53

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

APLICACIÓN DE LA ESTADÍSTICA EN LA TOMA DE DECISIONES PARA EL RECONOCIMIENTO DE PATRONES MEDIANTE REDES NEURONALES.

T E S I S

QUE PARA OBTENER EL TÍTULO DE ACTUARIO

P R E S E N T A

XÓCHITL LÓPEZ GÓMEZ

DIRECTOR: DR. JORGE ANTONIO ASCENCIO GUTIÉRREZ

2001

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

MAT. MARGARITA ELVIRA CHÁVEZ CANO
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis:
Aplicación de la Estadística en la Toma de Decisiones para el Reconocimiento de Patrones Mediante Redes Neuronales

realizado por Xóchitl López Gómez

con número de cuenta 8838745-1 , pasante de la carrera de Actuaría.

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis Propietario	Dr. Jorge Antonio Ascencio Gutiérrez.
Propietario	M. en A. P. María del Pilar Alonso Reyes.
Propietario	Dr. Alipio Gustavo Calles Martínez.
Suplente	Mat. Mario Delgadillo Torres.
Suplente	Act. José Guadalupe Vázquez Vázquez.

Consejo Departamental de Matemáticas.

M. en C. José Antonio Flores Díaz.

A mis padres, maestros
y amigos que me han
ayudado a ser mejor

Índice

Introducción

El hombre a través de los tiempos ha pensado en la forma de cómo resolver los problemas de manera más rápida y eficiente. Una forma de hacerlo es crear una máquina que imite las funciones del cerebro humano.

En 1943 un neurobiólogo Warren Mc Culloch y un estadístico Walter Pitts publicaron un artículo titulado "A Logical Calculus of Inminent in Nervous Activity", el cual inspiró a muchos científicos a desarrollar la Inteligencia Artificial.

En 1956 se organizó la primera conferencia sobre la Inteligencia Artificial, en donde investigadores de todo el mundo dieron a conocer sus adelantos y discutieron algunos problemas que tenían. Posteriormente las graves limitaciones computacionales y tecnológicas de esa época ocasionaron el decaimiento de esta área. Con el surgimiento de las computadoras que trabajan en paralelo resurge el interés por las redes neuronales. En la actualidad se ha retomado el problema original el cual consiste en la Toma de Decisiones.

Las redes neuronales se pueden ocupar en diferentes áreas como son: ingeniería, finanzas, medicina, electrónica, etc, de manera especial se enfocará esta tesis a resolver el problema de reconocimiento de patrones de nanoestructuras de oro, el cual consiste en identificar entre un conjunto de patrones experimentales las diferentes clases y los rasgos característicos de cada clase.

Este problema surge debido a que en la actualidad los investigadores ocupan muchas horas para identificar a qué clase pertenecen y en algunas ocasiones es necesario utilizar otros métodos de caracterización como Difracción de Rayos X (XRD) y Microscopía Electrónica de Transmisión (TEM) lo que requiere ser especialista en diferentes técnicas.

Para ello se propone un sistema inteligente el cual sea capaz de realizar la clasificación de forma rápida y objetiva ocupando una modificación del modelo K-NN (K Nearest Neighborg) propuesto por Dudani. En esta red se

requiere de una estructura multicapa, a la cual se le asignan pesos aleatorios que se irán ajustando en el transcurso de aprendizaje. Durante este proceso se le proporcionan a la red imágenes de alta resolución de oro, primero simuladas y luego conjuntamente simuladas y experimentales.

Cuando se realiza este proceso se crea una función, la cual ocupando los rasgos característicos (número de lados, número de ángulos iguales, número de direcciones de líneas paralelas y algunas otras características) de las nanoestructuras, se le asigna al patrón desetiquetado, un lugar en el espacio de variables; posteriormente se toma en cuenta la cercanía de sus k vecinos, se asignan pesos, se suman estos considerando la clase a la que pertenecen, y se le asigna (patrón desetiquetado) la clase con mayor peso.

¿Qué es una red neuronal?

El hombre se ha dado cuenta de la gran ventaja que tendría al crear un sistema inteligente, que fuera capaz de aprender y reproducir tareas semejantes a las que se le han enseñado. Para ello se ha inspirado en el dispositivo de cálculo más complejo conocido por la humanidad, el pensamiento, el cual tiene la capacidad de resolver problemas en un tiempo corto.

Se puede decir que una red neuronal es una emulación muy simplificada de la forma de actuar de las neuronas humanas, en la cual se conectan entre si un conjunto de computadoras de manera similar a como se encuentran unidas las neuronas del cerebro.

En el cerebro humano la unidad fundamental es la neurona, quien tiene la capacidad de enviar información a través del axón y recibirla por medio de las dendritas.

El cerebro humano está constituido por aproximadamente 10 mil millones de neuronas que se encuentran densamente interconectadas de la siguiente manera: el axón se conecta a las dendritas por una unión conocida como sinapsis. La transmisión es un proceso químico en el cual la intensidad depende de la cantidad de químico liberado (figura 0.1).

Neurona

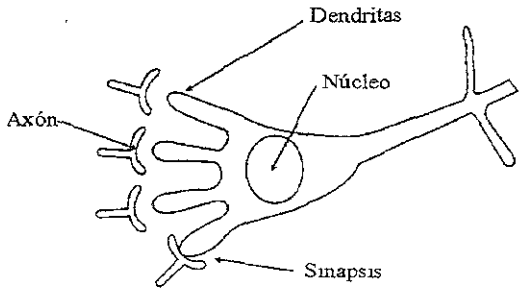

Figura 0.1

Varios investigadores han estudiado el funcionamiento del cerebro humano y han llegado a la conclusión que al cerebro humano le ayuda tener muchas conexiones entre las neuronas para la toma de decisiones, por la estructura al tomar una decisión, se realiza después de considerar varias alternativas.

A diferencia del cerebro humano, la unidad principal en una red neuronal artificial se llama elemento del procesamiento.

La red neuronal tiene por lo general muchas unidades de entrada pero una sola de salida, la cual puede aplicarse a otras unidades de entrada de la red. Está constituida por un conjunto de procesadores conectados adecuadamente según el problema que se desee resolver. Esquemáticamente cada procesador se representa por un nodo, y las conexiones por flechas que indican la dirección del flujo de la información de la red. Una red se encuentra constituida por varias capas, las cuales son un conjunto de elementos del procesamiento. Se considera que existen tres capas: la capa de entrada que es donde se presentan los datos a la red y se almacenan; la capa de salida, respuesta es donde se dan los datos de entrada, y la capa o capas intermedias entre la capa de entrada y de salida llamada capa oculta [1] (figura 0.2).

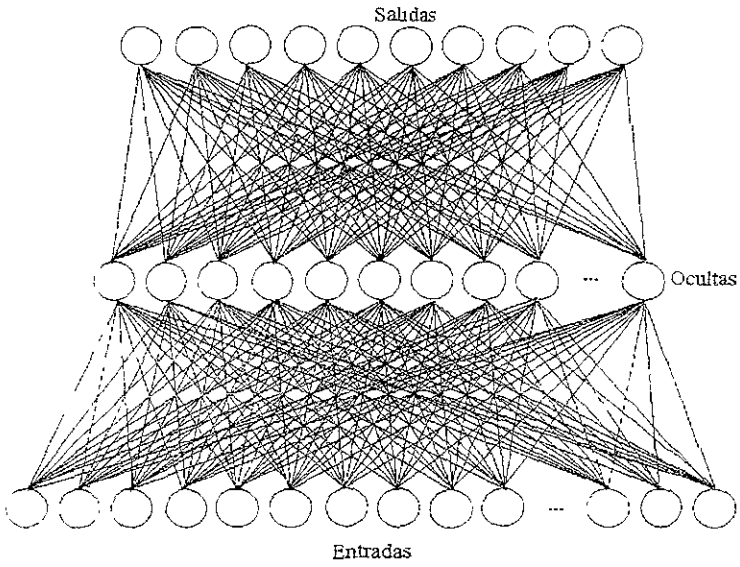

Figura 0.2

La notación es la siguiente:

Si se tiene la entrada i-ésima que procede de la j-ésima salida, se denotará como X_j. En este caso la intensidad se encuentra asociada al peso de la conexión.

El peso de la conexión procedente del j-ésimo nodo y que llega al i-ésimo nodo se denota mediante W_{ij} (figura 0.3).

A cada una de las capas se le representa en forma de vector, donde cada una se refiere a un elemento del procesamiento. Por ejemplo, si se tienen "n" procesadores en la capa de salida entonces se representa el vector mediante

$$X = \left(x_1, x_2, \ldots\ldots\ldots x_n\right)^t \qquad (0.1)$$

donde cada unidad representa una medición (variables, rangos, atributos, características e indicadores). Cada vector representa matemáticamente un punto en el espacio n dimensional.

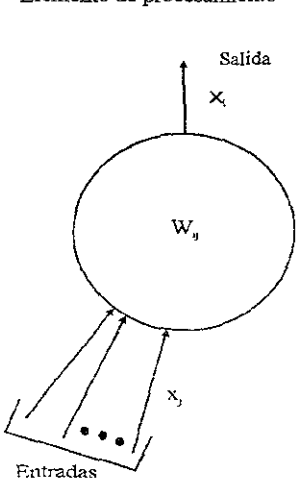

Figura 0.3

De manera similar se puede representar un vector de pesos. Supóngase que se tienen n nodos en una capa, los cuales van a una capa de m nodos, entonces se representara la i-ésima unidad de peso como (figura 0.4):

$$W_i = \left(w_{i1}, w_{i2}, w_{i3}, \ldots\ldots\ldots w_{in}\right)^t \tag{0.2}$$

donde el primer subíndice indica el nodo al que va a llegar la información, y el segundo subíndice del nodo del que provienen [2].

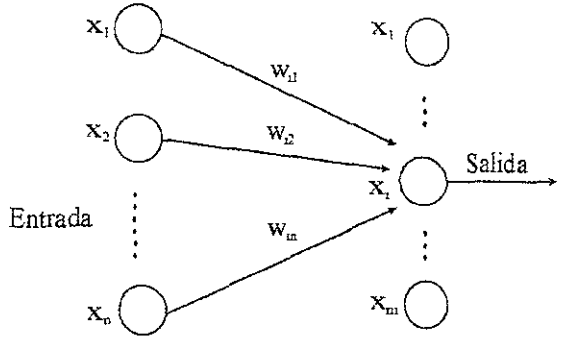

Figura 0.4

Se calcula el valor $neto_i$ de la siguiente manera:

Si se tienen n capas de salida, las cuales cada una provienen de m entradas (figura 0.5):

$$neta_j = \sum_j w_{ij} x_j \tag{0.3}$$

Con este cálculo se determina el valor de activación, el cual depende de la evaluación anterior de activación, por lo que se representa:

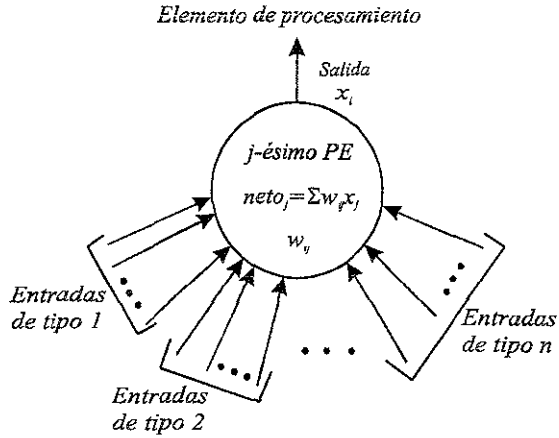

Figura 0.5

$$\alpha_i(t) = F_i(\alpha_i(t-1), neta_i(t)) \tag{0.4}$$

En un principio se le asigna a la red pesos aleatorios que se ajustan durante el proceso de entrenamiento, esto consiste, por lo general, en presentarle a la red varias veces un conjunto de patrones que sean representativos. Si la red ha sido entrenada adecuadamente será capaz, después en el proceso de funcionamiento, de identificar los rasgos característicos de la clase a la que pertenece el patrón que se le proporcione [3].

Taxonomía

Existen diferentes redes neuronales [4], las cuales toman decisiones de diferentes maneras. Se ha intentado agrupar éstas, ocupando sus características más importantes (figura 0.6). Lo primero que se examina es la clase de entrada y se tienen dos opciones, continua y discreta; posteriormente se considera el tipo de aprendizaje supervisado y aprendizaje no supervisado. En el aprendizaje supervisado durante el entrenamiento de la red, se ocupa la información de la clase a la que pertenece el patrón de entrada que se le proporciona, esto le ayuda a determinar los errores de clasificación con los cuales se hace un reajuste para luego realizar el proceso de retroalimentación.

Cuando se ocupa el aprendizaje no supervisado se le asigna a la red pesos aleatorios, después se le presenta un conjunto de patrones de entrada característicos con los cuales la red va ajustando pesos para realizar una clasificación correcta. En este proceso no es indispensable saber a qué clase pertenece cada patrón de entrenamiento.

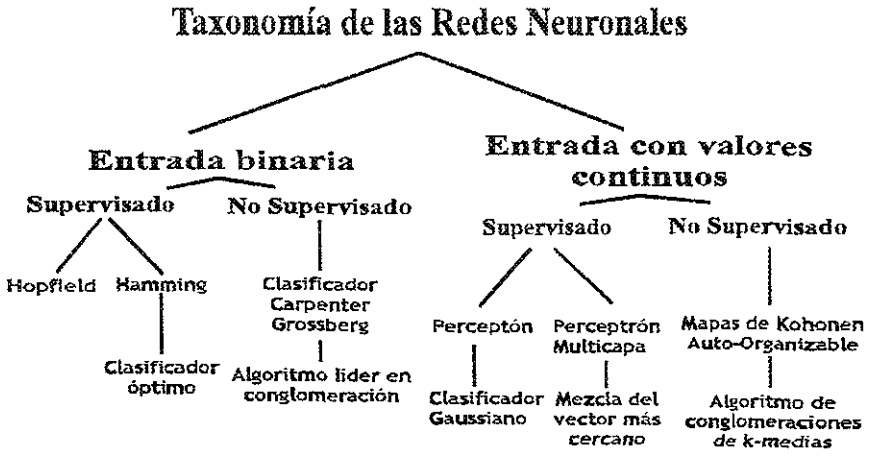

Figura 0.6

Es necesario presentarle varias veces el conjunto de patrones de entrenamiento, por lo que el aprendizaje es lento.

Actualmente se aplica el entrenamiento supervisado para solucionar problemas como reconocimiento de imágenes y de voz, control de motores y sobre todo en problemas de optimización.

El caso de estimación no paramétrico se ocupa en problemas de reconocimiento de patrones (voz, texto. imágenes y señales etc), codificación de datos, comprensión de imágenes y resolución de problemas de optimización como el del viajero.

Con lo anterior es posible concluir que una red neuronal es una emulación muy simplificada de la forma de actuar del cerebro humano.

La unidad fundamental en una red es el elemento de procesamiento.

La red tiene por lo general muchas unidades de entrada pero una sola de salida, ya que al cerebro humano le ayuda mucho el tener varias alternativas (unidades de entrada) para tomar la mejor decisión (unidad de salida). A las conexiones que existen entre los nodos se les asignan pesos aleatorios, los cuales se irán ajustando en un proceso denominado "de aprendizaje", donde se presenta varias veces a la red un conjunto de patrones de entrada y la red en este proceso, ajusta los pesos de conexión.

La notación es la siguiente; si se tiene la entrada i-ésima que procede de la j-ésima salida, denotada como X_j. En este caso la intensidad se encuentra asociada al peso de la conexión procedente del j-ésimo nodo y que llega al i-ésimo nodo W_{ij}.

Capítulo 1

1.1 Redes Neuronales

Existen distintos modelos dentro de las redes neuronales clásicas, éstos toman decisiones de diferente manera.

Los tres modelos que pueden resolver mejor el problema de reconocimiento de patrones son la red de Hopfield, Retropropagación y Kohonen, es por ello que a continuación se describirá la estructura, funcionamiento y aprendizaje, así como sus ventajas y limitaciones al realizar la clasificación de patrones experimentales de nanoestructuras de oro.

Estos tres modelos trabajan de distinta forma:

A la *red de Hopfield* se le proporcionan de manera previa los pesos de conexión, y en los otros dos modelos se permite que la red determine por sí misma cuales son los pesos de conexión más adecuados, durante un proceso denominado de aprendizaje.

La *red de Retropropagación* ocupa la derivada para seleccionar la dirección en la que se realizan los cambios de pesos, en comparación con la *red de Kohonen* que es una red competitiva que determina un ganador y ajusta los pesos de los patrones que se encuentran cerca.

Todos estos modelos pueden reconocer patrones que se encuentran distorsionados o con ruido; la red de Hopfield tiene la desventaja que si este patrón se encuentra trasladado o rotado no lo reconocerá, situación que no ocurre con los otros dos modelos.

Estas redes tienen diferentes ventajas y limitaciones cada autor propone una forma particular para realizar la toma de decisiones.

1.2 Hopfield

EL cerebro humano realiza muchas funciones importantes, entre ellas se encuentra la función de asociación, un ejemplo de ésta es la asociación que se

realiza al ver la cara de un amigo con su nombre.

La red de Hopfied tiene una gran importancia histórica, ya que con base en ella se impulsó el desarrollo de las demás pensando en la forma de mejorarla (con ello surgen redes como Backpropagation, Kohonen y otras).

Hopfield ha creado una red que se basa en la memoria qué es asociada [5]. El espacio de pesos de la red de Hopfield también denominado espacio de Hammington toma los valores de +1 ó -1, en cada uno de los elementos del espacio. Con ello se toman en cuenta dos características de interés y se representan mediante la siguiente expresión.

$$H^N = \left\{ X = (X_1, X_2, \ldots \quad X_n)^t \in \Re^n : X_i \in \left(\begin{matrix}+\\-\end{matrix} 1\right) \right\} \qquad (1.1)$$

Este espacio contiene 2^n puntos, si se tienen dos puntos en el espacio de Hammington se puede calcular su distancia. Tomando en cuenta primero la fórmula de la distancia euclidiana entre dos puntos

$$d = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \ldots (X_n - Y_n)^2} \qquad (1.2)$$

posteriormente como X_i y Y_i sólo toman el valor de +1 o -1 entonces $(X_i - Y_i)^2$ tomarán los valores 0 ó 4, es decir;

$$(X_i - Y_i)^2 = \begin{cases} 0 & X_i = Y_i \\ 4 & X_i \neq Y_i \end{cases} \qquad (1.3)$$

por lo que la distancia euclidiana se puede representar como

$$\begin{aligned} d &= \sqrt{4h} \\ d &= 2\sqrt{h} \end{aligned} \qquad (1.4)$$

Donde h = número de componentes distintos de X e Y (h se denomina distancia de Hamming).

Se le pide al modelo una restricción adicional, que consiste en que los vectores

X_i sean ortogonales entre sí, es decir, que los vectores sean los suficientemente diferentes. De lo contrario no se podrá asegurar una correcta asociación entre los vectores de entrada y de salida. Esta red tiene la característica de que siempre converge.

Se ocupa este modelo para solucionar problemas como el reconocimiento de voz, pero sobre todo para problemas de optimización, como es el caso del problema del viajante o encontrar el mínimo de una ecuación matemática, se ocupa también para la resolución de ecuaciones, así como para problemas del reconocimiento de imágenes; en este campo la red puede reconstruir imágenes distorsionadas pero tiene la limitante de que si la imagen se encuentra trasladada o rotada, la red no podrá seleccionar correctamente la imagen [6].

Cuando se desea encontrar el mínimo de una expresión matemática la red funcionará adecuadamente siempre y cuando la función sólo tenga un mínimo, de lo contrario la red no sabrá distinguir entre los mínimos local y el mínimo global.

1.2.1 Estructura

La estructura de esta red consiste en dos capas con el mismo número de unidades de entrada y de salida. Es decir con n-unidades de entrada y n-unidades de salida [7] (figura 1.1).

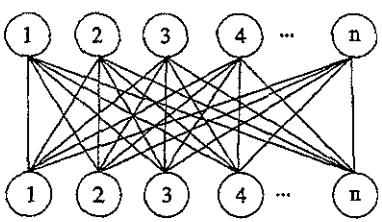

Figura 1.1

La estructura de esta red se puede reducir a una sola capa con n unidades. Existen dos estructuras de este tipo ligeramente diferentes, la primer estructura consiste en que cada una de las n unidades se encuentra interconectada con las n-1 unidades faltantes (figura 1.2).

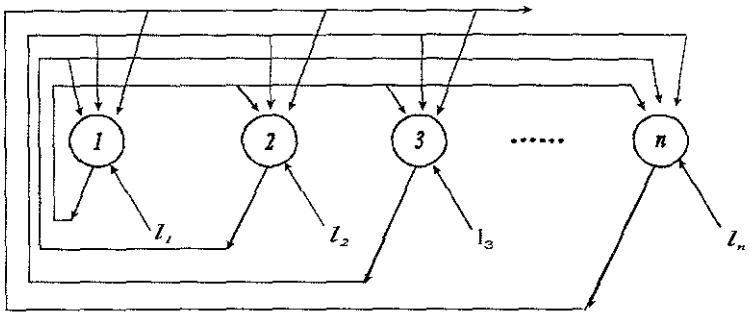

Figura 1.2

La segunda estructura se basa en que cada unidad se encuentra conectada no sólo con las n-1 unidades restantes sino también con ella misma (figura 1.3).

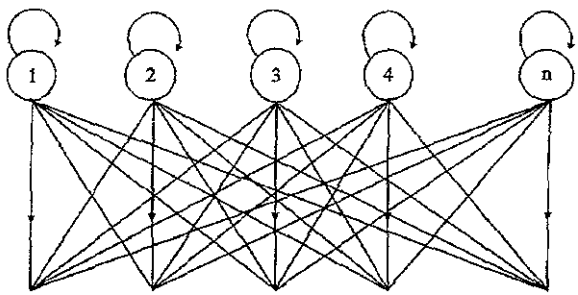

Figura 1.3

1.2.2 Convergencia

En la red de Hopfield es posible asegurar la convergencia del sistema.

Existe una forma para verificar que el sistema va a converger, para ello se definirá primero la función de Layapunov o función de energía la cual asegura que la función tenga límite.

Definición.

Se dice que una función es de Layapunov si es una función acotada, la cual disminuye el valor de la función en cada interacción.

Intuitivamente lo que hay que probar para asegurar la convergencia consta de

tres partes:

1.- Todo cambio de x a y durante el proceso BAM (Memoria Bidireccional Asociativa) da lugar a una disminución de E (energía).

2.- E posee un límite inferior que se encuentra dado por

$$E_{min} = -\sum_{ij} |w_{ij} \tag{1.5}$$

3.- Cuando cambia E debe de hacerlo en una cantidad finita

Se propone a la función E como una función de Layapunov. El inciso 1 asegura que la función E tome valores cada vez menores y en el inciso 2 se propone el límite. El inciso 3 asegura que el número de pasos para encontrar el límite sea finito y no se tenga una disminución pequeña de la función E, la cual haga que se necesite un número de pasos infinito para alcanzar el límite.

1.2.3 Funcionamiento

A diferencia de otras redes, la red de Hopfield calcula los pesos de conexión entre las unidades de la capa de entrada y de salida por anticipado.

En el proceso interactivo lo que se va actualizando son los valores de cada una de las unidades de entrada y de salida hasta que se estabilicen, es decir, hasta que no cambien los valores de las unidades de la capa de entrada y de salida.

Se propone como matriz de peso a *w* para la red que tiene *n unidades* de entrada y *m unidades* de salida

$$w = y_1 x_1^t + y_2 x_2^t + \ldots + y_L x_L^t , \tag{1.6}$$

t traspuesta.

L número de pares de vectores

Si la red tuviera sólo n unidades de entrada entonces se calcula la matriz de

peso w como:

$$w = x_1 x_1^t + x_2 x_2^t + \ldots + x_L x_L^t \tag{1.7}$$

y se le denomina memoria autoasociada.

El primer paso que realiza la red es calcular w, siguiendo el criterio antes mencionado. Posteriormente se realiza la propagación de y a x. En este proceso se calcula el valor $neto^y$

$$neto^y = wx \tag{1.8}$$

$$neto_i^y = \sum_{j=1}^{n} w_{ij} x_j \tag{1.9}$$

si se tiene la estructura de retroalimentación se le suma al valor $neto^y$ al valor de cada unidad de retroalimentación I_j, por lo que se tiene la siguiente ecuación:

$$neto_i^y = \sum_{j=1}^{n} w_{ij} x_j + I_j \tag{1.10}$$

Después se cambian cada uno de los valores resultantes por +1 o -1 tomando en cuenta el siguiente criterio

$$y_i(t+1) = \begin{cases} +1 & neto_i^y > 0 \\ y_i(t) & neto_i^y = 0 \\ -1 & neto_i^y < 0 \end{cases} \tag{1.11}$$

el vector que resulta tiene el mismo número de unidades que el vector de salida inicial por lo que se propondrá como actualización de las unidades de salida, denotándose éste como,

$$y_{nuevo} = (y_1, y_2, \ldots, y_n). \tag{1.12}$$

Luego se calcula la energía del sistema tomando en cuenta la fórmula

$$E = -y_{nuevo}^t w x_0 \tag{1.13}$$

Aprovechando la actualización de los pesos de salida (y_{nuevo}) se realiza la propagación de x a y que consiste en obtener el valor

$$neta^x = w^t y \tag{1.14}$$

$$neto_i^x = \sum_{j=1}^{m} w_{ij}^t y_j \tag{1.15}$$

y posteriormente se pasa el vector resultante a +1 ó -1 ocupando la siguiente ecuación

$$x_i(t+1) = \begin{cases} +1 & neto_i^x > 0 \\ x_i(t) & neto_i^x = 0 \\ -1 & neto_i^x < 0 \end{cases} \qquad (1.16)$$

Este vector resultante, como en el caso anterior, es el que se sugerirá como renovación del vector de entrada (x_{nuevo})

$$x_{nuevo} = (x_1, x_2, \ldots\ldots x_m). \qquad (1.17)$$

Se determina después la energía del sistema con la fórmula

$$E = -y^t_{nuevo} \; wx_{nuevo} \; . \qquad (1.18)$$

El siguiente paso es realizar la actualización de y a x, y posteriormente la actualización de x a y, siguiendo el mismo procedimiento. Se termina este proceso en el instante en que no haya cambios entre la unidad de entrada y la de salida, que será en el momento en que se tenga la energía mínima.

La finalidad principal de la función de energía, es asegurarse que el sistema vaya disminuyendo la energía hasta llegar al mínimo determinado de antemano, por la matriz de peso (parte 2 del teorema de convergencia). El proceso se puede comparar con el lanzamiento de una bola en un declive, pero con la diferencia que en este transcurso cuando la bola llega al mínimo carece de energía.

En un principio Hopfield ocupó el modelo binomial (0,1), en el cual se toma en cuenta que cada una de las componentes de los vectores tienen o no tienen la

característica de interés, mientras que en el modelo binario (1,-1), se consideran dos características de interés. Si se requiere pasar del modelo binario al modelo binomial se utiliza la siguiente ecuación

$$w = \left(2v_1 - \vec{1}\right)\left(2v_1 - \vec{1}\right) + \left(2v_2 - \vec{1}\right)\left(2v_2 - \vec{1}\right) + \\ \ldots + \left(2v_n - \vec{1}\right)\left(2v_n - \vec{1}\right) \tag{1.19}$$

en donde $\vec{1}$ es el vector que tiene el valor de 1 en todas sus componentes. La expresión

$$\left(2v_i - \vec{1}\right)$$

transforma el vector binario v_i en el vector bipolar equivalente, así que la matriz de peso es la misma que se calcula con los vectores bipolares originales.

El siguiente reemplazo importante que se realiza es cambiar cada uno de los valores obtenidos por 0 y -1 tomando en cuenta el siguiente criterio

$$y_i(t+1) = \begin{cases} +1 & neto_i^y > 0 \\ y_i(t) & neto_i^y = 0 \\ -1 & neto_i^y < 0 \end{cases} \tag{1.20}$$

Las interpretaciones estadísticas así como los cálculos del modelo bipolar y binomial son distintos por lo que se debe de tenerse cuidado al realizar el cambio de modelos.

La red de Hopfield tienen como limitante el ocupar una gran cantidad de neuronas y de conexiones para almacenar la información, por lo que si se

introduce demasiada información la red puede converger a valores diferentes a los requeridos [8].

A continuación se dará un ejemplo numérico del funcionamiento de la red de Hopfield.

Se propónen los siguientes vectores de entrada (x_1,x_2) y de salida (y_1,y_2).

$$x_1^t = \left(1,-1,-1,1,-1,1,1,-1,-1,1\right)^t$$

$$y_1^t = (1,-1,-1,-1,-1,1)^t$$

$$x_2^t = \left(1,1,1,-1,-1,-1,1,1,-1,-1\right)^t$$

$$y_2^t = (1,1,1,1,-1,-1)^t$$

se multiplica y_1 por x_1 y y_2 por x_2

$$y_1 * x_1^t = \begin{bmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

$$y_2 * x_2^t = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 \end{bmatrix}$$

Luego se suman $y_1 x_1^t + y_2 x_2^t$ y se obtiene la matriz de peso w

$$w=\begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ -2 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 \end{bmatrix}$$

Tomando en cuenta a los vectores iniciales x_o y y_o

$$x_o^t = \left(-1,-1,-1,1,-1,-1,-1,-1,-1,1\right)^t$$

$$y_o^t = \left(1,1,1,1,-1,-1\right)^t,$$

se realiza la propagación de y a x.

En este proceso se calcula el valor

$$neto^{y^t} = \begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ -2 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$neto^{y^t} = \left(4,-8,-8,8,-4,8,4,-8,-4,8\right)$$

Posteriormente se cambian los valores de cada unidad a (-1) y (1) usando la fórmula (1.11) y se tiene

$$y_{nuevo}^{t} = \left(1,-1,-1,-1,-1,1\right)^{t}.$$

El siguiente paso que realiza la red es calcular la energía ocupando la ecuación (1.13)

$$E = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix} * \begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ -2 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 \end{bmatrix} * \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

E=-32

Aprovechando el valor obtenido de y_{nuevo} se hace la propagación de x a y, y se calcula el valor neto^{x} con la fórmula (1.14)

$$neto^{x} = \left(4,-8,-8,8,-4,0,-8,-4,8\right)$$

después se cambian los valores a (-1) y (1) usando la ecuación (1.16) y se tiene

$$x_{nuevo} = \left(1,-1,-1,1,-1,1,1,-1,-1,1\right)$$

Se calcula la energía del sistema con la fórmula (1.18)

E=-64.

Por la parte 2 del teorema de convergencia se sabe que el valor que tiene la energía mínima es -64 por lo que se ha llegado al valor mínimo, si se siguiera realizando el proceso de actualización de los valores de x y y éstos permanecerían iguales ya que no es posible llegar a una energía menor .

Se puede notar que los cambios de energía en el sistema al principio son

grandes, pero a medida que se realiza la propagación de x a y y de y a x los cambios son cada vez menores.

Se concluye que la red de Hopfield no realiza el proceso de aprendizaje, ya que se dan de antemano los pesos de conexión. En el proceso interactivo lo que se encuentra es el mínimo tomando en cuenta la energía del sistema en cada actualización.

Este sistema tiene la ventaja de que siempre converge, por lo que la actualización se realizará hasta que no haya cambios en las unidades de entrada y de salida. Tiene la utilidad de que es un buen método para encontrar mínimos siempre y cuando éste sea único. Posee la limitante de que si los vectores de entrada que se le proporcionan, no son lo suficientemente diferentes entonces la red no relacionará correctamente los vectores de entrada con los vectores de salida.

Otra restricción del sistema es que sólo puede almacenar un número reducido de información ya que de lo contrario proporcionará información errónea y en el caso de reconocimiento de patrones, no conseguirá recordar la imagen introducida de antemano si ésta se encuentra trasladada o rotada, pero tiene la ventaja de que sí podrá realizar la reconstrucción, si la imagen se encuentra distorsionada.

1.3 Retropropagación (backpropagation)

Esta red surge por la necesidad de tener un sistema inteligente que sea capaz de realizar el reconocimiento de tramas complejas, es decir, que conecte tramas arbitrarias de entrada y las semeje con tramas aprendidas previamente e ignore el ruido.

Es posible que reconozca patrones similares a los que aprendió pero no puede reconocer patrones nuevos o diferentes.

Tiene la gran ventaja de ocupar un algoritmo sigmoidal, lo cual le permite almacenar un gran número de señales, pero tiene la limitante, como todos los sistemas que ocupan el método de gradiente descendente, que puede llegar a

un mínimo local en lugar de uno global.

Se ocupa esta red para resolver problemas como: la clasificación de vocales, formación de reglas para la relación de letras y el procesamiento no lineal de señales, entre otros.

1.3.1 Estructura

La estructura de la red se encuentra formada por tres capas: una capa de entrada, una o más capas ocultas y una capa de salida. En esta red, la propagación es hacia adelante y todas las capas se encuentran completamente interconectadas, por lo que no existen conexiones de retroalimentación, ni hay una conexión que salte de una capa para ir a una anterior (figura 1.4).

El número de capas ocultas se determina con base a la experiencia sobre modelos similares, y por lo tanto, no existe una regla establecida.

La idea principal consiste en ocupar el menor número de capas ocultas. Si la red no converge cabe la posibilidad de que se necesiten más nodos ocultos, si converge se puede probar con un número menor de nodos ocultos y determinar un tamaño final basándose en el rendimiento global de la red.

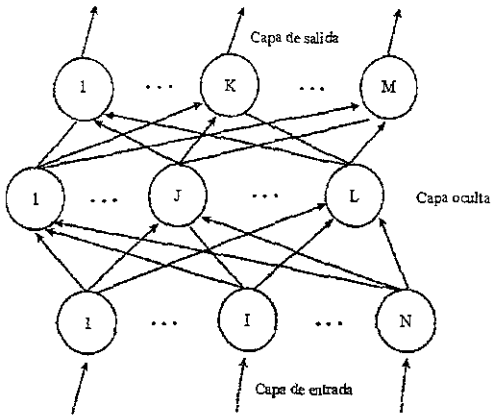

Figura 1.4

1.3.2 Aprendizaje

El proceso de aprendizaje tiene como fin que la red se organice a si misma, dando los pesos más adecuados mediante la capa oculta, de tal modo que la red aprende a reconocer distintas características del espacio total, esto con el fin de que cuando se le presente a la red una trama arbitraria de entrada con ruido o incompleta sepa clasificarla. Para ello se realiza un proceso interactivo descendiente en el cual se va minimizando el error en cada interacción, mediante el desplazamiento adecuado de un punto inicial sobre la superficie hasta llegar al mínimo. Este proceso hace que no importe haber hecho una buena aproximación inicial.

El proceso se inicia dando valores aleatorios pequeños a los pesos de conexión. Posteriormente se le presenta a la red un vector de entrada, $x_p = \left(x_{p1}, x_{p2} \ldots x_{pN}\right)^t$ el cual distribuye los valores a las unidades de la capa oculta. En la capa oculta se calcula el valor neto de cada unidad, el cual depende del peso que se le haya dado a cada unidad en la capa de entrada y se calcula mediante la siguiente ecuación

$$neta_{pj}^{h} = \sum_{i=1}^{n} w_{ji}^{h} X_{pi} + \theta_{j}^{h} \qquad (1.21)$$

Se debe suponer que existen L unidades en la capa oculta.

w_{ji}^{h} Es el peso de la conexión procedente de la *i*-ésima entrada.

θ_{j}^{h} Es un término de tendencia, también conocido como conexión de peso, este término sólo toma los valores 0 ó 1, tendrá el valor de uno cuando el vector de entrada corresponda a la clase, de lo contrario tomará el valor de cero.

h El índice *h* corresponde a la magnitud de la capa oculta.

Este valor depende del peso que se le haya dado a la unidad en la capa de

entrada.

El siguiente paso es calcular la *salida de la capa oculta* tomando en cuenta el valor neto de la capa oculta.

$$i_{pj} = f_j^h\left(neta_{pj}^h\right) \tag{1.22}$$

Posteriormente se calcula el *valor neto de salida*, utilizando el valor de salida de la capa oculta.

$$neta_{pk}^o = \sum_{J=1}^{L} w_{kI}^o i_{pJ} + \theta_k^o \tag{1.23}$$

Se obtiene después el valor de la capa de salida.

$$o_{pk} = f_k^o\left(neta_{pk}^o\right) \tag{1.24}$$

a este valor también se le denomina *valor de salida obtenido*, y representa la magnitud de salida.

En este momento se tiene la primera aproximación del peso correcto. Para hacer una comparación entre el valor de salida deseado y el valor de salida obtenido se calcula la diferencia, a esta ecuación se le denomina el *error de cada unidad*,

$$\delta_{pk} = \left(y_{pk} - o_{pk}\right) \tag{1.25}$$

p	***p*-ésimo vector de entrada.**
k	***k*-ésimo vector de salida.**
Y_{pk}	**Valor de salida deseado.**
o_{pk}	**Valor de salida obtenido.**

al error de cada unidad se eleva al cuadrado

$$\delta_{pk}^{2} = \left(y_{pk} - o_{pk}\right)^{2} \tag{1.26}$$

y después se suman todas las unidades de salida y se tiene el ***error total.***

$$E_{p} = \frac{1}{2}\sum_{k=1}^{M} \delta_{pk}^{2} \tag{1.27}$$

Se multiplica por $1/2$.

El error total es el que trata de minimizar. Para este fin se considerará a E_p como un espacio de pesos.

El siguiente paso que realiza la red es notar cuál es el error que origina cada una de las unidades de la capa de salida con una propagación hacia atrás, se revisa cuál es el error que proporciona cada una de las unidades de la capa oculta. Lógicamente el error que da la capa oculta es sólo una fracción del error total ya que a la capa oculta se aúna el error de la capa de salida.

Basándose en el error que suministra cada unidad, se actualizan los pesos de conexión, la dirección que debe de seguirse para que el error sea mínimo se encuentra dado por la derivada de E_p con respecto de w_{kj}^{0}.

Es por ello que a continuación se calculará la derivada de E_p.

Por las ecuaciones (1.26) y (1.27) se tiene

$$= \frac{1}{2}\sum_{k=1}^{M} \left(y_{pk} - o_{pk}\right)^{2}$$

Ocupando la ecuación (1.24)

$$= \frac{1}{2} \sum_{k=1}^{M} \left(y_{pk} - f_k^o \left(neta_{pk}^o \right) \right)^2 \qquad (1.28)$$

Sacando la derivada de E_p con respecto de w_{kj}^o

$$\frac{\partial E_P}{\partial w_{kj}^o} = \frac{1}{2} 2 \sum_{k=1}^{M} \left(y_{pk} - f_k^o \left(neta_{pk}^o \right) \right) * \left(- \frac{\partial f_k^o \left(neta_{pk}^o \right)}{\partial w_{kj}^o} \right)$$

$$= - \sum_{k=1}^{M} \left(y_{pk} - f_k^o \left(neta_{pk}^o \right) \right) * \left(\frac{\partial f_k^o \left(neta_{pk}^o \right)}{\partial w_{kj}^o} \right)$$

Multiplicando por un uno

$$\frac{\partial E_P}{\partial w_{kj}^o} = - \sum_{k=1}^{M} \left(y_{pk} - f_k^o \left(neta_{pk}^o \right) \right)$$

$$* \left(\frac{\partial f_k^o \left(neta_{pk}^o \right)}{\partial \left(w_{kj}^o \right)} \right) * \left(\frac{\partial \left(neta_{pk}^o \right)}{\partial \left(neta_{pk}^o \right)} \right)$$

$$= - \sum_{k=1}^{M} \left(y_{pk} - f_k^o \left(neta_{pk}^o \right) \right)$$

$$* \frac{\partial \, f_k^o\left(neta_{pk}^o\right)}{\partial \left(neta_{pk}^o\right)} * \frac{\partial \left(neta_{pk}^o\right)}{\partial \left(w_{kj}^o\right)} \qquad .(1.29)$$

Obs 1

$$neta_{pk}^o = \sum_{j=1}^{L} w_{kj}^o I_{pJ} + \theta_k^o$$

Empleando el valor de $neta_{pk}^o$ **para obtener la derivada**

$$\frac{\partial \left(neta_{pk}^o\right)}{\partial \, w_{kj}^o} = \frac{\partial \left(\sum_{j=1}^{L} w_{kj}^o I_{pJ} + \theta_k^o\right)}{\partial \, w_{kj}^o} = \sum_{j=1}^{L} I_{pj}$$

Sea

$$\sum_{j=1}^{L} I_{pJ} = i_{pJ}$$

Sustituyendo la ecuación (1.29) en la Obs. 1. se tiene

$$\frac{\partial \, E_p}{\partial \, w_{kj}^o} = -\sum_{k=1}^{M} \left(y_{pk} - f_k^{o'}\left(neta_{pk}^o\right)\right) * f_k^{o'}\left(neta_{pk}^o\right) * i_{pJ} \qquad (1.30)$$

La magnitud del cambio de pesos será proporcional a la derivada, por lo que a continuación se determinará la forma como se *actualizan los pesos de salida*

$$w_{kj}^o\left(t+1\right) = w_{kj}^o\left(t\right) + \Delta_p w_{kj}^o\left(t\right) \qquad (1.31)$$

$$\Delta_p w^o_{kj} = \eta\left(y_{pk} - o_{pk}\right) f_k^{o'}\left(neta^o_{pk}\right) i_{pj} \tag{1.32}$$

Donde η es un parámetro de la velocidad de aprendizaje.

Nótese que uno los requisitos que se le pide a f_k^o es que sea derivable [9]. Existen dos funciones de salida que son muy ocupadas, las cuales son

$$f_k^o\left(neta^o_{jk}\right) = neta^o_{jk} \tag{1.33}$$

$$f_k^o\left(neta^o_{jk}\right) = \left(1 + e^{-neta^o_{jk}}\right)^{-1} \tag{1.34}$$

La primera función, es una función lineal y la segunda es una función sigmoidal o logística. La selección adecuada de la función de salida, depende de la forma en que se deseen representar los datos de salida, si se desea que los datos de salida sean binarios se utilizará una función sigmoidal, ya que está limita la salida y es casi biestable. En otro caso se puede ocupar indistintamente cualquiera de las dos funciones.

La derivada de las dos funciones de salida propuesta son respectivamente

$$f_k^{o'}\left(neta^o_{jk}\right) = 1 \tag{1.35}$$

Sea $neta^o_{jk} = x$

$$f_k^{o`}(x) = -\left(1 + e^{-x}\right)^{-2} \frac{\partial\left(1 + e^{-x}\right)}{\partial x}$$

$$= -\left(1+e^{-x}\right)^{-2}\left(\frac{\partial(1)}{\partial x} + \frac{\partial\left(e^{-x}\right)}{\partial x}\right)$$

$$= -\left(1+e^{-x}\right)^{-2}\left(\frac{\partial\left(e^{-x}\right)}{\partial x}\right)$$

$$= -\left(1+e^{-x}\right)^{-2}\left(e^{-x}\frac{\partial(-x)}{\partial x}\right)$$

$$= \left(1+e^{-x}\right)^{-2}e^{-x}$$

$$= \frac{1}{\left(1+e^{-x}\right)} * \left(\frac{e^{-x}}{1+e^{-x}}\right)$$

$$= \frac{1}{1+e^{-x}} * \left(1 - \frac{1}{1+e^{-x}}\right)$$

$$= f_k^o(x)\left(1 - f_k^o(x)\right)$$

$$\therefore f_k^{o'}(x) = f_k^o(x)\left(1 - f_k^o(x)\right)$$

Ocupando la ecuación (1.32) se tiene

$$\therefore f_k^{o'}\left(neta_{jk}^o\right) = f_k^{o'}\left(neta_{jk}^o\right)\left(1 - f_k^{o'}\left(neta_{jk}^o\right)\right)$$

y usando la ecuación (1.24)

$$f_k^{o'}\left(neta_{jk}^o\right) = o_{pk}\left(1 - o_{pk}\right) \qquad (1.36)$$

Tomando en cuenta a las ecuaciones (1.31) y (1.32) se substituye el valor f_k^o de las dos funciones de salida, y se tiene que la actualización de los pesos de la capa de salida lineal es

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \eta\left(y_{pk} - o_{pk}\right) i_{pj} \tag{1.37}$$

y la actualización de los pesos para la capa de salida sigmoidal

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \eta\left(y_{pk} - o_{pk}\right)\left(1 - o_{pk}\right) o_{pk}\, i_{pj} \tag{1.38}$$

Para reducir la ecuación de actualización de pesos se define

$$\delta_{pk}^{o} = \left(y_{pk} - o_{pk}\right) f_k^{o'}\left(neta_{pk}^{o}\right)$$

por la ecuación (1.25)

$$\delta_{pk}^{o} = \delta_{pk} f_k^{o'}\left(neta_{pk}^{o}\right) \tag{1.39}$$

Si se utiliza la ecuación anterior y se substituye en la ecuación (1.31) y (1.32) se tiene

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \eta\delta_{pk}^{o} i_{pj} \tag{1.40}$$

El siguiente paso que realiza la red es una propagación hacia atrás para actualizar los pesos de la capa oculta basándose en los datos actualizados de la capa de salida.

Por la estructura que tiene la red es posible pensar que existe una relación entre la capa de salida y la capa oculta, lo cual se comprobará a continuación

$$E_P = \frac{1}{2}\sum_{k=1}^{M}\left(y_{pk} - o_{pk}\right)^2$$

Empleando la ecuación (1.24) se tiene

$$= \frac{1}{2}\sum_{k=1}^{M}\left(y_{pk} - f_k^o\left(neta_{pk}^o\right)\right)^2$$

Por la ecuación (1.23)

$$= \frac{1}{2}\sum_{k=1}^{M}\left(y_{pk} - f_k^o\left(\sum\left(w_{kj}^o * i_{pj} + \theta_k^o\right)\right)\right)^2$$

$$\therefore E_p = \frac{1}{2}\sum_{k=1}^{M}\left(y_{pk} - f_k^o\left(\sum\left(w_{kj}^o * i_{pj} + \theta_k^o\right)\right)\right)^2 \qquad (1.41)$$

Se sabe que i_{pj} depende del peso de la capa oculta. Se aprovechará esto para calcular el gradiente de E_p respecto a los pesos de la capa oculta.

$$E_p = -\frac{1}{2}2\sum_{k=1}^{M}\left(y_{pk} - f_k^o\left(neta_{pk}^o\right)\right) * \left(-\frac{\partial\left(neta_{pk}^o\right)}{\partial\, w_{ji}^h}\right)$$

Por la ecuación (1.24)

$$= -\sum_{k=1}^{M}\left(y_{pk} - o_{pk}\right) * \left(\frac{\partial\, o_{pk}}{\partial\, w_{ji}^h}\right)$$

$$= -\sum_{k=1}^{M}\left(y_{pk} - o_{pk}\right)*\left(\frac{\partial\, o_{pk}}{\partial\, neta^{o}_{pk}}\right)*$$

$$\left(\frac{\partial\, neta^{o}_{pk}}{\partial\, \imath_{pj}}\right)*\left(\frac{\partial\, \imath_{pj}}{\partial\, neta^{h}_{pj}}\right)*\left(\frac{\partial\, neta^{h}_{pj}}{\partial\, w^{h}_{ji}}\right) \qquad (1.42)$$

Sustituyendo la igualdad de la ecuación (1.21) para obtener la derivada

$$\frac{\partial\, neta^{o}_{pj}}{\partial\, i_{pj}} = \frac{\partial\left(\sum_{k=1}^{n} w^{o}_{kj}\, \imath_{pj} + \theta^{o}_{k}\right)}{\partial\, i_{pj}} = \sum_{k=1}^{n} w^{o}_{kj} \qquad (1.43)$$

Utilizando la ecuación (1.22) para sacar la derivada se tiene

$$\frac{\partial\, \imath_{pj}}{\partial\, neta^{h}_{pj}} = \frac{\partial\, f^{h}_{j}\left(neta^{h}_{pj}\right)}{\partial\, neta^{h}_{pj}} = f^{h'}_{j}\left(neta^{h}_{pj}\right) \qquad (1.44)$$

Empleando la ecuación (1.21) y desarrollando la derivada

$$\frac{\partial\, neta^{h}_{pj}}{\partial\, w^{h}_{ji}} = \frac{\partial\left(\sum_{J=1}^{n} w^{h}_{ji} X_{pi} + \theta^{h}_{J}\right)}{\partial\, w^{h}_{ji}} \qquad (1.45)$$

Sea

$$\sum_{j=1}^{n} X_{pi} = x_{pi}$$

Ocupando las ecuaciones (1.43), (1.41), (1.45) y sustituyéndolas en la ecuación (1.42) se tiene

$$\frac{\partial E_p}{\partial w_{ji}^h} = -\sum_{k=1}^{M} \left(y_{pk} - o_{pk}\right) * \left(f_k^{o'}\left(neta_{pk}^o\right)\right) * \left(w_{kj}^o\right) * \left(f_j^{h'}\left(neta_{pj}^h\right)\right) * x_{pi} \tag{1.46}$$

Con la derivada de E_p con respecto de w_{ji}^h se puede realizar la actualización de los pesos de la capa oculta mediante las siguientes ecuaciones

$$w_{ji}^h\left(t+1\right) = w_{ji}^h\left(t\right) + \Delta_p w_{ji}^h\left(t\right) \tag{1.47}$$

$$\Delta_p w_{ji}^h = \eta \; f_j^{h'}\left(neta_{pj}^h\right) * \left(x_{pi}\right) * \sum_{k=1}^{M} \left(y_{pk} - o_{pk}\right) * \left(f_k^{o'}\left(neta_{pk}^o\right)\right) * w_{kj}^o \tag{1.48}$$

Donde η es la velocidad de aprendizaje.

Sustituyendo la ecuación (1.48) en la fórmula (1.39) se tiene

$$\Delta_p w_{ji}^h = \eta f_{pj}^{h'}\left(neta_{pj}^h\right) * \left(x_{pi}\right) * \sum_{k=1}^{M} \delta_{pk}^o * \left(w_{kj}^o\right)$$

Para reducir términos se define δ_{pj}^h

$$\delta^{h}_{pj} = f^{h'}_{j}\left(neta^{h}_{pj}\right)\left(\sum_{k=1}^{M} \delta^{o}_{pk}\right) * \left(w^{o}_{kj}\right) \qquad (1.49)$$

y se sustituye en las ecuaciones (1.48) y (1.49).

$$w^{h}_{ji}(t+1) = w^{h}_{ji}(t) + \eta \delta^{h}_{pj} x_{pi} \qquad (1.50)$$

Al realizar la actualización de los pesos de la capa de oculta se tiene un nuevo punto en el espacio de pesos al cual se le realiza el mismo proceso como si fuera un vector de entrada. Este proceso termina en el momento en que la red llega, en el espacio de pesos a un mínimo, no se puede asegurar que éste sea un mínimo global y no es necesario que ello ocurra en todas las aplicaciones, puede ser suficiente con que se encuentre en un mínimo local, en el cual el error que se tenga sea menor o igual a un número preestablecido.

La velocidad de convergencia se encuentra dada por pequeños incrementos debido a que no se sabe si se está lejos o cerca del mínimo en el espacio de pesos, y si se toman incrementos grandes se corre el riesgo de pasar por encima del mínimo, sin embargo si se dan pequeños incrementos es más factible asegurar la convergencia, aunque la red tendrá que hacer una gran cantidad de interacciones para converger. La constante que determina la velocidad de convergencia es η, a la cual se le denomina constante de proporcionalidad o taza de aprendizaje y se encuentra en el intervalo [0.05, 0.25].

En esta red se ocupa la regla Delta Generalizada [10], la diferencia entre el método de mínimos cuadrados y la regla Delta Generalizada, estriba en que en el método de mínimos cuadrados se realiza la diferencia al cuadrado entre el valor de salida deseado y el valor de salida obtenido. Este mismo proceso se ocupa en la regla Delta Generalizada, con la diferencia que en este caso se van acumulando los cambios y se suma la actualización de los pesos.[11]

El proceso de la diferencia del cuadrado, entre la salida obtenida y la salida

deseada, se repite hasta que el error sea aceptablemente bajo. La desventaja de este método es que necesita una gran cantidad de memoria para almacenar.

La forma en que se seleccionan los datos que se requieren para entrenar a la red, está en función de la experiencia del investigador pero se darán los lineamentos principales que se deben de seguir en la sección de datos de entrenamiento.

1.3.3 Funcionamiento

Durante el proceso de funcionamiento la red ya ha ajustado los pesos de la capa oculta, de tal modo que cuando se le presenta un vector de entrada tiene la capacidad de extraer las características fundamentales del vector y proporcionar una salida que se encuentre cercana al valor que proporciona el error mínimo. Por lo que la red tendrá que hacer un número reducido de iteracciones para llegar al error mínimo.

El funcionamiento de la red se inicia dando un vector de entrada, este vector se propaga a través de la capa oculta y la capa de salida y se calculan los valores netos de cada una de esas capas, de forma similar como se hizo en el proceso de aprendizaje, ocupando las fórmulas (1.21), (1.22), (1.23), y (1.24), se obtiene así la unidad de salida, posteriormente se calcula la diferencia del valor de salida deseado y el valor de salida obtenido y se eleva al cuadrado, fórmula (1.26), y después se suma cada una de las unidades, fórmula (1.27). Posteriormente se determina cual es el error que aporta cada una de las unidades de la capa oculta y se realiza la actualización de la capa de salida usando las fórmulas (1.31) y (1.32) y de la capa oculta con las relaciones (1.47) y (1.48) ocupando la técnica de la derivada, obteniendo un nuevo punto el cual se trata como si fuera un vector de entrada aplicando así el mismo procedimiento, se finaliza cuando se obtiene en el espacio de pesos el mínimo.

1.3.4 Datos de Entrenamiento

La forma de seleccionar y preparar un conjunto de datos para entrenar una

red dependen de la experiencia del investigador en el entrenamiento de las redes neuronales.

Por lo general del total de datos que se tienen sólo se selecciona un conjunto reducido para entrenar a la red, los demás datos se ocupan para verificar que la red ha sido entrenada adecuadamente.

En ocasiones es necesario presentarle a la red algunos datos con ruido ya que esto ayuda a que la red converja.

Si se realiza un entrenamiento adecuado la red será capaz de hacer una buena generalización e ignorar los datos irrelevantes. Con ello se tendrá una adecuada clasificación. De lo contrario, si se entrena a la red con una sola clase de vectores de entrada o con datos insuficientes, la red no efectuará una clasificación adecuada ya que no identificará con claridad la clase a la que pertenecen los datos de entrada proporcionados.

No se debe de entrenar una red dándole primero todos los patrones de una clase y posteriormente los de la siguiente clase, ya que la red se olvidará de los datos aportados en la clase anterior [12].

1.3.5 Aplicación

Como se ha mencionado de manera previa este modelo puede ocuparse en diferentes áreas, a continuación se presenta un ejemplo de una red para la clasificación de señales electrocardiográficas (ECG).

La señal ECG se obtiene de un programa creado en la Universidad de Leuven.

Después se utiliza un paquete comercial de análisis ECG el cual proporciona 39 características entre las cuales se encuentran la amplitud y diámetro de los puntos QRS, duración eje del QRS, elevación o depresión del ST, área bajo el QRS, área bajo la onda T, la edad y el sexo de la persona entre otros, tal como se muestra en la figura 1.5.

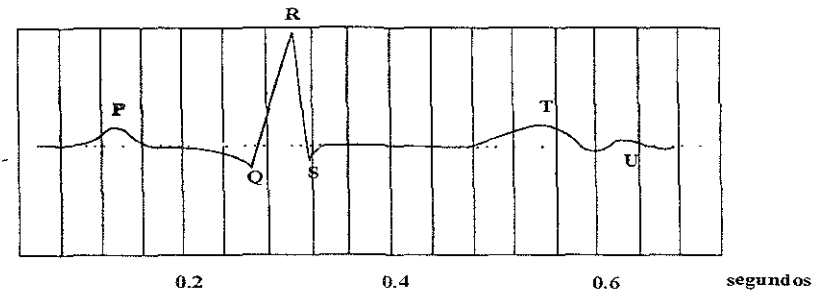

Figura 1.5

Con estos datos es posible detectar los siguientes tipos cardiacos.

1. Normal.
2. Hipertrofia ventricular izquierda (LVH).
3. Hipertrofia ventricular derecha (RVH).
4. Hipertrofia bi-ventricular (BVH).
5. Infarto de miocardio anterior (AMI).
6. Infarto de miocardio inferior (IMI).
7. Infarto de miocardio combinado(MIX).

Se puede entrenar a la red tomando en cuenta las 39 características para identificar los 7 tipos cardiacos mencionados previamente.

La estructura de la red se compone de 3 capas. La capa de entrada esta compuesta por 39 neuronas, se utiliza una neurona por cada una de las características que son utilizadas para hacer el diagnostico. La capa intermedia fue probada con diferente numero de neuronas siendo el número optima de 7 para este tipo de problema el que se encontró. Por último, se tiene la capa de salida que se compone de 7 neuronas, cada una de estas es empleada para cada uno de los 7 tipos de resultados cardiacos.

Durante el proceso de entrenamiento se le presentaron a la red las 39 características como vectores de entrada, para que la red ajuste pesos entre capas y así proporcionará un vector de salida.

En este caso se contó con 3266 señales, de las cuales se seleccionaron aleatoriamente 2446 para entrenar a la red y 820 señales para verificar su

buen funcionamiento.

La Red de Backpropagation ocupa un algoritmo sigmoidal, lo que le permite almacenar una gran cantidad de información. Se realiza un proceso de aprendizaje en el cual se le muestra a la red un conjunto de vectores de entrenamiento, con los cuales la red ajusta los pesos de conexión y las características fundamentales para que durante el funcionamiento se le dé un vector de entrada arbitrario y proporcione un vector de salida que se encuentre cerca del mínimo en el espacio de pesos. En el proceso de aprendizaje se realiza un ajuste de pesos mediante el desplazamiento adecuado de un punto sobre el espacio de pesos, tomando en cuenta que la mejor dirección para obtener el error mínimo se encuentra dada por la derivada.

Esta red puede reconocer vectores de entrada con ruido o incompletos, tiene la desventaja de que no puede aprender patrones nuevos después del proceso de aprendizaje. Puede llegar a un mínimo local en lugar de a un mínimo global, lo cual no es una limitación grave si el mínimo aporta un error que sea lo suficientemente pequeño.

1.4 Mapas Autorganizados *(Kohonen)*

En el cerebro humano la información se encuentra organizada en zonas, las cuales se perciben a través de los órganos sensoriales y se representan internamente por mapas bidimensionales [13]; un ejemplo de esta organización es la que tiene el sistema visual, el cual se ha detectado que se encuentra organizado en zonas de la corteza cerebral (capa externa del cerebro). Para imitar este proceso Kohonen ha propuesto un sistema inteligente, en el que se realizan las funciones de organización y agrupación de la información en zonas.

Esta red requiere de un proceso de aprendizaje, que consiste en darle un conjunto de patrones de entrada, con los cuales va ajustando los pesos entre las unidades de entrada y de salida. Durante el funcionamiento, la red determina primero, la clase a la que pertenece el vector de entrada y luego seleccionará

un elemento de la clase, el cual es el que tiene la menor diferencia en valor absoluto entre el vector de entrada y el vector de pesos.

Se utiliza esta red principalmente en las áreas de la investigación donde los datos etiquetados son escasos pero los datos sin etiquetar son abundantes, es ocupada también para el reconocimiento de patrones (voz, texto, imágenes, señales, etc.), y para la resolución de problemas de optimización (problema del agente viajero).

1.4.1 Estructura

La estructura de la red se encuentra integrada por dos capas; un vector de entrada que se encuentra constituido por N unidades relacionadas por impulsos hacia adelante con las M unidades de salida, las cuales forman un espacio bidimensional (figura1.6)

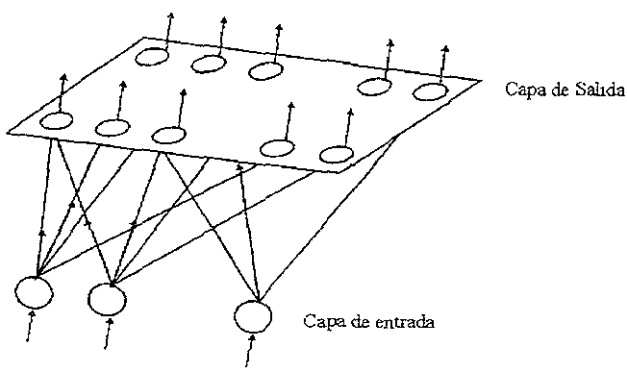

Figura 1.6

Esta estructura le permite a la red tener cierta influencia entre las neuronas laterales, aunque éstas no se encuentren conectadas físicamente, lo cual ocasiona que se formen vecindades o zonas que pueden ser rectangulares, circulares, hexagonales, o de cualquier otro tipo de polígono regular [14].

1.4.2 Funcionamiento

El proceso se inicia dándole a la red un vector de entrada, el cual visita a cada una de las unidades de las clases de salida y se activa la clase a la que pertenece el vector de entrada.

Posteriormente como se trata de una red competitiva se selecciona una unidad ganadora quien tiene el menor valor absoluto entre el vector de entrada $E_k = \left(e_1^{(k)}, \ldots\ldots e_N^{(k)} \right)$ y el vector de peso $W_j = \left(w_1^{(k)}, \ldots\ldots w_n^{(k)} \right)$ de las conexiones entre cada una de las neuronas de entrada y las neuronas de salida.

$$s_j = \begin{cases} 1 & MIN E_k - W_j = MIN \left[\sum_{i=1}^{N} e_i^k \right] \\ 0 & e.o.c. \end{cases} \qquad (1.51)$$

La finalidad principal de este proceso es determinar cuál es el vector de entrada más parecido al vector de entrada proporcionado, así como la neurona que se activa, y la zona a la que pertenece.

1.4.3 Aprendizaje

El proceso de aprendizaje de la red sirve para fijar los pesos entre las conexiones de entrada y de salida. Esta red tiene un aprendizaje supervisado, por lo que el proceso de aprendizaje inicia dándole valores aleatorios pequeños entre las unidades de salida y se fijan las zonas de las vecindades. Posteriormente se le presenta a la red un patrón de entrada que se desea que aprenda representado en forma de vector de entrada.

Como se trata de un aprendizaje competitivo se determina una unidad ganadora, la cual es la que tiene la menor diferencia al cuadrado entre el vector de entrada y el vector de peso. La diferencia se eleva al cuadrado ya que se podría tener el caso en que al sumar las diferencias positivas y negativas se compensen los valores y dieran cero, y ésta no sería la mejor diferencia entre el vector de entrada y de salida. Por lo que se ocupa la siguiente ecuación

$$d = \sum_{i=1}^{N} \left(x_i^k - w_{ji} \right)^2 \qquad 1 \le j \le M \tag{1.52}$$

x_i^k Componente i-ésimo de la k-ésima entrada.

w_{ji} Peso de la conexión de la i-ésima capa de entrada y la j-ésima unidad de salida

$1 \le j \le M$ Suponiendo que se tienen M capas de salida.

El siguiente paso después de haber localizado la unidad ganadora, es realizar un reajuste entre los pesos de las unidades de salida de la unidad ganadora y las unidades laterales. Con ello se asocia la información a una vecindad o zona de salida.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)\left[e_i^k - w_{ji}(t) \right] para\ j \in Zona_{j^*}(t) \tag{1.53}$$

$Zona_{j^*}(t)$ es la zona de vecindad alrededor de la vencedora j^* en la que se encuentra las neuronas cuyos pesos son actualizados.

Posteriormente se le presenta a la red un conjunto de patrones de entrenamiento, el cual será necesario que se le muestren un gran número de veces $(500 \le t \le 10000)$. El $\alpha(t)$ es un término de ganancia o coeficiente de aprendizaje, el cual toma valores entre 0 y 1; cuando se le ha presentado a la red un gran número de veces el conjunto de patrones de entrenamiento el valor de ajuste $\alpha(t)$ se irá reduciendo y será casi cero. La ecuación que sigue

$\alpha(t)$ con respecto al tiempo es la siguiente

$$\alpha(t) = \frac{1}{t} \qquad \alpha(t) = \left[1 - \frac{1}{\alpha_2}\right] \tag{1.54}$$

Siendo α_1 un valor 0.1 ó 0.2 y α_2 un número próximo al número total de interacciones del aprendizaje, suele tomar el valor de 10000.

El proceso de presentar todo el juego de patrones debe repetirse por lo menos 500 veces.

Durante el proceso de aprendizaje se va reduciendo el diámetro de la vecindad ganadora y como consecuencia se tiene una mejor selección de la unidad ganadora con respecto al vector de entrada que se le presenta, lo ideal sería que para cada vector de entrada que se le da a la red, exista una unidad ganadora (figura 1.7).

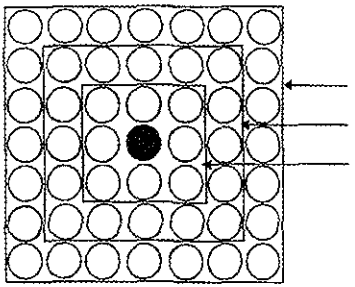

Figura 1.7

Cuando se realiza el *proceso de entrenamiento* se le debe mostrar a la red patrones de diferentes clases, es decir, no se le deben de presentar primero todos los patrones de una misma clase y luego los de la otra porque al terminar el entrenamiento la red no recordará los patrones iniciales [15].

1.4.4 *Aplicación*

Un ejemplo de la aplicación de este modelo fue utilizado para diseñar una forma de reducir el tamaño de una imagen ya sea para almacenarla o

transmitirla, tal como se explica a continuación.

En este caso se divide una imagen en 3 imágenes monocolor y posteriormente se identifica el grado de intensidad de cada píxel.

Lo primero que se hace es convertir la imagen a una escala de RGV (rojo, verde y azul).

Posteriormente se divide esta imagen en los 3 colores básicos (rojo, verde y azul) lógicamente la combinación de estos 3 colores dan lugar a la imagen original.

En una imagen de colores cada uno de los pixeles se encuentran codificados por 24 bits, con la separación de los colores de la imagen, cada uno de los pixeles se encuentran constituidos por 8 bits.

Luego se identifica el grado de intensidad de cada píxel, el cual tiene 256 posibilidades en cada una de las 3 imágenes monocolor obtenidas.

Para identificar el grado de intensidad de cada píxel se propone un sistema inteligente, el cual tenga 3 neuronas de entrada que identifiquen el grado de intensidad de los 3 colores básicos y 256 neuronas correspondiente a las 256 posibilidades del grado de intensidad.

En el proceso de aprendizaje lo primero que de hace es asignar pesos aleatorios pequeños, el siguiente paso es asignar la clase y las zonas de estas.

Se le presenta un vector de entrada a la red, esta calcula el valor de d ocupando la ecuación 1.52.

Por comodidad para la explicación

Sea

$$d_n = \qquad n = 1,\ldots\ldots, m \tag{1.55}$$

donde

d_n Valor de d generado por el vector n-ésimo.

M Número de patrones de entrenamiento.

Por lo tanto por la ecuación 1.55 se le asigna al valor obtenido de d la notación d_1.

Se le da al patrón de entrenamiento cualquier clase y posición ya que no tiene con quien comparar, enseguida se le presenta otro vector de entrada de manera análoga se calcula el valor de d_2, la red identifica que tan parecido es este nuevo vector con respecto al anterior, esto lo hace tomando en cuenta la diferencia entre el valor d_1 y d_2.

Si el valor de d_1 es igual al valor de d_2 entonces se le asigna la misma clase y posición que la de d_1. Si d_1 tiene un valor muy diferente a d_2 entonces se coloca muy alejado con respecto a d_1 y como consecuencia estará en otra clase. Se realiza un ajuste de pesos entre las neuronas vecinas y con esto se asigna la zona del nuevo patrón. Si el vector de entrada d_2 tiene un valor parecido o cercano a d_1 entonces se coloca cerca del primer patrón y como consecuencia se encontrará dentro de la misma clase.

Luego se le presenta a la red el tercer vector de entrada, la red calcula el valor de d_3 compara este con el de d_1, si este valor es parecido entonces se coloca cerca de la neurona de salida asociada al valor obtenido de d_1, de lo contrario se compara con el valor de d_2 si este valor son cercanos entonces se coloca a d_3 cerca del lugar asignado al segundo patrón. Si el valor de d_3 es muy diferente al valor de d_1 y de d_2 entonces a d_3 se le coloca lejos de los dos primeros patrones y se le asigna una nueva clase.

Para el resto de los m vectores se realiza el mismo proceso descrito anteriormente para su clasificación en las neuronas de la capa de salida Es necesario presentarle a la red varias veces el conjunto de patrones de entrenamiento con la finalidad de verificar que se han asignado correctamente los pesos.

Después del proceso de entrenamiento la red será capaz de identificar la intensidad de cada píxel de la imagen mediante el proceso de funcionamiento en el cual se le presenta a la red un vector de entrada, la red identifica la

clase a la que pertenece y calcula el valor de d identifica cual de los elementos de la clase tiene el valor más parecido y se asigna esta intensidad de píxel.

En esta red se tiene la gran ventaja de que puede tener un conjunto pequeño de patrones etiquetados y muchos sin etiquetar. Es factible almacenar y procesar una gran cantidad de información, requiere de un aprendizaje no supervisado, lo que permite que ella ajuste los pesos de conexión y determine la clase a la que pertenece cada patrón de entrenamiento durante un proceso denominado de aprendizaje.

Tiene dos limitaciones: la primera es que la red no puede aprender patrones nuevos o diferentes después del proceso de aprendizaje, y la segunda (la cual se encuentra asociada al tipo de aprendizaje no supervisado), que tiene un largo proceso de aprendizaje ya que se le debe presentar el conjunto de patrones que debe aprender por lo menos 500 veces, con el fin de que se estabilicen los pesos de conexión .

Capitulo 2

2.1 Redes Neuronales Estadísticas

En la actualidad es necesario juntar varias áreas del conocimiento con la finalidad de tener mejores soluciones, en este caso se aunarán las ventajas de la estadística con la de los Sistemas Inteligentes. Debe de recordarse que las redes Neuronales tienen fuertes raíces estadísticas y fue Chow [16] en 1957 el primero en realizar la formalización entre las redes neuronales con la Teoría Estadística de Decisión, con la finalidad de obtener un clasificador que sea óptimo, sin embargo se ha desarrollado tanto el área de las redes neuronales que se le ha considerado como una rama independiente, pero requiere de la estadística para que sus resultados tengan validez.

Por ejemplo, uno de los problemas principales en el área de Reconocimiento de Patrones es asignar una función la cual tenga como dominio el conjunto de mediciones o atributos y como imagen la clase a la que pertenece cada patrón, esta función tiene que tener la característica que divida el espacio de variables en clases, que cubra todo el espacio, y que las clases sean disyuntas entre si, lo cual asegura que cada uno de los patrones siempre tengan una clase.

Dentro de cada clase existen pequeñas diferencias (variabilidad intrínseca), las cuales no siempre se deben a errores del proceso sino a variaciones dentro de la clase. Para determinar si un patrón pertenece o no a una clase se ocupa la teoría estadística de decisión. Para asignar esta función se debe de tomar en cuenta la información que se tenga a priori del problema específico.

Si se conoce la función de distribución de cada clase se puede ocupar el método de mínimos cuadrados con la finalidad de minimizar el error y tener una clasificación que sea óptima.

Si se desconoce la función de distribución es necesario ocupar un método no paramétrico el cual depende en su totalidad de que la muestra que se tenga sea representativa de lo que se desea enseñar a la red, corresponderá la eficiencia también a la cantidad de patrones de entrenamiento que se tenga tanto para

que la red aprenda adecuadamente como para que el método no paramétrico que se ocupe sea eficiente.

Cuando es necesario ocupar un método de estadística no paramétrica porque no se cumplen los supuestos de la estadística paramétrica [17] como:

- las poblaciones que se estudian tienen una distribución normal
- la varianza de cada una de las poblaciones es pequeña

en el caso en que alguno de estos supuestos no se pueda verificar, es necesario ocupar la estadística no paramétrica, con esta estadística es posible encontrar algunos parámetros (media, varianza, etc.) del modelo que se esté estudiando. Mientras mayor sea la población que se tenga se podrá proponer una función de densidad, invariablemente si se tiene una población pequeña y no se conoce la función de densidad de la población se tendrá que ocupar un método no paramétrico.

Es posible ocupar una rama de la estadística, el muestreo para obtener una muestra de la población que sea representativa, es decir, seleccionar un conjunto de patrones que representen a todos los existentes (universo) y en la misma proporción.

Se puede emplear la estadística, para reducir el conjunto de patrones seleccionados quitando los patrones que aporten la misma información mediante el análisis multivariado.

En este capítulo se presenta el funcionamiento de dos redes estadísticas, la red de Boltzmann y el método NN y algunas de sus modificaciones, en estas dos redes se requiere de un proceso de aprendizaje, y posteriormente durante el funcionamiento la red será capaz de clasificar patrones experimentales.

En la red de Boltzmann se tiene una función de salida no determinística, es decir, que se le puede dar el mismo vector de entrenamiento y éste puede proporcionar diferentes vectores de salida. Se tiene la ventaja en esta red que se puede asegurar la convergencia al mínimo global, gracias a que la elección de los pesos que se modifican es aleatoria y a que se propone una función de

temperatura que tiene la finalidad de incrementar la energía para que la red escape de un mínimo local a uno global.

Posteriormente se revisará el proceso de aprendizaje y funcionamiento del modelo NN. En este modelo es posible que se le de a la red las características que debe de tomar en cuenta, esto ayuda a que la red tenga mucha precisión en la clasificación.

Han surgido con el tiempo algunas modificaciones para mejorar este modelo como por ejemplo métodos para mejorar el rendimiento del clasificador, o cuando se tienen casos específicos como el de seleccionar de una muestra de entrenamiento grande una muestra que sea representativa, o la opción en que algún patrón de entrenamiento se encuentre mal clasificado por el costo o la dificultad que implica la clasificación, al final se menciona cuando los tamaños de las clases de la muestra de entrenamiento se encuentran en diferente proporción. Se mencionará a continuación cada uno de los modelos a su autor o autores, así como las ventajas y limitaciones de cada uno de los modelos.

2.2 Boltzmann

La red de Boltzmann como las redes de Backpropagation y Kohonen, poseen un periodo de entrenamiento en el cual, se le proporciona a la red un conjunto de tramas de entrada con la finalidad de identificar los rasgos característicos de diferentes patrones que se le dan, para que posteriormente en el proceso de funcionamiento sea capaz de identificar el patrón que se le proporciona tomando en cuenta los rasgos característicos e ignore el ruido o las partes faltantes. Los componentes de la red de Boltzmann son binarios (0,1).

La red de Boltzmann, es menos empleada que la red de Backpropagation y de Hopfield debido a su lentitud, aunque tiene la gran ventaja de asegurar que la red converja al mínimo global en lugar de a un mínimo local. La arquitectura de monocapa en la red de Boltzmann tiene las mismas aplicaciones que el modelo de Hopfield, y el modelo con arquitectura de multicapa se usa de manera similar a la red de Backpropagation .

2.2.1 Arquitectura

Existen dos arquitecturas diferentes en la red de Boltzmann, la primera es la red de terminación de Boltzmann (Boltzmann completion network) [18], la cual consta de una sola capa, en la que se distinguen N neuronas visibles y N+P neuronas ocultas (figura 2.1). Las conexiones en la red se establecen en ambos sentidos y los pesos entre las conexiones son

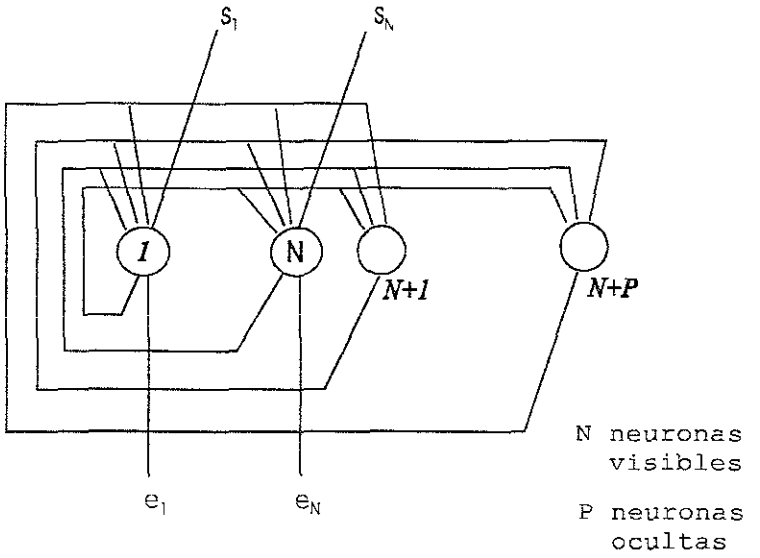

Figura 2.1

simétricas por lo que se cumplen $w_{ij}=w_{ji}$

La otra arquitectura es la red de entrada-salida de Boltzmann (Boltmann input-output network), en la cual se encuentran tres capas: la capa de entrada con N neuronas, la capa oculta que contiene N+P neuronas, y la capa de salida con N+P+M neuronas (figura 2.2).

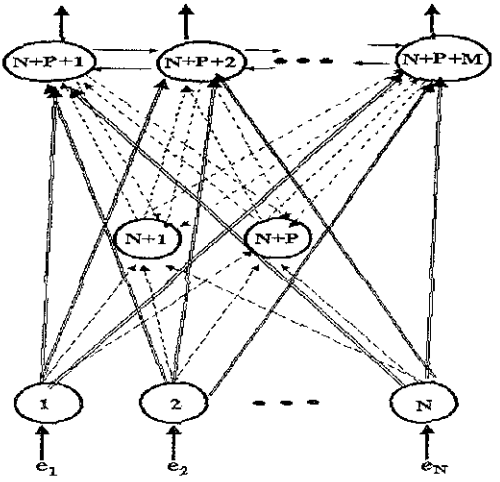

Figura 2.2

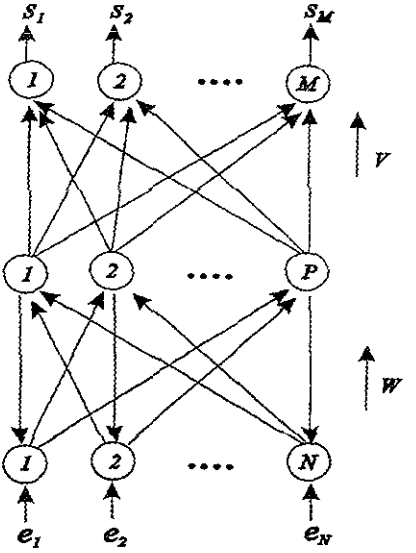

Figura 2.3

Las conexiones de la red tienen ambos sentidos, algunos autores ocupan un solo sentido hacia adelante (Boltzmann Feedforward Network) [19] (figura 2.3).

2.2.3 Proceso de Temperatura

La red de Boltzmann propone un incremento en la temperatura, para con ello aumentar la función de energía y sacar a la red de un mínimo local a un mínimo global.

Durante este proceso de descenso mientras menor sea la temperatura [20], la red tendrá menos oportunidad de realizar cambios y como consecuencia será menos probable que cambie de un mínimo local a un mínimo global. Si se disminuye la temperatura apresuradamente para que la red sea mas rápida se corre el riesgo que la red quede atrapada en un mínimo local (figura 2.4).

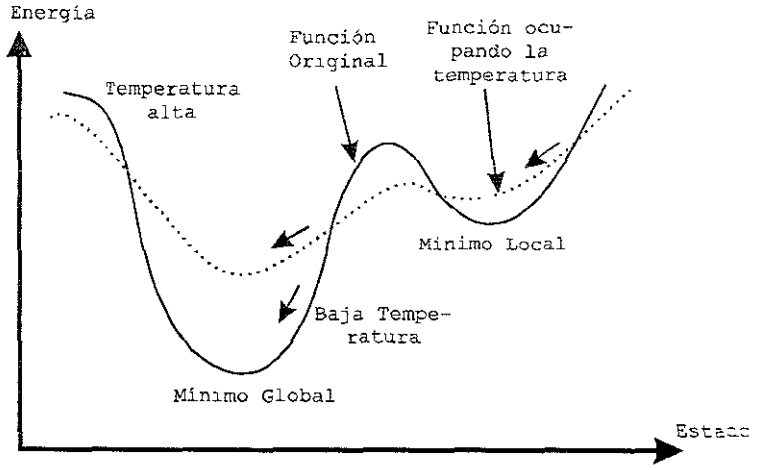

Figura 2.4

Para determinar qué funciones son adecuadas para incrementar la temperatura y decrementarla paulatinamente, varios investigadores han propuesto diferentes funciones [21], entre las más importantes se encuentra, la función de templado por intervalos , en la cual se le asigna a cada intervalo de tiempo cierta temperatura (figura 2.5)

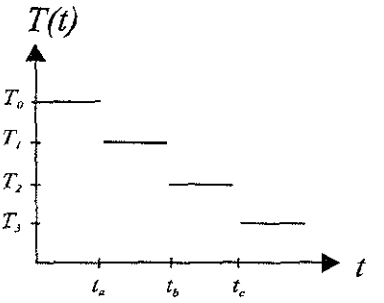

Figura 2.5

$$T(t) = \begin{matrix} T_0 & 0 \le t \le t_a \\ T_1 & t_a \le t \le t_b \\ T_2 & t_b \le t \le t_c \\ T_3 & t \le t_c \end{matrix} \qquad (2.1)$$

Los estudios de German y German mostraron que una buena forma de decrementar la temperatura es ocupando la inversa del logaritmo de la temperatura, por lo que propuso la función (figura 2.6):

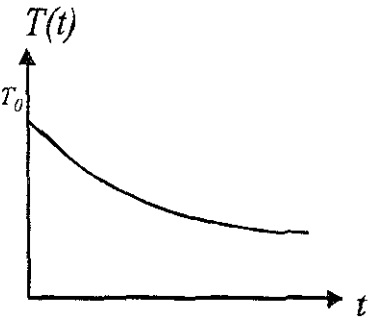

Figura2.6

$$T(t) = \frac{T_0}{1 + ln(t)} \qquad (2.2)$$

donde T_o es la temperatura inicial y t el tiempo, ésta es la función más ocupada para la red de Boltzmann [22]. Otros autores propusieron la inversa de la

temperatura que es la función que se emplea regularmente para la red de Cauchy (figura 2.7).

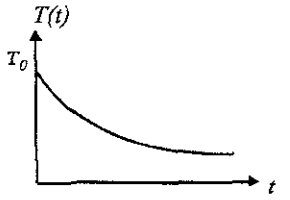

Figura2.7

$$T(t) = \frac{T_0}{1+t} \tag{2.3}$$

2.2.4 Aprendizaje de las Redes con Arquitectura Monocapa.

Se inicia el proceso dando valores aleatorios a los pesos (w_{ij}) de conexión entre la capa visible y la capa oculta. Posteriormente se le presenta a la red un vector de

entrenamiento $E_1 = \left(e_1^{(1)}, \ldots\ldots e_N^{(1)}\right)$, **el cual se transfiere de la capa de entrada a la capa oculta. Luego se ajusta la temperatura que está en relación del tiempo, el cual inicialmente toma el valor t=0. Se realiza un reajuste de los pesos. Se selecciona una neurona oculta aleatoria, como se trata de una red estocástica la función de activación es no determinística, lo cual quiere decir que se pueden asignar valores diferentes de salida a una misma entrada.**

Cuando se tiene una red de este tipo, se desea saber el valor de salida exacto en cada momento, para ello se ocupa la técnica de simulación, la cual consiste en seleccionar un número aleatorio que se encuentre en el intervalo [0,1] con una distribución uniforme. Se considera que la neurona está activa si el número aleatorio seleccionado es menor o igual que la probabilidad neta de entrada de *i*, esta probabilidad depende de la temperatura

$x \in [0,1]$

$$\mathcal{P}_{net_i}\left(s_i = 1\right) = \frac{1}{1 + e^{net_i/T}} \tag{2.4}$$

$$s_i = \begin{cases} 1 & P_{net_i}\left(s_i = 1\right) \geq x \\ 0 & e.o.c \end{cases} \tag{2.5}$$

Posteriormente se selecciona al azar otra neurona oculta y se repite el mismo proceso. El número de veces que se repite este proceso depende de la cantidad de elementos que existen en la capa oculta ya que se da opción a que todas las unidades tengan oportunidad de ser seleccionadas por lo menos una vez. Por lo que el ciclo de procesamiento, si se tienen diez unidades, es mayor o igual a diez. Como se selecciona aleatoriamente una neurona oculta para el proceso no se puede asegurar que cada una de las neuronas ocultas haya sido seleccionada. Se termina el ciclo también en el momento en que no haya variación en la energía de la red en dos o tres interacciones

$$\left(\Delta E = net_i = \sum w_{ij} s_j\right). \tag{2.6}$$

Cuando termina el proceso se registran los valores de salida de la neurona $\left(s_1^{(1)}, \ldots\ldots, s_n^{(k)}\right)$

$$P_{ij}^{+} = \frac{1}{k} \sum_{k=1}^{k} s_i^{(k)} s_j^{(k)} \tag{2.7}$$

Se calcula el valor de P_{ij}.

Se realiza el mismo proceso pero ahora se desbloquean las neuronas visibles por lo que ahora podrán ser seleccionadas.

Se hace un reajuste de pesos en la red

$$\Delta w_{ij} = \eta\left(P_{ij}^{+} - P_{ij}^{-}\right) \tag{2.8}$$

La idea principal en realizar este ajuste, consiste en que la red, cuando se encuentre en una situación estable no variará los pesos entre las neuronas visibles y ocultas.

Se ocupa el mismo proceso de la red antes mencionado hasta que Δw_{ij} sea lo suficientemente pequeño.

2.2.5 Aprendizaje de las Redes con Arquitectura Multicapa

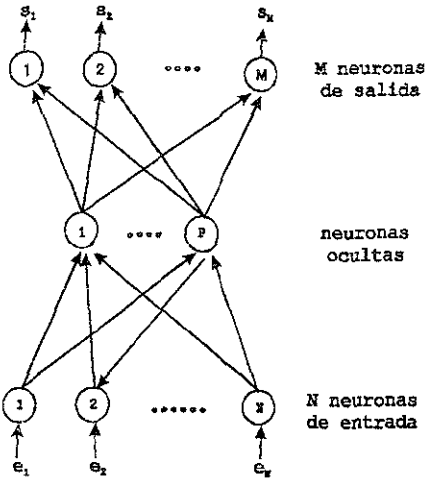

Figura 2.8

A diferencia de la red con arquitectura monocapa, en la red con arquitectura multicapa se divide la capa visible en capa de entrada y capa de salida, se dispone como en la red monocapa de una capa intermedia a la cual se le denomina capa oculta (figura 2.8). La capa de entrada sólo se ocupa para introducir los valores a la red, y las neuronas de salida son las que se bloquean y desbloquean. En esta red, después de ajustar la función de tiempo se calcula la diferencia de la salida deseada s_k y la salida obtenida s_k^h y se le denomina como $Error_1$

$$Error_1 = \frac{1}{2}\sum_{k=1}^{M}\left(s_k - s_k^h\right)^2 \qquad (2.9)$$

El siguiente paso que realiza la red es seleccionar aleatoriamente un peso entre

la capa de entrada y oculta o la capa oculta y de salida, y modificar levemente el valor. Luego se calcula el error que tiene la red realizando esta modificación y se ocupa la fórmula (2.8), a este error se le llama $Error_2$.

Después se asigna

$$\Delta E = Error_2 - Error_1 \tag{2.10}$$

Si $\Delta E > 0$ entonces se acepta la modificación del cambio de pesos seleccionados. Si $\Delta E < 0$ se decide tomando en cuenta la técnica de simulación (utilizada en la red monocapa), con la diferencia que en este caso, la función de probabilidad es la función de densidad de Boltzmann.

El otro tipo de aprendizaje con arquitectura multicapa difiere de la red monocapa en que la primera después de seleccionar una neurona oculta invierte su estado de activación

$$s_{no_j}(t) = \begin{cases} 1 & si \quad s_{no_j}(t-1) = 0 \\ 0 & si \quad s_{no_j}(t-1) = 1 \end{cases} \tag{2.11}$$

lo que no sucede en la red monocapa. Luego se calcula la energía global del sistema

$$\Delta E_j = \sum_{i=1}^{N} w_{ji} s_j + \sum_{h=1}^{M} v_{hj} s_j \tag{2.12}$$

Posteriormente el otro cambio que se tiene es que en esta red se calcula

$$p_{ij}^{+} \text{ y } q_{ij}^{+}$$

$$p_{ij}^{+} = \frac{1}{k} \sum_{k=11}^{k} Cor\left[e_i^{(k)} s_{no_j}^{(k)}\right] \tag{2.13}$$

$$q_{ij}^{+} = \frac{1}{k} \sum_{k=11}^{k} Cor\left[s_{no_j}^{(k)} s_h^{(k)}\right] \tag{2.14}$$

donde

Cor *Correlación*

Se selecciona un peso w ocupando el mismo proceso con la diferencia que la energía del sistema se calcula

$$\Delta E_j = \sum_{i=1}^{N} w_{ji} s_j \tag{2.15}$$

y posteriormente se calculan p_{ij}^{-} y q_{ij}^{-}.

2.2.5.1 Funcionamiento

Se inicia el funcionamiento de la red calculando la temperatura la cual depende del tiempo, e inicialmente se le da el valor de t=0. Posteriormente se asignan los valores de salida a la red, si se tiene una estructura de una sola capa, la visible se inicializa con el vector de entrada que se le proporciona, sin embargo si se tiene una red con varias capas se le asignan valores aleatorios a la salida de la red 0 y 1.

Después se selecciona aleatoriamente una neurona oculta y se determina la salida ocupando la técnica de simulación. Este proceso se realiza tantas veces como neuronas ocultas tenga la red, esto con la finalidad de dar la posibilidad de ser seleccionados por lo menos una vez. No se puede asegurar que cada una de las neuronas ocultas ha sido seleccionada ya que la selección de las neuronas ocultas es aleatoria. Luego se incrementa el tiempo t=t+1 y se repite todo el proceso.

2.2.6 Diferencias entre la Red de Boltzmann y de Cauchy.

La red de Cauchy [23] es un método mejorado de la red de Boltzmann, ocupa la misma arquitectura y método de funcionamiento que la red de Boltzmann pero tiene diferente función de probabilidad y función de temperatura. La distribución de probabilidad es la propuesta por Cauchy

$$P_{net}(s_i = 1) = \frac{1}{2} + arctan\left(\frac{net_i}{T}\right) \tag{2.16}$$

y la función de temperatura es (figura 2.7)

$$T(t) = \frac{T_0}{1+t} \tag{2.17}$$

Con estos cambios se logra una red que tiene mayor probabilidad de obtener una salida 1, por otra parte, es una red que en menor tiempo llega al mínimo global, debido a que es más rápido el descenso de la función de temperatura.

El autor de esta red ha demostrado, que ocupando la función de probabilidad y de temperatura, la red siempre converge al mínimo global.

Esta red tiene la ventaja de que puede aprender, de un conjunto de patrones dados en el proceso de aprendizaje, sus rasgos característicos y posteriormente, en el transcurso del funcionamiento, la red será capaz de identificar los rasgos característicos de cada uno de los patrones de entrenamiento que se le den e ignorar el "ruido".

Por otra parte durante el aprendizaje se tiene la ventaja de que no se requiere de la derivada de la función como en el caso de la red de Backpropagation, ya que en muchas ocasiones encontrar la derivada es mucho más complicado y requiere de más operaciones aritméticas para su cálculo que cuando se calcula la función. También debe de tomarse en cuenta que calcular la derivada en algunas ocasiones resulta más complicado que resolver el problema original.

Los cambios aritméticos para realizar reajustes son aleatorios así como la activación o no activación de una neurona. Esta aleatoriedad aunada a un proceso de temperatura ayuda a la red a converger al mínimo global. La limitante que presenta esta red es que requiere de la disminución paulatina de la temperatura, lo que la hace lenta.

2.3 Modelo NN

2.3.1 Vecino más cercano

A la regla del *vecino más cercano* se le denomina NN por sus sílabas en inglés Nearest Neighbor, este modelo tiene un método de clasificación supervisado y es no paramétrico.

Para que la red realice un aprendizaje adecuado se debe de tomar en cuenta los siguientes tres pasos.

1.- Identificación de los rasgos característicos.

2.-Realizar un clasificador, el cual tome los rasgos característicos de los patrones de entrenamiento para enseñar a la red como va a realizar el proceso de clasificación de un patrón, tomando en cuenta el problema práctico que se tenga para diseñar una regla que sea adecuada.

3.-Efectuar la clasificación de un patrón desconocido utilizando la red.

En este modelo se le enseña a la red a crear una función, la cual tome en cuenta los rasgos característicos de los patrones de entrenamiento que se le proporcionan y los mande a una sección diferente del espacio muestra, la cual constituirá cada una de las clases.

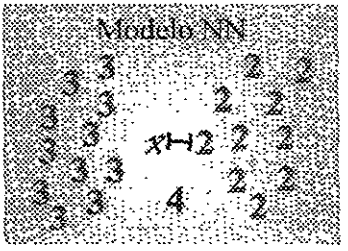

Figura 2.9

Posteriormente durante el funcionamiento de la red se clasificará un nuevo patrón, en este proceso lo primero que hace la red es identificar los rasgos característicos y ocupando la función los manda a cierto lugar en el espacio muestral, tomando en cuenta este lugar se asigna a este nuevo patrón x_i la clase w_k del vecino más cercano (figura 2.9).

Para ello se define la distancia como

$$d(X, x)) = mind(X, x_i) \qquad i = 1,2,.....n \tag{2.18}$$

Donde *d(*,*)* denota cualquier métrica conveniente, definida en el espacio p-dimensional de las variables.

Con esta regla se tiene como supuesto, que dos patrones que se encuentran cerca en el espacio de variables pertenecen a la misma clase con una alta probabilidad esta observación fue hecha por Devijiver y Kittler (1982) [24].

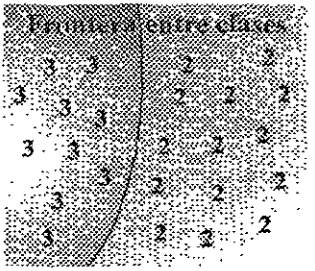

Figura 2.10

En este modelo se puede construir la frontera entre clases, para crear ésta se toma en cuenta la distancia de cada uno de los elementos de la muestra en el espacio de variables, con esto se sabe la clase a la que pertenece un patrón desconocido por el lugar en que se encuentra en el espacio de variable (figura 2.10).

2.3.1.1Ventajas

-Desde un punto de vista computacional es relativamente más fácil realizar un programa que haga la clasificación, y no se requiere de grandes recursos computacionales.

Se puede determinar las cotas de error en especial la cota superior de error, tomando en cuenta la probabilidad óptima de error, son pocos los modelos no paramétricos en los que se tiene esta prueba, esta investigación la realizó Cover y Hart (1967) [25] ellos se basan en las propiedades de convergencia del modelo NN.

2.3.1.2 Limitaciones

-Con este método sólo se tienen una aproximación de la distribución, lo cual sucede por tratarse de un método "elemental". El método no se utiliza para estimaciones de ningún tipo.

-Este modelo es sensitivo al orden en que se codifican los patrones de entrenamiento, pero no a la trayectoria.

Existen algunos problemas que han surgido con la práctica como por ejemplo:

1.-Extracción de los rasgos característicos para llevar a cabo la clasificación.

2.-Cuando se requiere mejora el método de clasificación

3.-El caso en que se tienen una población limitada.

4.-Como reducir el tamaño de una población y sacar una muestra que sea representativa.

5.-La situación en que los patrones no se encuentran clasificados en una sola clase o cuando esta labor es una tarea difícil y costosa.

6.-Los patrones de entrenamiento se encuentran en diferentes proporciones en cuanto al tamaño de la clase.

Algunas de estas limitaciones ya han sido estudiadas y se han propuesto soluciones las cuales se desarrollan a continuación, también se mencionará sus ventajas y limitaciones, ya que en muchas ocasiones con base a éstas se propone un nuevo modelo.

2.2 Método para mejorar el Rendimiento del Clasificador

2.4.1 Modelo k-NN

Para mejorar el método de clasificación, surge la idea de tomando en cuenta la clase a la que pertenecen los k vecinos más cercanos y asignar la clase que tenga mayor número de representantes, esta regla fue propuesta por Fix y Hodges (1951), estos autores tienen la importancia que son los primeros que proponen un método no paramétrico con una función de distribución multivariada.

Modelo k-NN

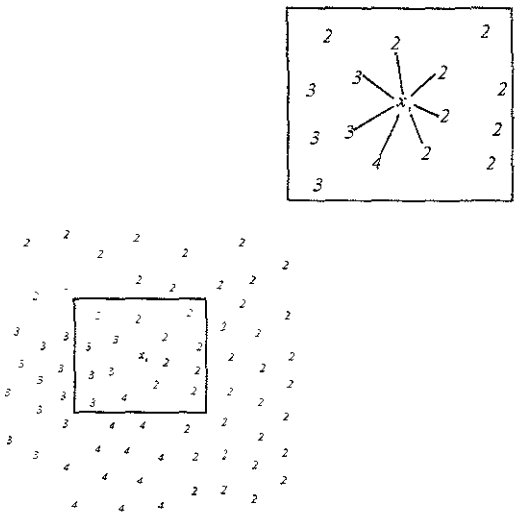

Figura 2.11

En este modelo es necesario formar una vecindad la cual se encuentra constituida por k-vectores de entrenamiento de los cuales se conoce la clase a la que pertenece cada uno de estos patrones, en el centro de la vecindad se encuentra un patrón desetiquetado a este patrón se le asigna la etiqueta de la clase que tenga mayor frecuencia, si existe un empate entre dos o más clases, entonces se asigna la clase aleatoriamente, tomando en cuenta la clase con mayor número de representantes (figura 2.11).

Para este modelo es necesario asignar el número de patrones de entrenamiento que forma la vecindad, este número depende del problema específico que se esté resolviendo.

2.4.1.1 Ventajas

-Es un modelo no paramétrico.

-Cuando se asigna un patrón de entrenamiento en el espacio de variables no se corre el riesgo de exactitud en la asignación.

-Se tiene la facilidad de tener diferentes vecindades con varias formas.

-Algunos modelos del vecino más cercano pueden ser obtenidos a partir del modelo de Bayes, a este modelo se le denomina de máxima probabilidad y realiza una excelente clasificación siempre y cuando se haya hecho un buen proceso de aprendizaje.

-Cuando existe un empate entre clases este modelo es el mejor para realizar el proceso de clasificación.

2.4.1.2 Limitaciones

-Requiere mayores recursos computacionales que el modelo NN.

-Se tienen las ventajas y limitaciones de un método no paramétrico.

2.4.2 Regla NN con entrenamiento editado.

En este modelo se tiene la finalidad de disminuir el error lo cual trae como consecuencia un incremento en la confiabilidad del modelo. Wilson (1972) [26] propone la regla NN basada en un entrenamiento editado en el cual se realizan los siguientes pasos:

-Para cada una de las muestras se encuentran sus k vecinos.

-Posteriormente se asigna la clase x_i tomando en cuenta la clase que tenga el mayor número de representantes, entre los k vecinos o se decide de manera aleatoria en caso de existir un empate entre clases.

-Se repite el paso uno y dos nuevamente y se eliminan todos aquellos patrones que hallan sido clasificados en clases diferentes.

Durante el proceso de funcionamiento, la red ocupará el modelo NN para realizar la clasificación de un patrón desconocido

2.4.3 Regla NN con Opción a Rechazo.

Cuando se tiene un modelo de este tipo se supone que el error es mayor en el momento en que se tienen todos los patrones que cuando se eliminan algunos, se considera también que los patrones que se eliminan posteriormente se podrán someter a un análisis más profundo.

En Hellman (1970) tomando en cuenta los supuestos antes mencionados propone la regla (2,2) NN de clasificación. La cual consiste en que si la clase de los dos patrones más cercanos pertenecen a la misma clase y se cumple que el valor de w que se define a continuación:

$$w = \frac{\textit{distancia de } x_i \textit{ a su primer vecino}}{\textit{distancia de } x_i \textit{ a su segundo vecino}} \qquad (2.19)$$

es menor que T, el cual se encuentra predeterminado entonces se le asigna esta clase al patrón desconocido.

Si la clase de los dos vecinos más cercanos no es la misma entonces no existe evidencia suficiente para asignar una clase al patrón desconocido x_i.

Luego se propuso la generalización de esta regla llamada la regla (k, k) NN que requiere que sus k vecinos representen a la misma clase para no rechazarla.

2.5 Métodos para reducir el Tamaño de la Muestra.

2.5.1 Reducción del Tamaño de la Muestra

En algunas ocasiones existen muestras que son muy grandes, lo cual trae como consecuencia que se requiera de altos recursos computacionales, y que existan muestras que sean imposibles de ocupar para entrenar a la red por sus altos recursos computacionales.

En muchas ocasiones es necesario analizar el conjunto de muestras que se presentan a la red como representantes de cada clase, ya que no es necesario proporcionarle patrones que aporten la misma información (suponiendo que se tiene una muestra grande), mediante el análisis multivariado o ocupando el muestreo para obtener una muestra que sea representativa.

Debe de cuidarse también que el número de patrones representantes de cada clase sea más o menos el mismo número, en caso contrario posteriormente en una sección de este capítulo se mencionarán algunos cuidados que se deben de tener para una clasificación óptima.

Algunos autores han propuesto métodos para reducir el tamaño de la muestra, como, Hart (1968) [27] propone un método en el cual supone que todos los patrones de entrenamiento se encuentran ordenados por algún proceso. Posteriormente se hacen dos grupos a los cuales se les llama conservar y desechar, se coloca el primer patrón de entrenamiento en conservar y los demás en desechar. Como la finalidad del algoritmo de Hart es desechar patrones de entrenamiento, se considerará como éxito el desechar un patrón, por lo que si un patrón resulta bien clasificado se descartará y por analogía se considera como fracaso conservar el patrón de entrenamiento. Este proceso se repite con cada uno de los patrones de entrenamiento, es decir para i = 1, ..., n_g.

Se repite este proceso hasta que se tenga un ciclo y no halla ningún cambio en los grupos de conservar y desechar El algoritmo termina cuando se ha tenido un ciclo y no ha habido cambios entre el grupo de conservar y desechar o

cuando todos los patrones de entrenamiento se encuentran en conservar. Lógicamente se queda con el conjunto que se ha formado en conservar.

2.5.1.1Ventajas

-Es un buen método para eliminar patrones que tienen características similares.

-Se tiene una buena aproximación para encontrar un subconjunto que sea consistente.

2.5.1.2 Limitaciones

-Existe una fuerte dependencia de orden para seleccionar el patrón de conservar, lo cual ocasiona que la solución no sea única.

-Con el algoritmo propuesto por Hart, hay patrones de frontera que no se encuentran representados en la muestra seleccionada, y sin embargo se selecciona patrones que son innecesarios para realizar una correcta clasificación.

-Se tiene la posibilidad que la clase que se seleccione se encuentre tendida hacia el patrón inicial que se dio.

2.5.2 Método Híbrido o Compuesto.

Se puede considerar que existen dos métodos para reducir el tamaño de la población, el método editado para disminuir el tamaño de la población en un 20%, y el subconjunto consistente el cual reduce la población en un 70%, lógicamente los dos porcentajes antes dados varían de acuerdo al problema específico que se esté resolviendo. Tomak (1976) [28] se le ocurrió que si primero se ocupa el método editado y posteriormente se obtiene una muestra consistente entonces se tendría un mejor clasificador ya que se tendría una muestra más limpia, esta observación la hace notar al final de su artículo, pero no fue verificada hasta que Devijiver y Kittler (1982) realizan un experimento aplicando estos dos métodos al cual le llamaron "Método Híbrido" en este

experimento se reduce la población en un 90% y se tiene una disminución en la clasificación errónea.

2.5.3 Conjunto Selectivo

Riter y sus colegas proponen un método el cual tiene la gran ventaja que no se basa en la intuición, ya que ellos definen de forma clara y precisa sus objetivos, y realizan la formalización matemática adecuada para obtenerlos.

Su método tiene la finalidad de obtener un subconjunto, el cual tenga las característica de frontera del conjunto original. Ello resalta la importancia de que el subconjunto tenga esta característica, ya que con ello se puede garantizar la mejor aproximación, desafortunadamente los modelos posteriores no toman en cuenta esta característica.

Los criterios que proponen estos autores son:

1. - El subconjunto debe ser consistente.

2. - Todos los integrantes del subconjunto deben de encontrarse más cercanos a un elemento de la misma clase que a cualquier elemento de otra clase.

3. - No debe haber un conjunto que satisfaga 1 y 2 y que tenga más elementos.

2.5.3.1Ventajas

-Este modelo resuelve el problema de orden del algoritmo de Hart, desafortunadamente ésta no es la única limitación que se tiene.

-Debe de notarse que el criterio 1 es innecesario, ya que todo subconjunto que satisface el criterio 2 son consistentes.

2.6 Situaciones Imperfectamente Supervisadas.

En algunas ocasiones es complicado tener identificado cada uno de los patrones que integran la muestra de entrenamiento, debido a que es una tarea difícil y costosa. Ésta es la causa por la que pueden existir patrones que se encuentren como representantes de una clase a la que en realidad no

pertenecen. En este caso se ocupa un método intermedio entre el método supervisado y no supervisado al cual se le denomina método imperfectamente supervisado.

2.6.1 Vecindad Mutua

El método llamado valor de vecindad mutua propuesto por Gowda y Krishna (1979) [29], tiene como objeto principal ocupar el conocimiento adquirido en el proceso de aprendizaje con la información suministrada para la identificación.

Se define el valor de la vecindad mutua (MNV, por sus silabas en inglés Mutual Neighborhood Value) como sigue: Sea x_i el q- ésimo vecino más cercano de x_j donde q representa el q- ésimo lugar en orden creciente tomando en cuenta la distancia con respecto a x_i, de igual manera se determina el q- ésimo vecino de x_j, tomando en cuenta estos dos valores se tiene que:

$$MNV = m + q \tag{2.20}$$

donde m y q varían de 0 a n-1.

En este proceso se considera que x_j es vecino de x_i si se encuentra dentro de sus k vecinos más cercanos con respecto a su distancia, de lo contrario se dice que x_j no es vecino de x_i.

Se define V_{ij} que representa el peso de x_j con respecto de x_i como:

$$V_{ij} = \frac{1}{MNV(x_i, x_j)} \qquad x_i \neq x_j \tag{2.21}$$

donde los pesos que se asignan toman en cuenta la distancia.

El algoritmo propuesto por Gowda y Krishma tiene dos etapas, la primera consiste en obtener para cada x_i

$$V_{i1} = \frac{1}{MNV(x_i, x_j)} \tag{2.22}$$

donde x_j es uno de los k vecinos más cercanos a x_i y que representa la clase 1.

Luego de manera similar se calcula

$$V_{i2} = \sum_{t} \frac{1}{MNV(x_i, x_t)} \tag{2.23}$$

donde x_t es uno de los k vecinos más cercanos de x_i y representa a la clase 2 posteriormente se calcula el valor de LMN (Label Mutual Neighborhood) donde

$$LMN = \begin{cases} 1 & V_{i1} > V_{i2} \\ 0 & e.o.c. \end{cases} \tag{2.24}$$

Si la vecindad LMN coincide con la vecindad que tenía originalmente x_i entonces se le asigna esta vecindad a x_i y se ubica en el grupo A, en caso contrario se le asigna a x_i la clase original que tenía y se coloca en el grupo B. Posteriormente de que se ha hecho este procedimiento para todos los x_i de la muestra de entrenamiento se tiene en el grupo A la muestra editada.

En la segunda etapa ocupando los patrones que se encuentran en el grupo B, se buscan sus k vecinos más cercanos del grupo A. Luego se genera una nueva etiqueta NLMN de acuerdo con el mayor número de patrones que pertenezcan a una misma clase de los k vecinos más próximos de x_i

Si LMN=NLMN entonces a x_i se le asigna esta etiqueta, en caso contrario se le asigna a x_i la clase original que tenía. Al finalizar este proceso, la muestra tiene el mismo número de patrones solo que algunos habrán cambiado de identificación.

2.6.2 Edición General

El modelo de la Edición General fue propuesto por Browman (1978). En este modelo se buscan los k vecinos más cercanos de x_i, si se tiene por lo menos k* vecinos que pertenecen a una misma clase entonces se le asigna esta clase a x_i (no importa la clase original que haya tenido), de lo contrario se elimina a x_i.

Se ha notado que existe una fuerte dependencia entre el número de patrones que se encuentran mal clasificados originalmente con el número de patrones nuevos que estarán erróneamente clasificados.

2.6.2.1 Limitaciones

-Existe la posibilidad que un patrón que en un principio se encuentra bien clasificado después de este proceso se encuentre en una clase a la que no pertenece.

Posteriormente se ha empleado este método en forma reiterada con la finalidad de tener una muestra menos contaminada, es decir, una muestra con un menor número de clasificaciones erróneas. Algunos autores se han ocupado de estudiar cual es la cantidad de repeticiones ideal y encontraron que es tres, puesto que después de este número el error que se tiene, tiende a estabilizarse por lo que de la cuarta reincidencia en adelante los cambios que se tienen en la distancia del error son pequeños.

En algunos experimentos reales se observó que existe un grupo de patrones de entrenamiento los cuales sufren recursivamente una reidentificación, ya que pasan de la clase g a la clase h en la primera interacción, y en la segunda aplicación un grupo de ellos pasa de la clase h a la clase g, después en la tercera renovación un conjunto de patrones de entrenamiento pasan de la clase g a la h, y así sucesivamente, siendo el grupo de patrones de entrenamiento que pasan de una clase a otra cada vez menor, a este efecto se le denominó efecto de "péndulo", y se notó que el conjunto de patrones que saltan de una clase a otra se eliminan cuando se aplica el método de Wilson.

2.7 Ponderación de la Regla de Clasificación

Una de las críticas que se le han hecho al modelo k-NN es que no hace una diferencia entre los k vecinos más cercanos. Es por ello que Dudani (1976) [30] propone un modelo el cual realiza la diferencia de cada uno de los patrones de entrenamiento, tomando en cuenta sus distancias de cada uno de ellos con respecto al patrón x_i (patrón desetiquetado).

En este modelo durante el proceso de aprendizaje se crea una función, la cual ocupando los rasgos característicos (los cuales serán asignados) proporciona un lugar en el espacio de variables, posteriormente tomando en cuenta los k vecinos y la cercanía de éstos, se registra la clase a la que pertenecen, se ordenan de forma creciente, luego se le asigna un peso tomando en cuenta la clase a la que pertenecen cada uno de los k vecinos, y se asigna a x_i la clase que tenga mayor peso.

Para asignar el peso de cada patrón Dudan propone dos reglas:

a) Ponderación simple

$$w_j = \frac{1}{d_j} \qquad si \quad d_j > 0 \tag{2.25}$$

b) Ponderación según el orden

$$w_j = k - j + 1 \tag{2.26}$$

en este caso k toma valores desde k hasta 1.

Dudani realizó estudios comparando la regla mayoritaria con alguna de sus tres métodos propuestos y notó que cuando existe empate entre clases la regla k-NN tiene un mejor comportamiento que cualquiera de los métodos propuestos por Dudani.

2.8 Método utilizado en el caso en que el Tamaño de las Clases es de Diferente Proporción

Brown y Koplowitz (1979) [31] se preocuparon por saber el comportamiento de la regla NN en el momento en que la proporción de patrones de entrenamiento representantes de cada clase son muy diferentes. La motivación de este análisis fueron los estudios hechos por Levine y sus colegas (1973) los cuales analizaron como se comportan las poblaciones con distribución uniforme multivariada con igual probabilidad a priori, sus conclusiones a grandes rasgos son:

a) La mejor selección de la cantidad de patrones representantes de cada clase, es la que se encuentra en función de la probabilidad a priori.

b) Si una clase tiene un número pequeño de representantes no es necesario "maquillar" esta deficiencia, ya que se tienen mejores resultados cuando ambas poblaciones tienen un número pequeño de representantes.

Brown y Koplowitz rechazaron esta última aseveración ya que esto implica que si existen elementos en exceso es necesario eliminarlos antes de ocupar la regla NN, lo cual no parece razonable ya que se desecharía información.

Su idea consiste en incrementar la distancia de los patrones que se encuentran más representados, con la finalidad de que todos los patrones tengan la misma oportunidad de ser seleccionados.

La diferencia con respecto de otros modelos consiste en la forma de calcular la distancia. Ahora la distancia del patrón x a un patrón de entrenamiento x_o que representa la clase i ésima se calcula:

$$d\left(x, x_0\right) = \left(\frac{n_i}{p_i n}\right)^{1/p} d\left(x, x_0\right) \qquad (2.27)$$

donde

n_i	es la cantidad de representantes de la clase i- ésima en la ME.
p_i	es la probabilidad a priori de la clase i- ésima
n	es la cantidad total de patrones en la ME.
p	es la dimensión del espacio, o cantidad de variables que se consideran.

Sobre este tema aún existen muchas dudas, por ejemplo Hardin (1994) muestra preocupación por la forma de calcular la probabilidad a priori, ya que tomando en cuenta esta probabilidad es la mejor forma de seleccionar el número de representantes (patrones de entrenamiento) de cada clase.

Capítulo 3

3.1 Clasificación de Nanoestructuras de Oro

Para realizar la clasificación de nanoestructuras mediante imágenes de Alta Resolución se sugiere a la red de Dudani, la cual tiene la ventaja de tener un método de aprendizaje imperfectamente supervisado lo que permite tener algunos patrones mal clasificados debido a que una imagen tenga rasgos característicos de dos clases distintas lo que dificulta su clasificación.

Se propone durante el proceso de aprendizaje, la función que sea más adecuada, la cual tome los rasgos característicos (número de lados, número de ángulos iguales, número de direcciones de líneas paralelas y algunas otras características más específicas que permitan diferenciar por lo general dos patrones que se encuentren en la misma clase) de la nanoestructuras que se le proporciona y los mande a un lugar en el espacio de variables y posteriormente tomando en cuenta los k vecinos y la cercanía de éstos con el vector x_i (patrón desetiquetado) se asigna una clase.

3.1.1 Nanoestructura

Una nanoestructura es un conjunto de átomos de escala 10^{-9}m los cuales en algunas ocasiones se comportan como un solo átomo y en otros casos como una macromolécula.

Las nanoestructuras se agrupan de diferentes maneras, hasta el momento el principal fundamento que se tiene para esta agrupación es que todo sistema tiende a un estado de mínima energía, en el cual alcanza su mayor estabilidad. La forma en que se agrupan tiene una gran importancia ya que ésta se encuentra relacionada a sus propiedades y como consecuencia en lo que se puede aplicar. Las nanoestructuras metálicas generalmente tienen agrupamientos geométricos como tetraedro, icosaedro y decaedro.

Se obtiene información de las nanoestructuras a través de espectros, imágenes o gráficas de difracción. El aparato que se ocupan principalmente es el

Microscopio Electrónico de Transmisión de Alta Resolución debido a que proporciona una excelente imagen y da una buena resolución, también proporciona el orden de las columnas de los átomos lo que ayuda a tener un correcto análisis de las nanoestructuras, por lo que es preferido.

Algunos autores han pensado que para realizar una caracterización completa es necesario producir partículas con tamaños, formas y estructuras controladas y analizar sus propiedades; actualmente se han estudiado partículas mayores a 40A° pero con estructuras menores a este rango actualmente se analizan.

Para realizar un estudio de partículas que se encuentren en orientaciones de bajo índice es complicado debido a que se colocan en un soporte de carbón amorfo lo que hace que se encuentren partículas con diferentes orientaciones, por lo que surge la necesidad de conocer la nanoestructuras en diferentes orientaciones. Para que este estudio sea sistemático se ha hecho un catálogo [32] con imágenes de HREM las cuales se obtienen del Cerius^2, el cual permite tener imágenes simuladas con orientaciones controladas así como el tamaño (estas imágenes simuladas se ocuparán para entrenar a la red).

Estas imágenes simuladas permiten que la clasificación de las imágenes experimentales sea menos complicada, ya que se pueden comparar las imágenes simuladas con las imágenes experimentales. Se sugiere en este capitulo una forma rápida y objetiva de como realizar esta clasificación entrenando a una red.

3.1.2 Arquitectura

La arquitectura que se sugerirá es una red multicapa, la cual tenga una capa de entrada, capa oculta y capa de salida, lo cual le permite a la red dar múltiples pesos a los diferentes rasgos característicos que se tomen en cuenta, para que la red identifique la clase y la posición de la imagen que se le proporciona.

A la red en un principio se le asignan pesos aleatorios los cuales va ajustando durante el proceso de aprendizaje.

3.1.3 Entrenamiento

Para entrenar a la red se emplearán imágenes de Alta Resolución de Oro. Primero se le enseñarán a la red patrones de entrenamiento simulados, ya que este es el caso de la idealidad en el que todos los patrones de entrenamiento se encuentran con todas sus características, con esto se pretende que la red extraiga los rasgos característicos del conjunto de patrones de entrenamiento que se le proporcionan, posteriormente se juntan los patrones de entrenamiento simulados y experimentales en estos últimos en algunas ocasiones los patrones se encuentran incompletos o con ruido lo que hace que a la red le cueste mas trabajo identificarlos. Para obtener un subconjunto de patrones simulados se ocupará el Muestreo Aleatorio Simple con el conjunto *total de patrones simulados y con estos se entrenará a la red, posteriormente* se unen los patrones simulados y experimentales y se realiza el mismo proceso.

3.1.4 Muestreo Aleatorio Simple.

La finalidad de este muestreo es seleccionar entre un conjunto de patrones (población) de entrenamiento un subconjunto, el cual se pretende que tenga todas las características del conjunto de patrones de entrenamiento y en la misma proporción (figura 3.1), esto se realizará si se tiene un conjunto grande de patrones de entrenamiento.

Muestreo Aleatorio Simple

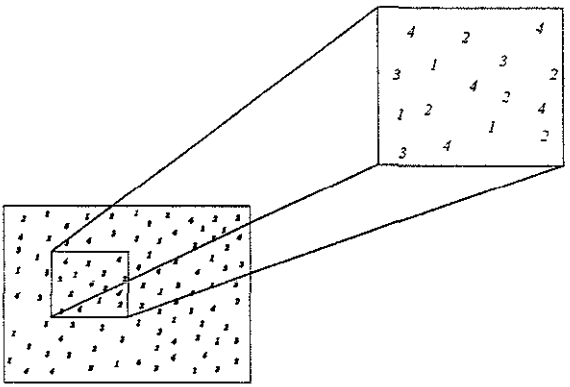

Figura 3.1

Se supondrá que el conjunto de patrones que se tiene no se encuentra tendida a una clase o posición, es decir, se considera que cualquier clase y posición se pueden obtener con la misma probabilidad. Para tener un subconjunto de patrones de entrenamiento se ocupa el muestreo aleatorio simple. Si no se tiene este supuesto es conveniente analizar el tipo de población que se tiene y ocupar otros método de muestreo como por ejemplo el muestreo por conglomerados.

Se supone que se tiene una población de N elementos y se quiere obtener una muestra de n de ellos.

Para obtener la muestra es conveniente enumerar cada uno de los patrones de *entrenamiento* a los cuales se les considera como la población N. Posteriormente para seleccionar la muestra se pueden ocupar dos métodos:

- La Tabla de Números Aleatorios.
- Algún paquete estadístico que genere con la opción de Random la cantidad de números aleatorios que se requieran.

Con estos dos métodos se tiene la ventaja de la aleatoriedad y se elimina la posibilidad de tener un sesgo.

El uso de la Tabla de Números Aleatorios es el siguiente, se selecciona cualquier posición de esta tabla considerando los últimos dígitos de esta posición, se debe de tomar en cuenta que este número y los que se seleccionan para la muestra sean menores que N.

Posteriormente se obtienen los números que integran la muestra tomando en cuenta los números que se encuentran abajo de la columna del número seleccionado, también se pueden considerar los números que se encuentran en el *mismo renglón.*

Los patrones de entrenamiento se seleccionan sin reemplazo y cada una de las extracciones es independiente, por lo que la probabilidad de cada extracción es

$$\frac{1}{n} \qquad i = 0, 1, 2, \ldots, n-1 \tag{3.1}$$

El número total de todas las posibles muestras que se pueden tener es

$$\binom{N}{n} = \frac{N}{(N-n)!n!} \tag{3.2}$$

donde

N Población total o conjunto de patrones.
n Tamaño de la muestra o subconjunto de patrones.

Se considera que las muestras que tienen los mismos patrones de entrenamiento aunque en distinto orden son iguales.

La probabilidad de obtener una muestra particular está dada por:

$$\frac{1}{\binom{N}{n}} \tag{3.3}$$

3.1.4.1 Fórmulas más usadas.

3.1.4.1.1 Media Muestral

Cuando se ocupa el Muestreo Aleatorio Simple se obtiene una muestra que tiene el mismo comportamiento que la población, por lo que es necesario conocer algunos parámetros como la media

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{3.4}$$

y_i i-ésimo elemento de la muestra.
n Número de unidades en la muestra

3.1.4.1.2 La Varianza con Respecto a la Media.

$$\hat{V}(\bar{y}) = \left(\frac{N-n}{n}\right)\frac{S^2}{n} \tag{3.5}$$

donde

$$S^2 = \frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{n-1} \tag{3.6}$$

s^2 **Varianza muestral de la característica de interés.**
N Total de elementos en la población.

3.1.4.1.3 Intervalo de Confianza.

Debido a que se no se sabe con precisión el valor de la media poblacional sería apropiado proporcionar un intervalo de confianza en el cual se asegure con cierto nivel que se encuentra el valor de la media poblacional.

$$y \pm Z_{(1-\alpha/2)} \sqrt{\left(\frac{N-n}{N}\right)\frac{S^2}{n}} \tag{3.7}$$

$Z_{(1-a/2)}$ **Distribución normal.**

3.1.4.1.4 Tamaño de la Población.

En algunas ocasiones se realiza una muestra con la finalidad de estimar el total de la población

$$\hat{Y} = N\bar{y} \tag{3.8}$$

3.1.4.1.5 Varianza de la Población.

La varianza muestral del total de la población se encuentra determinada por

$$\hat{V}(\hat{Y}) = N^2\left(\frac{N-n}{N}\right)\frac{S^2}{n} \tag{3.9}$$

3.1.4.1.6 Intervalo de Confianza.

El intervalo de confianza para el total de la población es

$$\hat{Y} \pm NZ_{(1-\alpha/2)}\sqrt{\left(\frac{N-n}{N}\right)\frac{S^2}{n}} \tag{3.10}$$

Cuando se realiza el Muestreo es necesario determinar el tamaño de muestra, ya que si se tiene una muestra más grande de lo necesario esto implica que se ocupa más tiempo y dinero del necesario, si la muestra es muy pequeña ésta no representa la población que se desea, o bien no posee las características deseadas de estimación.

Debe de notarse que se requiere de la varianza poblacional para el cálculo del tamaño de muestra, este valor se puede obtener de una muestra piloto o de una encuesta previa.

3.1.4.1.7 Tamaño de la Muestra.

El tamaño de la muestra se encuentra determinado por

$$n = \frac{n_o}{1 + \frac{n_o}{N}} \tag{3.11}$$

en la cual

$$n_o = \frac{Z^2_{(1-\alpha/2)}\sigma^2}{d^2} \tag{3.12}$$

$$\sigma^2 = \frac{\sum_{i=1}^{N}\left(Y_i - \bar{Y}\right)^2}{N-1} \tag{3.13}$$

$$\bar{Y} = \frac{\sum_{i=1}^{N} Y_i}{N} \tag{3.14}$$

Y_i i-ésimo elemento de la población.

$\bar{Y}$ Media Poblacional.

σ^2 **Varianza poblacional de la característica.**

d^2 **Precisión.**

En este problema específico se ocupará la red de Dudani ya que se considera que cada patrón de entrenamiento y cada posición es una muestra diferente; por ejemplo el octaedro truncado [111] y el octaedro truncado [011] son muestras distintas, si se deseara que la red proporcionará el grupo al que pertenece partículas cúbicas (fcc) en esta clase se encuentra el cuboctaedro y sus truncamientos en diferentes grados los cuales son el octaedro truncado y el tetracaidecaedro, en otra clase se encuentra el decaedro y icosaedro.

Se tendría el caso en que una de las clases se encuentre más representada que las otras dos, por lo que es apropiado ocupar la red de Brow y Koplowitz quienes estudiaron este acontecimiento, aunque para calcular la distancia, la cual incrementan en los patrones de entrenamiento que se encuentran más simbolizados es necesario saber la probabilidad a priori de cada clase y la forma de obtener ésta aún no se encuentra bien estudiada, como lo hizo notar Hardin, por lo que sería apropiado ocupar la red de Boltzmann.

Para que la red Dudani realice un aprendizaje adecuado se debe de tomar en cuenta como primera característica el número de lados

4 lados	Cuboctahedral	[011]
6 lados	Cuboctahedral	[011], [111]
	Octaedro Truncado	[011]
	Tetrakaidecaedral	[011]
	Decahedro de Marks	[110]
	Icosahedral	[001], [112]
8 lados	Octaedro Truncado	[001]
	Tetrakaidecaedral	[001]
	Decahedro de Marks	[021]
10 lados	Icosahedral	[11-1]
12 lados	Octaedro Truncado	[111]
	Tetrakaidecaedral	[111]
15 lados	Decahedro de Marks	[001]

si el patrón de entrenamiento tiene 4, 10, 15 lados entonces se tiene

Cuboctahedral [001], Icosahedral [11-1], y Decahedro de Marks [001] respectivamente.

Si el número de lados es 6 entonces para identificar qué clase y qué posición se tomará en cuenta si el patrón tiene todos los ángulos iguales y se forma con estos una clase y otra si por lo menos uno de sus ángulos es diferente:

$\alpha^o = \beta^o = \gamma^o$	Cuboctahedral	[011], [111]
	Icosahedral	[001], [1-12]
$\alpha^o \neq \beta^o = \gamma^o$	Octaedro Truncado	[011]
ó	Tetrakaidecaedral	[011]
$\alpha^o = \beta^o = \gamma^o$	Decahedro de Marks	[110]

Esta misma característica se ocupará cuando se tienen 8 lados, pero en este caso no es necesario ya que todos los patrones que se encuentran en esta clase tiene por lo menos un ángulo diferente y entonces todos quedarían en la misma clase.

La tercera característica en el caso en que se tiene una imagen con 6 lados, es el número de direcciones de líneas paralelas que se encuentra definidas y se toman en cuenta tres clases: 0 direcciones de líneas paralelas, 1 dirección de líneas paralelas, 2 o más direcciones de líneas paralelas.

a‖b‖c	Cuboctahedral	[011]
a⋪b⋪c	Cuboctahedral	[111]
	Icosahedral	[001], [1-12]
a‖b‖c	Octaedro Truncado	[011]
	Tetrakaidecaedral	[011]
a⋪b‖c o a‖b⋪c	Decahedro de Marks	[110]

Con esta característica se coloca en una clase al Cuboctahedral [001].

Se tiene en una misma clase al Cuboctahedral [011] y Icosahedral [001], [1-12]. se toma en cuenta como cuarta característica el que la imagen tenga un átomo central. Con esta característica se coloca en una clase al Icosahedral [001], el cual no tiene un átomo central y en otra clase al Cuboctahedral [111] y Icosahedral [1-12], para distinguir estos dos últimos se divide el hexágono en 6

partes partiendo del punto central de la imagen, si todos las secciones son iguales entonces es un Cuboctahedral [111], si por lo menos una sección es diferente entonces es un Icosahedral [1-12]..

Para el caso en el que se tiene por lo menos un ángulo diferente y una dirección de líneas paralelas se clasifica el Decahedro de Marks [110], si se tienen 2 o más líneas paralelas entonces se encuentra en la clase al Octaedro Truncado [011] y Tetrakaidecaedral [011], para colocar estos dos últimos patrones en clases diferentes se toma en cuenta el número de elementos que hay en la imagen en uno de los lados mas pequeños si son 2 entonces es un Octaedro Truncado [011] y si son 3 entonces es el Tetrakaidecaedral [011]

Se tomara en cuenta esta misma clasificación cuando se tiene una imagen con 8 lados

a||b||c	Octaedro Truncado	[001]
	Tetrakaidecaedral	[001]
a\Nb\Nc	Decahedro de Marks	[121]

Tomando en cuenta esta característica se tiene al Decahedro de Marks [110] el cual no tiene ninguna línea paralela.

La tercera característica que se ocupa para diferencia el Octaedro Truncado [001] y Tetrakaidecaedral [001] es el número de elementos que se requieren para formar un triángulo rectángulo, tomando en cuenta uno de los lados que haya resultado más pequeño, si el número de elementos que se requieren para formar un triángulo rectángulo son 4 entonces es un Octaedro Truncado [001], y si son 6 entonces es un Tetrakaidecaedral [110].

Un ejemplo de como se realiza la clasificación del Cuboctahedral [001], se muestra en la (figura 3.2) en la cual el número de lados en la primera característica son 6, todos los ángulos son iguales y se tienen más de dos líneas paralelas, posteriormente en el último paso la función asigna un lugar en el espacio muestral.

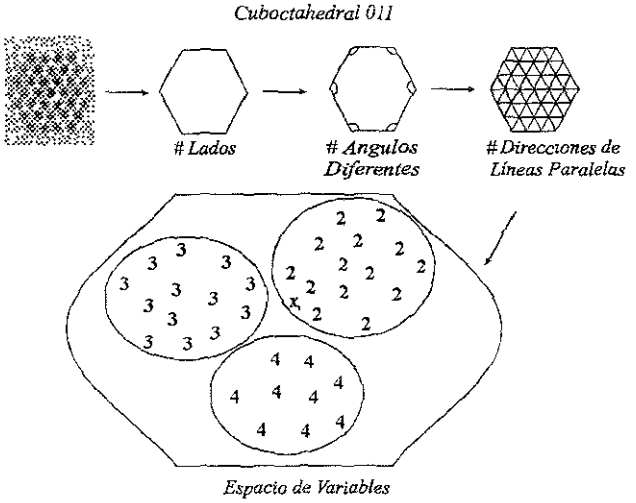

Figura 3.2

Con esto lo que se hace es darle a la red las características que debe de tomar en cuenta para crear una función la cual proporcione un lugar en el espacio de variables para x_i (patrón desetiquetado), posteriormente se identifican cuales son los k vecinos del patrón x_i y se calcula la distancia con respecto de este, se registran las clases a la que pertenecen cada uno de los k vecinos (figura 3.3), luego se ordenan las distancias de forma creciente y se les asignan pesos ocupando la siguiente fórmula:

$$w_j = \frac{1}{d_j} \quad si \quad d_j > 0 \tag{3.15}$$

Después se suman los pesos tomando por clase y se asigna al patrón x_i, la clase que tenga la mayor suma.

Con estos rasgos lo que se hace es darle a la red las características que debe de tomar en cuenta para crear una función la cual proporcione un lugar en el espacio muestral.

Modelo de Dudani

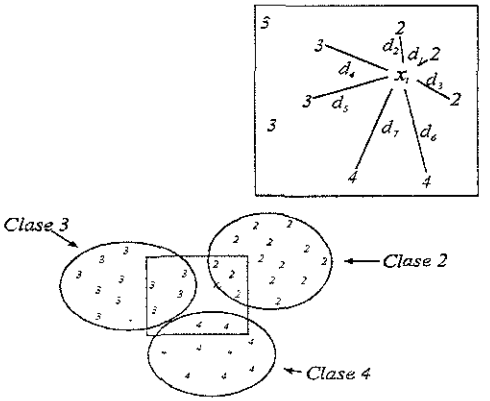

Figura 3.3

Durante el proceso de funcionamiento se le dará a la red un patrón experimental y ésta lo clasificará tomando en cuenta los rasgos característicos de cada nanoestructura como el número de lados, ángulos, líneas paralelas, y algunas otras características las cuales dependen del patrón que se este clasificando (las cuales ya se mencionaron en el proceso de aprendizaje), con estas características la red propondrá un lugar en el espacio de variables, posteriormente se identifica el vecino más cercano del patrón x_i y se le asigna esta clase al patrón x_i .

Es importante hacer notar que otros autores no han resuelto este problema, de esta forma, por lo que se tendrá que experimentar cuando se entrene a la red con el número más adecuado de vecinos que se deben de tomar en cuenta para que la clasificación sea óptima. Si existe un empate entre clases es preferible ocupar el método k-NN ya que realiza una mejor clasificación.

Se juzga que en esta red el factor de ajuste que se tiene es la clase a la que pertenecen los k vecinos, se tiene como supuesto que dos patrones que se encuentran cerca en el espacio de variables pertenecen a la misma clase con una alta probabilidad.

Puede considerar que este es uno de los mejores modelos ya que da pesos tomando en cuenta la distancia a la que se encuentra x_i con respecto a cada uno de sus k vecinos y éste es uno de los factores más importantes.

Tiene el beneficio de que es un modelo no paramétrico por lo que durante el proceso de aprendizaje se creará la función que se requiera.

Al ocupar este modelo se tiene la ventaja que la red ocupa un método imperfectamente supervisado (ver sección 2.6) por lo que no requiere que todos los patrones de entrenamiento se encuentren correctamente clasificados, se tiene la opción que algunos no se encuentren bien clasificados debido a que tengan características de dos clases, aunque se ha demostrado que mientras menor sea el número de patrones mal etiquetado mejor será el aprendizaje y viceversa.

Para clasificar imágenes de Alta Resolución de Nanoestructuras de Oro se ocupa la red de Dudani y se utiliza una estructura multicapa, a la cual se le asignan pesos aleatorios, los cuales se irán ajustando en el transcurso del aprendizaje, durante este proceso se le proporciona la red imágenes de Alta Resolución primero simuladas ya que éste es el caso en el que los patrones de entrenamiento se encuentran con todas sus características y después se juntan patrones simulados y experimentales en los cuales existen patrones incompletos o con ruido.

Del conjunto de patrones que se tengan se hará una selección ocupando el Muestreo Aleatorio Simple con la finalidad de tener un subconjunto que sea representativo. Durante el aprendizaje, la red crea una función ocupando los rasgos característicos que se le den, con esta función la red provee un lugar en el espacio de pesos al patrón x_i (desetiquetado), luego se toma en cuenta la clase de los k vecinos y la cercanía de estos con respecto al patrón x_i y se le da diferentes pesos, posteriormente se suman cada uno de los k vecinos considerando la clase a la que pertenecen y se le asigna al patrón x_i la clase con mayor peso.

3.1.5 Muestreo Aleatorio por Conglomerados.

Se ocupa este método cuando los elementos de la población que se desea estudiar, se encuentran separados en grupos independientes.

Debe de tomar en cuenta en el momento de seleccionar una muestra de la

donde

$$s_h^2 = \frac{\sum_{i=1}^{N_h}\left(y_{hi} - \bar{y}_h\right)^2}{N_h - 1} \tag{3.19}$$

s_h^2 **Varianza del h-ésimo estrato.**

3.1.5.1.3 Intervalo de Confianza.

$$\bar{\bar{y}} \pm Z_{(1-\alpha/2)}\sqrt{\frac{1}{N^2}\sum_{h=1}^{L}\left(\frac{N_h - n_h}{N_h}\right)\frac{s_h^2}{n_h}} \tag{3.20}$$

$Z_{(1-\alpha/2)}$ **Valor en tablas del intervalo del intervalo de confianza.**

3.1.5.1.4 Tamaño de la Población.

$$\hat{Y} = \sum_{h=1}^{L} N_h \bar{y}_h \tag{3.21}$$

3.1.5.1.5 Varianza de la Población.

$$V\left(\hat{Y}\right) = \sum_{h=1}^{L} N_h^2\left(\frac{N_h - n_h}{N_h}\right)\left(\frac{s_h^2}{n_h}\right) \tag{3.22}$$

3.1.5.1.6 Intervalo de Confianza.

$$\hat{Y} \pm NZ_{(1-\alpha/2)}\sum_{h=1}^{L} N_h^2\left(\frac{N_h - n_h}{N_h}\right)\left(\frac{s_h^2}{n_h}\right) \tag{3.23}$$

3.1.5.1.7 Tamaño de la Muestra.

$$n = \frac{N \sum_{h=1}^{L} N_h s_h^2}{N^2 d^2 \cdots + \sum_{h=1}^{L} N_h s_h^2} \qquad (3.24)$$

d^2 **Precisión**

s_h^2 **Varianza muestral del h-ésimo estrato.**

Conclusiones

Se presentarán a continuación las ventajas y limitaciones que se tendrían al ocupar tanto las redes clásicas como los modelos estadísticos, para resolver un problema de reconocimiento de patrones.

El reconocimiento de patrones consiste en determinar las diferentes clases a las que pertenecen un conjunto de patrones experimentales de nanoestructuras de oro, ocupando diferentes redes neuronales para que la clasificación sea rápida y objetiva.

Hopfield

Esta red realiza el proceso de asociación, esto implica que puede reconocer patrones aunque éstos se encuentren con ruido o distorsionados, asegura la convergencia, la falta de esta prueba no significa que la red no va a converger o que no funcione adecuadamente, sólo que no hay garantía de que ello ocurra. La principal importancia de esta red es histórica ya que basándose en ella y tratando de mejorar sus limitaciones se desarrollan las demás redes.

Dentro de sus limitaciones se encuentra el que no se puede almacenar una gran cantidad de información, no realiza un proceso de aprendizaje, ya que se dan de antemano los pesos de conexión, no distingue entre mínimos locales y globales, si se le presenta un patrón que se encuentre un poco rotado en comparación con los patrones que tiene almacenados, entonces la red no sabrá a qué clase pertenece.

Por lo tanto, es posible considerar a esta red como un antecedente en la búsqueda de una mejor propuesta al problema, ya que esta red tiene un espacio limitado para guardar datos, es posible que la red converja a un mínimo local en lugar de a un mínimo global y no realiza el proceso de aprendizaje.

Retropropagación

Se puede almacenar una gran cantidad de información ya que ocupa por lo general un algoritmo sigmoidal; tiene un proceso de aprendizaje lo que le

permite realizar un ajuste de pesos, con estos pesos la red forma un plano al cual se le llama espacio de pesos, puede ocurrir que la red converja a un mínimo local del espacio de pesos en vez de a un mínimo global, esta limitación no es tan grave siempre y cuando al mínimo local al que se haya llegado se encuentre lo suficientemente cerca del mínimo global y como consecuencia aporte un error que sea pequeño.

La dirección en la que se realizan los cambios en el espacio de pesos depende de la derivada, esto ayuda a que los desplazamientos que se realizan tengan la mejor dirección. Realiza el reconocimiento de patrones experimentales aunque éstos se encuentren trasladados o rotados, así como cuando la imagen tiene ruido.

Puede llegar a un mínimo local en lugar de a un mínimo global. Necesita el cálculo de la derivada lo cual requiere frecuentemente de grandes operaciones aritméticas, lo cual trae como consecuencia que se necesite de grandes recursos computacionales y en muchas ocasiones se requiere un mayor número de operaciones para calcular f'(x) que para calcular f(x).

Aunque este método es rápido no siempre asegura que se tiene la mejor solución, ya que la red puede converger a un mínimo local.

Kohonen

En esta red se determinan las clases por vecindades, se asignan inicialmente pesos aleatorios entre la capa de entrada y la capa de salida, y se permite que la red se adapte a sí misma y ajuste los pesos más adecuados durante el proceso de aprendizaje. En este proceso se le pueden dar un conjunto pequeño de patrones etiquetados y muchos sin etiquetar.

Se tiene una red competitiva, por lo que tanto en el proceso de aprendizaje como de funcionamiento cuando a la red se le proporciona un nuevo patrón, ésta compara sus características con los patrones existentes y determina como ganador al patrón que sea más parecido, si se encuentra en el proceso de aprendizaje, entonces la red realizará un reajuste de pesos entre los vecinos

del patrón que resulte ganador. Puede identificar patrones que se encuentren trasladados o rotados, tiene la ventaja de que no requiere de la derivada.

El aprendizaje que se realiza en esta red es lento, debido a que se le debe proporcionar varias veces el conjunto de patrones de entrenamiento, para que la red realice un ajuste adecuado de pesos.

Por lo que se puede inferir, que aunque el proceso de aprendizaje es lento esta red puede asegurar una buena clasificación de nuestro patrón.

Boltzmann

A esta red se le puede dar un conjunto pequeño de datos etiquetados y muchos sin etiquetar. Tiene un proceso de aprendizaje, en el cual la red identifica los rasgos característicos de cada clase.

Se propone un incremento de la temperatura para aumentar la función de energía y sacar a la red de un mínimo local a un mínimo global, con esto se asegura que el error que se tiene en la clasificación de un patrón sea la mínima. Los cambios de peso entre las conexiones en este caso son aleatorios.

Durante el funcionamiento la red será capaz de reconocer patrones que se encuentren trasladados o rotados. Este modelo tiene dos grandes ventajas con respecto a los modelos antes mencionados, la primera es que no requiere de la derivada para tener la mejor dirección (Retropropagación) en la que se realicen los cambios para llegar al mínimo. La segunda es que asegura la convergencia al mínimo global. Tiene la limitante que se debe de realizar una disminución pequeña de los cambios, lo que hace que la red sea lenta.

En consecuencia, es posible aseverar que aunque esta red es muy lenta asegura la convergencia al mínimo global.

Dudani

Dentro de las redes estadísticas se encuentra el modelo NN, a este modelo se le han hecho varias modificaciones, y se han propuesto nuevos modelos con la finalidad de resolver algunos problemas específicos que se han presentado en la práctica. El modelo que mejor se adapta para resolver el problema de

reconocimiento de patrones es el modelo propuesto por Dudani, por lo que se consideran sus ventajas y limitaciones.

La red de Dudani, a diferencia de las demás redes tiene la ventaja que se le puede proponer cuáles son los rasgos característicos (número de lados, número de ángulos iguales, número de direcciones de líneas paralelas y algunas otras características), que se deben tomar en cuenta para determinar las diferentes clases, esto hace que la red tenga mucha precisión al realizar la clasificación, tiene un aprendizaje imperfectamente supervisado el cual le permite que los patrones que se le den durante el proceso de aprendizaje puedan estar mal clasificados, aunque se ha demostrado que el buen aprendizaje de la red está en proporción del número de patrones bien clasificados que se le han proporcionado, se tiene la opción de tener un número reducido de patrones mal clasificados debido al costo o al tiempo que esto implica.

Esta red puede reconocer patrones que se encuentren trasladados o rotados e ignora el ruido. Durante el proceso de aprendizaje la red crea una función, tomando en cuenta los rasgos característicos, proporciona un lugar en el espacio de pesos, posteriormente se identifican los k-vecinos más cercanos y se les asignan pesos dependiendo de la cercanía a la que se encuentre cada uno de los k vecinos de x_i. Luego se suman los pesos tomando en cuenta la clase a la que pertenecen y se asigna la clase que tenga mayor número de representantes. En el proceso de funcionamiento, la red identifica los rasgos característicos y proporciona un lugar en el espacio de variables, asignando al patrón x_i la clase del vecino más cercano.

Tiene como desventaja el que si existe un empate entre clases, es mejor el modelo K-NN ya que realiza una mejor clasificación.

Es necesario tomar en cuenta que otros autores no han resuelto este problema de esta forma, por lo que cuando se entrene a la red se debe de experimentar con el número más adecuado de vecinos más cercanos.

Se puede concluir que la red de Dudani es el mejor modelo para resolver el problema de reconocimiento de patrones de nanoestructuras de oro, ya que es

una red muy precisa debido a que se le puede proporcionar los rasgos característicos de cada clase, lo cual es una gran ventaja.

Es factible ocupar la red de Kohonen cuando se desea entrenar una red la cual ella misma identifique cuáles son los rasgos característicos de cada clase y sea por lo tanto totalmente objetiva.

Cuando se le presenta a la red un patrón nuevo después de que ha realizado el aprendizaje, ninguna de las redes podrá caracterizarla y crear una nueva clase, debido a que durante el proceso de aprendizaje, todas las redes realizan el ajuste de pesos entre las conexiones, por lo que no podrán incorporar un nuevo patrón.

En el momento que se le presenta a la red de Kohonen un nuevo patrón, como es una red competitiva dará como respuesta la clase que más se le parezca. Es posible realizar una modificación a esta red tomando en cuenta el error, si este error es menor o igual a un número asignado previamente, entonces se acepta la clase del patrón, de lo contrario se coloca una nueva clase con los patrones de clase desconocida.

Si se ocupa la Red de Dudani para clasificar a un patrón tendrá que coincidir con el conjunto de rasgos característicos, de lo contrario la red no asignará ninguna clase.

Debe de recordarse que para entrenar una red neuronal es necesario tomar en cuenta los criterios de un especialista, como es el caso de la red de Dudani, la cual basa su precisión en los rasgos característicos que se le den, esto ocurre de manera similar en la red de Kohonen en la cual se requiere de un experto para seleccionar los patrones que se consideran como representantes de cada clase y con los cuales se entrenará a la red.

En esta tesis se da la pauta para que alguna persona que esté familiarizada con la programación en paralelo, considere las ventajas y limitaciones que se han propuesto, así como la forma en la que toman decisiones cada una de las redes y con ello programe una red neuronal.

Es posible entrenar a la red a partir de otros métodos de caracterización como Patrones de Difracción, en el cual la red puede realizar la comparación y decidir si la imagen del patrón experimental corresponde al modelo y a la imagen del patrón simulado.

Con esto se propone un nuevo método, el cual realiza una clasificación plena, rápida y objetiva por medios automatizados.

Referencias

1. -Vazquez Z Eduardo, *"¿Que son las redes neuronales?"*, Tópicos de la Investigación y Posgrado, vol 2, no 3, pp 11-18, Febrero 1992.

2. -Hilera González José Ramón y Mártinez Hernando Victor José, *Redes Neuronales Artificiales (Fundamentos Modelos y Aplicaciones)*, Editorial Addison Wesley, 1991.

3. -Fermann James A, Skapura David M, *Redes neuronales (Algoritmos, Aplicaciones y Técnicas de Programación)*, Editorial Addison Wesley, 1991.

4. -Lippmann Ricardo P, *"An Introduction to Computing with Neural Nets"*, Magazine IEEE, pp 4-22, Abril 1987.

5. -Hopfield John J y Tank David W, *"Neural computation of decisions in optimization problems"*, Biological Cybernetics, vol 52, pp 141-152, 1985.

6. -Hopfield John J y Tank David W, *"Computing with neural circuits: A model"*, Science, vol 233, pp 625-633, Agosto 1986.

7. -Hopfield J, *"Neural Networks and physical systems with emergent collective computational abilities"*, Proceedings of the National Academy of Sciences, vol 79, pp 2554-2558, 1982.

8. -Kuh Anthony y Dickinson Bradley W, *"Information capacity of associative memories"*, IEEE Transactions on Information Theory, vol 35, pp 59-68, Enero de 1989.

9. -Gorman R. Paul y Sejnowski Terrence J, *"Analysis of hidden units in a layered network trained to classify sonar targets"*, Neural Networks, vol 1, pp 76-90, 1988.

10. -Terrence J. Sejonowski y Rosenberg Charles R, *"Parallel Networks that learn to pronounce"* , Complex Systems, vol 1, pp 145-168, 1987.

11. -Mc Clelland James y Rumelhart David, *Explorations in Parallel Distributed Procesing*, vol 1 y 2, Cam Cambridge 1986.

12. -Hecht-Nielsen Robert, *Neurocomputing*, Adison Wesley, 1990.

13. -Kohonen T, *"Self-organized formation of topologically correct feature maps"*, Biological Cybernetics, vol 43, pp 59-69, 1982.

14. -Thiran Patrick y Hasler Martin, *"Self-Organization of a One-Dimensional Kohonen Network With Quantized Weights and Imputs"*, Neural Networks, vol 7, pp 1427-1439, 1994.

15. –Kohonen Teuvo, *"Self-organization and Associative Memory"*, Springer Series in Information Sciences. Springer-Verlag, vol 8, 1984.

16. -Chow C. K, *"An optimum character recognition system using decision functions"*, IRE Trans. Elec. Comp. (EG-6), pp 247-245,1957.

17. -Mood Alexander M, Graybill Franklin A, Boes Duane C, *Introduction to the Theory of Statistics*, Mc Graw Hill International Editions, 1974.

18. -Hinton G, Ackley D y Sejnowski T, *"Boltzmann machines: Constraint networks than learn"*, Carnegie-Mellon University, Departament of Computer Science Report (CUM-CS), pp 84-119, 1984.

19. -Maren A. J, y Harston C. T, *Handbook of Neural Computing Applications*, Academic Press, 1990.

20. -Sears Francisco W, Zemansky Marks W, Young Hugh D., *"Física Universitaria"*, Addison Wesley Iberoamericana, 379-381, 1988.

21. -Brostow Witold, *Introducción a la Ciencia de los Materiales*, Limusa, pp 43-80, 1981.

22. -Shu S, Bilven S y Belina J, *"Training of Feedfoward Neural Networks Architecture for Feature Recognition of Abnormal ECG Waveforms"*, Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol 13, no 2, pp 1395-1396, 1991.

23. -Gutzmann K, *"Combinatorial Optimization Using a Continuous State Boltzmann Machine"*, Proceedings of the IEEE First Int. Conf. on Neural Networks, vol 3, pp 721-734, 1987.

24. -Devijver P. E, Kittler J, *Pattern Recognition: a statistical approach*, Prentice Hall, 1982.

25. -Cover T. M, Hart P. E, *"Nearest Neighbor pattern Classification"*, IEEE Trans Info. Theory (IT-13), pp 21-27, 1967.

26. -Wilson D. L, *"Asymptotic properties of nearest neighbor rules using edited data"* , IEEE Transon Syst, Man and Cryber (SMC-2),PP 408-421, 1972.

27. -Hart P. E, *"The Condensed Nearest Neighbor ruler"*, IEEE Trans Info Theory (IT-14), pp 505-516, 1986.

28. -Tomek I, *"An experiment with the edited nearest neighbor ruler"*, IEEE Trans Syst, Man and Cyber (SMC-6), pp 448-452, 1976.

29. -Gowda K. C y Krishna G. *"Editing and error corrections using the concept of mutual nearest neighborthood"*, Proc Int Conf on Cyber and Society, Denver 1979.

30. -Brown T.A y Koplowitz J, *"The weighted nearest neighbor ruler for class dependent samplesizes"*, IEEE Trans. Info. Thery (IT-25), pp 617-619, 1979.

31. -Dudani S.A, *"The distance-weighted k nearest neighbor ruler"*, IEEE Trans Syst, Manand Cyber (SMC-6), pp 325-327, 1976.

32. -Ascencio J. A, Gutierrez-Wing C, Espinosa M. E, Marin M, Tehuacanero S, Zorrilla C, Yacamán M. Jose, *"Structure determination of small particles by HREM imaging theory and experiment"*, Surface Science. vol 396, pp 349-368.

Bibliografía

Libros

1. -Brostow Witold, *Introducción a la Ciencia de los Materiales*, Limusa, pp 43-80, 1981.

2. -Devijver P. E, Kittler J, *Pattern Recognition: a statistical approach*, Prentice Hall, 1982.

3. -Fermann James A, Skapura David M, *Redes neuronales (Algoritmos, Aplicaciones y Técnicas de Programación)*, Editorial Addison Wesley, 1991.

4. -Hecht-Nielsen Robert, *Neurocomputing*, Adison Wesley, 1990.

5. -Hilera González José Ramón y Mártinez Hernando Victor José, *Redes Neuronales Artificiales (Fundamentos Modelos y Aplicaciones)*, Editorial Addison Wesley, 1991.

6. -Maren A. J, y Harston C. T, *Handbook of Neural Computing Applications*, Academic Press, 1990.

7. -Mc Clelland James y Rumelhart David, *Explorations in Parallel Distributed Procesing*, vol 1 y 2, CamCambridge 1986.

8. -Mood Alexander M, Graybill Franklin A, Boes Duane C, *Introduction to the Theory of Statistics*, Mc Graw Hill International Editions, 1974.

Revistas

1. -Ascencio J. A, Gutierrez-Wing C, Espinosa M. E, Marin M, Tehuacanero S, Zorrilla C, Yacamán M. Jose, *"Structure determination of small particles by HREM imaging theory and experiment"*, Surface Science, vol 396, pp 349-368.

2. -Brown T.A y Koplowitz J, *"The weighted nearest neighbor ruler for class dependent samplesizes"*, IEEE Trans. Info. Thery (IT-25), pp 617-619, 1979.

3. -Chow C. K, *"An optimum character recognition system using decision functions"*, IRE Trans. Elec. Comp. (EG-6), pp 247-245,1957.

4. -Cover T. M, Hart P. E, *"Nearest Neighbor pattern Classification"*, IEEE Trans Info. Theory (IT-13), pp 21-27, 1967.

5. -Dudani S.A, *"The distance-weighted k nearest neighbor ruler"*, IEEE Trans Syst, Manand Cyber (SMC-6), pp 325-327, 1976.

6. -Gorman R. Paul y Sejnowski Terrence J, *"Analysis of hidden units in a layered network trained to classify sonar targets"*, Neural Networks, vol 1, pp 76-90, 1988.

7. -Gowda K. C y Krishna G. *"Editing and error corrections using the concept of mutual nearest neighborthood"*, Proc Int Conf on Cyber and Society, Denver 1979.

8. -Gutzmann K, *"Combinatorial Optimization Using a Continuous State Boltzmann Machine"*, Proceedings of the IEEE First Int. Conf. on Neural Networks, vol 3, pp 721-734, 1987.

9. -Hart P. E, *"The Condensed Nearest Neighbor ruler"*, IEEE Trans Info Theory (IT-14), pp 505-516, 1986.

10. -Hinton G, Ackley D y Sejnowski T, *"Boltzmann machines: Constraint networks than learn"*, Carnegie-Mellon University, Departament of Computer Science Report (CUM-CS), pp 84-119, 1984.

11. -Hopfield J, *"Neural Networks and physical systems with emergent collective computational abilities"*, Proceedings of the National Academy of Sciences, vol 79, pp 2554-2558, 1982.

12. -Hopfield John J y Tank David W, *"Computing with neural circuits: A model"*, Science, vol 233, pp 625-633, Agosto 1986.

13. -Hopfield John J y Tank David W, *"Neural computation of decisions in optimization problems"*, Biological Cybernetics, vol 52, pp 141-152, 1985.

14. -Kohonen T, *"Self-organized formation of topologically correct feature maps"*, Biological Cybernetics, vol 43, pp 59-69, 1982.

15. -Kohonen Teuvo, *"Self-organization and Associative Memory"*, Springer Series in Information Sciences. Springer-Verlag, vol 8, 1984.

16. -Kuh Anthony y Dickinson Bradley W, *"Information capacity of associative memories"*, IEEE Transactions on Information Theory, vol 35, pp 59-68, Enero de 1989.

17. -Lippmann Ricardo P, *"An Introduction to Computing with Neural Nets"*, Magazine IEEE, pp 4-22, Abril 1987.

18. -Sears Francisco W, Zemansky Marks W, Young Hugh D., *"Física Universitaria"*, Addison Wesley Iberoamericana, 379-381, 1988.

19. -Shu S, Bilven S y Belina J, *"Training of Feedfoward Neural Networks Architecture for Feature Recognition of Abnormal ECG Waveforms"*, Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol 13, no 2, pp 1395-1396, 1991.

20. -Terrence J. Sejonowski y Rosenberg Charles R, *"Parallel Networks that learn to pronounce"* , Complex Systems, vol 1, pp 145-168, 1987.

21. -Thiran Patrick y Hasler Martin, *"Self-Organization of a One-Dimensional Kohonen Network With Quantized Weights and Imputs"*, Neural Networks, vol 7, pp 1427-1439, 1994.

22. -Tomek I, *"An experiment with the edited nearest neighbor ruler"*, IEEE Trans Syst, Man and Cyber (SMC-6), pp 448-452, 1976.

23. -Vazquez Z Eduardo, *"¿Que son las redes neuronales?"*, Tópicos de la Investigación y Posgrado, vol 2, no 3, pp 11-18, Febrero 1992.

24. -Wilson D. L, *"Asymptotic properties of nearest neighbor rules using edited data"* , IEEE Transon Syst, Man and Cryber (SMC-2), pp 408-421, 1972.