neuronales más comunes y sus principales características y aplicaciones. Las redes de Retropropagación y Teoría de resonancia adaptativa se estudiarán con más detalle en las secciones 4.2.1 y 3 respectivamente.

2.5.1 Red de Hebb.

Desarrollada por Donald Hebb. El aprendizaje de Hebb fue la primer regla de aprendizaje desarrollada para las redes neuronales artificiales [Hebb, 1949]. Hebb propone que el aprendizaje ocurre por la modificación de la fuerza de la sinapsis (pesos) de forma tal que si dos neuronas interconectadas se activan al mismo tiempo, entonces el peso entre estas neuronas se incrementa.

La red de Hebb tiene una arquitectura de capa sencilla con alimentación hacia adelante y es del modo supervisado. Es aplicable en la solución de algunas funciones lógicas y reconocimiento de caracteres.

2.5.2 Perceptrón.

Fue construida por Frank Rosenblatt [Rosenblatt, 1958]. Es un modelo computacional que simula la operación de la retina, con un aprendizaje asociador que presenta capacidad para el reconocimiento de patrones del sistema visual.

Es una red de tres capas; la capa de entrada funciona como una retina, son las unidades sensoriales que envían los datos a la siguiente capa. La capa intermedia funciona como unidades asociadoras y están conectadas a la capa de salida que funciona como unidad de respuesta.

Se aplica en la clasificación de formas, reconocimiento de caracteres y sistemas de visión en robótica.

2.5.3 ADALINE/MADALINE.

ADALINE proviene de ADaptive Linear Element, y MADALINE es para Múltiples ADALINES en paralelo.

Estos modelos fueron contruidos por Bernald Widrow y Marcian Hoff [Widrow et al, 1960].

Son redes de tres capas con funciones de transferencia de umbrales. Estas redes han sido aplicadas a una gran variedad de problemas como: prueba clásica en control adaptativo, arreglos de ruido adaptativo en telecomunicaciones y reconocimiento de patrones.

2.5.4 Retropropagación.

Fue desarrollada inicialmente por Paul Werbos [Werbos, 1974]. Posteriormente fue modificada simultáneamente por David Parker, Davis Rumelhard, Hinton y Williams [Parker, 1985; Rumelhart et al, 1986].

Actualmente la red de retropropagación es la más popular. Es una red multicapa con alimentación hacia adelante y es del modo supervisado. Su principal limitación es el

proceso de aprendizaje, ya que este requiere de un considerable procesamiento.

Esta red calcula la diferencia entre la salida real y la salida deseada. Usando este error, se modifican los pesos. La secuencia de operación incluye dos pasos: un paso hacia adelante para calcular el error y entonces un paso de regreso para modificar los pesos, tal que el error sea decrementado.

Puede ser aplicada para compresión de datos, reconocimiento de patrones, control de brazos de robot y representación del conocimiento.

2.5.5 Memoria Asociativa Bidireccional (BAM).

Desarrollada por Bart Kosko [Kosko, 1988] la BAM almacena un conjunto de asociación de patrones por la suma de matrices de correlación bipolar.

Su arquitectura consiste en dos capas de neuronas conectadas por trayectorias de conexión de pesos direccionales. Las iteraciones de la red envían señales de regreso y hacia adelante entre las dos capas hasta que todas las neuronas estén en equilibrio.

2.5.6 Máquina de Boltzman.

Las redes neuronales de la máquina de Boltzman fueron introducidas por Jeffrey Hinton y Terry Sejnowsky [Hinton et al, 1983]. Los estados de las unidades son valores binarios, con transiciones de estado probabilísticos.

La arquitectura de una máquina de Boltzman consiste de un conjunto de unidades y un conjunto de conexiones bidireccionales entre pares de unidades. Se aplica en reconocimiento de patrones.

2.5.7 Contrapropagación.

Las redes de Contrapropagación fueron desarrolladas por Robert Hecht-Nielsen [Hetch-Nielsen, 1987]. Son redes multicapas basadas en una combinación de capas de entradas, de agrupamiento y de salida.

Una red de Contrapropagación aproxima sus pares de vectores de entrenamiento por la construcción adaptativa de tablas de asignación look-up.

Las redes de Contrapropagación pueden ser usadas para la compresión de datos, para aproximar funciones o asociar patrones.

2.5.8 Mapas de auto-organización.

Desarrollada por Teuvo Kohonen [Kohonen, 1989]. Estas redes están construidas para “aprender” sin que se de la respuesta correcta para el patrón de entrada.

Esta red auto-asociativa es de capa sencilla, recurrente y altamente conectada. Los pesos deberán ser inicializados y los pesos y las entradas deberán ser normalizados o

ajustados a alguna referencia estándar. Los elementos de procesamiento compiten para tener el privilegio del aprendizaje. En una regla de aprendizaje “el ganador toma todo”, el nodo con la respuesta más alta y sus vecinos más cercanos ajustarán sus pesos. Conforme pasa el tiempo el tamaño del vecindario puede ser reducido. Los vecinos dan respuestas similares y se inicia una organización global.

Puede ser aplicada para el reconocimiento de voz, para generar música por computadora y en problemas de optimización.

2.5.9 Teoría de Resonancia Adaptativa (ART).

La Teoría de Resonancia Adaptativa fue desarrollada por Gail Carpenter y Stephen Grossberg . Entre las arquitecturas de redes que se desarrollaron con esta teoría se tiene ART1 [Carpenter et al, 1987a], diseñada para la clasificación de vectores binarios, otra es ART2 [Carpenter et al, 1987b], que acepta vectores de valores continuos; ART3, la cual puede portar búsqueda paralela o prueba de hipótesis en niveles de red jerárquicos [Carpenter et al, 1990]; recientemente el modelo de ART1 fue utilizado para diseñar una arquitectura de red jerárquica llamada ARTMAP que puede lograr rápidamente una auto-organización [Carpenter et al, 1991a]; y finalmente la red FuzzyART [Carpenter et al, 1991b] la cual es una modificación de ART1 que integra operaciones de lógica difusa para manejar vectores continuos como entrada.

Estas redes clasifican las entradas usando aprendizaje no supervisado. Los patrones de entrada pueden ser presentados en cualquier orden. Cada vez que un patrón es presentado,

se elige una unidad que lo clasifique y los pesos de esa unidad son ajustados para que la unidad de clasificación aprenda de ese patrón.

Se utiliza para el reconocimiento de patrones.

2.5.10 Neocognitrón.

Desarrollada por Fukushima, Miyake e Ito [Fukushima et al, 1983]. Es un ejemplo de una red jerárquica en la cual existen muchas capas.

Las redes Neocognitrón son entrenadas usando aprendizaje supervisado. El Neocognitrón fue diseñado para reconocer caracteres escritos a mano. El propósito de la red es hacer que su respuesta sea independiente a las variaciones de la posición y estilo en el cual el caracter esté escrito.

3. Teoría de Resonancia Adaptativa (ART)

3.1 Dilema de la estabilidad-plasticidad.

Una de las características más importantes de la memoria humana es la habilidad para aprender conceptos nuevos sin necesariamente olvidar los aprendidos en el pasado.

La mayoría de las redes neuronales que se han desarrollado tienden a olvidar la información pasada si tratan de agregar nueva información a su base de conocimientos [Freeman, 1994].

En particular, para la clasificación de patrones, típicamente se procede a entrenar el sistema con un conjunto de patrones ejemplares. Durante el entrenamiento, la información es codificada en el sistema mediante el ajuste de los valores de sus pesos. Una vez que la red se entrenó adecuadamente, el sistema está listo para ser puesto en producción y no se permiten modificaciones adicionales a los pesos.

Este escenario de operación es aceptable para problemas cuyo dominio tiene definido sus límites y es estable. Bajo tales circunstancias es posible definir un conjunto adecuado de entradas de entrenamiento para que el problema sea resuelto. Desafortunadamente en muchas situaciones reales, el ambiente no está limitado y no es estable.

Considere un ejemplo sencillo. Suponga que se entrena una red para reconocer siluetas

de ciertas clases de aviones. Las imágenes apropiadas pueden ser coleccionadas y usadas para entrenar la red, donde el tiempo que consume esta tarea es tan grande como el tamaño de la red requerida. Después de que la red ha aprendido es capaz de reconocer todos los aviones, el período de entrenamiento ha terminado y ya no se permiten modificaciones a los pesos. Si en el futuro, otro avión de la misma clase está operando, se puede desear agregar su silueta en el conocimiento de la red. Para hacer esto, se deberá reentrenar la red con el nuevo patrón y los patrones previos, ya que si entrenamos únicamente la nueva silueta, la red aprenderá este nuevo patrón, pero posiblemente olvidará los patrones previamente aprendidos.

Este problema es llamado **Dilema de estabilidad-plasticidad** [Carpenter et al, 1987a]. Este dilema es resuelto por todos los sistemas inteligentes capaces de adaptarse autónomamente en tiempo real a cambios inesperados en su mundo.

El dilema de la estabilidad-plasticidad puede ser planteado como una serie de preguntas [Carpenter et al, 1988]:

- ¿Cómo puede un sistema de aprendizaje permanecer adaptativo (plástico) en respuesta a entradas significantes, permaneciendo estable a entradas irrelevantes?
- ¿Cómo realiza los cambios entre modos de estabilidad y plasticidad?
- ¿Cómo puede retener la información aprendida mientras continúa aprendiendo nuevas cosas?

En respuesta a estas preguntas, Stephen Grossberg y Gail Carpenter desarrollaron la

Teoría de Resonancia Adaptativa (ART), que sugiere una solución a este problema. ART es una extensión de los esquemas de aprendizaje competitivos. En los sistemas competitivos los nodos compiten entre ellos, basados en algún criterio, y se dice que el ganador es el nodo que clasifica el patrón de entrada. Pueden aparecer inestabilidades en estas redes en forma tal que diferentes nodos puedan responder al mismo patrón de entrada en diferentes ocasiones. Más aún, posteriores aprendizajes pueden modificar lo aprendido si el ambiente no es estadísticamente estacionario o si se presentan nuevas entradas.

Una clave para resolver el dilema de la estabilidad-plasticidad es agregar un mecanismo de retroalimentación entre la capa competitiva y la capa de entrada de una red. Este mecanismo de retroalimentación facilita el aprendizaje de nueva información sin destruir la información pasada, automáticamente intercambia los modos de estabilidad y plasticidad y la estabilización de la codificación de las clases es completada por los nodos. Como resultado de esta aproximación surge la Teoría de Resonancia Adaptativa.

3.2 Descripción de la red ART.

La estructura básica de **Resonancia Adaptativa** es presentada como dos capas en la figura 3.1 [Grossberg, 1982]. Las flechas gruesas indican que las dos capas están completamente interconectadas con pesos modificables.

aplicación sencilla de un vector de entrada. Los pesos asociados con las conexiones *bottom-up* y *top-down* entre F_1 y F_2 son trazos de memoria de largo plazo (LTM por sus siglas en inglés) ya que codifican la información que permanece en la red por un período de tiempo extendido.

Son agregados a la red dos elementos más, la unidad de desactivación, A , y el subsistema de control de ganancia, G .

Existen tres posibles fuentes de entrada a F_1 , el vector de entrada I , una señal excitatoria de G y el patrón de esperanza que viene de F_2 , pero sólo dos son usadas al mismo tiempo. Las unidades en F_1 están construidas de tal forma que pueden activarse solamente si dos de las tres posibles fuentes de entrada están activas. Esta característica es llamada **Regla 2/3**.

La unidad de desactivación es responsable de sensar igualaciones entre patrones *bottom-up* y *top-down* en la capa F_1 . Puede ser modelado como un elemento de procesamiento simple con una salida a cada una de las unidades en la capa F_2 .

El subsistema de control de ganancia puede ser modelado como un elemento de procesamiento simple que recibe señales excitatorias de un patrón de entrada y puede ser inhibido por el patrón de esperanza que viene de la capa superior. G se combina con la regla 2/3 para permitir que F_1 distinga entre una señal de esperanza de arriba, y una señal de entrada de abajo.

Se puede pensar de la manera como operan las capas F_1 y F_2 en otra forma. El modelo básico de operación es el de una prueba de hipótesis. El patrón de entrada es pasado a F_2 , la cuál intenta reconocerlo. F_2 hace una suposición acerca de la categoría del patrón *bottom-up* y lo envía en la forma de patrón *top-down*, a F_1 . El resultado es entonces comparado con el patrón original; si la suposición es correcta, el patrón de prueba *bottom-up* y la suposición *top-down* mutuamente se refuerzan y mejoran. Si la suposición es incorrecta, F_2 crea otra suposición. Eventualmente el patrón podrá ser colocado en una categoría existente o aprendido como una nueva categoría. Así, F_2 forma una *hipótesis* de la categoría correcta para cada uno de los patrones de entrada; esta *hipótesis* es entonces comprobada enviándola de nuevo a la capa inferior para ver si los patrones *top-down* y *bottom-up* son suficientemente parecidos.

Si el patrón *top-down* no es suficientemente parecido al patrón *bottom-up*, la red automáticamente busca en los patrones almacenados por un mejor patrón. Este proceso de búsqueda continúa hasta que un patrón adecuado sea encontrado o hasta que el patrón de entrada sea aprendido como un nuevo elemento.

En la figura 3.2 se muestra una secuencia hipotética de eventos que pueden ocurrir en una red ART. El escenario es una operación simple de comparación de patrones durante la cual la red trata de determinar en cuál de los patrones almacenados clasifica un nuevo patrón de entrada.

En (a), el proceso inicia cuando un patrón de entrada (**I**) es presentado a las unidades

de la capa de entrada (F_1). Este patrón activa la unidad de desactivación (A) y al subsistema de control de ganancia (G). En F_1 se genera un patrón de activación STM (X) que envía una señal de inhibición a A. La red está estructurada para que esta señal inhibitoria cancele la señal excitatoria de I, de tal forma que A permanece inactiva. Esta misma actividad genera un patrón de salida (S) que es enviado a las unidades de la capa de almacenamiento (F_2), a través del filtro adaptativo *bottom-up* (W). Cada una de las unidades de F_2 recibe el vector de salida S , de F_1 , y calculan sus valores de entrada neta (T) de la manera usual, sumando los productos de los vectores de entrada y los pesos de las conexiones. En respuesta a las entradas de F_1 un patrón de actividad (Y) se desarrolla a través de F_2 . En (b), este patrón genera un patrón de salida (U) de F_2 . Este patrón envía una señal inhibitoria a G. G está configurado de manera tal que si recibe una señal inhibitoria de F_2 , cesa su actividad. U también es enviado como un segundo patrón de entrada a las unidades de F_1 , a través del filtro adaptativo *top-down* (Z). U es transformado por el filtro *top-down* en un patrón V , también llamado patrón de esperanza. Si V no es suficientemente parecido a I , entonces un nuevo patrón de actividad X^* es generado en F_1 . En (c), A es activado y envía una señal de desactivación al patrón Y en F_2 . Esta señal inhibe a esta unidad y se hace necesaria para prevenir que el mismo nodo gane la competencia durante el siguiente ciclo de comparación. En (d), después de que Y es inhibido, la esperanza *top-down* es eliminada y X vuelve a ser instalado en F_1 . Ahora X nuevamente genera un patrón W a F_2 , pero ya que Y está inhibido W puede activar un patrón diferente Y^* , en F_2 .

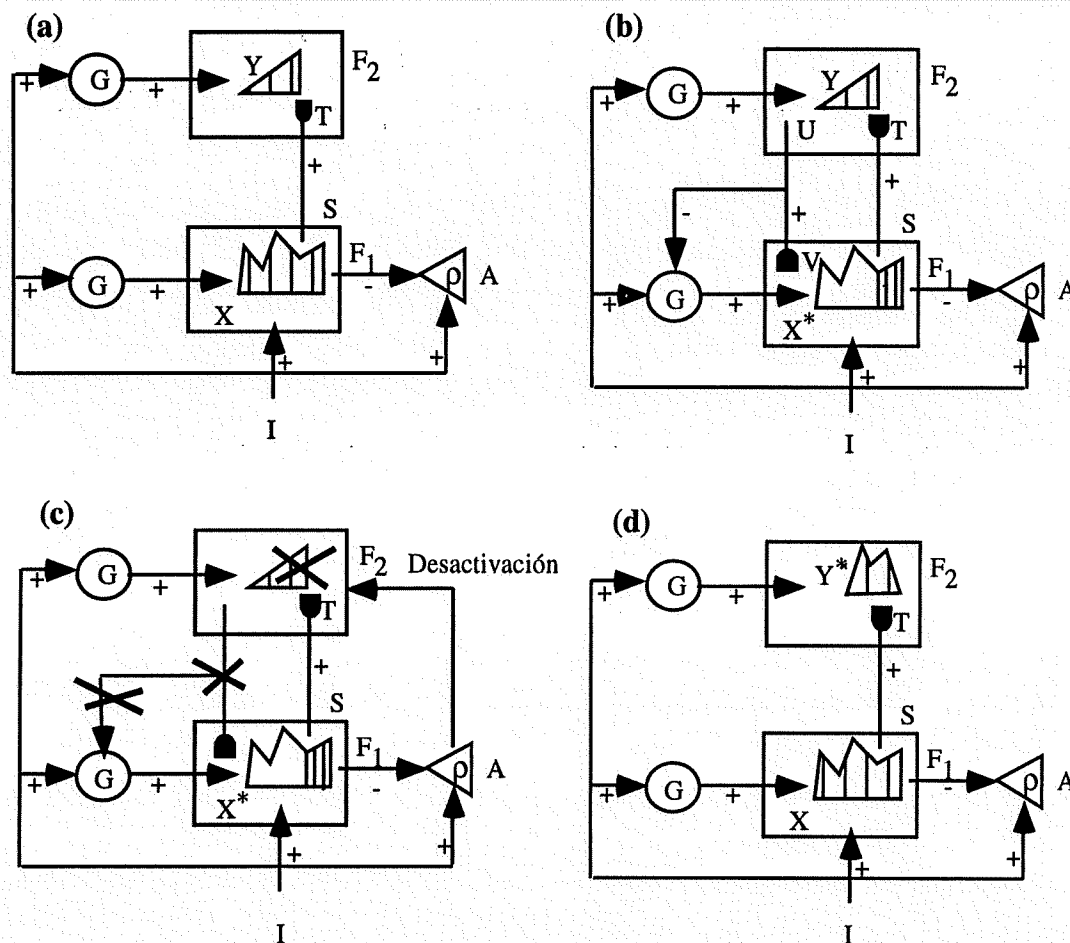


Figura 3.2 Ciclo de comparación de patrones de una red ART. El proceso implica la presentación de un patrón de entrada en (a), trata de igualar los patrones en (b), se da una desactivación en (c) y finalmente se reconoce el patrón en (d). (Adaptado de Carpenter et al, 1991b).

Este ciclo de comparación continúa hasta que un reconocimiento se de, o hasta que F_2 agote todos los patrones previamente almacenados. Si no se dio ningún reconocimiento la red asignará un nodo no comprometido e iniciará el aprendizaje de un nuevo patrón. Cuando se da un reconocimiento, no hay una señal de desactivación y la red se fija en un estado de resonancia.

3.3 ART1

La arquitectura de ART1 comparte la estructura de la figura 3.1. Se hace notar que todas las entradas para ART1 deberán ser vectores binarios, esto es, sus componentes deben ser elementos del conjunto $\{0,1\}$. Esta restricción puede ser una limitante de la red, pero existen muchos problemas para los cuales sus datos pueden ser expresados en forma binaria.

3.3.1 Funcionamiento de ART1

Suponga que I_i indica la entrada a la i -ésima unidad de v_1 de F_1 , $i=1,\dots,n$ y u_j indica la j -ésima unidad de F_2 , $j=1,\dots,m$; se dice que x_i denota la actividad o trazo STM, de v_i . Suponga que la actividad x_i excita una señal S_i en las trayectorias de v_1 a F_2 . Se dice que la señal neta de v_1 a u_j es $S_i W_{ij}$, donde W_{ij} denota el peso adaptativo *bottom-up* o trazo LTM, de la trayectoria de v_i a u_j . Entonces la señal total de F_1 a u_j es:

$$T_j = \sum_{i=1}^n S_i W_{ij}$$

La unidad con la activación más alta ($t_k = \max\{t_j : j=1,\dots,m\}$) es seleccionada como la categoría a la cual pertenece la entrada S_i y este valor es igualado a 1 mientras que las otras unidades son igualadas a cero. La selección de t_k , también llamada hipótesis, será

comprobada realizando una comparación con la *esperanza* de la categoría seleccionada $\{V\}$. Esta esperanza es equivalente a un conjunto de pesos *top-down*. Estos pesos pueden ser representados como una matriz $[Z]$ tal que la esperanza puede ser calculada por:

$$\{V\} = [Z] \{S\}$$

La activación en F_1 , $\{S\}$ y la esperanza de la categoría propuesta, $\{V\}$ son entonces comparadas. Una igualdad entre las dos señales ocurre cuando la razón de la normal entre estos dos vectores es mayor o igual que un parámetro ρ :

$$\frac{|V|}{|S|} \geq \rho$$

donde $|\cdot| \equiv |X| = \sum_{i=1}^n X_i$, y ρ es el parámetro de vigilancia que es usado para determinar qué tanto el ejemplar $\{S\}$ pertenece a la categoría representada por el prototipo $\{V\}$. Si no se cumple esta condición, la categoría seleccionada se deshabilita, esto es, y_k es igualado a -1 hasta que $\{S\}$ sea clasificado. De aquí que una nueva unidad y_j en F_2 es seleccionada como hipótesis. Entonces el proceso de crear una esperanza y compararla con $\{S\}$ es repetido hasta que se elija un y_j que cumpla la condición, o que todas las unidades de F_2 sean rechazadas. Si el segundo caso ocurre, significa que la red es incapaz de clasificar el ejemplar.

Cuando existe un reconocimiento del ejemplar, se dice que la red está en un estado de resonancia, es seleccionada una categoría para el patrón de entrada y la red aprende adaptando los trazos LTM representados por los pesos *bottom-up* y *top-down*, $[W]$ y $[Z]$, respectivamente. Las adaptaciones son realizadas como sigue:

Para los pesos *top-down*

$$Z_{ij}(t+1) = (Z_{ij}(t)S_i) \beta + (1 - \beta) Z_{ij}$$

donde $\beta \in [0, 1]$ corresponde a la tasa de aprendizaje; en el aprendizaje rápido $\beta = 1$ y la ecuación se resume en:

$$Z_{ij}(t+1) = Z_{ij}(t)S_i$$

Para los pesos *bottom-up*

$$W_{ij}(t+1) = \frac{|Z_{ji}(t)S_i|}{\alpha + |Z_{ji}|}$$

donde $\alpha > 0$ es el parámetro de elección.

3.4 ART2

ART2 difiere de ART1 en la naturaleza de los patrones de entrada: ART2 acepta vectores de componentes continuos (o escala de gris) así como componentes binarios.

En primera instancia se incrementa la complejidad en el nivel de procesamiento de F_1 . El nivel F_1 de ART2 consiste de varios subniveles y varios subsistemas de control de ganancia. El procesamiento en F_2 es el mismo en ART2 como en ART1.

3.4.1 Arquitectura de ART2

La Figura 3.3 [Carpenter et al, 1987b] muestra una arquitectura de la red ART2. La capa F_1 es dividida en seis subcapas, **w**, **x**, **u**, **v**, **p** y **q**. Cada uno de los nodos etiquetados por **G** es una unidad de control de ganancia que envía una señal inhibitoria a cada una de las unidades que alimenta. Todas las subcapas, así como la capa **r** de la unidad de desactivación, tienen el mismo número de unidades. Las capas individuales están conectadas unidad a unidad, esto es, las capas no son completamente interconectadas con excepción de las conexiones *bottom-up* y *top-down*.

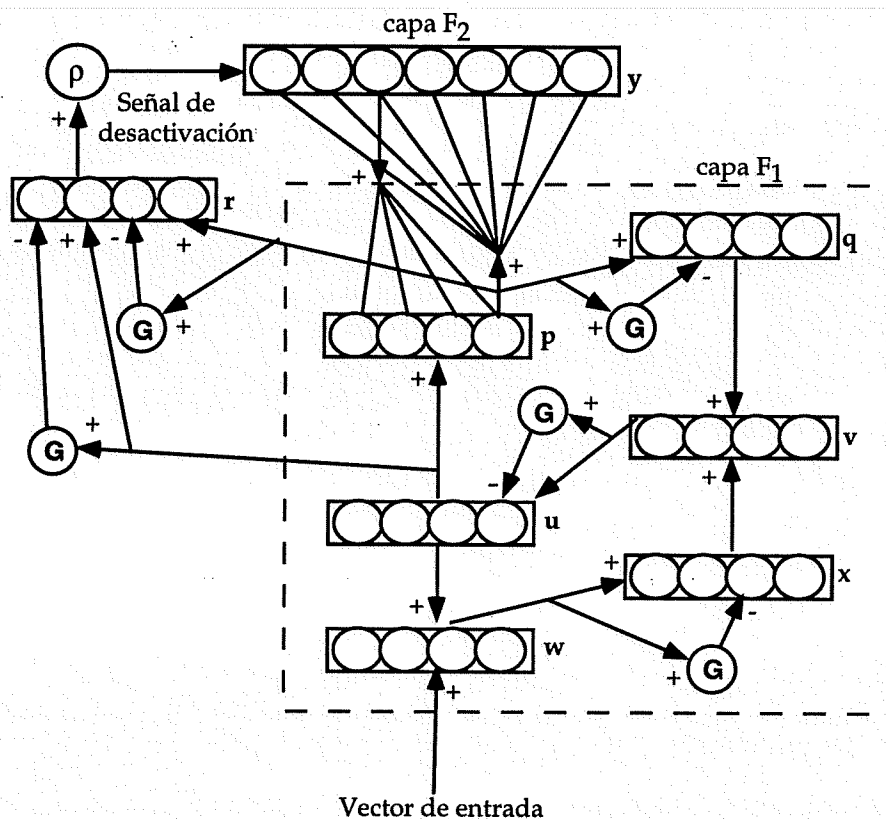


Figura 3.3. Arquitectura de ART2. Al igual que ART1, ART2 cuenta con 2 capas de procesamiento de elementos, un sistema de ganancia y un sistema de desactivación. Para que ART2 pueda trabajar con vectores analógicos la capa F_1 es dividida en un número de subcapas que contiene conexiones de alimentación hacia adelante y retroalimentación.

Además, en el sistema de desactivación se requiere muestrear la información para determinar el grado de igualdad entre el patrón de entrada y el patrón de esperanza, para lograrlo, se agrega un vector r de la misma magnitud que las subcapas de F_1 .

3.4.2. Funcionamiento de ART2.

Las activaciones de cada una de las subcapas de F_1 están dadas por una ecuación de la forma:

$$x_k = \frac{J_k^+}{A + DJ_k^-}$$

donde A y D son constantes diferentes de cero. J_k^+ y J_k^- representan señales excitatorias e inhibitorias, respectivamente.

Los valores de esta ecuación varían de acuerdo a la subcapa que se esté considerando. La tabla I muestra todas las cantidades apropiadas para cada una de las subcapas de F_1 así como la capa del sistema de desactivación.

Basados en la tabla I, las actividades de cada una de las subcapas en F_1 pueden ser resumidas por las siguientes ecuaciones:

$$w_i = I_i + au_i$$

$$x_i = \frac{w_i}{\|w\|}$$

Tabla I. Factores en la ecuación de activación para cada una de las subcapas y la capa r. I_i es el i-ésimo componente de el vector de entrada. Los parámetros a, b, y c son contantes. y_j es la actividad de la j-ésima unidad en la capa F_2 y $g(y)$ es la función de salida en F_2 .

Capa	A	D	J_i^+	J_i^-
w	1	1	$I_i + au_i$	0
x	1	1	w_i	$\ w\ $
u	0	1	v_i	$\ v\ $
v	1	1	$f(x_i) + bf(q_i)$	0
p	1	1	$u_i + \sum g(y_j)z_{ij}$	0
q	0	1	p_i	$\ p\ $
r	0	1	$u_i + cp_i$	$\ u\ + \ cp\ $

$$v_i = f(x_i) + bf(q_i)$$

$$u_i = \frac{v_i}{\|v\|}$$

$$p_i = u_i + \sum_j g(y_j)W_{ij}$$

$$q_i = \frac{p_i}{\|p\|}$$

la forma de la función $f(x)$, determina la naturaleza del contraste que hubo en F_1 . Esta función puede ser

$$f(x) = \begin{cases} 0 & 0 \leq x \leq \theta \\ x & x > \theta \end{cases}$$

donde θ es un constante positiva menor que uno.

Las entradas *bottom-up* son calculadas como en ART1:

$$T_j = \sum p_i W_{ij}$$

La función de salida de F_2 está dada por:

$$g(y_i) = \begin{cases} d & T_j = \max_k \{T_k\} \\ 0 & \text{en otro caso} \end{cases}$$

Ahora se puede reescribir la ecuación para el procesamiento en la subcapa p de F_1

como:

$$p_i = \begin{cases} u_i & \text{si } F_2 \text{ está inactivo} \\ u_i + dW_{ij} & \text{si el } J\text{-ésimo nodo está inhibido} \end{cases}$$

3.4.3 Trazos LTM.

Las ecuaciones de los trazos LTM, *bottom-up* y *top-down*, están dadas por:

top-down ($F_2 \rightarrow F_1$)

$$\frac{d}{dt} Z_{ij} = g(y_i)[p - Z_{ij}]$$

bottom-up ($F_1 \rightarrow F_2$)

$$\frac{d}{dt} W_{ij} = g(y_i)[p_i - W_{ij}]$$

si F_2 realiza una elección, implica que el J -ésimo nodo F_2 está activo, entonces:

$$Z_{ij} = W_{ij} = \frac{u_i}{1-d}$$

donde se asume que $0 < d < 1$.

3.4.4 Sistema de desactivación.

De la tabla I y la ecuación de activación, se puede construir la ecuación para las actividades de los nodos de la capa r del sistema de desactivación:

$$r = \frac{u_i + cp_i}{\|u\| + \|cp\|}$$

por lo tanto la condición de desactivación es:

$$\frac{\rho}{\|r\|} > 1$$

donde ρ es el parámetro de vigilancia como en ART1.

3.5 Lógica difusa.

La lógica difusa trabaja con proposiciones que pueden ser ciertas en un grado de certeza de 0 a 1. Este grado de certeza difiere de la probabilidad ya que los eventos en las

probabilidades son mutuamente exclusivos y no pueden sumar más que 1.

La lógica difusa puede ser mejor entendida en el contexto de membresía de conjunto. Por ejemplo para decir que el día de hoy va a llover en un grado de certeza de 0.8, puede ser expresado como el par ordenado (hoy,0.8), donde el primer elemento es el candidato y el segundo un valor entre 0 y 1 llamado grado de membresía en el conjunto. Los conjuntos difusos son conjuntos en los cuales los miembros son presentados como pares ordenados que incluyen información en grados de membresía.

Las operaciones tradicionales que se realizan en conjuntos son la *unión*, en la cual se toman todos los elementos que existen en ambos conjuntos; y la *intersección*, en la cual se toman los elementos que estén en un conjunto y en otro. En el caso de conjuntos difusos con la unión se trata de encontrar el grado de membresía que un elemento tiene en un nuevo conjunto difuso.

La unión de dos conjuntos difusos se da con los grados de membresía de cada elemento. El grado de membresía de un elemento en la unión de dos conjuntos difusos es el máximo valor de sus grados de membresía. Si a, b, c y d tienen los grados de membresía en los conjuntos difusos A y B como $A = (0.9, 0.4, 0.5, 0)$ y $B = (0.7, 0.6, 0.3, 0.8)$, entonces $A \cup B = (0.9, 0.6, 0.5, 0.8)$.

El grado de membresía en la intersección de dos conjuntos difusos es el mínimo, o el valor más pequeño de sus grados de membresía. Para los conjuntos A y B ya definidos

$$A \cap B = (0.7, 0.4, 0.3, 0).$$

El complemento de un conjunto difuso se obtiene como sigue: si el grado de membresía para un miembro es 0.8, entonces que un miembro no esté en el conjunto es el grado $1 - 0.8 = 0.2$. Con los conjuntos anteriores $A' = (0.1, 0.6, 0.5, 1)$ y $B' = (0.3, 0.4, 0.7, 0.2)$.

Note que en conjuntos no difusos $A \cap A' = \emptyset$ y en el caso de conjuntos difusos $A \cap A' \neq \emptyset$.

3.6 Fuzzy ART.

3.6.1 Descripción de Fuzzy ART.

El modelo de Fuzzy ART [Carpenter et al, 1991b] fue desarrollado a partir del modelo de ART1 para ser capaz de un aprendizaje estable y clasificar categorías en respuesta a patrones de entrada continuos o binarios.

Fuzzy ART incorpora cálculos de la teoría de conjuntos difusa como el operador MIN ($\wedge$) y MAX ($\vee$). Fuzzy ART incorpora además características de todos los sistemas ART, como la comparación de patrones entre la entrada *bottom-up* y los vectores de prototipos aprendidos *top-down*.

Existe una analogía entre las operaciones de ART1 y Fuzzy ART [Carpenter et al,

1991b]. La tabla II resume cómo las operaciones en ART1 de elección de categorías, comparación, búsqueda y aprendizaje trasladan a las operaciones de Fuzzy ART reemplazando el operador de intersección de la teoría de conjuntos ($\cap$) de ART1 por la conjunción de la teoría de conjuntos difusos, u operador MIN ($\wedge$). Para el caso especial de entradas binarias y aprendizaje rápido, los cálculos de Fuzzy ART son idénticos a los de la red neuronal ART1. El algoritmo de Fuzzy ART también describe dos características opcionales, una concerniente al aprendizaje y la otra al preprocesamiento de las entradas.

Tabla II. Analogía entre ART1 y Fuzzy ART. En ART1 se utiliza la Intersección lógica AND ($\cap$) y para ART difuso se utiliza el mínimo AND difuso ($\wedge$).

ART1 (Binario)	FUZZY ART (Analógico)
<u>Elección de categoría</u>	
$T_j = \frac{ I \cap W_j }{\alpha + W_j }$	$T_j = \frac{ I \wedge W_j }{\alpha + W_j }$
<u>Criterio de comparación</u>	
$\frac{ I \cap W }{ I } \leq \rho$	$\frac{ I \wedge W }{ I } \leq \rho$
<u>Aprendizaje rápido</u>	
$W_j^{(\text{nuevo})} = I \cap W_j^{(\text{viejo})}$	$W_j^{(\text{nuevo})} = I \wedge W_j^{(\text{viejo})}$

3.6.2 Operaciones opcionales.

Para algunas aplicaciones, es útil combinar un aprendizaje rápido inicialmente con una tasa muy baja de olvido. Llamamos a esta opción *compromete-rápido*

recodificación-lenta. Esta combinación de propiedades retiene el beneficio del aprendizaje rápido; particularmente, una respuesta adecuada a entradas que pueden ocurrir raramente y una respuesta a las que se presentan constantemente puede ser dada rápidamente. La operación de recodificación-lenta también previene que características que están ya incorporadas en los prototipos de las categorías sean erróneamente borradas en respuesta a ruido o entradas parciales. Con la opción de recodificación-lenta sólo una persistencia estadística cambia una relación de una característica a una categoría.

La opción de preprocesamiento de las entradas se refiere a la normalización de los patrones de entrada. Un procedimiento de normalización llamado **codificación-complemento** será mostrado más adelante (sección 6.2.1).

En los sistemas Fuzzy ART se maneja un rango de parámetro llamado **límite conservativo**. En este límite, una entrada siempre selecciona una categoría en la que el vector de pesos es un subconjunto difuso de la entrada, si tal categoría existe. Dada tal elección, no ocurren cambios de pesos durante el aprendizaje; de aquí el nombre de límite conservativo ya que los pesos aprendidos son conservados lo más posible.

Para vectores continuos el grado para el cual y es un subconjunto difuso de x está dado por el término:

$$\frac{|x \wedge y|}{|y|}$$

En el límite conservativo de Fuzzy ART, la función de elección T_j en la ecuación de elección refleja el grado para el cual el vector de pesos W_j es un subconjunto difuso de el vector de entrada I . Si

$$\frac{|I \wedge W_j|}{|W_j|} = 1,$$

entonces W_j es un subconjunto difuso de I , y la categoría j es un subconjunto difuso elegido para la entrada I . En este caso, no ocurren recodificaciones si j es seleccionado ya que $I \wedge W_j = W_j$.

4. Clasificación de imágenes de satélite.

4.1 Métodos tradicionales.

Las técnicas de sensado remoto se utilizan para analizar información espectral de energía electromagnética reflejada por los objetos. Esto es de utilidad en estudios ambientales ya que es posible el observar áreas de amplia cobertura de manera simultánea y en forma periódica.

La interpretación de las imágenes de satélite es llevada a cabo mediante dos tipos de métodos [Richards, 1986]: uno basado en el procesamiento digital de imágenes usando computadoras (análisis cuantitativo), y el segundo basado en la inspección visual, donde la información contextual, tal como la proximidad a las regiones conocidas es tomada en cuenta por el interprete para definir la naturaleza y localización de los diferentes tipos de regiones (análisis cualitativo).

Una aplicación importante en la interpretación de datos de sensado remoto es la identificación de tipos de terreno. La clasificación automática de cubiertas de terrenos es de particular importancia en áreas de difícil acceso y que necesitan ser monitoreadas con mucha frecuencia. La identificación de terrenos es una tarea adecuada para los métodos cuantitativos de análisis, ya que la computadora es capaz de usar todos los componentes espectrales obtenidos del sensor, mientras que el análisis cualitativo no lo permite dadas las limitaciones del ojo humano.

Tradicionalmente se han utilizado clasificadores de tipo Bayesiano basados en la suposición teórica de que su función de densidad de probabilidad se comporta como una distribución de Gauss. Esta dependencia en un modelo probabilístico específico es la principal limitante de este tipo de clasificadores.

Los métodos de clasificación se dividen en métodos supervisados y no supervisados.

4.1.1 Técnicas de clasificación supervisadas

La clasificación supervisada es el procedimiento más utilizado para un análisis cuantitativo de imágenes de sensado remoto. Existen algoritmos que son capaces de representar los píxeles en tipos o clases de suelo. Estos algoritmos se basan en modelos de distribución de probabilidades de las clases de interés. En estos, el espacio multispectral es dividido en regiones de clases específicas usando la localización óptima de las superficies [Richards, 1986].

Los pasos que siguen las técnicas supervisadas son:

1. Decidir los tipos de suelo en que la imagen va a ser segmentada (agua, desierto, vegetación, etc.).
2. Elegir los píxeles representativos o prototipos para cada una de las clases deseadas. Estos píxeles forman el *conjunto de entrenamiento*. Este conjunto puede ser establecido por medio de visitas de campo, análisis de mapas o utilizando fotografías aéreas mediante

fotointerpretación.

3. Se utilizan los datos de entrenamiento para determinar los parámetros del algoritmo de clasificación particular que será usado; estos parámetros pueden ser las propiedades del modelo de probabilidad usado o las ecuaciones que definen las particiones en el espacio multiespectral.

4. Usando el clasificador entrenado, se clasifican cada uno de los píxeles de la imagen en los tipos de suelo deseados (*clases informativas*).

Las técnicas de clasificación supervisada más conocidas son: Clasificador de Máxima similitud y el Clasificador de Distancias Mínimas [Richards, 1986]

4.1.2 Técnicas de clasificación no supervisadas

Las técnicas de clasificación no supervisada clasifican la imagen en clases desconocidas y es tarea del analista etiquetar estas clases posteriormente.

Los píxeles son asignados a grupos que están representados por símbolos para indicar que esos píxeles pertenecen a la misma agrupación o *clase espectral*. Si algunos píxeles pueden ser identificados por un tipo de suelo en particular, entonces todos los píxeles con el mismo símbolo pueden ser asociados con esa clase.

En las técnicas no supervisadas el analista no participa, sino hasta que se completa la

clasificación de toda la imagen. Estas técnicas son utilizadas especialmente cuando los datos de entrenamiento para la clasificación supervisada no pueden ser obtenidos o cuesta mucho obtenerlos. También son utilizadas para determinar el número de clases espectrales que pueden ser consideradas para una clasificación supervisada posterior.

4.2 Redes neuronales para la clasificación de imágenes.

La capacidad de aprendizaje de las redes neuronales representa una alternativa a los clasificadores tradicionales de tipo Bayesiano, y tienen la ventaja de que no requieren de un conocimiento previo de la distribución estadística de las clases, ya que los modelos de redes neuronales están libres de una distribución, esta es una característica importante ya que las distribuciones estadísticas pueden cambiar en clases de datos así como en la fuente de los mismos.

Otra ventaja de las redes neuronales sobre los clasificadores tradicionales es que éstas no requieren de una especificación previa de la influencia que cada una de las fuentes de datos debe tener para el proceso de clasificación, de aquí que las redes neuronales son preferibles para clasificación de datos de sensado remoto de múltiples fuentes [Benediktsson et al, 1990].

En la figura 4.1 se presenta un esquema para la clasificación de imágenes del modo supervisado [Favela et al, 1994]. Primero se forma el conjunto de entrenamiento a partir de pixeles ejemplares de tipos de suelo conocidos. La información espectral de estos puntos sirve como entrada a la red y el tipo de suelo sirve como la salida esperada para ese

punto. Se procede a entrenar a la red hasta que se obtiene un margen de error deseable. La red puede ser probada usando puntos del conjunto de entrenamiento que no hayan sido utilizados durante el entrenamiento. Y finalmente, los valores espectrales de cada uno de los píxeles en la imagen alimentan la red para obtener su clasificación.

Este proceso es una aproximación general para la clasificación de imágenes utilizando redes neuronales del modo supervisado. Para el modo no supervisado el proceso se reduce al último paso.

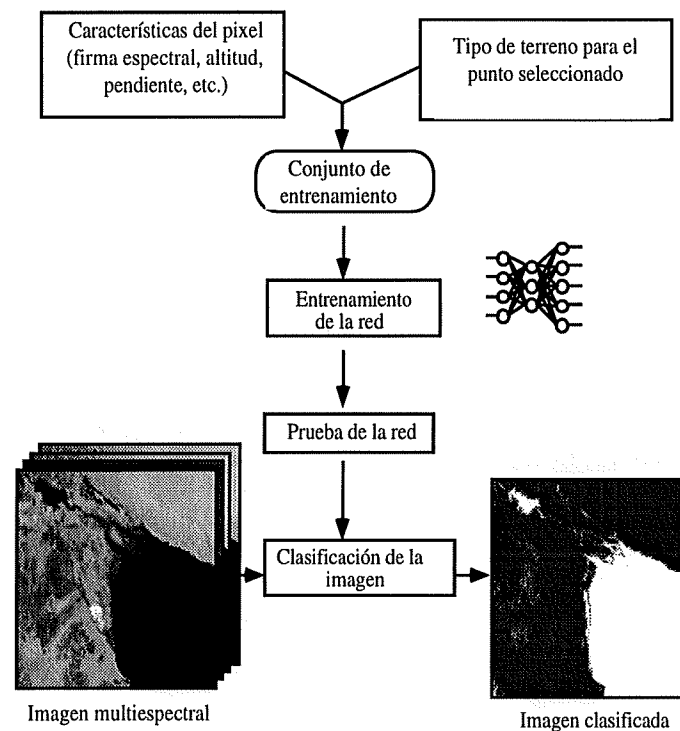


Figura4.1. Proceso de clasificación con redes neuronales.

En las secciones siguientes se describen los procesos de clasificación usando las redes

neuronales de Retropropagación y FuzzyART.

4.2.1 Red Neuronal de Retropropagación.

Las redes de retropropagación trabajan en modo supervisado. Deben de contener al menos tres capas, una capa de entrada, una capa de salida y al menos una capa intermedia [Ruiz, 1995].

El algoritmo de aprendizaje que utiliza es la Regla Delta Generalizada, la cual se basa en la modificación continua de los pesos entre las conexiones de las neuronas para minimizar la diferencia entre la salida obtenida y la salida deseada (delta) para minimizar el error. La modificación de los pesos se lleva a cabo por medio de la Regla del Gradiente descendente, esto es, los pesos son modificados en una cantidad proporcional a la primera derivada del error con respecto a los pesos.

El proceso de entrenamiento de la red consiste de dos pasos. Primero se debe aplicar un patrón de entrenamiento a la red que se denota por X_p . La entrada total a una neurona se calcula por medio de la sumatoria de los productos punto entre los pesos de las conexiones asociadas a la neurona y la salida de la neurona de la capa anterior, para el caso de la capa de salida. Para la capa intermedia es reemplazada la salida de la neurona por el componente del vector de entrenamiento aplicado. Esto puede expresarse matemáticamente de la siguiente forma:

$$\text{net}_{pj} = \sum_{i=0}^N w_{ji} i_{pi}$$

donde net_{pj} es la entrada neta a la neurona, w_{ji} es el peso de la conexión de la neurona i a la neurona j , e i_{pi} es la salida de la neurona i , o el componente i del patrón de entrenamiento $\mathbf{X}_p$ según sea el caso.

Para el cálculo de la salida de una neurona se toma el resultado de evaluar la función de activación de la neurona en el valor de entrada obtenido para dicha neurona. Esto es :

$$o_{pj} = f_j(\text{net}_{pj})$$

donde f_j es la función de activación, o_{pj} es la salida de la neurona j y net_{pj} es la entrada a dicha neurona. La función de activación utilizada es la función sigmoideal:

$$f(\text{net}) = (1 + e^{-\text{net}})^{-1}$$

Estas fórmulas son aplicadas a todas las neuronas de cada capa hasta obtener una salida de la red. Esta salida es comparada entonces con la salida deseada y el error es calculado como:

$$E_p = \frac{1}{2} \sum_{j=1}^M (y_{pj} - o_{pj})^2$$

donde y_{pj} es la salida esperada para la neurona de salida j y o_{pj} es la salida obtenida de dicha neurona.

El segundo paso en el entrenamiento consiste en la retropropagación del error para realizar la modificación de los pesos. Esta modificación de los pesos se hace como se dijo anteriormente por medio de la regla Delta Generalizada y se tiene que :

$$\Delta w_{ji} \propto - \frac{\partial E_p}{\partial w_{ji}}$$

por lo tanto la modificación de los pesos será :

$$w_{ji}(t + 1) = w_{ji}(t) + \Delta_p w_{ji}(t) + \alpha \Delta_p w_{ji}(t - 1)$$

Los parámetros involucrados en el cálculo de esta ecuación son :

- Tasa de aprendizaje : determina la velocidad con la que la red aprende.
- Momentum (α) : se utiliza para incrementar la velocidad del aprendizaje.
- Ruido : término adicional para incluir entradas con ruido y permitir el posterior reconocimiento de entradas con ruido.

Para formar el conjunto de patrones de entrenamiento, se seleccionan puntos de la imagen (píxeles) que representan a las clases por reconocer.

Cada uno de los vectores de entrenamiento consta a su vez de dos vectores. El primer vector representa el patrón de entrada que se espera que la red aprenda. Este vector de entradas consta de n valores, cada uno de los cuales corresponde al valor del punto seleccionado (pixel), en cada una de las n bandas. Los valores obtenidos para cada pixel son normalizados para que estos valores estén dentro del intervalo $(0,1)$.

El vector de salidas en cada patrón de entrenamiento consiste de x valores (número de clases). Para representar los x vectores de salidas deseados se asigna un 0.95 a un sólo nodo, y cero a los demás nodos, de acuerdo a la posición asignada a la clasificación. Por ejemplo, si se tiene que $x=15$ el vector de salidas será :

$$(0.95, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

como salida esperada para la primera clase deseada y el vector

$$(0, 0.95, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

para la segunda clasificación y así sucesivamente.

Los resultados obtenidos utilizando una red de retropropagación para la clasificación de imágenes son muy semejantes a los obtenidos por métodos tradicionales [Ruiz, 1995] y tienen la ventaja sobre éstos que pueden ser agregados más información como entrada tal como los valores de altitud, pendiente y algunas otras característica de la zona sin ningún problema y los resultados que se obtendrían serían mucho más exactos.

4.2.2 Red Neuronal Fuzzy ART.

La red de Fuzzy ART fue desarrollada por Carpenter y Grossberg [Carpenter y Grossberg, 1991b]. Es una red del modo no supervisado que utiliza operaciones de lógica difusa para la clasificación de patrones de entrada con valores continuos. Esta red es explicada con detalle en la sección 3.6.

La arquitectura de la red utilizada fue de 4 neuronas en la capa de entrada, correspondientes a cada banda espectral, y 15 neuronas en la capa de salida, número máximo de tipos de terreno identificados en un trabajo que ha servido como marco de referencia para evaluar el algoritmo de clasificación [Torres, 1994].

A partir de la red neuronal ART1 se hicieron las modificaciones de la Tabla I mediante operaciones difusas, es decir cambiando el operador de intersección de la teoría de conjuntos ($\cap$), por el operador de intersección de la teoría de conjuntos difusos, $\text{MIN}(\wedge)$.

Cada uno de los valores espectrales de las bandas fueron divididos entre el valor máximo que obtiene el sensor, para el caso del MSS es de 127, de tal forma que los valores obtenidos para alimentar la red estuvieran en el intervalo de (0,1).

Deben definirse los parámetros que definen el comportamiento de la red, los principales

son:

- *laza de aprendizaje*: la cual define la velocidad en que la red realiza el proceso de aprendizaje, es decir la modificación de los pesos.

- *parámetro de vigilancia*: el cual marca el grado de exactitud o exigencia que se le pide a la red para que acepte a un vector de entrada como elemento de una clase.

Además de estos parámetros existen otros, los cuales son explicados en el capítulo 6 donde se muestran también los resultados obtenidos de las pruebas de clasificación y los efectos que cada uno de los parámetros tienen en la red.

5. Sistema para la clasificación de imágenes multiespectrales (NeuroClas)

Se desarrolló un sistema para la clasificación de imágenes de satélite por medio de redes neuronales del modo supervisado y no supervisado, Retropropagación y Fuzzy ART respectivamente.

El sistema fue desarrollado sobre el sistema XWindows y es descrito en este capítulo.

5.1 El sistema XWindows.

El sistema XWindows es un ambiente de programación para estaciones de trabajo. Éste ofrece un ambiente rico y complejo al programador de aplicaciones. El fundamento de X es un sistema a base de ventanas. La interfaz de este sistema con el mundo exterior utiliza el protocolo de red X [Jones, 1989].

Los programas de aplicación de X ordinariamente no usan el protocolo de red directamente sino que trabajan a través de la interfaz de programación. El sistema XWindows provee un paquete de subrutinas en lenguaje C conocido como Xlib con el propósito de dar una interfaz a las aplicaciones con el protocolo de red.

La mayoría de las aplicaciones de X son escritas en lenguaje C y hacen llamadas a Xlib.

Los programas de aplicación interactúan con la estación de trabajo por:

- Eventos recibidos de la estación.
- Reaccionando apropiadamente a estos eventos.

5.2 NeuroClas.

NeuroClas es un sistema desarrollado para la clasificación de imágenes multiespectrales con redes neuronales del modo supervisado y no supervisado, además de técnicas adicionales para el pre-procesamiento de las imágenes fuente y postprocesamiento de las imágenes obtenidas por la clasificación.

El sistema trabaja en estaciones de trabajo bajo sistema operativo UNIX. Está programado en C y hace uso de las librerías de X11 para definir la interfaz gráfica.

Inicialmente se realizaron programas independientes para la clasificación de imágenes con las redes neuronales de Retropropagación y FuzzyART. Estos programas fueron programados en C++ y utilizaban clases para definir las estructuras de las redes (Figura 5.1a). Se desarrolló un programa en C utilizando las librerías de X11 para desplegar las imágenes clasificadas que se generaban con los programas de clasificación. Posteriormente se decidió integrar los programas en un sólo sistema para facilitar el proceso de clasificación. No se logró integrar los programas orientados a objetos con las funciones que hacían llamadas a las librerías de X11, por lo cual fue necesario pasar las clases a estructuras (Figura 5.1b) y modificar las funciones que antes estaban definidas

como métodos de las clases.

```
class art
{
    int          n_inputs; // # de neuronas en la capa de entrada
    int          m_outputs; // # de neuronas en la capa de salida
    VECTOR       inputs;    // vector de entrada
    VECTOR       outputs;   // vector de salida
    MATRIZ       topdown;   // pesos de la capa de salida a la de entrada
    MATRIZ       bottomup; // pesos de la capa de entrada a la de salida
    float        beta;      // tasa de aprendizaje
    float        rho;       // factor de vigilancia
    float        alfa;      // parámetro de elección

public:
    void entrenar(); // método de la clase para la clasificación
};
```

(a)

```
struct art
{
    int          n_inputs; // # de neuronas en la capa de entrada
    int          m_outputs; // # de neuronas en la capa de salida
    VECTOR       inputs;    // vector de entrada
    VECTOR       outputs;   // vector de salida
    MATRIZ       topdown;   // pesos de la capa de salida a la de
entrada
    MATRIZ       bottomup; // pesos de la capa de entrada a la de
salida
    float        beta;      // tasa de aprendizaje
    float        rho;       // factor de vigilancia
    float        alfa;      // parámetro de elección
};

void entrenar(); // función que realiza el entrenamiento
```

(b)

Figura 5. 1. a) Declaración de la clase art. b) Declaración de la estructura art.

Posteriormente se agregaron técnicas de histogramas que sirvieran como

preprocesamiento para las imágenes antes de ser clasificadas. Además se incluyeron procedimientos para la modificación de la paleta de colores para que la imagen clasificada pudiera ser mejor apreciada. La implementación del sistema es explicada con más detalle en el apéndice A.

El sistema está compuesto de tres módulos principales (figura 5.2) un módulo de **preprocesamiento** que pretende mejorar la apariencia de la imagen o resaltar algunos aspectos de la información contenida; el segundo corresponde a la **clasificación de la imagen** con redes neuronales; y el tercero con funciones de **postprocesamiento** para dar una “maquillada” a la imagen, de tal forma que las clases encontradas sean mejor apreciadas.

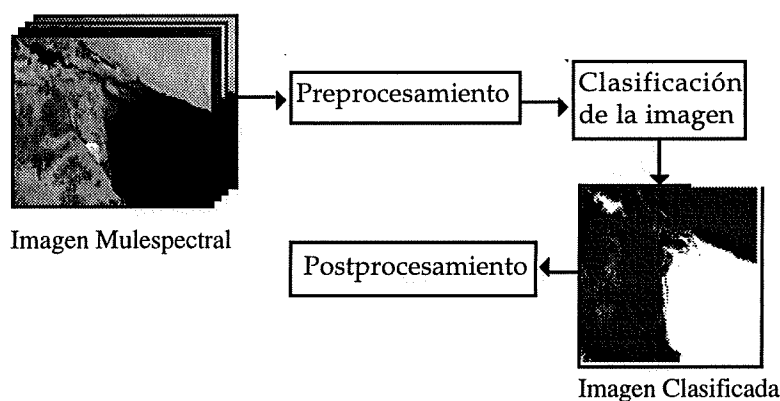


Figura 5.2. Módulos principales de NeuroClas.

5.2.1 Preprocesamiento.

Este módulo consiste en técnicas de realce de imágenes puntuales a partir del histograma de la imagen, mediante el uso de una función de transformación.

El histograma de una imagen es simplemente un conjunto de números que determinan el porcentaje de píxeles que una imagen tiene en sus niveles de gris [Richards, 1986].

Considere una imagen de 512 x 512 con 128 niveles de gris. Esta imagen tiene un total de 262,144 píxeles. Se tiene que n_i es el número de píxeles en la imagen que tienen el i -ésimo nivel de gris, G_i . El histograma de la imagen es entonces definido como el conjunto de números dados por:

$$h_i = \frac{n_i}{n_t}$$

para $i = 0, \dots, (M-1)$,

donde

M es el número de niveles de gris,

n_i es número de píxeles con el nivel de gris G_i

n_t es el total de píxeles en la imagen.

El realce de la imagen se lleva a cabo utilizando una función de transformación del modo:

$$s = T(r)$$

donde

r es el valor original,

T es la función de transformación aplicada y

s el valor resultante de la transformación.

Las funciones de transformación que han sido incluidas en este sistema son:

- *Transformación lineal*. También conocida como estiramiento de contraste. En esta se hace una ampliación del intervalo dinámico en la imagen. La función lineal es del tipo $y = ax + b$.

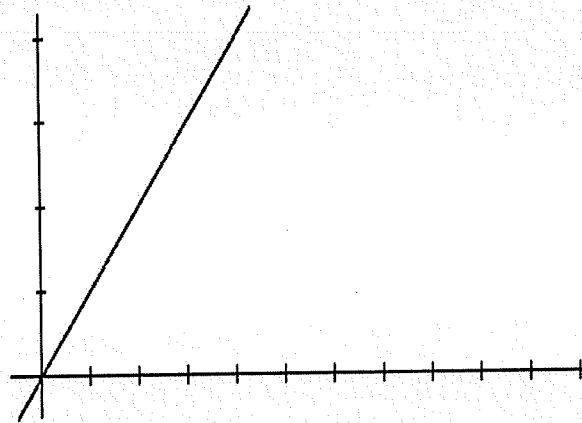


Figura 5.3 Gráfica de una transformación lineal.

- *Transformación exponencial*. Esta transformación utiliza la función exponencial, $y = e^x$ (Figura 5.4). Esta transformación se recomienda para imágenes claras.

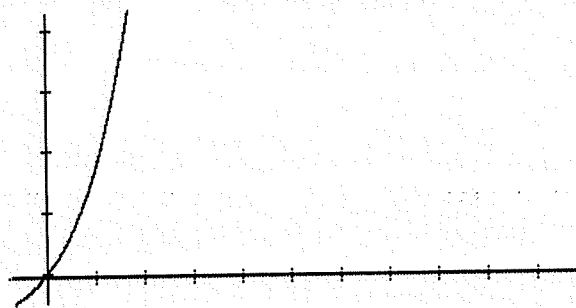


Figura 5.4 Gráfica de la función de transformación exponencial.

- *Transformación logarítmica.* Se utiliza la función logarítmica, $y = \ln x$ y se aplica para imágenes oscuras.

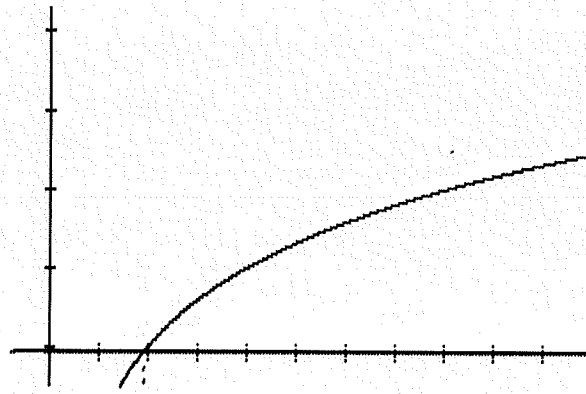


Figura 5.5 Gráfica de la función de transformación logarítmica.

Por ejemplo la transformación del nivel de gris 45 con una recta con pendiente 0.6 utilizando la fórmula lineal es:

$$y = 0.6 * 45 = 27$$

por lo tanto el nivel de gris 45 toma el valor de 27 por lo tanto la imagen será más oscura.

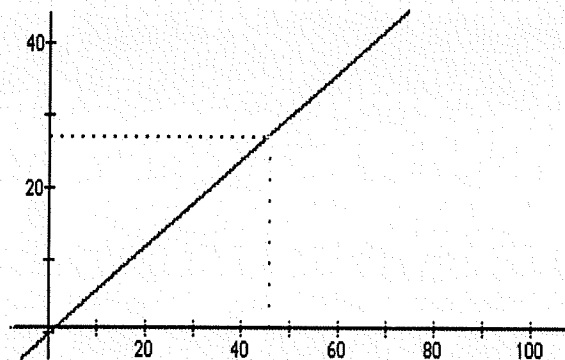


Figura 5.6 Ejemplo de una transformación lineal con una recta de pendiente 0.6. Las líneas punteadas muestran la transformación del tono 45.

5.2.2 Clasificación.

La parte de clasificación es explicada con detalle en el capítulo 4 y está compuesta de un submódulo de aprendizaje basado en redes neuronales FuzzyART y un submódulo de clasificación supervisado que utiliza la red de Retropropagación.

5.3.3 Postprocesamiento

En este módulo se implementó una función que toma el contexto del pixel para dar un suavizado a la imagen de tal forma que las clases queden mejor definidas y así facilitar aún más su interpretación. Este procedimiento es una primera aproximación que toma los 8 pixeles vecinos de cada pixel y si en ese conjunto no hay una cantidad suficiente del mismo tono que el pixel central, se cambia su tono al que predomine en el conjunto.

5.3 Descripción de Neuroclas.

NeuroClas fue diseñado con un ambiente de ventanas y botones que se manejan con el ratón de manera que la interacción con el usuario sea más sencilla.

Se tiene una ventana principal (Figura 5.7) la cual contiene los procedimientos que el sistema puede operar. Se crean ventanas para el despliegue de la imagen las cuales tienen las características de las ventanas del ambiente en el cual se encuentra corriendo el sistema. Además se tienen ventanas de diálogo (Figura 5.8) donde se pedirán los datos requeridos para los procesamientos, así como nombres de archivos, etc.. Si se presenta un error en el sistema se despliega una ventana de mensajes que indica el tipo de error detectado. La

forma de operar al sistema NeuroClas es explicado en el apéndice B.



Figura 5.7 Ventana principal de Neuroclas.

6 . Resultados y discusiones.

En este capítulo se muestran las características de la imagen con que fueron realizadas las pruebas. Se dan a conocer algunos de los problemas que se tuvieron en las primeras pruebas y la forma en que fueron solucionados. Finalmente se presentan los resultados obtenidos con algoritmo de clasificación FuzzyART.

6.1 Características de la imagen clasificada.

La imagen de prueba que se utilizó con el clasificador neuronal corresponde a la región deltaica del Río Colorado. Esta imagen fue adquirida por el sensor LANDSAT MSS, en el verano de 1979 y tiene un tamaño de 2048x2048 pixeles. Consta de cuatro bandas espectrales (4, 5, 6 y 7) corregidas radiométrica y geométricamente mezcladas en un sólo archivo en formato BIL (formato de entrelazado de renglones). Los detectores de las bandas responden al nivel de luz solar reflejada por la superficie terrestre. Las señales analógicas obtenidas son muestreadas y digitalizadas para obtener los niveles de brillo (pixeles) los cuales varían entre 0 y 127 exepтуando la banda 7, que va de 0 a 63. Cada una de las bandas tiene una resolución espacial de 79 x 79 metros (Figura 6.1).

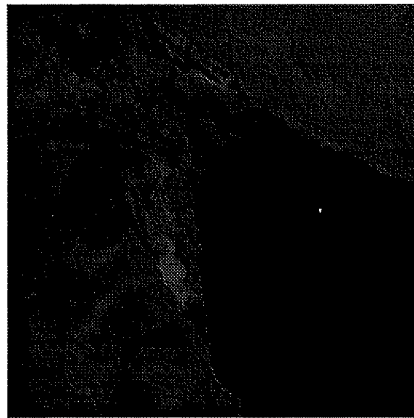


Figura 6.1 Banda espectral utilizada como entrada a la red (Banda 4).

6.2 Pruebas de clasificación.

Las pruebas se hicieron variando los parámetros que definen el comportamiento de la red. Estos son:

Parámetro de elección (α). Este parámetro es utilizado en la decisión de elección de categoría. Debe ser mayor que cero.

Tasa de aprendizaje (β). Este parámetro nos define la rapidez de aprendizaje de la red, es decir, las modificaciones que se harán a los pesos si se da un reconocimiento de una categoría.

Factor de vigilancia (ρ). Este factor indica la exigencia de la red para asignar una entrada a una categoría.

Número de neuronas en la capa de salida. La capacidad que tendrá la red para asignar puntos a estas categorías.

Durante las primeras pruebas realizadas se obtuvieron resultados muy deficientes, ya que la red no reconoció las clases espectrales en la imagen. Esto debido a que al iniciar el reconocimiento de la imagen (pixel por pixel) cada uno de los puntos que inicialmente va utilizando para su clasificación es colocado en una clase de salida diferente (neurona de la capa de salida). Por lo tanto, al ir encontrando más clases diferentes ya no tiene lugar para asignarla, esto es, la memoria de la red se ha agotado.

Este problema fue denominado por Carpenter y Grossberg como *proliferación de categorías* [Carpenter y Grossberg, 1991], y consiste en el agotamiento inmediato de todas las unidades en la capa de salida impidiendo que la red pueda reconocer nuevas clases espectrales para un nuevo patrón de entrada, esto es, esa entrada no es reconocida.

Con la finalidad de resolver este problema se modificó el algoritmo de Fuzzy ART para incorporar la normalización de entradas o codificación complemento y por otro lado la opción de compromete-rápido recodificación-lenta.

6.2.1 Opción de normalización de la entrada

Moore [Moore, 1989] describió un problema de *proliferación de las categorías* que puede ocurrir en algunos sistemas ART continuos cuando un gran número de entradas

distorcionan la normal de los vectores de pesos. La proliferación de las categorías es eliminada en Fuzzy ART si las entradas son normalizadas. Una regla de normalización sugerida, llamada **Codificación complemento**, logra la normalización mientras se preserva la amplitud de la información.

Codificación complemento representa la *respuesta on* y la *respuesta off* para representar un patrón de entrada. Las celdas *on* en el prototipo nos dan las características que están presentes en el ejemplar, mientras que las celdas *off* nos dan las características que están ausentes. Para definir esta operación en una forma sencilla, decimos que **a** representa una respuesta *on*. El complemento de **a**, denotado por **a^c**, representa la respuesta *off*, donde

$$a_i^c \equiv 1 - a_i$$

La entrada **I** para la codificación complemento es el vector de dimensión $2n$.

$$\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) \equiv (a_1, \dots, a_n, a_1^c, \dots, a_n^c)$$

Note que

$$|\mathbf{I}| = |(\mathbf{a}, \mathbf{a}^c)|$$

$$= \sum_{i=1}^n a_i + (n - \sum_{i=1}^n a_i)$$

$$= n$$

La opción de normalización de las entradas (II) es agregada a los parámetros que definen la red.

6.2.2 Opción de compromiso rápido, recodificación lenta.

Para una codificación eficiente de conjuntos con entradas ruidosas se usa un $\beta = 1$, cuando J es un nodo no comprometido. Entonces se toma $\beta < 1$ después de que la categoría es elegida. Entonces $W_J^{(\text{nuevo})} = I$ la primera vez que la categoría J es activada y cada vez que se asigna otra entrada a esta categoría se va decrementando la tasa de aprendizaje un $\Delta\beta$ hasta llegar a 0.

El decremento de la tasa de aprendizaje ($\Delta\beta$) es tomado como un nuevo parámetro para definir el comportamiento de la red.

6.3 Resultados.

Algunos de los resultados obtenidos de las pruebas realizadas variando los parámetros definidos en el capítulo anterior son mostrados en esta sección.

Para cada uno de los parámetros se tiene una tabla donde se varía su valor.

Tabla III. Variación del factor de vigilancia ρ .

I	ρ	β	$\Delta\beta$	α	# de clases encontradas
si	0.8	1.0	0.1	0.5	3
si	0.85	1.0	0.1	0.5	4
si	0.90	1.0	0.1	0.5	6
si	0.95	1.0	0.1	0.5	6
si	0.995	1.0	0.1	0.5	12

a) De la tabla III notamos que la red es muy susceptible al cambio del factor de vigilancia, ρ , ya que con pequeñas modificaciones a este parámetro se obtuvieron resultados muy diferentes. Para valores cercanos a 1 la red obtuvo más clases que para valores más pequeños. Para clasificaciones obtenidas el mejor ρ fue de 0.95.

Tabla IV. Variación de la tasa de aprendizaje β .

I	ρ	β	$\Delta\beta$	α	# de clases encontradas
si	0.95	0.80	0.1	0.5	3
si	0.95	0.90	0.1	0.5	6
si	0.95	0.90	0.0	0.5	-
si	0.95	1.0	0.1	0.5	7

b) La tasa de aprendizaje inicial para los nodos β_j con la que se obtuvieron mejores resultados fue de 1, es decir, iniciar con un aprendizaje rápido y cada vez que un pixel sea asignado a la clase j decrementar su correspondiente β_j .

Tabla V. Variación de $\Delta\beta$.

I	ρ	β	$\Delta\beta$	α	# de clases encontradas
si	0.95	1.0	0.0	0.5	-
si	0.95	1.0	0.1	0.5	7
si	0.95	1.0	0.3	0.5	2
si	0.95	1.0	0.5	0.5	3

c) La opción de rápido aprendizaje y recodificación lenta es indispensable para que se realice la clasificación ya que en las pruebas realizadas sin esta opción no se logró clasificar la mayor parte de la imagen. El mejor $\Delta\beta$ se encontró en el rango de [0.05 - 0.1]. Para valores más grandes la red identifica muy pocas clases y no se obtiene una buena clasificación.

Tabla VI. Variación del parámetro de elección α .

I	ρ	β	$\Delta\beta$	α	# de clases encontradas
si	0.95	1.0	0.1	0.005	7
si	0.95	1.0	0.1	0.5	7
si	0.95	1.0	0.1	5.0	7
si	0.95	1.0	0.1	20.0	8
si	0.95	1.0	0.1	40.0	8

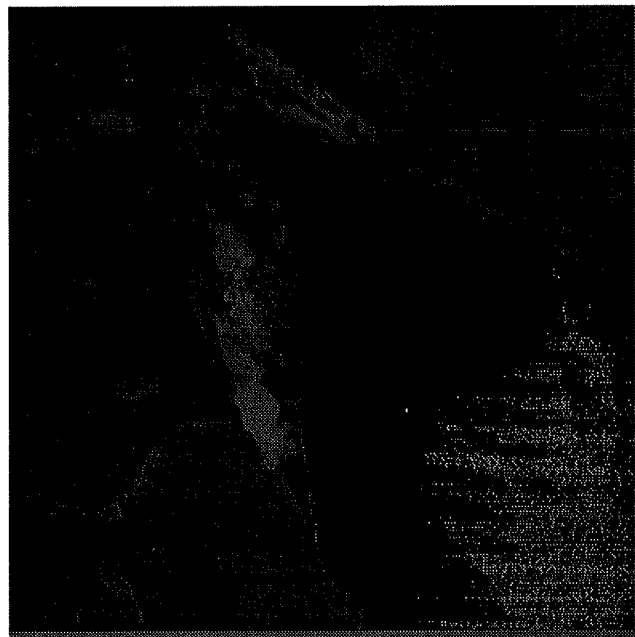
d) La variación del parámetro de elección, α , no tuvo un efecto importante en los resultados obtenidos, ya que con grandes variaciones de este parámetro ($0 < \alpha \leq 10$) se tienen resultados muy semejantes. Se recomienda que el valor de este parámetro sea cercano a 0.

e) Con la opción de codificación complemento (normalización de la entrada, IIII) se identificó un mayor número de clases manteniendo los otros parámetros fijos. Por otra parte, esta normalización da un efecto de suavizado a la imagen clasificada que define mejor las clases, eliminando ruido.

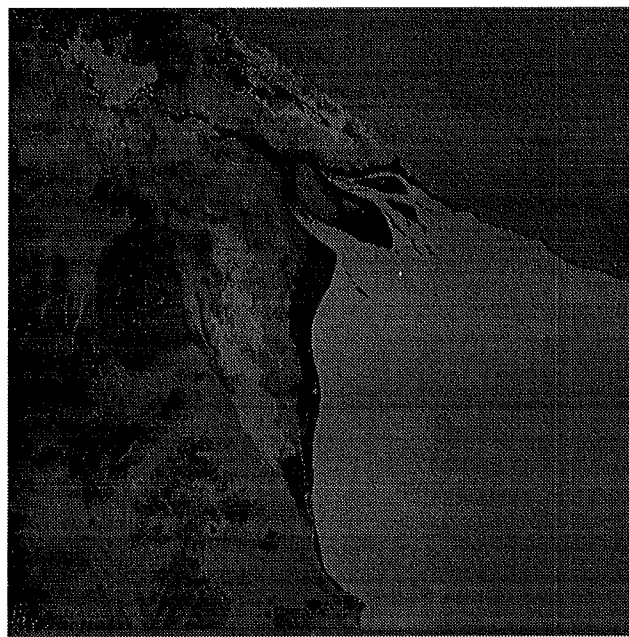
La figura 6.2 muestra algunas de las mejores imágenes obtenidas de la clasificación. Estas imágenes utilizan α de 0.5 e inician con un rápido aprendizaje disminuyendo esta tasa un $\Delta\beta = 0.1$. En (a) no se normaliza la entrada y tiene un ρ de 0.95. En (b) si se normaliza la entrada y tiene un ρ de 0.995.

Tabla VII. Resultados obtenidos sin la normalización de la entrada.

I	ρ	β	$\Delta\beta$	α	# de clases encontradas
no	0.95	1.0	0.0	0.5	-
no	0.90	1.0	0.0	0.5	-
no	0.95	0.9	0.0	0.5	-
no	0.95	1.0	0.1	0.5	12
no	0.9	1.0	0.1	0.5	8



(a)



(b)

Figura 6.2 Resultado de una prueba de clasificación con el algoritmo FuzzyART. En (a) se tiene un valor para p de 0.95 y no se normaliza la entrada, se encontraron 12 clases. En (b) se tiene un p de 0.995 y si se normaliza la entrada. En ambos se tiene un α de 0.5, un β de 1.0 y un $\Delta\beta$ de 0.1.

7. Conclusiones y recomendaciones.

7.1 Conclusiones.

Los resultados obtenidos mediante el algoritmo FuzzyART fueron contrastados con los obtenidos por [Torres, 1994] utilizando clasificadores tradicionales (Figura 7.1). Ambos resultados muestran patrones de distribución espacial muy semejantes para las cubiertas de terreno en la región deltaica del Río Colorado. En base a esto se puede concluir que las redes neuronales representan una buena alternativa a los clasificadores tradicionales ya que muestran ciertas ventajas sobre de ellos, como son:

a) Su fácil uso. Se tiene un control del comportamiento de la red ya que los parámetros que la definen pueden ser modificados de manera sencilla e intuitiva.

b) La robustés de sus parámetros. Las redes neuronales permiten como entrada información heterogénea (alturas, pendientes, fuentes múltiples) sin requerir una especificación previa de cada una de las fuentes.

c) Oportunidad de paralelismo. Debido a su arquitectura es muy sencillo lograr paralelizar las operaciones de las redes neuronales logrando reducir el tiempo de cómputo.

En particular, la red neuronal Fuzzy ART tiene la gran ventaja de ser un sistema autónomo, capaz de continuar aprendiendo nuevos patrones sin olvidar los que ha aprendido en el pasado. Respondiendo rápidamente a patrones presentados anteriormente y reconociendo como un nuevo patrón a una entrada nunca antes presentada.

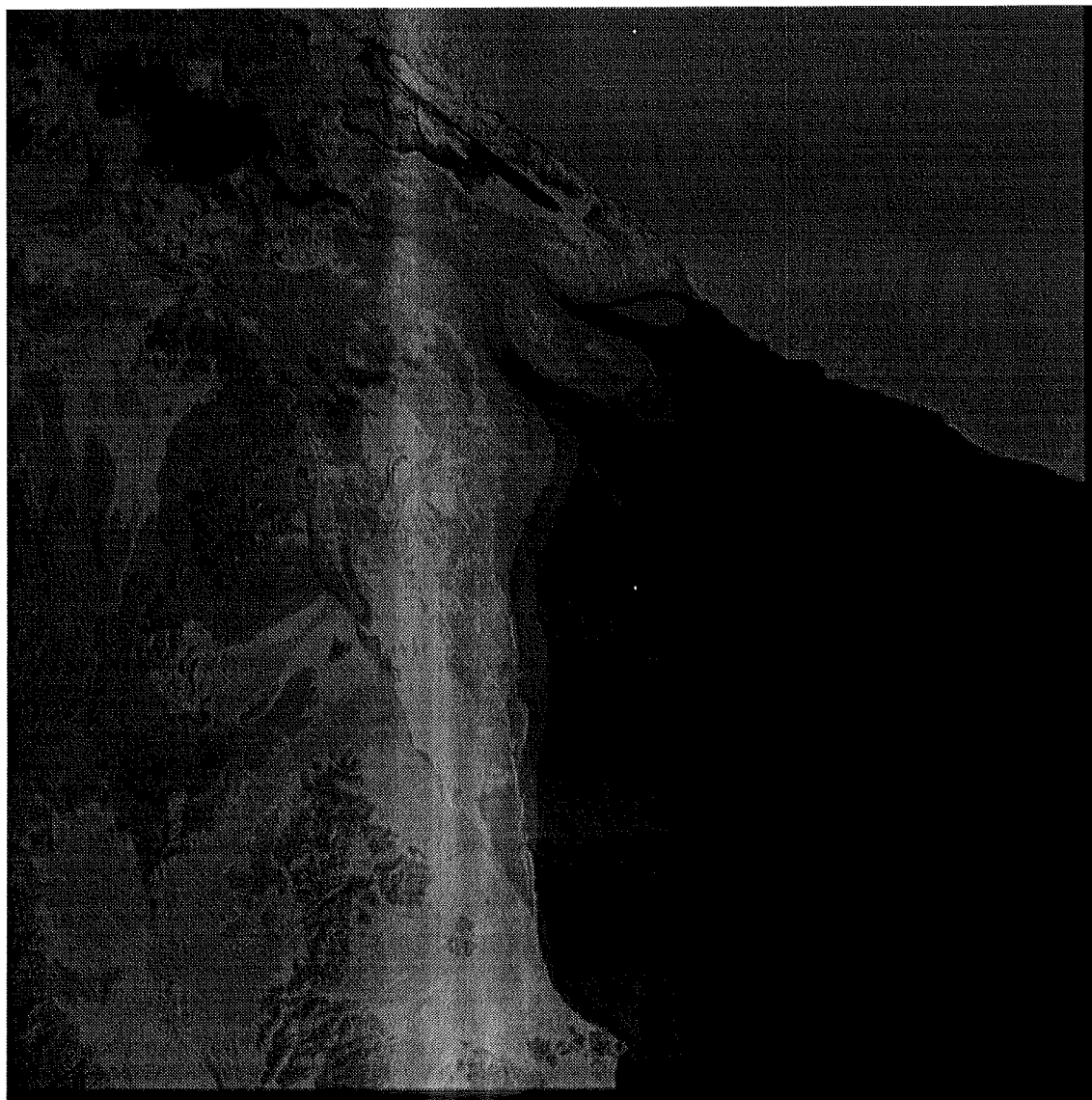


Figura 7.1 Prueba de clasificación con métodos tradicionales. Como entrada se tiene un archivo en formato BIL de 8 bandas las 4 de verano (utilizadas con la red neuronal) más 4 de invierno, tomadas por el sensor LANDSAT MSS de la zona deltaica del Río Colorado.

7.2 Aportaciones.

Se demostró que FuzzyART puede ser usado con gran éxito en la clasificación de imágenes multiespectrales y la utilizad de las operaciones opcionales sugeridas por sus creadores.

Se realiza un análisis de la influencia de los parámetros que definen la red FuzzyART, logrando tener un buen control de su comportamiento.

Se desarrolló el sistema NeuroClas, el cual es muy sencillo de manejar y puede ser utilizado por cualquier persona que esté interesada en la clasificación de imágenes sin importar el tamaño de sus imágenes y la fuente de sus datos. Este sistema puede ser solicitado para su uso por correo electrónico a **mmeza@cicese.mx**.

7.3 Recomendaciones.

El sistema desarrollado para la clasificación de imágenes con las redes de Retropropagación y Fuzzy ART está en su primera etapa de implementación, es decir, existen muchas técnicas de procesamiento digital que pueden ser añadidas al sistema, ya sea para el preprocesamiento aplicando filtros a las imágenes para eliminar ruido o problemas de bandeo típicos en las imágenes de satélite. Así como técnicas de postprocesamiento, para la evaluación de las clases obtenidas por la clasificación. Además el sistema puede incluir mejoras a la interfaz con el usuario para permitir operaciones de visualización como agrandar o reducir la imagen (zoom's), creación de paletas de colores,

un encabezado de la imagen más completo conteniendo posiblemente las operaciones realizadas a la imagen (filtros, técnicas de histogramas, cambio de paleta, etc).

Las operaciones de clasificación pueden ser paralelizadas, de esta manera el tiempo que consume esta tarea se reduciría considerablemente.

Además de la clasificación de imágenes las redes neuronales como hemos visto en el capítulo 2 pueden ser implementadas para una gran variedad de aplicaciones. En medicina están siendo utilizadas para la detección de enfermedades como tumores o esquizofrenia [Gross, 1994] a partir de imágenes tomadas por rayos X en distintas zonas del cerebro.

8. Referencias citadas.

- Benediktsson, J. A, Swain, P. H. y Ersoy, O. K. 1990. "Neural network approaches versus statistical methods in classification of multisource remote sensing data" *IEEE Trans. Geosci. Remote Sensing*, 28: 540-551.
- Carpenter, G. y Grossberg, S. 1987a. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37: 54-115.
- Carpenter, G. y Grossberg, S. 1987b. ART 2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26: 4919-4930.
- Carpenter, G. y Grossberg, S. 1988. The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *Computer*, 21: 169-181.
- Carpenter, G. y Grossberg, S. 1990. ART3 hierarchical search: Chemical transmitters in self-organizing pattern recognition architectures. *International Joint conference on Neural Networks (Washington, DC) II*: 30-33.
- Carpenter, G., Grossberg, S. y Reynolds, J. H. 1991a. ARTMAP: A self-organizing neural network architecture for fast supervised learning and pattern recognition. *International Joint conference on Neural Networks (Seattle) I*: 863-868.

- Carpenter, G. y Grossberg, S. 1991b. Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks*, 4: 759-771.
- Fausett, L. 1994. Fundamentals of neural networks, architectures, algorithms and applications. Englewood Cliffs, NJ: Prentice-Hall.
- Favela, J. Torres, J., Hidalgo, H. y Granillo, R. 1994. Load Balancing for Neural-Network Classification of Remote Sensing Data in a Heterogeneous Network of Workstations. 7th International conference on Parallel and Distributed Processing.
- Freeman J. y Skapura, M., 1994. Neural networks, algorithms, applications, and programming techniques. Addison Wesley Publishing Company.
- Fukushima, K., Miyake, S. e Ito T. 1983. Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 13: 826-834.
- Gross, M. 1994. Subspace methods for the visualization of multidimensional data sets. *Scientific visualization*. 171-185.
- Grossberg, S. 1976. Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, 23: 187-202.

Grossberg, S. 1982. *Studies of Mind and Brain. Volume 70 of Boston Studies in the Philosophy of Science.* D. Reidel Publishing Company.

Hara, Y., Atkins, R., Yueh, S., Shin, R. y Kong, J. A. 1994. Application of Neural Networks to Radar Image Classification. *IEEE Trans. Geosci. Remote Sensing*, 32(1):100-109.

Hebb, D. O. 1949. *The Organization of Behavior.* New York: John Wiley & Sons. Introduction.

Hecht-Nielsen, R. 1987. Counterpropagation Networks. *Applied Optics*, 26(23): 4979-4984.

Heerman, P. D. y Khazenie, N. 1992. "Classification of multispectral remote sensing data using back-propagation neural network", *IEEE Trans. Geosci. Remote Sensing*, 30:81-88.

Hinton, G. E. y Sejnowski, T. J. 1983. Optimal Perceptual Inference. Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition, Washington DC.* I:448-453.

Jones, O. 1989. *Introduction to the X Window system.* Prentice Hall, Englewood Cliffs, New Jersey.

- Kohonen, T. 1989. Self-Organization an associative memory (3rd ed.), Berlin: Springer-Verlag.
- Kosko, B. 1988. Bidirectional associative memories. *IEEE Transactions on Systems, man, an Cybernetics*, 18: 49-60.
- Moore, B. 1989. ART1 and pattern clustering. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 Connectinist Models Summer School*. 174-198. San Mateo, CA: Morgan Kaufmann Publishers.
- Nelson, M. y Illingworth, W. T. 1991. *A Practical Guide to Neural Nets*. Addison-Wesley Publishing Company, Inc.
- Parker D. 1985. Learning Logic. Technical Report TR-87, Cambridge, MA: Center for computational Research in Economics and Management Science, MIT.
- Richards, J. 1986. Remote sensing digital image analysis, Springer-Verlag.
- Rosenblatt, F. 1958. The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65: 386-408.
- Ruiz, D. 1995. Redes neuronales y su aplicación on la clasificación de imágenes. Trabajo de titulación, UABC.

Rumelhart, D. E., Hinton J. L. y Williams R. J. 1986. Learning Internal Representations by Error Propagation. *Parallel Distributed Processing*, vol. 1 chapter 8.

Werbos, P. (1974). *Beyond Regression: New Tools For Prediction and Analysis in the Behavioral Sciences* (Ph. D. thesis). Cambridge, MA: Harvard U. Committee an Applied Mathematics.

Widrow, B. y Hoff, M. E. Jr. (1960). Adaptive Switching Circuits. IRE WESCON Convention Record, part. 4, pp. 96-104.

Apéndice A. Descripción del sistema NeuroClas.

El sistema NeuroClas está compuesto por varios programas escritos en lenguaje C que hacen llamadas a Xlib. Cada uno de estos programas contiene funciones específicas que pueden ser utilizadas por otras funciones.

Se tiene un programa *Make* que compila los programas por separado creando archivos objeto (".o") que serán encadenados para obtener el ejecutable.

El programa principal, NeuroClas.c, tiene las siguientes secciones:

#include, los archivos de encabezado incluidos definen funciones y tipos de datos predefinidos. Los archivos son: stdio.h, string.h, stdlib.h, malloc.h, X11/Xlib.h, X11/X.h, X11/cursorfont.h, X11/Xutil.h, X11/Xatom.h, "Estructuras.h"

Declaraciones, se declaran variables para el manejo del ambiente, identificadores de ventanas y botones, estructuras de ventanas e imágenes. Algunas de las principales declaraciones son:

```
Display  *mydisplay; //define el despliegue
int      myscreen;    //define la pantalla a utilizar
XFontStruct *font;    //Estructura para los fonts
```

```

Cursor    creloj,ccenter,
          chand, carrow; //variables para definir los cursores

EstVentana    wimagen,whisto,
              wtool,...    //Estructuras para las ventanas del sistema
EstImagen imagen    //Estructura para la imagen
int    idimg,idh,
        idp,...    //identificadores para las ventanas
int    idb,idf,
        ids,...    //identificadores para los botones
.
.

```

Inicialización, se abre una conexión de despliegue, se crean los cursores, las ventanas y los botones:

```

.
// se abre el despliegue
mydisplay = XOpenDisplay(" ");
// se define la pantalla
myscreen = DefaultScreen(mydisplay);
// se define el cursor
ccenter = XCreateFontcursor(mydisplay,XC_center_ptr);
.
.
// se crea la ventana principal
wtool = CreaVentana(0,0,BUTTON_WIDTH*2 + MAX_WIDTH + 24,
                  FontHeight, ccenter, "NeuroClas V 1.0", lightgrey);
.
.

```

```
// se crea el botón de Imagen

idimg = CreatePushButton(mydisplay, window, x, y, BUTTON_WIDTH,
                          height, gc, font, "Imagen", IniciaImg);
```

```
.
```

Maapeo de ventanas, Cuando queremos hacer visible en pantalla una ventana invocamos:

```
.
```

```
// visualizamos la ventana en pantalla
```

```
XMapRaised(mydisplay, wtool.w);
```

Ciclo principal para la lectura de eventos, el ciclo principal lee un evento y chequea si fue en alguna de las ventanas de NeuroClas:

```
.
```

```
while(flag)
```

```
{
```

```
    // lee el siguiente evento
```

```
    XNextEvent(mydisplay, &myevent);
```

```
    //primero chequea si fue en alguno de los botones
```

```
    ToolEvent(&myevent);
```

```
    // chequea si fue en alguna de las ventanas del sistema
```

```
    switch(myevent.xany.window)
```

```
    {
```

```

// en la ventana de la imagen?
case wimagen.w:      EventosImg(idimg,&myevent);
                    break;

// en la ventana del histograma?
case whisot.w: EventosHisto(idh,&myevent);
                    break;

// en la ventana de la imagen clasificada por Retropropagación
case wclasback.w:    EventosImg(idbc,&myevent);
                    break;

// en la ventana de la imagen clasificada por Fuzzy ART
case wclasfuzzy.w:   EventosImg(idfc,&myevent);
                    break;

// en la ventana de la imagen principal
case wpaleta.w:      EventosImg(idp,&myevent);
                    break;
}
}
:
:

```

Las funciones llamadas en este ciclo controlan los eventos de la ventana correspondiente. Una función de eventos contiene lo siguiente:

```

// tipo de evento ocurrido
switch (myevent -> type)
{

```

```
//Repinta la ventana en un evento de expose
case Expose:
    ...
    break;

// Procesa las presiones del botón del mouse
case ButtonPress:
    ...
    break;

// Procesa los movimientos del mouse
case MotionNotify:
    ...
    break;

//Procesa si se libera un botón del mouse
case BottonRelease:
    ...
    break;
}

.
```

Terminación, se destruyen los recursos y se cierra el canal de despliegue:

```
.
```

```
// se destruye la ventana principal
```

```
XDestroyWindow(mydisplay,wtool.w);  
  
// se destruye la ventana de la imagen principal  
  
XDestroyWindow(mydisplay,wimagen.w);  
  
// se libera el mapa de colores  
  
XFreeColormap(mydisplay,mapa);  
  
// se cierra el despliegue  
  
XCloseDisplay(mydisplay
```

Apéndice B. Funcionamiento de Neuroclas.

Este sistema trabaja en estaciones de trabajo con sistema operativo UNIX.

NeuroClas fue diseñado con un ambiente de ventanas y botones que se manejan con el ratón de manera que la interacción con el usuario sea más sencilla.

Se tiene una ventana principal (Figura 5.7) la cual contiene los procedimientos que el sistema puede operar. Se crean ventanas para el despliegue de la imagen, las cuales tienen las características de las ventanas del ambiente en el cual se encuentra corriendo el sistema. Además se tienen ventanas de dialogo (Figura 5.8) donde se pedirán los datos requeridos para los procesamientos, así como nombres de archivos, etc.. Para dar un dato es necesario presionar un botón del ratón dentro del espacio para escritura y mantener en ese espacio el ratón mientras que el dato sea introducido. Si se presenta un error en el sistema se despliega una ventana de mensajes que indica el tipo de error detectado.

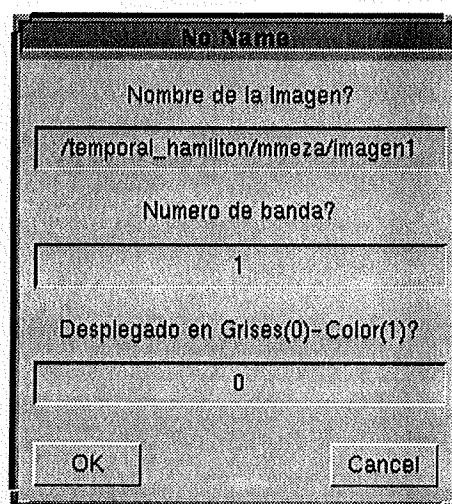
Las operaciones del sistema son:

- Opción de imagen.
- Opción de Histograma.
- Opción de Salvar Histograma.
- Opción de Paleta.
- Opción de Suavizado.
- Opción de Clasificación con Retropropagación.

- Opción de Clasificación con Fuzzy ART.

- **Opción de Imagen.**

Al oprimir el botón de *imagen* el usuario podrá desplegar una imagen en una ventana. Mediante una caja de dialogo (Figura 9.1), el usuario especificará el nombre del encabezado de la imagen, el número de banda a desplegar y si el despliegue de la imagen será en color o en tonos de gris.



The dialog box has a title bar that says "No Name". It contains three text input fields. The first field is labeled "Nombre de la Imagen?" and contains the text "/temporal_hamilton/mmeza/imagen1". The second field is labeled "Numero de banda?" and contains the number "1". The third field is labeled "Desplegado en Grises(0)-Color(1)?" and contains the number "0". At the bottom of the dialog box are two buttons: "OK" and "Cancel".

Figura B-1 Obtención de los datos necesarios para el despliegue de una imagen.

El nombre del encabezado de la imagen debe de ser el mismo que el de la imagen más una terminación .etc, este archivo debe de contener las dimensiones de la imagen, largo y ancho, el número de bandas y el número de colores o tonos de la misma, en el orden descrito.

La imagen que se desea desplegar debe de estar en formato BIL (formato de entrelazado de renglones), esto es, las n bandas de la imagen están entrelazadas en un solo archivo.

Cuando se tiene una imagen desplegada y se desea ver otra imagen, es necesario despresionar el botón de imagen para quitar la imagen actual.

- **Opción de Histograma.**

Mediante esta opción se hace un preprocesamiento de la imagen desplegada, no esta por de mas mencionar que se debe de tener una imagen desplegada para poder utilizar esta opción.

El manejo de histograma se puede hacer de forma lineal, exponencial o logarítmica, según se desee y se especifique en la ventana de dialogo que aparecerá al oprimir el botón. Las opciones se especifican por un número donde el 0 es lineal, 1 es exponencial y 2 logarítmico.

Apareciendo la ventana del histograma, este puede ser manejado de forma interactiva por medio de puntos de control, que en el caso del histograma lineal (Figura 9.2), se encuentran en los extremos de las rectas (se contemplan tres rectas), y para el exponencial y el logarítmico se encuentra en el centro.

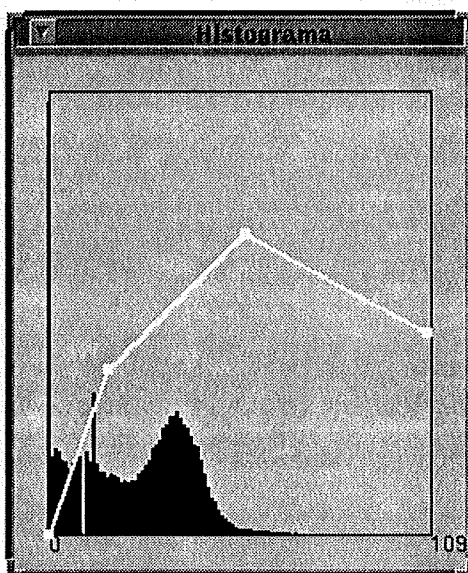


Figura B-2 Manejo de histograma lineal. Se tienen puntos de control que forman las rectas para la transformación.

Para seleccionar los puntos de control se debe de posicionar el cursor sobre los mismos, el cambio de cursor a una forma de $\leftrightarrow$, nos indicará que nos encontramos en la zona en la que podemos mover el punto, presionando un botón del ratón se podrá realizar el movimiento de dicho punto. El ajuste de la imagen de acuerdo al movimiento de los puntos de control será visto en forma automática. Al despresionar el botón del ratón los puntos de control dejarán de ser ajustados.

Para quitar la ventana del histograma sólo es necesario despresionar el botón de *histograma*. El ajuste hecho a la imagen no será salvado por medio de este botón, así que si se desea salvar este cambio es necesario presionar el botón de *Salva Histo* antes de quitar la ventana del histograma.

- **Opción de Salvar Histograma.**

Por medio de esta opción, se permite al usuario salvar los ajustes hechos en la imagen por medio del histograma. Los ajustes se realizan únicamente en la banda desplegada, no en las n bandas. Para poder utilizar esta opción, es necesario tener la ventana del histograma activa (desplegada).

- **Opción de Paleta.**

Esta opción nos permite cambiar la paleta de la imagen desplegada. La imagen debe estar en colores, de otra manera el cambio no podrá ser apreciado.

Al oprimir este botón, aparecerán dos ventanas, en la primera de ellas se tendrá la gama de colores disponibles (ventana de paleta). En la segunda ventana (ventana de características) se desplegará información acerca de las coordenadas del punto y de su valor numérico en la imagen (Figura 9.3). El despliegue de características será visible solo cuando se mueva el cursor dentro de la ventana que contiene a la imagen a la que se desea cambiar de paleta.

Para realizar el cambio de paleta, se deberá de presionar el botón del ratón sobre un punto de la imagen, el cual debe ser del color que se desea cambiar. A continuación, se deberá de presionar un botón del ratón sobre el nuevo color elegido.

Esta opción puede ser aplicada sobre la ventana de la imagen principal y sobre las

ventanas de las imágenes clasificadas.

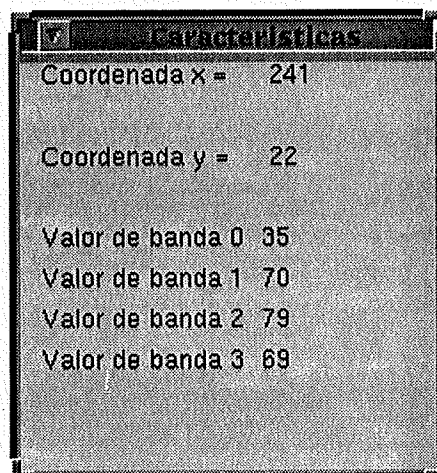


Figura B-3 Características de los puntos de la imagen.

- **Opción de Suavizado.**

Se tiene la opción de dar un suavizado a una imagen clasificada. Es necesario que la imagen sea cargada con la opción de *imagen*, y al presionar el botón de suavizado, se verá sobre la imagen desplegada un cambio, donde se definen mejor las clasificaciones de suelo.

- **Opción de Clasificación con Retropropagación.**

En esta opción se contemplan una serie de botones que nos permiten realizar el proceso de clasificación de una imagen de satélite por medio de una red neuronal de retropropagación. Las opciones de los botones son:

- Entrenar.
- Clasificar.
- Puntos.
- Leer Red.
- Salvar Red

- Entrenar.

En esta opción se crea y entrena la red neuronal. Al presionar el botón aparecerá una ventana de dialogo (Figura 9.4), en la cual se nos piden los parámetros necesarios para entrenar la red. Primeramente debemos de especificar el número de capas que contendrá la red, como mínimo se debe de especificar tres, la capa de entrada, la capa de salida y al menos una capa interna. A continuación se nos piden el número de neuronas por cada capa interna, por ejemplo, si la red va a tener cuatro capas, se debe de especificar el número de neuronas para las dos capas intermedias, estos número deberán ir separados por espacios, sin ningún símbolo entre ellos (comas, puntos,etc), así podemos poner :

5 6 6 4 9 etc.

A continuación se piden los parámetros de aprendizaje, ruido y momentum, los cuales deben de ser menor que 1.

Dialog box titled "No Name" with the following fields and values:

- Numero de Capas de la red?
- Neuronas por c/capa interna?
- Tasa de aprendizaje? 0.5
- Momentum? 0.3
- Ruido? 0.05
- Nombre del archivo de puntos?
- Numero de ciclos de entrenamiento 5000

Buttons: OK, Cancel

Figura B-4 Ventana de diálogo para introducir los parámetros necesarios para entrenar la red de Retropropagación.

Se pide el nombre del archivo que contiene los puntos de entrenamiento para la red. Este archivo debe de contener el número de puntos por cada clase, el numero de características de entrada y el numero de clases de salida esperada, así como el nombre de la clasificación, y los pares de entrenamiento correspondientes.

Por último se pide el número de ciclos de entrenamiento para la red.

- Clasificar.

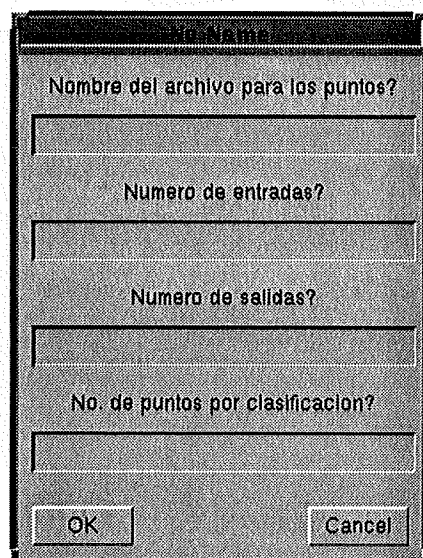
Por medio de este botón se inicia la clasificación de la imagen. Cabe mencionar que para poder realizar la clasificación debe tenerse una imagen desplegada, así como una red activa. La red puede ser creada antes, por medio del botón de entrenamiento o leída de disco por medio del botón de Leer Red.

Por medio de una ventana de dialogo se pide el nombre con el que la imagen clasificada será salvada en disco.

Durante la clasificación, se podrá ver el número de línea en la cual va el proceso, esto en la consola. Al termino de la clasificación, la imagen será desplegada en una ventana que contendrá además de la imagen, el nombre de las clasificaciones y el color asociado a dicha clasificación. Podrá aplicarse sobre esta ventana la opción de Paleta para cambiar los colores de las clasificaciones. El uso de la opción de paleta es descrito anteriormente.

- Puntos.

Mediante esta opción se crea el archivo con los puntos de entrenamiento para una red. Al presionar el botón, aparecerá una ventana de dialogo (Figura 9.5) en la cual se piden el número de características de entrada, el número de clasificaciones de salida esperado y cuantos puntos por clasificación se van a seleccionar, así como el nombre para dicho archivo.



The dialog box is titled "File Name". It contains four input fields, each preceded by a label: "Nombre del archivo para los puntos?", "Numero de entradas?", "Numero de salidas?", and "No. de puntos por clasificacion?". At the bottom, there are two buttons labeled "OK" and "Cancel".

Figura B-5 Diálogo para pedir los puntos de entrenamiento.

Para esta opción se despliega una ventana (Figura 9.3) en la cual se da información de las coordenadas de un punto así como su valor en cada una de las n bandas de la imagen. Esta información es desplegada al posicionar y mover el cursor sobre la ventana de la imagen principal.

Por medio de una ventana de diálogo se pide el nombre de la clasificación de la cual se van a seleccionar los puntos de entrenamiento.

Para seleccionar un punto de entrenamiento, dado el nombre de la clasificación a la cual pertenece, se debe de presionar el botón del ratón sobre el punto elegido y el patrón de entrenamiento para dicho punto será generado automáticamente y grabado en el archivo.

Cuando se completan el numero de puntos especificados para la clase actual, se procede

a pedir el nombre de la siguiente clase. Al completar el número de clases simplemente se oculta la ventana de características.

- Leer Red.

En esta opción se permite el leer una red ya entrenada de disco, la cual será entonces la red activa y con la que se realizarán las clasificaciones posteriores mientras no se lea otra red o se entrene una.

Mediante una ventana de diálogo se pide el nombre de la red que se quiere leer.

- Salvar Red.

En esta opción se permite almacenar la red activa en disco y así posteriormente permitir su uso, obteniéndola con el botón de Leer Red.

Por medio de una ventana de diálogo (Figura 9.6) se especifica el nombre con el cual será salvada la red.

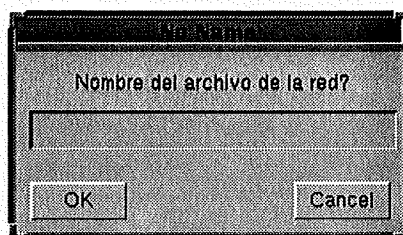


Figura B-6 Obtención del nombre para salvar la red activa.

- **Opción de Clasificación con Fuzzy ART.**

En esta opción se contemplan una serie de botones que nos permiten realizar el proceso de clasificación de una imagen de satélite por medio de una red neuronal Fuzzy ART. Se tienen diferentes operaciones para esta red, estas son:

- Crear Red.
- Clasificar.
- Salvar Red.
- Leer Red.

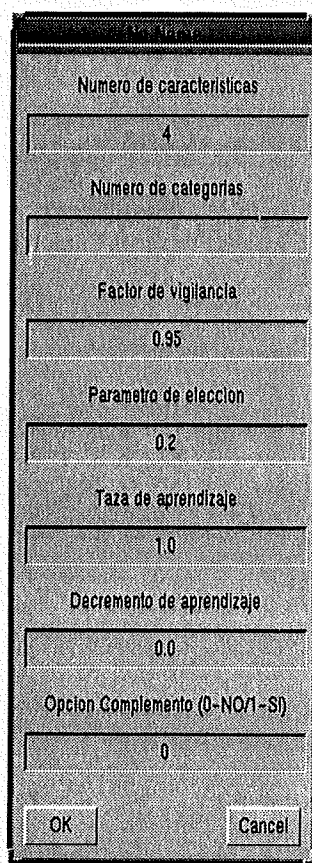
- **Crear Red.**

Antes de iniciar el proceso de clasificación debemos tener una red creada. Al presionar este botón aparece una caja de diálogo (Figura 9.7) donde se piden los valores iniciales para la estructura de la red, estos datos son: No. de atributos de entrada $n > 0$, No. de clases de salida $m > 0$, Factor de vigilancia $\rho \in [0,1]$, Parámetro de elección $\alpha > 0$, y Tasa de aprendizaje $\beta \in [0, 1]$.

- **Clasificar.**

Por medio de este botón se inicia la clasificación de la imagen. Cabe mencionar que

para poder realizar la clasificación debe tenerse una imagen desplegada, así como una red activa. La red puede ser creada antes, por medio del botón de entrenamiento o leída de disco por medio del botón de Leer Red.



The image shows a vertical dialog box titled "FuzzyART". It contains several input fields with labels and values:

- Numero de características: 4
- Numero de categorías: (empty)
- Factor de vigilancia: 0.95
- Parametro de eleccion: 0.2
- Taza de aprendizaje: 1.0
- Decremento de aprendizaje: 0.0
- Opcion Complemento (0-NOT-SI): 0

At the bottom, there are two buttons: "OK" and "Cancel".

Figura B-7 Diálogo para la obtención de los parámetros de la red neuronal FuzzyART.

Por medio de una ventana de dialogo se pide el nombre con el que la imagen clasificada será salvada en disco.

Durante la clasificación, se podrá ver el número de línea que se está clasificando en

consola. Al termino de la clasificación, la imagen será desplegada.

- **Salvar Red.**

Una vez que se ha completado la clasificación de la imagen, es posible salvar la estructura de la red que está activa, así como los valores de sus pesos. De este modo para clasificaciones posteriores se podrá utilizar una red que ya tiene un aprendizaje previo de tipos de suelo y lograr una clasificación más inmediata

Por medio de una ventana de diálogo (Figura 9.6) se especifica el nombre con el cual será salvada la red.

- **Leer Red.**

En esta opción se permite el leer una red que contiene un aprendizaje almacenado, la cual será entonces la red activa y con la que se realizaran las clasificaciones posteriores mientras no se lea otra red o se cree una.

Mediante una ventana de dialogo se pide el nombre de la red que se quiere leer.