

**CARACTERIZACIÓN Y ANÁLISIS COMPARATIVO DE TRES MODELOS  
BASADOS EN LAS TÉCNICAS PCA PARA EL RECONOCIMIENTO DE  
ROSTROS**

**JESUS ALFONSO CERON LOMBANA**

**ANEXOS**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE SISTEMAS  
POPAYÁN  
2006**

# INDICE

---

<b>ANEXO 1</b>	<b>GLOSARIO</b>	<b>3</b>
----------------	-----------------	----------

---

<b>ANEXO 2</b>	<b>COMPLEMENTO AL MARCO TEÓRICO</b>	<b>5</b>
----------------	-------------------------------------	----------

---

<b>2.1. Procesamiento Digital de Imágenes</b>	<b>5</b>
2.1.1. Historia	5
2.1.2. Introducción	7
2.1.3. Imágenes Digitales	8
2.1.3.1. Dispositivos de adquisición de Imágenes	9
2.1.3.2. Almacenamiento de imágenes digitales	12
2.1.4. Modelo general para el procesamiento de imágenes	15
2.1.5. Procesamiento Digital de Imágenes	17
2.1.5.1 Segmentación	17
2.1.5.2. Transformaciones de la imagen	18
2.1.5.3. Realce	18
2.1.5.4. Efectos	18
2.1.5.5. Efectos especiales	19
2.1.5.6. Vista	19
2.1.5.7. Filtros	19
<b>2.2. Redes Neuronales Artificiales RNA</b>	<b>25</b>
2.2.1. Historia	25
2.2.2. Fundamentación Biológica	26
2.2.3. Red de Neuronas Artificiales	29
2.2.3.1 Redes con Aprendizaje Supervisado	29
2.2.3.2. Redes con Aprendizaje No Supervisado	29
2.2.4. Estructura Básica de la Red	30
2.2.5. Implementación de Redes Neuronales Artificiales	31
2.2.5.1. Realización de Redes Neuronales	31
2.2.5.2. Herramientas software de desarrollo	31
2.2.5.3. Neuro-computadores de propósito especial y de propósito general	32
2.2.5.4. Implementación microelectrónica (VLSI).	32
2.2.6. Perceptrón Multicapa	32
2.2.6.1. Introducción	32
2.2.6.2. Arquitectura del Perceptrón Multicapa	33
2.2.6.2.1 Parámetros que determinan el aprendizaje de la Red	34
2.2.6.2.2. La función de activación	34
2.2.6.2.3. Tasa de aprendizaje	35
2.2.6.2.4. Momento	35
2.2.6.2.5. "Bías" o polarización	35
2.2.6.2.6. Ruido	36

2.2.6.2.7. Tamaño de las capas de neuronas	36
2.2.6.2.8. Conjunto de entrenamiento	36
2.2.6.3. Método de aprendizaje del perceptrón multicapa	37
2.2.7. Redes Neuronales de Funciones de Base Radial RBF ( <i>Radial Basis Function</i> )	39
2.2.7.1. Introducción	39
2.2.7.2 Fritzke: aprendizaje rápido adaptativo del modelo RBF	40
2.2.7.2.1 Iniciación de la red	40
2.2.7.2.2 Adaptación de los centros y los pesos	40
2.2.7.2.3 Inserción de nuevos centros	41
2.2.7.2.4 Finalización del entrenamiento	41
2.2.8. Mapas Auto-Organizados De Kohonen – Som ( <i>Self-Organizing Maps</i> )	41
2.2.8.1. Introducción	41
2.2.8.2. Un Ejemplo Gráfico	43
<b>ANEXO 3 DIAGRAMAS DE SECUENCIA</b>	<b>47</b>
<hr/>	
<b>3.1. Diagramas de Secuencia</b>	<b>47</b>
3.1.1. Análisis de Principales Componentes (PCA)	47
3.1.2. Captura de Imágenes	49
3.1.3. Identificación de Rostros	50
3.1.4. Filtros	50
<b>ANEXO 4 DETALLES DE IMPLEMENTACIÓN DE LA RNA</b>	<b>52</b>
<hr/>	

## ANEXO 1

### GLOSARIO

**Acometer:** Empezar, intentar.

**Análogo:** Información presentada de manera secuencial y continua.

**Anatomía:** Ciencia que estudia la forma y estructura externa e interna de los seres vivos, y especialmente del cuerpo humano.

**Anatómico:** De la anatomía o relativo a ella.

**Aprendizaje:** Adquisición de conocimientos, especialmente en algún arte u oficio.

**Biometría:** Estudio estadístico y matemático de los fenómenos biológicos.

**Cara:** Parte anterior de la cabeza, desde la frente a la barbilla:

**Conciso:** [Enunciado] poco rebuscado y preciso.

**Condiciona:** Que incluye o conlleva una condición o requisito.

**Convexo:** [Línea o superficie] curva cuya parte más prominente está del lado del que mira.

**Cotejar:** Confrontar una cosa con otra u otras.

**Densidad:** Espesor, concentración.

**Derrotero:** Camino tomado para lograr el fin propuesto.

**Difuso:** Impreciso, borroso, poco claro.

**Digital:** [Aparato o instrumento] que mide cantidades y las representa con números dígitos. Señal que utiliza valores discretos en lugar de un espectro continuo de valores.

**Discriminar:** Separar, diferenciar una cosa de otra.

**Discriminatorio:** Que discrimina.

**Distorsión:** Deformación de la onda de imágenes, sonidos o señales durante su propagación.

**Eigen:** En alemán propio, apropiado.

**Entrenamiento:** Adiestramiento y preparación física y técnica que se realiza para perfeccionar el ejercicio de una actividad deportiva o lúdica.

**Específico:** Que distingue una especie o una clase de elementos de otra.

**Esencial:** Sustancial, imprescindible. Fundamental o de gran relevancia.

**Fiable:** que es probable que funcione bien o sea segura.

**Fraudes:** Engaño que se realiza eludiendo obligaciones legales o usurpando derechos con el fin de obtener un beneficio.

**Gesto:** Movimiento del rostro o de las manos con que se expresa algo. Semblante, cara, rostro.

**Histograma:** Representación gráfica de una distribución de frecuencias por medio de rectángulos.

**Heurística:** Arte de inventar o descubrir.

**Identificación:** Reconocimiento de la identidad de alguien. Consideración de dos cosas distintas para que aparezcan como una misma.

**Incremental:**

**Irrelevante:** Que carece de relevancia o importancia.

**Método:** Modo estructurado y ordenado de obtener un resultado, descubrir la verdad y sistematizar los conocimientos.

**Neuroglia:** Células no excitables que sostienen las neuronas del sistema nervioso central (neuro = nervio; glia = pegamento).

**Neurona:** Célula diferenciada perteneciente al sistema nervioso, capaz de propagar el impulso nervioso a otra neurona. Está compuesta por una zona de recepción, las dendritas, y otra de emisión o salida, el axón o neurita.

**Password:** Clave secreta o personal con que se accede a la información contenida en un ordenador o en una red informática.

**Patrón:** Modelo que sirve de muestra para sacar otra cosa igual.

**Preferencia:** Primacía o ventaja que una persona o cosa tienen sobre otra por su valor o importancia. Inclination, predilección o elección de una cosa o persona entre varias.

**Preferentemente:** Con preferencia.

**Rasgo:** Cada una de las facciones del rostro humano.

**Retroalimentación:** Comunicación entre el instructor o el sistema y el aprendiz, como resultado de una acción o proceso.

**Rostro:** Cara de las personas.

**Sensores:** Dispositivo formado por células sensibles que detecta variaciones en una magnitud física y las convierte en señales útiles para un sistema de medida o control.

**Sináptico:** Que pertenece o afecta a la zona de contacto entre dos neuronas. Relativo o perteneciente a la sinapsis. El espacio sináptico es el espacio que, en la unión neuromuscular (UNM), está entre el extremo de la célula nerviosa y la célula muscular. A través del espacio sináptico "viaja" la acetilcolina (ACh).

**Técnica:** Procedimiento o el conjunto de procedimientos que tienen como objetivo obtener un resultado determinado, ya sea en el campo de la ciencia, de la tecnología, de las artesanías o en otra actividad.

**Umbral:** Valor a partir del cual empiezan a ser perceptibles los efectos de un agente físico.

**Verificación:** Comprobación de la verdad o autenticidad de algo. Realización de algo que se dijo o se había pronosticado.

## ANEXO 2

### COMPLEMENTO AL MARCO TEORICO

#### 2.1. Procesamiento Digital de Imágenes

##### 2.1.1. Historia

A diferencia del estudio de los mecanismos de la visión humana, el procesamiento y análisis de imágenes digitales nacieron en el momento en que se dispone de recursos tecnológicos para captar y manipular, en forma de matrices de valores, gran cantidad de información espacial.

Históricamente, la primera vez que se hizo uso de las técnicas de procesamiento de imágenes fue en los años veinte, cuando se transmitieron imágenes de fotografías periodísticas a través de cable submarino entre Londres y New York [1] referencias en la monografía.

Esto supuso que el tiempo entre emisión y recepción de las imágenes pasase de una semana (por barco) a tres horas (por cable). Algunos problemas iniciales para la mejora de la calidad de la imagen estaban relacionados con la selección del método de impresión y la distribución de los niveles de gris. Los primeros sistemas, años veinte, eran capaces de codificar cinco niveles de gris distintos. En 1929 ya se utilizaban quince. Obviamente estábamos en lo que podría considerarse el comienzo de la formación, captación, muestreo, cuantificación, codificación y visualización de imágenes [2].

Podría decirse que el procesamiento de imágenes como tal comienza en los años 1950-1960 debido a la combinación de dos hechos, por una parte la aparición de las computadoras digitales y por otra los Programas Espaciales, fundamentalmente de los Estados Unidos. Como combinación de estos hechos aparecen las imágenes como matrices de valores listas para ser procesadas por el computador. El objetivo inicial era muy concreto: mejorar la calidad visual de dichas imágenes [3].

Para mejorar la calidad visual, un objetivo claro era la eliminación del ruido en las imágenes captadas y/o transmitidas. Este objetivo se aborda inicialmente con la utilización de filtros de medias, filtrado en el dominio de las frecuencias y, con posterioridad, filtrado no lineal (la literatura en estos temas es enorme, ver, por ejemplo, *Pratt* [4] y *Rosenfeld y Kak* [5]). Obviamente la eliminación del ruido es un problema típico de procesamiento de imágenes que ha ido evolucionando tanto en la aplicación a otros muchos campos, como en la monetización del propio ruido en función del dispositivo utilizado. Mientras que las técnicas mencionadas para eliminar el ruido se han mantenido, y evolucionado, han aparecido también nuevas aproximaciones. En particular mencionaremos el uso de modelos piramidales [6], el trabajo de Cai [7] en el contexto de alisamiento por difusión [8] para el uso de

difusión anisotrópica<sup>1</sup> en problemas de procesamiento de imágenes, Witkin [9] y Lindeberg [10] para el uso de representaciones espacio-escala en procesamiento de imágenes y las descomposiciones basadas en Wavelets [11, 12], sin olvidar el desarrollo de técnicas de eliminación de ruido aplicadas a imágenes en color [13]. Obviamente, como veremos más adelante, estas nuevas metodologías no han sido aplicadas sólo a la eliminación de ruido.

Junto con la eliminación de ruido, otro problema típico de las imágenes espaciales es que aparecían borrosas al observador, bien por el proceso de captación en el Espacio, por la observación desde la Tierra o como ocurrió en 1990 por un error imposible de ocurrir como fue el pulido del espejo del Telescopio Espacial Hubble con un aparato defectuoso y que fue posteriormente utilizado para comprobar que el pulido había sido correcto. Aparecen así los métodos de restauración de imágenes que utilizan filtros inversos, los filtros de Wiener [14], y Kalman y posteriormente, desde la comunidad de ingeniería eléctrica y electrónica, la regularización [15, 16]. Es importante notar que la restauración (de convolución o filtrado inverso) se comenzó a aplicar a las imágenes de las misiones espaciales Ranger, Surveyor y Mariner a mediados de los años sesenta y que los filtros iniciales provenían de filtros eléctricos y análisis de series temporales.

A comienzos de los años ochenta, la comunidad estadística introduce la aplicación de los modelos bayesianos a problemas de procesamiento y análisis de imágenes. Inspirados por los trabajos de Grenander [17, 18], los hermanos Geman [19] y posteriormente Besag [20] publican sus trabajos en restauración y análisis de imágenes desde un punto de vista bayesiano, ver también Ripley [21]. Aunque, desde el punto de vista de la solución, el problema a resolver utilizando el paradigma bayesiano puede considerarse igual al de la regularización, el aspecto novedoso y con el que nos encontraremos a lo largo del presente artículo, es la introducción de información a priori en el problema que se intenta resolver, el libro de (Katsaggelos, 1991) [16] contiene contribuciones sobre restauración de imágenes desde la perspectiva bayesiana y de regularización. Para un recorrido por los temas de restauración ver, por ejemplo, los libros de texto Umbaugh [22], Castleman [12], Jain [23] y González y Woods [1].

Junto con las técnicas de eliminación de ruido y restauración aparecen las llamadas técnicas de mejora de imágenes. Aunque tanto la eliminación de ruido como la restauración pueden entenderse como técnicas de mejora, la literatura en procesamiento de imágenes suele incluir dentro de este epígrafe las técnicas de procesamiento del histograma y de filtrado, en el dominio espacial y en el de las frecuencias (excluyendo o incluyendo eliminación de ruido) junto con su extensión a imágenes en color [1, 23].

---

<sup>1</sup> Difusión donde la difusividad es un tensor y no un escalar. Al proyectarse el gradiente sobre el tensor, en la ecuación de difusión, se potencia el proceso de decisiones en dirección perpendicular a la componente principal de mayor auto valor. [32]

Un tema más, que se suele considerar parte del procesamiento es la compresión de imágenes. Podríamos decir que el comienzo de la compresión de imágenes se sitúa en 1950 cuando se solicita la patente del código predictivo (Cutler, 1952) [24], esta lleva asociada la publicación de los trabajos de Harrison [25] sobre la aplicación de la predicción lineal a televisión y el trabajo de Huffman [26] sobre la construcción de códigos eficientes de longitud variable. Junto a estos trabajos es necesario mencionar el trabajo de Gabor [27] que llevaría al análisis de resolución variable tiempo/escala y al interés posterior en wavelets. Igualmente importante fue el trabajo de Shannon [27] que proporcionó la base teórica para la compresión eficiente. Con posterioridad el campo de la compresión de imágenes y video ha evolucionado enormemente [30], al ser estas las tecnologías que permiten (*enabling technologies*) el desarrollo de otras tecnologías, como por ejemplo multimedia.

Sin embargo, surge la pregunta de por qué se puede considerar la compresión parte del procesamiento de imágenes. La razón es simple, por una parte la compresión era, al menos inicialmente, una forma de procesamiento orientada a que las imágenes ocupasen menos espacio para transmisión y almacenamiento. Con el paso del tiempo esta utilidad sigue vigente, pero además han surgido otras aplicaciones como la eliminación de artificios en imágenes y video comprimidos, un campo en continua expansión [1, 12, 22, 23, 30, 31].

A lo largo del recorrido por el procesamiento de imágenes se ha hablado de operaciones en el dominio espacial o en el dominio de las frecuencias. La razón es clara, muchas operaciones de procesamiento se pueden realizar bien sobre la propia imagen o transformando esta a un dominio alternativo, realizando el procesamiento en el dominio alternativo para posteriormente realizar la transformación inversa. Surge así el uso de las transformadas de Larhunen-Loeve, Fourier, Walsh, Hadamard, Coseno discreta, Wavelets, entre otras.

### **2.1.2. Introducción**

Hace no mucho las posibilidades de los equipos de captura y procesamiento digital eran bastante limitadas y los costos de tiempo y de procesamiento bastante prohibitivos. Ante lo cual en muy pocas áreas se prestaba atención al potencial que las herramientas par el manejo de imágenes digitales ofrecían. La explotación de estas herramientas se había quedado restringida a algunas secciones de investigación y desarrollo de aplicaciones software se orientaba hacia problemas donde el presupuesto era vasto. En la actualidad es posible explotar plataformas de bajo costo y obtener resultados de gran calidad y crear aplicaciones de gran utilidad, versátiles y flexibles, así como aplicaciones de software de propósito específico para atender las diversas necesidades de los especialistas.

Es posible citar gran cantidad de ejemplos donde el procesamiento de imágenes ayuda a analizar, deducir y tomar decisiones. Entre otras áreas en la cuales se han desarrollado herramientas de gran utilidad podemos mencionar las siguientes:

Medicina, Fisiología, Biometría, Astronomía, Ciencia Ambiental, Robótica Metalúrgica, Física, Electrónica, Biología y el OCR (*optic Character Recognition*).

El procesamiento de imágenes puede considerarse como un tipo especial del procesamiento digital en dos dimensiones, el cual se usa para revelar información sobre imágenes y que involucra hardware, software y soporte teórico.

Procesamiento de imágenes es el término usado para denominar las operaciones desarrolladas sobre un conjunto de datos de imagen para mejorarlas de alguna forma, para ayudar a su interpretación o para extraer algún tipo de información útil de ella.

### 2.1.3. Imágenes Digitales

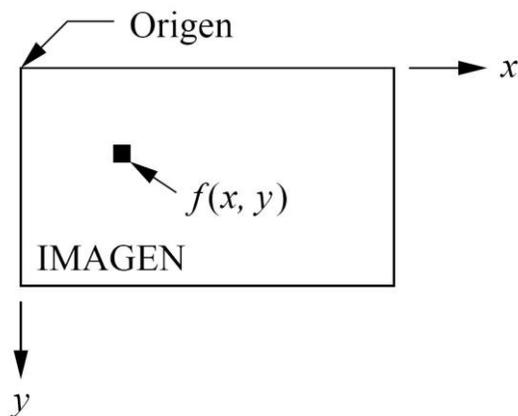


Imagen 2.2 Convención de Ejes

Para representar una imagen monocromática (blanco y negro o tonos de grises) en la computadora, se requiere de una matriz (un arreglo de números de dos dimensiones, o sea, varios renglones de datos, en donde sus elementos pueden ser localizados por su posición dentro de los renglones y las columnas), así a cada elemento de la matriz le corresponde un *píxel*. El valor de cada elemento es una tonalidad en una escala de grises, en donde el rango de valores de estos píxeles, puede estar entre 0 y 255. Para representar este rango de valores en la computadora son necesarios 8 bits. En caso de que una imagen sea solamente blanco y negro, será necesario 1 *bit*, el cual indicará con un 0 si el *píxel* es negro y con 1 si el *píxel* es blanco. Esto es típico en imágenes en donde únicamente exista texto, como en el caso de un fax [34].

Cuando se quiere representar una imagen en color, es necesario contar con tres matrices, una para cada color básico (rojo, verde y azul). También es posible contar con una matriz y una serie de consideraciones, para las cuales será necesario poseer tablas de colores o mejor conocidas como paleta de colores, en donde la cantidad de bits para generarlas, será lo que determine la cantidad de

colores de la imagen, por ejemplo, si se usan 4 bits solamente podrán generarse 16 colores, con 8 bits 256 colores y en el caso de 24 bits se producirán 16 millones de colores. La imagen 1.1 muestra un ejemplo, para entender como varía la calidad de una imagen conforme se modifica la cantidad de bits para representar un tono de un píxel, en escalas o tonos de grises.

### 2.1.3.1. Dispositivos de adquisición de Imágenes

#### 2.3.1.1 CCD (Coupled Charge Device)

Un CCD es un dispositivo similar a un microchip con una serie de elementos fotosensibles (*píxel*) de polisilicio. Cada uno de estos elementos es capaz de generar una señal eléctrica proporcional al número de fotones (intensidad luminosa) que ha recibido.

Existen dos tipos de CCD según la disposición de los elementos fotosensibles: lineales y de superficie. En los primeros los *píxel* se ordenan de modo lineal. Para obtener una imagen es preciso desplazar el CCD por la superficie de la imagen. Este tipo de CCD es el que se utiliza en los escáneres.

Los CCD de superficie o "*array*" son los que se utilizan en las cámaras de vídeo o de fotografía digital. Tienen dispuestos los *píxeles* en forma de matriz y permiten capturar toda la imagen al mismo tiempo [35].

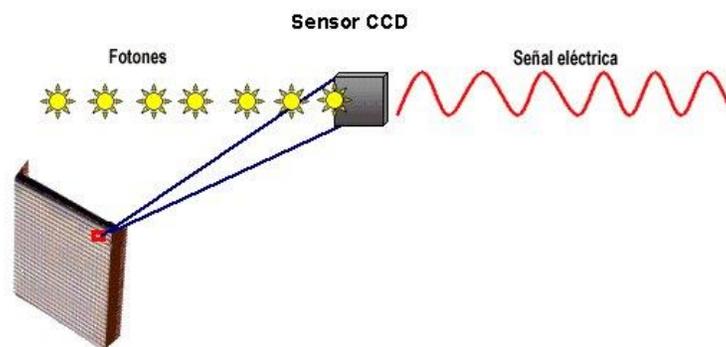


Imagen 1.3 Proceso de captura de imagen con CCD [35]

Los dispositivos CCD capturan imágenes monocromas. Para poder capturar imágenes en color con un CCD es necesario descomponer previamente la luz en los tres colores básicos del espectro: rojo verde y azul y recoger la información de cada uno de estos colores en un *píxel*. Para conseguir esto se utilizan tres métodos:

- **CCD Único.** Un CCD con píxel sensibles a diferentes colores (rojo, verde y azul).

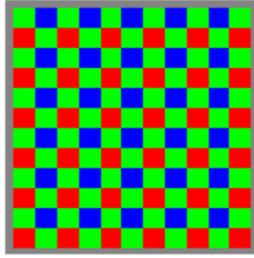


Imagen 1.4 CCD Único

- Tres CCDs. Cada uno sensible a uno de los colores. El sistema lleva un prisma que manda las tres componentes del espectro (roja, verde y azul) a cada uno de los CCDs.

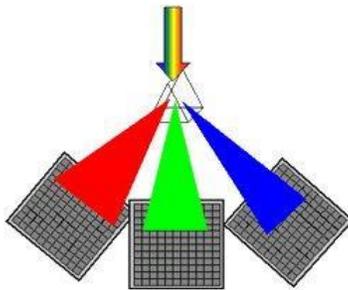


Imagen 1.5 Tri CCD

- CCD único con triple exposición. Se coloca delante del CCD un filtro para cada uno de los colores del espectro y se toman tres imágenes de la misma zona.

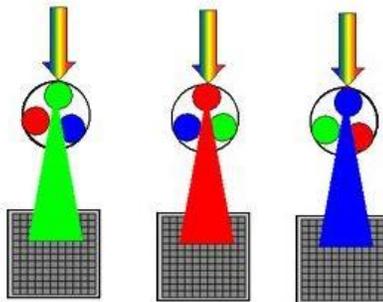


Imagen 1.6 CCD único con triple exposición

Las imágenes de mayor calidad se obtienen con el segundo método. El tercero permite obtener una calidad similar al primero, pero en objetos estáticos ya que se necesita el triple de tiempo para obtener la imagen en color.

Los escáneres al contar con un sensor lineal permiten una mayor resolución de imagen, sin embargo el tiempo de digitalización es mucho más lento y en la mayoría de los casos sólo podemos digitalizar imágenes planas (fotografías,

transparencias, radiografías, etc.) Por el contrario los tiempos de digitalización de las cámaras son mucho más rápidos y nos permiten captar cualquier tipo de escenas, aunque la resolución espacial de la imagen sea menor.

En el caso de imágenes en color la intensidad puede considerarse como un vector tridimensional cuyas componentes son las intensidades en las tres bandas espectrales: rojo, verde y azul. Hablándose en este caso de una imagen multibanda, puesto que la información de la imagen color se desglosa en tres imágenes correspondientes a cada una de las bandas del espectro visible. Para cada una de las bandas se utilizan 8 bits de información, 24 bits en total, lo que nos dan más de 16 millones de posibles combinaciones de colores

El otro modelo corresponde a los sistemas basados en *reflexión*, este es el caso de los *scanners* de cama plana. Estos dispositivos en vez de tener un arreglo bidimensional de *CCD's*, sólo tienen un arreglo lineal de éstos. Mediante un sistema óptico se envía luz desde una lámpara a la imagen y es recibida en un arreglo lineal de detectores y un subsistema de desplazamiento basado en un motor. La imagen 1.8 muestra el principio de operación del dispositivo.

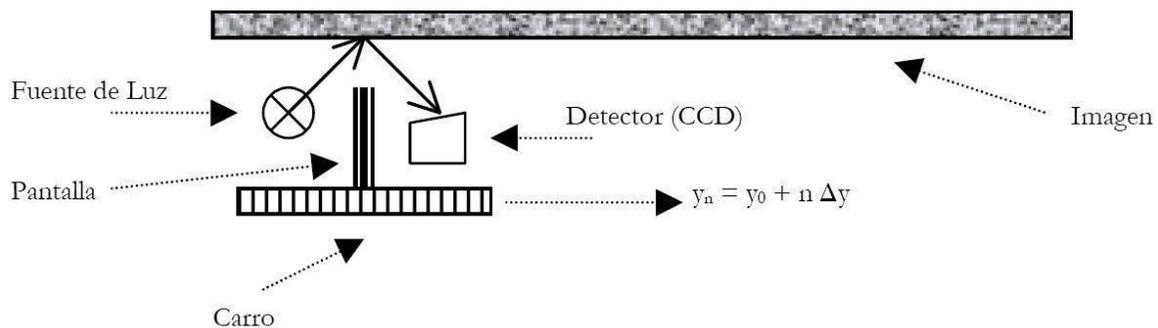


Imagen 1.8 Sistema de registro por reflexión

Es claro que dependiendo del tipo de luz que se utilice y las propiedades de reflectividad de la imagen el registro variará. El objetivo de la pantalla es evitar que la luz de la fuente de luz llegue directamente al detector, de tal forma que lo que éste registra es la luz que se ha reflejado en la imagen, la cual contiene información de ella. El "carro" se mueve mediante un motor de pasos  $\Delta y$ . El sistema se compone de un cierto número de *CCD's* en la dirección perpendicular al moviendo del carro, los cuales registran de forma "paralela" la información de la luz reflejada a lo "ancho" de la imagen, éste arreglo de datos se almacena en forma de "renglones" en la matriz de digitalización. Y para cada "paso" del motor se hace el cambio de renglón generándose así las columnas de la matriz de datos. Cuando se ha recorrido toda la imagen se procede al almacenamiento.

El principio de las cámaras digitales es también la reflexión de la luz, solo que la fuente es externa al dispositivo (la cámara). En general puede haber varias fuentes de luz, pero solo hay un sistema de registro.

En todos los casos antes descritos los datos sufren un proceso de discretización o cuantización. Este proceso se refiere al hecho de que la información registrada no es almacenada de manera exacta como un número real (los cuales son densos), sino como enteros, ya que el sistema luego de tomar el dato (en general analógico) lo pasa por un “Convertidos Analógico-Digital” (DAC). Este paso ocasiona una perdida en la precisión de los registros. En la imagen 1.9 se muestra una curva analógica digitalizada a 8 niveles.

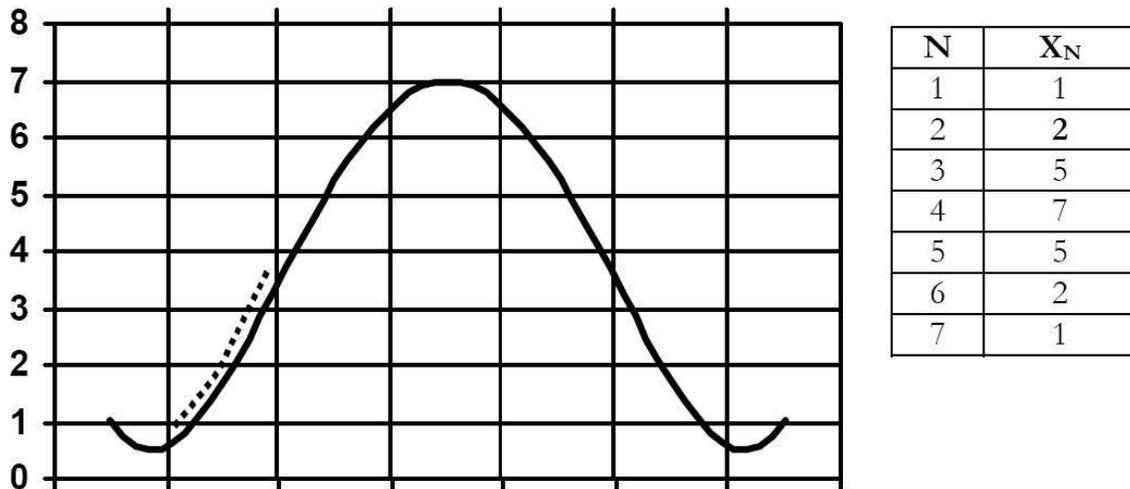


Imagen 2.9 Proceso de cuantización de datos

### 2.1.3.2. Almacenamiento de imágenes digitales

Una vez capturada la imagen necesitamos pasarla a un ordenador donde se almacenará y procesará. Si el dispositivo de captación genera una imagen digital (escáner o cámara digital) la imagen la podremos almacenar y procesar directamente. Si lo que genera es una imagen analógica deberemos de contar con una tarjeta digitalizadora o "*frame grabber*" que nos la convierta a formato digital.

Un "*frame grabber*" está formado por una serie de convertidores analógico/digitales, un área de memoria de imagen, una paleta de color o lut (*look-up table*) y una serie de convertidores digital analógico. La cámara genera una señal de vídeo que es digitalizada por medio de los convertidores analógico/digitales (A/D). Si cámara es en blanco y negro se utiliza un convertidor y si se trata de una cámara en color se utilizan tres, uno para cada banda. La imagen digitalizada es almacenada en el área de memoria. Para visualizarla se transforma la imagen digital en una señal analógica (señal de vídeo), mediante los convertidores digital/ analógico (D/A) y es visualizada en un monitor. La paleta de colores es una tabla donde a cada valor se asigna un color o un nivel de gris que será el que se utilice para representar la imagen en el monitor.

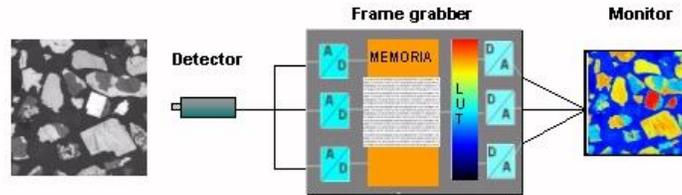


Imagen 2.10 Paso de imagen por *Frame Grabber*

Para almacenar una imagen en disco a memoria se han desarrollado una diversidad de formatos, los campos básicos a considerarse son los descriptores principales de la imagen que están definidos en la clase *ImagenenColor* o *Imagen* (que se relaciona con una imagen en tonos de gris).

Se almacena mínimamente: el ancho, el alto, la profundidad en bits y la matriz de píxeles. A esta representación se le llama un "Mapa de Bits" o simplemente *BitMap*.

Por ejemplo para guarda una imagen en tonos de gris adquirida de una fotografía de 2" por 3" con una resolución de 300 *dpi* (*dots per inch*) y una profundidad en bits de 8, se requiere un espacio de almacenamiento de:

$$N_1 = (2 \times 300)(3 \times 300) \text{ bytes} = 5.4 \times 10^5 \text{ bytes}$$

Que corresponde aproximadamente a ½ Mega Byte.

Si tomamos ahora una postal de (5" x 8") y la digitalizamos a 600 *dpi* en color a 24 bits, el espacio de almacenamiento requerido para el mapa de bits será:

$$N_2 = (5 \times 600)(8 \times 600) \text{ bytes} = 1.44 \times 10^7 \text{ bytes}$$

Que ahora corresponde a 13.73 MB. Puede notarse que el aumento de resolución al doble (300 → 600) y el aumento en el tamaño de la imagen en  $(\frac{5}{2}, \frac{8}{3}) \approx (2.5, 2.66)$  refleja el carácter no lineal de la transformación, donde  $N_2 = 2 \times 2 \times 2.5 \times (8/3) N_1 = 26.66 N_1$ , donde el crecimiento es aproximadamente 26.7 veces.

Debido al gran espacio que se requiere para el proceso, se han desarrollado formatos comprimidos para el almacenamiento de las imágenes. Estos utilizan diferentes técnicas para reducir el espacio de almacenamiento, algunas son: Método de árboles de Fuman, códigos de empaquetado por repetición de vecinos (*RLE*), manejo de autoreferencias (Métodos *LZ77*, *LZ78* y *LZW*) y métodos basados en la Transformada de Fourier, éstos últimos son los más eficientes en términos generales [36].

Con estos tamaños de imagen en unas pocas sesiones habremos acabado con la capacidad de un disco de varios Gigabytes. El tamaño de las imágenes también es un problema a la hora de transferirlas a través de una red. Las velocidades de transmisión de la red nos condicionan el tamaño de nuestras imágenes. Si estas velocidades no son muy altas deberemos utilizar formatos comprimidos.

Los formatos de imágenes más usuales son:

- **TIFF.** Es probablemente el formato de imagen más universal. Soporta imágenes de 24 bits por *píxel*, pudiendo ser comprimidas o no comprimidas. El formato comprimido utiliza algoritmos de compresión sin pérdida de información con relación al fichero original. - **GIF.** Es un formato comprimido muy utilizado para guardar iconos, gráficos e imágenes con pocos colores o en niveles de gris. Únicamente soporta imágenes de 8 bits por píxel (256 colores).
- **PCX:** establecido por ZsoftPC Paint Brush, es usado comúnmente por computadoras IBM compatibles. Éste es un formato que soporta 8 bits por *píxel* en color indexado.
- **BMP.** Es el formato de imágenes de Microsoft. Usado frecuentemente en programas de Windows. Soporta imágenes de 24 bits sin comprimir, por lo que su tamaño suele ser grande.
- **PICT.** Considerado el formato estándar de imágenes para Macintosh. Soporta imágenes de 24 bits por píxel.
- **GIF.** Es un formato comprimido muy utilizado para guardar iconos, gráficos e imágenes con pocos colores o en niveles de gris. Únicamente soporta imágenes de 8 bits por píxel (256 colores).
- **JPG.** Es el formato comprimido más popular, compatible con gran número de plataformas. Los datos son comprimidos para eliminar información no detectable por el ojo humano. La eficiencia de la compresión es excelente pudiendo llegar a 1/20 o 1/30 del original, lo que hace que sea un formato muy aconsejable para cuando deseamos almacenar un gran número de imágenes en un espacio de disco limitado, o cuando deseamos transmitir esas imágenes. Soporta imágenes en color real de 24 bits por *píxel*. Existen tres variantes dentro de este formato, *Baseline* o estándar es un formato reconocido por casi todos los visualizadores de páginas Web; *Baseline Optimized* optimiza la calidad de los colores pero puede producir un archivo ligeramente más grande que no es soportado por todos los *Browsers*, y el *Progressive* que provoca que la imagen se vaya desplegando gradualmente en una serie de pasos (el usuario define cuántos). Sin embargo, este formato es más grande en tamaño y requiere mayor memoria para verse, además de no estar soportado por todos los *Browsers*.

#### 2.1.4. Modelo general para el procesamiento de imágenes

González [33], modela los pasos fundamentales que se siguen en el procesado de imágenes como se representa en la imagen 1.11.

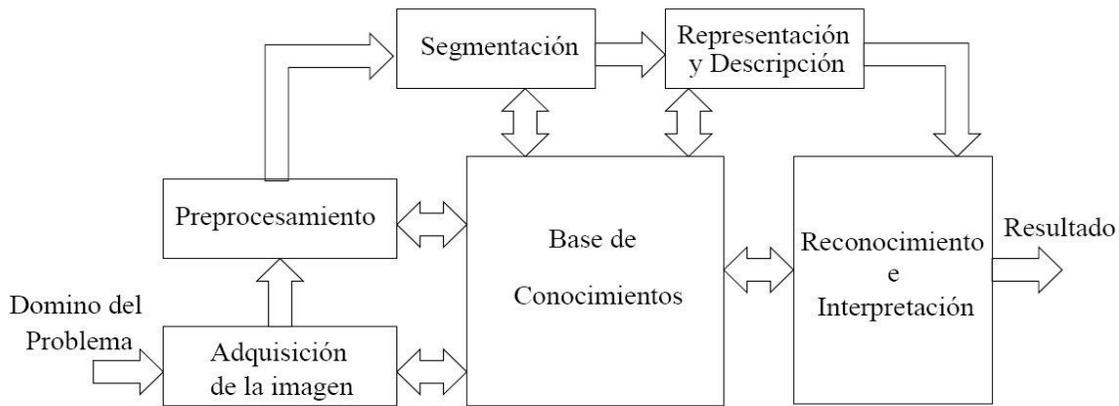


Imagen 2.11 Pasos fundamentales del procesamiento de imágenes [33]

Para ayudar a una conceptualización sencilla de este tema, se ilustrara con un ejemplo: se quieren usar las técnicas del procesado de imágenes para leer automáticamente la dirección en sobres de correo. Nuestro dominio del problema (ver imagen 1.11) en este ejemplo abarca los sobres del correo, y el objetivo es leer la dirección anotada sobre cada sobre.

Entonces, la salida deseada en este caso es una serie de caracteres alfanuméricos.

El primer paso del proceso es la adquisición de la imagen, esto es, digitalizarla. Algunos dispositivos con que se puede realizar esto se trataron en la sección anterior.

Después de que la imagen digital ha sido obtenida, el siguiente paso es el preprocesamiento. Su función es mejorar la imagen de manera que se incremente la oportunidad de éxito de los siguientes procesos. El preprocesamiento típicamente trata con técnicas para realizar el contraste y remover ruido.

El siguiente paso es la segmentación. Definida en forma amplia, la segmentación particiona una imagen de entrada en sus partes constituyentes u objetos. Generalmente, la segmentación automática es una de las tareas más difíciles en el procesamiento de imágenes. En términos del reconocimiento de caracteres, el rol de la segmentación es extraer caracteres individuales y palabras del fondo de la imagen.

La descripción llamada también selección de características, trata con extracción de los rasgos que resulta en alguna información cuantitativa de interés o

características que son básicas para diferenciar una clase de objetos con otra. En términos del ejemplo, para el reconocimiento de caracteres, descriptores tales como lagos (huecos) y bahías son rasgos poderosos que ayudan a diferenciar una parte del alfabeto de otra.

El último estado comprende al reconocimiento y la interpretación. El reconocimiento es el proceso que etiqueta, o asigna un nombre, a un objeto basándose en la información que proveen sus descriptores. La interpretación involucra la asignación de significado a un conjunto de objetos reconocido. En términos del ejemplo, identificar un carácter como, digamos, una c, requiere la asociación de los descriptores para este carácter con la etiqueta c.

La base de conocimientos mantiene todo el conocimiento que se tiene acerca del problema tratado. Este conocimiento puede ser simple, teniendo solo las regiones de la imagen donde la información de interés se conoce que está localizada, con lo que se limita la búsqueda que conduce a esa información. La base de conocimientos también puede ser compleja, tal como una lista interrelacionada de todos los defectos mayores posibles en un problema de inspección de materiales o una base de datos de imágenes conteniendo imágenes de satélite en alta resolución de una región en conexión con aplicaciones de detección de cambios. Además de guiar la operación de cada módulo del proceso, la base de conocimientos también controla la interacción entre módulos. Representando con flechas de doble punta en la imagen 1.11. Con esto se indica que la comunicación entre módulos del proceso generalmente está basada sobre conocimiento previo de que resultado se podría esperar. Por ejemplo, para que una máquina concluya que una cadena de caracteres es un código postal, el sistema debe estar dotado del conocimiento necesario para reconocer el significado de la localización de la cadena con respecto a otros componentes en un campo correspondiente a una dirección. Este conocimiento no sólo guía la operación de cada módulo, sino también ayuda en las operaciones de retroalimentación entre módulos a través de la base de conocimientos. Como en el caso, una cadena de números en la localización correcta pero que consisten de sólo cuatro caracteres (uno de los cuales pudo no haberse reconocido) podría guiar al módulo de interpretación que “sospeche” que dos caracteres están unidos. Una retroalimentación que mande un mensaje, a través de la base de conocimientos, a la etapa de segmentación para que “mire” otra vez, es un ejemplo de desempeño de tareas en el procesamiento de imágenes.

Es importante hacer notar que los resultados del procesamiento pueden verse en la salida de cualquier paso de la imagen 1.11. También debe notarse que no todas las aplicaciones del procesamiento de imágenes requieren la complejidad de interacciones de la imagen 1.11. Numerosas aplicaciones caen fuera de este esquema. De hecho, no todos los módulos, son siempre necesarios. Por ejemplo, el realce de imágenes para interpretación visual humana rara vez va más allá de la etapa de preprocesamiento. En general, las funciones de procesamiento que incluyen reconocimiento e interpretación están asociadas con aplicaciones de

análisis de la imagen en la cual el objetivo es que se extraiga la información de una imagen en forma automática o semiautomática.

## **2.1.5 Procesamiento Digital de Imágenes**

### **2.1.5.1 Segmentación**

Cuando se trabaja una imagen digital, es necesario extraer la información semántica requerida (el significado de la imagen y sus partes) directamente de la representación digitalizada, por ello, se hace necesario el reconocimiento o separación de objetos. La segmentación de imágenes es una técnica de descomposición de imágenes en partes con significado para separar objetos del entorno o fondo (*background*) con objeto de distinguirlos e integrarlos en una descripción estructurada de la imagen original.

El proceso de segmentación consiste en la identificación de las regiones de objetos o límites, que se distinguen por los cambios de contrastes en la escala de grises o las texturas; con lo que facilitan la identificación de regiones con áreas homogéneas.

La segmentación por escala de grises o por texturas se utiliza para clasificar grupos de *píxel* como parte de un objeto o en el fondo.

Las técnicas mencionadas anteriormente forman parte del campo del procesamiento de imágenes. Éste es utilizado en diversas áreas como medicina, percepción remota, astronomía, química, biología, biometría, etcétera, por mencionar algunas.

Regularmente el software comercial para edición y modificación de imágenes, contiene una serie de operadores que en muchos casos, realizan algunas de las técnicas básicas. De la misma forma existe también software más especializado, el cual está principalmente enfocado a analizar y procesar cierto tipo de imágenes, como las médicas o de percepción remota.

Si bien es cierto que gran parte del procesamiento de imágenes está enfocado al ámbito científico, muchos de los operadores que contiene el software tanto comercial como público, pueden aplicarse por otra área importante, como lo es el diseño gráfico, que utiliza las herramientas del procesamiento de imágenes para darle un toque creativo y original a imágenes que pretenden lograr otro propósito que no sea la extracción de características de éstas. Es común que los diseñadores gráficos y la gente dedicada al campo de la multimedia, utilicen este tipo de herramientas para crear y manipular imágenes con fines artísticos.

### 2.1.5.2. Transformaciones de la imagen

- *Crop*: esta operación sirve para cortar o seleccionar una zona de interés en la imagen.
- *Flop*: invierte la imagen en el plano horizontal, es decir, da la apariencia de ponerla de cabeza.
- *Flip*: invierte la imagen pero ahora en el plano vertical, es decir, como si fuera el resultado de reflejarla en un espejo.
- *Rotate*: aplica una rotación a la imagen, tomando como eje de rotación el centro de ésta o algunos de los puntos en los vértices.

### 2.1.5.3 Realce

- *Hue*: modifica el color de la imagen, es similar a modificar el “tinte” de ésta.
- *Saturation*: es la cantidad de color presente en un determinado momento, por ejemplo, sirve para distinguir entre el rojo y el rosa.
- *Brightness*: modifica la intensidad o brillo de la imagen.
- *Gamma Correction*: la corrección del gamma se relaciona con modificar el factor de brillo y contraste, es similar a cambiar el ángulo de apertura de una lente en una cámara, mientras más grande sea el ángulo, mayor será la cantidad de luz, por tanto la imagen tendrá mayor brillo; en cambio mientras más pequeño sea el ángulo de apertura, menor será la cantidad de iluminación lo que generará una imagen oscura.
- *Equalize*: este operador utiliza el histograma de una imagen, con la finalidad de tratar de igualar las tonalidades de la imagen, es decir, lograr que los tonos de la imagen sean más uniformes.
- *Normalize*: efectúa una reasignación de valores a partir del valor más alto de cada banda de color; de esta manera modifica ese valor en relación con el valor más alto en la escala, para reasignar o remapear todos los demás valores de acuerdo con la que tenían originalmente.
- *Negative*: obtiene el negativo de la imagen. El proceso es similar a tener el negativo de una fotografía.
- *Grey Scale*: convierte una imagen de color en tonos de grises.

### 2.1.5.4. Efectos

- *Emboss*: realza los bordes verticales de una imagen.
- *Add Noise*: este operador añade ruido a la imagen, es decir, modifica la imagen al agregar puntos de manera aleatoria para que éstos contrasten con la imagen.
- *Sharpen*: permite realzar los bordes de la imagen sin que el fondo de ésta se afecte. Esta operación permite ver con mayor nitidez los bordes.
- *Blur*: al aplicar este operador a una imagen, ésta se verá borrosa, debido a que se efectúan promedio de *píxel* en una dirección, ya sea por columnas o por renglones, de manera que la imagen pierda nitidez.

- *Threshold*: separa o realza valores específicos de la imagen, es decir, a este operador se le indica el valor de un *píxel*, que tendrá la función de un umbral; de esta forma los valores que se encuentren por arriba o por abajo (según sea el caso) de ese umbral, serán modificados a un solo valor, ya sea un 0 (negro) o un 255 (blanco).
- *Edge Detect*: esta operación permite detectar bordes de una imagen, tanto horizontales como verticales.
- *Spread*: modifica una imagen de tal forma que se aprecie como si hubiese sido creada por muchos puntos.
- *Shade*: este operador funciona como si estuviera cambiando el punto de iluminación, realza las partes sombreadas de una imagen.

### 2.1.5.5. Efectos especiales

- *Swirl*: produce un efecto semejante al de un remolino, es decir, al aplicar este operador la imagen se deformará en forma de remolino.
- *Oil Painting*: este efecto hará que la imagen se modifique y se observe como si se hubiese pintado al óleo.

### 2.1.5.6. Vista

- *Zoom*: esta operación le permitirá acercarse o alejarse de la imagen.

### 21.5.7. Filtros

La aplicación de filtros se basa en la inspección de los vecinos de un *píxel*  $p[i, j]$  y una máscara, dada la vecindad de  $p[i, j]$

$p[i-1, j-1]$	$p[i, j-1]$	$p[i+1, j-1]$
$p[i-1, j]$	$p[i, j]$	$p[i+1, j]$
$p[i-1, j+1]$	$p[i, j+1]$	$p[i+1, j+1]$

donde los valores de  $p[a, b]$  representan el tono correspondiente a la coordenada  $(a, b)$  en la imagen en escala de gris o bien a cada plano de color de una imagen en codificación RGB (*Red, Green, Blue*).

Así si la máscara del filtro esta dada por,

$$m = A \begin{pmatrix} m[i-1, j-1] & m[i, j-1] & m[i+1, j-1] \\ m[i-1, j] & m[i, j] & m[i+1, j] \\ m[i-1, j+1] & m[i, j+1] & m[i+1, j+1] \end{pmatrix}$$

donde  $A$  se denomina peso de la transformación.

Al aplicar la máscara  $m$  sobre  $p[i, j]$ , se sustituirá el tono del *píxel*  $p[i, j]$  por

$$p[i, j] \leftarrow A \sum_{j=-1}^1 \sum_{i=-1}^1 p[i, j] m[i, j]$$

A esta familia de máscaras se les llama *máscaras de  $3 \times 3$*  por la dimensión de la matriz que las define. De igual forma se puede construir máscaras de  $5 \times 5$  y otras dimensiones. Para la implementación se suele usar vectores en vez de matrices.

#### 2.1.5.7.1 Filtros Básicos

##### 2.1.5.7.1.1 Promedio Estándar

Es el filtro más simple y se basa en el promedio aritmético de los primeros vecinos del *píxel*, la forma de la máscara es

$$m = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Este filtro en principio tiene las propiedades de: suavizar, eliminar ruido y difuminar bordes.

##### 2.1.5.7.1.2 Promedio Pesado

Este filtro se basa en el *promedio* aritmético de los primeros vecinos de un *píxel* dándole mayor peso al *píxel* del centro ( $N > 1$ ), la forma de la máscara es,

$$m = \frac{1}{N+8} \begin{pmatrix} 1 & 1 & 1 \\ 1 & N & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Los efectos de este filtro son similares al *promedio estándar* y su efecto depende del valor de  $N$ . A mayor valor de  $N$  se dará más importancia al *píxel* central.

#### 2.1.5.7.2 Filtros Realzantes

Estos filtros tienen como propósito destacar algunos detalles de las imágenes que se encuentren difuminados.

### 2.1.5.7.2.1. Paso Alto Básico

Para realizar el realce se da un peso alto al *píxel* central y a los primeros vecinos un peso negativo bajo.

$$m = \frac{1}{9} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ 1 & -1 & -1 \end{pmatrix}$$

Puede notarse que la suma de los pesos es cero, por lo tanto el valor resultante será pequeño o cero para zonas de tono constante. Este filtro realiza una mejor en los bordes de la imagen sobre un fondo oscuro. Una variante de este filtro es el *high-pass* o *enfático de frecuencias altas* que incrementa el peso del *píxel* central. La forma de la máscara es ( $A > 1$ ):

$$m = \frac{1}{9A} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9A-1 & -1 \\ 1 & -1 & -1 \end{pmatrix}$$

Mientras mayor sea  $A$  respecto a 1 el efecto de realce será mayor. El primer filtro expuesto es un caso particular con ( $A = 1$ ).

### 2.1.5.7.2.2. Suavizado

Este remueve las frecuencias altas (cambios bruscos entre los píxeles vecinos) y retiene las bajas frecuencias (cambios graduales entre píxeles vecinos).

$$m = \frac{1}{32} \begin{pmatrix} 1 & 3 & 1 \\ 3 & 8 & 3 \\ 1 & 3 & 1 \end{pmatrix}$$

### 2.1.5.7.2.3. Filtros Diferenciales

El proceso de promediado con diferentes pesos se puede considerar como un proceso de integración, puede verse en general que la imagen tiende a hacerse borroso por estos mecanismos. La transformación inversa, es decir la diferenciación, debe producir el efecto contrario: *aumentar la nitidez de la imagen*.

Dado que la operación sobre las imágenes es sobre los píxeles que la conforman, entonces la diferenciación puede ser entendida como un proceso de *diferencias finitas*, de tal forma que el proceso:

$$f' = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - F(x)}{\Delta x}$$

se puede aplicar entre dos píxeles como

$$\delta f_x = f(x+1) - f(x) \text{ o bien } \delta f_y = f(y+1) - f(y),$$

dependiendo de la dirección de interés.

Algunos de estos filtros se aplican a regiones de  $2 \times 2$ , dada la región,

$$\begin{array}{cc} z_1 & z_2 \\ z_3 & z_3 \end{array}$$

### 2.1.5.7.2.3. Bordes Simples

Esta es la transformación más elemental, se basa en la primera derivada y la regla sobre  $z_1$ , para los **bordes horizontales** es simplemente,

$$z_1 \leftarrow |z_2 - z_1|,$$

y para los **bordes verticales** tenderemos

$$z_1 \leftarrow |z_3 - z_1|$$

La idea del filtro es que al encontrarse una zona donde el tono sea constante el valor resultante será cero y donde se encuentren cambios, estos se reflejarán en la transformación, ya que esta será no nula.

#### 2.1.5.7.2.4. Gradiente en Valor Absoluto (L1)

Se calcula la combinación de las dos derivadas,

$$z_1 \leftarrow |z_2 - z_1| + |z_3 - z_1|,$$

debe tenerse cuidado cuando las dos diferencias corresponda al *máximo* del tono en la escala de trabajo, lo cual es válido para una operación sobre grises o para un plano de color *RGB*. Es claro que de haber una diferencia de esta clase en ambas direcciones (*X* e *Y*) la sustitución puede saturar el nivel permitido, ya que el resultado será dos veces el valor máximo. Por lo tanto una corrección que debemos hacer es incluir un factor para evitar la saturación. La expresión para hallar los bordes en las dos direcciones será entonces,

$$z_1 \leftarrow \frac{1}{2} (|z_2 - z_1| + |z_3 - z_1|),$$

o bien usando un operador *bit a bit*,

$$z_1 \leftarrow (|z_2 - z_1|) \text{ OR } |z_3 - z_1|,$$

Este segundo método no requiere el factor de normalización.

#### 2.1.5.7.2.5. Gradiente con Cuadrados (L2)

Se calcula la combinación de las dos derivadas mediante la norma L2 del gradiente,

$$z_1 \leftarrow 0.7071 \sqrt{(z_2 - z_1)^2 + (z_3 - z_1)^2},$$

donde el factor de peso 0.7071 corresponde a la normalización de  $1/\sqrt{2}$  para evitar la saturación.

#### 2.1.5.7.2.6. Gradiente de Robert's

Se calcula la combinación de las dos derivadas, pero en diagonal, estas se llaman derivadas cruzadas. Se pueden usar la norma L1 o bien la L2 para el gradiente, en la norma L1 se tendrá,

$$z_1 \leftarrow \frac{1}{2} (|z_4 - z_1| + |z_3 - z_2|),$$

y en L2

$$z_1 \leftarrow 0.7071 \sqrt{(z_4 - z_1)^2 + (z_3 - z_2)^2}.$$

La ventaja de esta definición es que se toman los cuatro píxeles del entorno, pues en la implementación simple el píxel  $z_4$  no participa en el cálculo de la derivada.

#### 2.1.5.7.2.7. Operadores de Prewitt

Esto se definen para entornos de  $3 \times 3$  y se basan en una aproximación de la derivada, la idea es tomar varios píxeles alrededor y tener un valor medio para la derivada con más información de los vecinos. Las máscaras son; para los bordes en X

$$D_x = A \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix},$$

y para los bordes en Y,

$$D_x = A \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix},$$

También se pueden aplicar las dos máscaras mediante la combinación OR.

#### 2.1.5.7.2.8. Laplaciano

Este es un filtro de paso alto que remueve los cambios finos y resalta los cambios bruscos, su forma es

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}.$$

Puede observarse que la suma de los términos es cero, por lo cual se acostumbra no usar un peso. Mas en general se debe tener cuidado con la saturación y prever el caso cuando el filtro arroje valores mayores que el máximo tono permitido.

## 2.2. Redes Neuronales Artificiales RNA

### 2.2.1. HISTORIA

Los intentos por imitar el funcionamiento del cerebro han seguido la evolución del estado de la tecnología. Por ejemplo, al finalizar el siglo XIX se le comparó con la operación de la bomba hidráulica; durante la década de 1920 a 1930 se intentó utilizar la teoría de la conmutación telefónica como punto de partida de un sistema de conocimiento similar al del cerebro. Entre 1940 y 1950 los científicos comenzaron a pensar seriamente en las redes neuronales utilizando como concepto la noción de que las neuronas del cerebro funcionan como interruptores digitales (*on - off*) de manera también similar al recién desarrollado computador digital. Así nace la idea de "revolución cibernética" que maneja la analogía entre el cerebro y el computador digital [50].

Las primeras explicaciones teóricas sobre el cerebro y el pensamiento fueron dadas por algunos filósofos griegos, como Platón y Aristóteles, quienes fueron apoyados después por Descartes y filósofos empiristas.

A finales del siglo XIX se logró una mayor claridad sobre el trabajo del cerebro debido a los trabajos de Santiago Ramón y Cajal en España, quien identificó la célula encargada de transmitir información dentro del sistema nervioso, la neurona, y Charles Scott Sherrington en Inglaterra, que trabajo en los puntos de conexión de las neuronas o sinápsis.

Alan Turing, en 1936, fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación, pero quienes primero concibieron algunos fundamentos de la computación neuronal fueron Warren McCulloch y Walter Pitts en 1943, después otras teorías iniciales fueron expuestas por Donald Hebb en 1949. Pero solo hasta 1957 Frank Rosenblatt comenzó el desarrollo del Perceptrón que es una generalización del modelo de McCulloch-Pitts, la red neuronal más antigua.

Más adelante apareció el modelo ADALINE, desarrollado por Bernard Widrow y Marcial Hoff entre 1959 y 1960. Stephen Grossberg es el más influyente y formal de todos los investigadores en Redes Neuronales Artificiales, realizó *Avalancha* en 1967. Shun-Ichi Amari sus estudios incluyen el tratamiento de Redes Neuronales Artificiales dinámicas y aleatoriamente conectadas, estudios de aprendizaje competitivo, así como el análisis matemático de memorias asociativas publicados entre 1971 y 1982. Marvin Minsky y Seymour Papert, después James Anderson desarrollo el Asociador Lineal, en Japón Kunihiko Fukushima empezó a trabajar sobre modelos espaciales y espacio-temporales para sistemas de visión y el cerebro, su trabajo más notable ha sido la creación de un paradigma de Red Neuronal Artificial multicapa para visión trabajo llamado *Cognitron*, trabajo realizado entre 1969 y 1980. Desde 1971 hasta 1977 Teuvo Kohonen trabajo con memorias asociativas y matrices de correlación y sistema de aprendizaje competitivo [51]. En 1982 John Hopfield describió un método de análisis del

estado estable en una red autoasociativa. Bart Kosko en 1987 creó una familia de paradigmas de Redes de Neuronas Artificiales que extienden a las auto asociativas de Hebb de una nivel, a dos niveles utilizando aprendizaje sin supervisión; y son capaces de converger a una solución mínima dada una matriz arbitraria.

### 2.2.2. Fundamentación Biológica

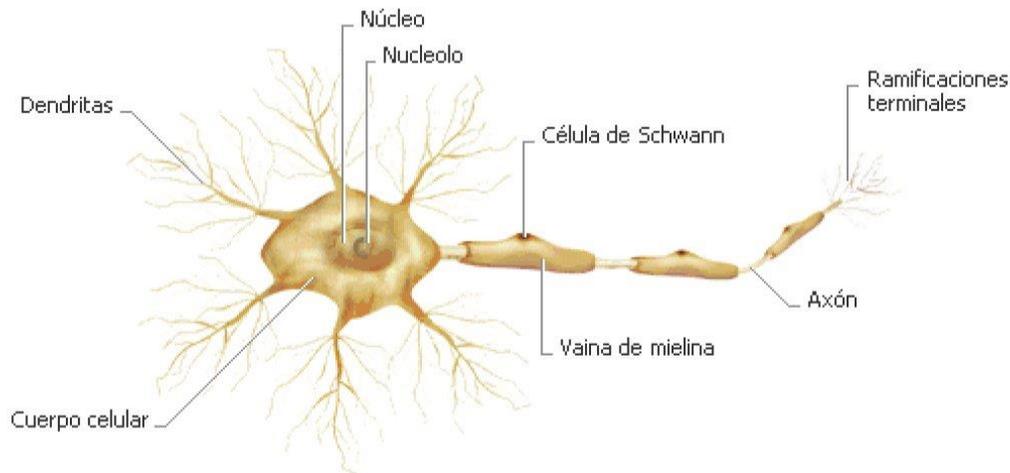


Imagen 2.3.1 Estructura Neurona biológica

Se estima que en cada milímetro del cerebro hay cerca de 50.000 neuronas y en cada cerebro existen alrededor de 100.000 millones de neuronas conectadas cada una de ellas con alrededor de 10.000 [52], es decir que cada actividad neuronal afecta a otras 10.000 neuronas, lo cual forma una red de tamaño enorme. Un dispositivo de características similares es imposible de fabricar con la tecnología actual. Los intentos más cercanos que se han llevado a cabo han sido con redes de solo un millón de procesadores en las cuales cada procesador se conectaba únicamente con sus ocho adyacentes, lo cual queda varios órdenes de magnitud por debajo de lo deseable. Esta es una de las grandes limitaciones que tienen los sistemas artificiales existentes en la actualidad.

El tamaño y la forma de las neuronas es variable, pero con las mismas subdivisiones que muestra la Imagen 3.1. El cuerpo de la neurona o Soma contiene el núcleo. Se encarga de todas las actividades metabólicas de la neurona y recibe la información de otras neuronas vecinas a través de las conexiones sinápticas.

Las dendritas son las conexiones de entrada de la neurona. Por su parte el axón es la salida de la neurona y se utiliza para enviar impulsos o señales a otras células nerviosas. Cuando el axón está cerca de sus células destino se divide en muchas ramificaciones que forman sinápsis con el soma o axones de otras células. Esta unión puede ser inhibitoria o excitadora según el transmisor que las

libere. Cada neurona recibe de 10.000 a 100.000 sinápsis y el axón realiza una cantidad de conexiones similar.

La transmisión de una señal de una célula a otra por medio de la sinápsis es un proceso químico. En él se liberan sustancias transmisoras en el lado del emisor de la unión. El efecto es elevar o disminuir el potencial eléctrico dentro del cuerpo de la célula receptora. Si su potencial alcanza el umbral se envía un pulso o potencial de acción por el axón. Se dice, entonces, que la célula se disparó. Este pulso alcanza otras neuronas a través de la distribución de los axones.

Aunque los axones puedan parecer hilos conductores aislados, no conducen los impulsos eléctricos de igual forma, como hilos eléctricos no serían muy valiosos, pues su resistencia a lo largo del eje es demasiado grande y a resistencia de la membrana demasiado baja; la carga positiva inyectada en el axón durante el potencial de acción queda disipada uno o dos milímetros más adelante, para que la señal recorra varios centímetros es preciso regenerar frecuentemente el potencial de acción a lo largo del camino la necesidad de reforzar repetidamente esta corriente eléctrica limita a unos 100 metros por segundo la velocidad máxima de viaje de los impulsos, tal velocidad es inferior a la millonésima de la velocidad de una señal eléctrica por un hilo de cobre.

Los potenciales de acción, son señales de baja frecuencia conducidas en forma muy lenta, estos no pueden saltar de una célula a otra, la comunicación entre neuronas viene siempre mediada por transmisores químicos que son liberados en las sinápsis. Un ejemplo de comunicación entre neuronas y del proceso químico de la liberación de neurotransmisores se ilustra en la Imagen 3.2.

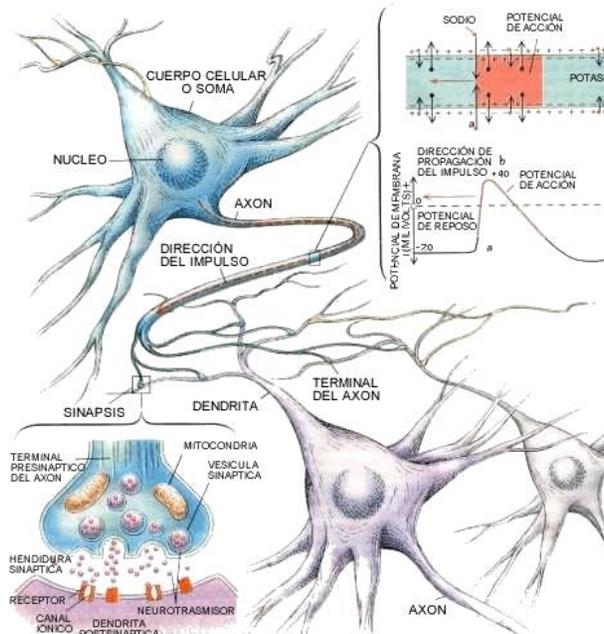


Imagen 2.3.2 Comunicación entre Neuronas [53]

Cuando un potencial de acción llega al terminal de un axón son liberados transmisores alojados en diminutas vesículas, que después son vertidos en una hendidura de unos 20 nanómetros de anchura que separa la membrana presináptica de la postsináptica; durante el apogeo del potencial de acción, penetran iones de calcio en el terminal nervioso, su movimiento constituye la señal determinante de la exocitosis sincronizada, esto es la liberación coordinada de moléculas neurotransmisoras. En cuanto son liberados, los neurotransmisores se enlazan con receptores postsinápticos, instando el cambio de la permeabilidad de la membrana [53].

Cuando el desplazamiento de carga hace que la membrana se aproxime al umbral de generación de potenciales de acción, se produce un efecto excitador y cuando la membrana resulta estabilizada en la vecindad el valor de reposo se produce un efecto inhibitor. Cada sinápsis produce sólo un pequeño efecto, para determinar la intensidad (frecuencia de los potenciales de acción) de la respuesta cada neurona ha de integrar continuamente hasta unas 1000 señales sinápticas, que se suman en el soma o cuerpo de la célula.

En la Imagen 3.3 se visualiza el proceso químico de una sinápsis y los diferentes elementos que hacen parte del proceso tanto en la neurona presináptica, como en la postsináptica.

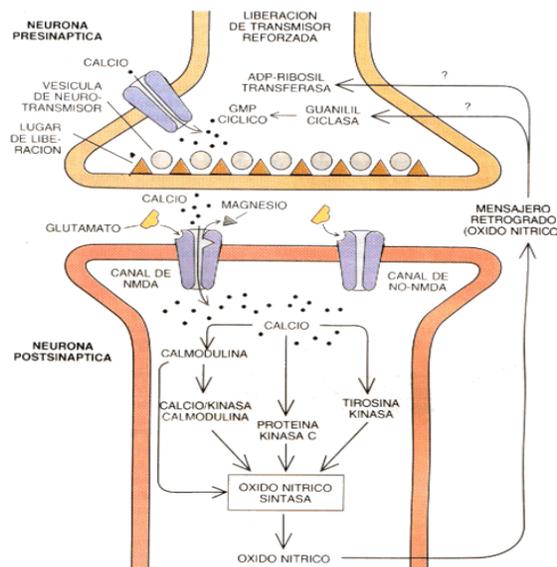


Imagen 2.3.3 Proceso químico de una sinápsis [53]

Nuestro comportamiento, inteligente o no, y el del resto de los animales superiores, siguen un esquema de este tipo. Y son estos mecanismos los que tratan de incorporar los modelos artificiales que ha ido apareciendo a lo largo de la historia de las Redes Neuronales Artificiales.

## 2.2.3. Red de Neuronas Artificiales

### 2.2.3.1. Redes con Aprendizaje Supervisado

El proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor comprueba la salida de la red y en el caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida se aproxime a la deseada.

Se consideran tres formas de llevar a cabo este tipo de aprendizaje:

- Aprendizaje por corrección de error: Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error.
- Aprendizaje por refuerzo: Se basa en la idea de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. La función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito=+1 o fracaso=-1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades.
- Aprendizaje estocástico: Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

### 2.2.3.2. Redes con Aprendizaje No Supervisado

Estas redes no requieren influencia externa para ajustar los pesos de las conexiones entre neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada es o no correcta, así que existen varias posibilidades en cuanto a la interpretación de la salida de estas redes.

En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado en el pasado. En otro caso podría realizar una codificación de los datos de entrada, generando a la salida una versión codificada de la entrada, con menos bits, pero manteniendo la información relevante de los datos, o algunas redes con aprendizaje no supervisado lo que realizan es un mapeo de características, obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada, de tal forma que si se presentan a la red información similar, siempre

sean afectadas neuronas de salidas próximas entre sí, en la misma zona del mapa.

En general en este tipo de aprendizaje se suelen considerar dos tipos:

- Aprendizaje Hebbiano: Consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación, así si las dos unidades son activas (positivas), se produce un reforzamiento de la conexión. Por el contrario cuando una es activa y la otra pasiva (negativa), se produce un debilitamiento de la conexión.
- Aprendizaje competitivo y cooperativo: Las neuronas compiten (y cooperan) unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, solo una de las neuronas de salida se active (alcance su valor de respuesta máximo). Por tanto las neuronas compiten por activarse, quedando finalmente una, o una por grupo, como neurona vencedora.

#### 2.2.4. Estructura Básica de la Red

En la imagen 3.4 se muestra un ejemplo de una neurona estándar que compone una red de neuronas artificiales. A la izquierda se ve una serie de entradas a la neurona; cada una llega de la salida de otra neurona de la red o del exterior. Una vez calculada la salida de una neurona, ésta se propaga, vía conexiones de salida, a las células de destino. Todas las conexiones de salida reciben el mismo valor de salida.

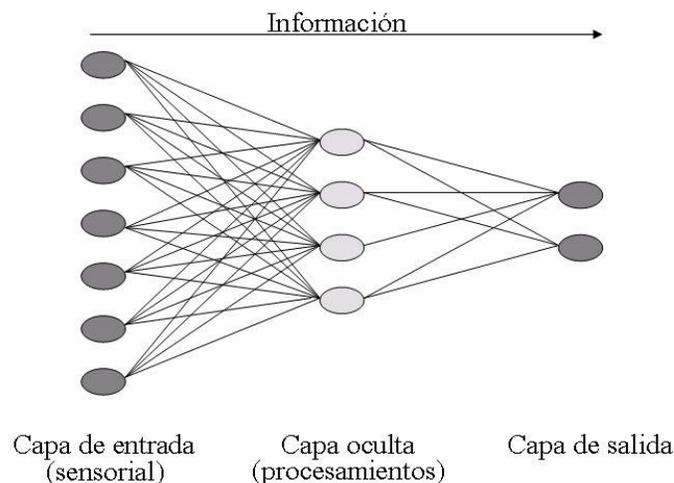


Imagen 3.5. Esquema de una Red de tres capas totalmente interconectadas [51]

## 2.2.5. Implementación de Redes Neuronales Artificiales

En la búsqueda de sistemas inteligentes en general, se ha llegado a un importante desarrollo del software, dentro de esta línea se encuentran algunos de los neuro-computadores más conocidos. Un neuro-computador es básicamente un conjunto de procesadores conectados con cierta regularidad que operan concurrentemente. En la actualidad ya existen una serie de neuro-computadores comerciales destinados a la realización de redes neuronales. Por otro lado la realización de Redes Neuronales Artificiales puede llevarse a cabo por medio de uno o varios circuitos integrados específicos, para así poder obtener una estructura que se comporte lo más similar posible a como lo haría una red neuronal. Otra tecnología que podría ser apropiada en la implementación de las redes neuronales es la tecnología electro óptica, con la ventaja de utilizar la luz como medio de transporte de la información, permitiendo la transmisión, masiva de datos.

### 2.2.5.1. Realización de Redes Neuronales

La realización más simple e inmediata consiste en simular la red sobre un ordenador convencional mediante un software específico. Es un procedimiento rápido, económico, e instituíble, pero su mayor desventaja radica en el hecho de que se intentan simular redes con un alto grado de paralelismo sobre máquinas que ejecuten secuencialmente las operaciones. Valores intrínsecos de las redes neuronales no pueden obtenerse de esta forma.

Otro método es la realización de redes neuronales a través de arquitecturas orientadas a la ejecución de procesos con un alto de paralelismo, tales como redes de *Transputers*<sup>2</sup>, arquitecturas sistólicas<sup>3</sup>, etc. Este método es una optimización del anterior, ya que el acelera el proceso, permitiendo una respuesta en tiempo real, pero el comportamiento real de la red sigue siendo simulado por una estructura ajena a la estructura intrínseca de una red neuronal.

Una tercera aproximación radicalmente distinta es la realización de redes neuronales mediante su implementación por uno o varios circuitos integrados específicos. Son los llamados chips neuronales. Las neuronas y las conexiones se emulan con dispositivos específicos, de forma que la estructura del circuito integrado refleja la arquitectura de la red. Se consigue de esta forma realizaciones que funcionan a alta velocidad.

### 2.2.5.2. Herramientas software de desarrollo

La comercialización de productos software es la forma más extendida para simular redes neuronales, debido a las ventajas citadas anteriormente. La diferencia entre

---

<sup>2</sup> Componente electrónico que permite a las máquinas lograr un paralelismo masivo en sus operaciones. Equivale a lo que es actualmente el transistor para las máquinas electrónicas en general, o sea un componente básico que puede ser fabricado en forma masiva y económica. El primero fue creado por la firma inglesa INMOS y presentado en La Feria Internacional de Componentes de 1983 en París.

<sup>3</sup> Arquitectura paralela que reemplaza un solo procesador con un arreglo de procesamientos regulares.

los distintos productos software radica en aspectos tales como el tipo y el número de arquitecturas de red que soporta, velocidad de procesamiento, interfaz gráfica, exportación de código C para el desarrollo automático de aplicaciones, etc.

Algunos de los productos comerciales son: NeuroSolution (Windows, OS9, OSX), ANSim (DOS), ExpertNet (DOS, Windows), Neuralesk (Windows), Neuralworks Pro II/Plus (DOS, OS/2, UNIX, VMS).

#### 2.2.5.3. Neuro-computadores de propósito especial y de propósito general

Como ya se mencionó anteriormente un neuro-computador es básicamente un conjunto de procesadores conectados con cierta regularidad que operan concurrentemente. Los de propósito general deben ser capaces de emular un gran número de modelos de red neuronal. Los de propósito especial han sido diseñados para implementar un modelo específico de red neuronal.

#### 2.2.5.4. Implementación microelectrónica (VLSI).

En general si la red ha estado previamente simulada y su configuración de conexiones perfectamente determinada, se busca la implementación sobre un circuito de conexiones fijas. La presentación del panorama que actualmente existe en el ámbito de la implementación VLSI de redes neuronales resulta complicada por la gran cantidad de aportaciones existentes, tanto en soluciones analógicas como digitales y con la alternativa de matriz de neuronas o de sinapsis.

### 2.2.6. PERCEPTRON *MULTILAYER*

#### 2.2.6.1. INTRODUCCIÓN

No obstante, la idea de combinar varios perceptrones sirvió como base para estudios posteriores realizados por Rumelhart, Hinton y Williams en 1986 [58]. Quienes reciben el mérito del éxito del algoritmo de aprendizaje *BackPropagation* o *BP*, dentro del grupo de trabajo PDP<sup>4</sup>, quienes lo presentaron a la comunidad internacional como una técnica útil de resolución de problemas complejos, lo que despertó el interés, no solo por el perceptrón, sino por el campo de la neurocomputación en general. Los importantes requisitos de cómputo que el algoritmo de *BP* precisa no podían ser satisfechos con los medios disponibles a principios de los setenta, por lo que el primer descubrimiento del *BP* [59] era quizás algo prematuro. En los años ochenta los computadores eran lo suficientemente potentes como para permitir la aplicación del algoritmo de *BP* a problemas de interés, lo cual permitió que el grupo PDP pudiera mostrar su gran potencial de la aplicabilidad a la resolución de tareas complejas.

---

<sup>4</sup>Grupo de Investigación de Procesamiento Distribuido Paralelo.

Diferentes autores [60, 61] han demostrado independientemente que el Perceptrón Multicapa es un aproximador universal, en el sentido de que cualquier función continua sobre un conjunto de  $R^n$  puede aproximarse con una Perceptrón Multicapa, con al menos una capa oculta de neuronas. Este resultado sitúa al Perceptrón Multicapa como una nueva clase de funciones (como pueden ser también los polinomios, las funciones trigonométricas, los splines, etc.), para aproximar o interpolar relaciones no lineales entre los datos de entrada y salida.

La habilidad de Perceptrón Multicapa para aprender a partir de un conjunto de ejemplos, aproximar relaciones no lineales, filtrar ruido en los datos, etc., hace que sea un modelo adecuado para abordar problemas reales, sin que esto indique que sean los mejores aproximadores universales.

Dentro del marco de las redes neuronales, el Perceptrón Multicapa es en la actualidad una de las arquitecturas más utilizadas en la resolución de problemas. Esto es debido, fundamentalmente, a su capacidad como aproximador universal, así como su fácil uso y aplicabilidad. Estas redes han sido aplicadas con éxito para la resolución de problemas en una gran variedad de áreas diferentes, como reconocimiento de habla [62], reconocimiento de caracteres ópticos [63], reconocimiento de caracteres escritos [64], control de procesos [65], modelización de sistemas dinámicos [66], conducción de vehículos [67], diagnósticos médicos [68], predicción de series temporales [69], etc.

Sin embargo, a pesar de ser una de las redes neuronales más conocidas y utilizadas, esto no implica que sea una de las más potentes y con mejores resultados en las diferentes áreas de aplicación. De hecho, el Perceptrón Multicapa posee una serie de limitaciones, como el largo proceso de aprendizaje para problemas complejos dependientes de un gran número de variables; la dificultad en ocasiones de codificar problemas reales mediante valores numéricos; la dificultad para realizar un análisis teórico de la red debido a la presencia de componentes no lineales y a la alta conectividad.

#### **2.2.6.2. Arquitectura del Perceptrón Multicapa**

Generalmente, todas las neuronas de una capa están conectadas a todas las neuronas de la siguiente capa. De este modo, las neuronas de la capa de entrada están conectadas a todas las neuronas de la primera capa oculta; las neuronas de la primera capa oculta se conectan a las neuronas de la siguiente capa, etc. Se dice entonces que existe conectividad total o que la red está totalmente conectada.

Aunque la mayor parte de los casos la arquitectura del Perceptrón Multicapa está asociada al esquema de la imagen 4.1, es posible también englobar dentro de este tipo de redes a arquitecturas con las siguientes características:

- Redes con conexiones de todas o ciertas neuronas de una determinada capa a neuronas de capas posteriores, aunque no inmediatamente posteriores.
- Redes en las que ciertas neuronas de ciertas capas no están conectadas a neuronas de la siguiente capa, es decir, el peso de la conexión es constante e igual a cero.

#### 2.2.6.2.1 Parámetros que determinan el aprendizaje de la Red

Cuando se trabaja con este tipo de redes neuronales artificiales, es necesario tomar una serie de decisiones, como por ejemplo:

- Cómo representar la información para crear los patrones de entrenamiento.
- Cómo determinar cuantas capas debe de tener la red neuronal artificial, y cuántas neuronas debe de tener cada una de las capas de dicha red.
- Qué valores deben de tener los parámetros que determinan el aprendizaje de la red: tasa de aprendizaje, momento, “*bias*” o polarización, etc.
- Cómo debe de realizarse el entrenamiento, etc.

En una red cada una de las capas tiene una función perfectamente definida. La capa de entrada recibe los vectores de entrada y los redistribuye a las neuronas de la capa intermedia. En este tipo de redes, esta capa no realiza ningún tipo de procesamiento sobre los datos, es sólo una receptora de información. Las neuronas de la capa intermedia transforman los vectores de entrada en vectores intermedios, que caracterizan los patrones de entrenamiento. La capa de salida, acepta un estímulo procedente de la capa intermedia y genera la respuesta de la red.

La habilidad de las redes perceptrón multicapa para caracterizar los patrones de entrenamiento, permite que la red genere su propio modelo de representación. El Teorema de Aproximación Universal [70, 71] se puede utilizar para demostrar que en este tipo de redes, sólo se necesita utilizar una capa de neuronas ocultas o intermedias para modelar una proyección no lineal entre los espacios de entrada y salida. El aumento del número de neuronas en la red, conlleva un aumento en la varianza de la misma, lo cual hace pensar que una red con una capa intermedia, puede generar mejores resultados que otra con dos o más capas, ya que cuanto menor sea la varianza mayor es la capacidad de generalización de la red [72].

#### 2.2.6.2.2. La función de activación

La función de activación es una de las partes más importantes de este tipo de redes, siendo en la mayoría de los casos una función de tipo sigmoideal. Cuanto mayor sea el gradiente de la función, más rápido se declinará la neurona por alcanzar un valor más próximo a 1 o a 0, y más complicado será cambiar su

respuesta si es equivocada. Por tanto, en casos en los que la red tiene que crear un modelo para solucionar un problema complejo, es más conveniente utilizar funciones de activación con una pendiente menor, y si por el contrario, los patrones de entrenamiento reflejan casos claros y problemas con soluciones sencillas, se puede utilizar una función con una pendiente mayor.

#### 2.2.6.2.3. Tasa de aprendizaje

La regla que rige el entrenamiento de las redes de perceptrón multicapa (determinada por una o más ecuaciones), caracteriza la forma sistemática en la que se modifican los pesos que unen las distintas capas de neuronas durante la etapa de entrenamiento.

Durante el entrenamiento, la tasa de aprendizaje  $h$  juega un papel fundamental, ya que la velocidad del aprendizaje y el tiempo que la red requiere para entrenarse, depende en gran medida de este parámetro. El valor de este parámetro tiene que estar entre 0 y 1, de otra forma la red nunca aprendería. Para mejorar las posibilidades de generalización de la red, es conveniente que a medida que la red vaya aprendiendo, el valor de la tasa de aprendizaje vaya disminuyendo hasta alcanzar un valor muy próximo a cero. Sin embargo, el valor de la tasa de aprendizaje no es crítico en la mayoría de los casos, a no ser que se trate de problemas excesivamente complejos.

#### 2.2.6.2.4. Momento

El objetivo del término momento  $m$ , es el de reducir el tiempo de entrenamiento de las redes neuronales. Además, al añadir el momento a la regla Delta de aprendizaje, se eliminan algunos de los factores que favorecen la aparición de mínimos locales. El momento hace que los pesos cambien su valor siempre en la misma dirección, a no ser que el sentido global del aprendizaje de la red indique un cambio de sentido. La inclusión del momento implica el almacenar los cambios realizados en cada uno de los pesos. La implementación más extendida del momento, es aquella que genera un valor acumulativo de la variación de los pesos a lo largo del entrenamiento para el total del conjunto de patrones. Una vez que todo el conjunto de entrenamiento se ha pasado por la red, se calcula la dirección general en la que los pesos se han desplazado y el momento se activa, así sucesivamente para cada una de las iteraciones por la red del conjunto completo de patrones de entrenamiento. El valor del momento puede mantenerse relativamente alto (siempre en el intervalo  $[0, 1]$ ), aunque este valor también debe reducirse a medida que la red va aprendiendo.

#### 2.2.6.2.5. “Bias” o polarización

En el perceptrón multicapa, la activación de cada una de las neuronas se produce al evaluar cada una de las funciones de activación, con los valores procedentes de las neuronas de la capa anterior. Las “bias” o polarizaciones, son unas neuronas que están unidas al resto de las neuronas de cada capa y que disparan siempre 1,

o dicho de otra forma, son unos pesos extras añadidos a las neuronas de las capas intermedias y a las de la capa de salida que proporcionan un nivel de activación independiente de la entradas y los vectores de entrenamiento. El valor de las “*bias*”, se modifica de la misma forma que el resto de los pesos de la red.

#### 2.2.6.2.6. Ruido

Todas las redes neuronales artificiales pueden llegar a memorizar los datos que intentan aprender. Una forma de impedir que esto suceda, es introducir ruido en los patrones de entrenamiento, de forma que la red nunca se entrene con exactamente el mismo patrón más de una vez. Las redes entrenadas con ruido son en general más robustas, y tienen más facilidad de enfrentarse a datos en los que hay ciertos componentes de ruido.

#### 2.2.6.2.7. Tamaño de las capas de neuronas

Definir el tamaño de las capas de neuronas es siempre difícil. Sólo la experiencia puede ayudar a definir el tamaño para la capa intermedia en función del número de neuronas de la capa de entrada, para que la red aprenda de forma rápida. Sin embargo, aprender rápido no implica aprender bien, en el sentido de crear un modelo adecuado de generalización.

Un número muy grande de neuronas en la capa intermedia, hace que la red requiera menos iteraciones de entrenamiento para aprender, aunque cada uno de los pasos requiere más tiempo de cómputo, ya que en general hay más pesos que ajustar. El número de neuronas en la capa intermedia tampoco debe de ser igual al de patrones de entrenamiento, ya que esto favorece la memorización de los patrones, puesto que la red hace que cada una de las neuronas de la capa intermedia se encargue de reconocer uno de los patrones de entrenamiento, en vez de generalizar a partir de casos individuales.

El número de neuronas debe de ser en general, menor que el número de patrones de entrenamiento para evitar problemas de memorización. El tamaño de la capa de entrada y salida, viene dado por la naturaleza del problema y el grado de precisión que se espera de la red.

#### 2.2.6.2.8. Conjunto de entrenamiento

El conjunto de datos utilizados para el entrenamiento de la red, es el que determina qué es lo que la red debería ser capaz de solucionar. Normalmente se reserva un número pequeño de patrones sin utilizar en el entrenamiento, para validar la capacidad de la red a la hora de solucionar el problema. Este conjunto de datos se denomina conjunto de validación.

El perceptrón multicapa tiene una gran capacidad de generalización. Generalizar en este sentido, significa que la red es capaz de crear un modelo a partir de los

datos del conjunto de entrenamiento, para ello obvia datos irrelevantes y resalta las similitudes generales que existen entre los casos. Un entrenamiento insuficiente de la red, hace que ésta no sea capaz de proporcionar respuestas claras, y por el contrario, un exceso de entrenamiento hace que la red memorice los vectores de entrenamiento. También es necesario que el conjunto de datos de entrenamiento sea representativo del problema, para que la red sea capaz de responder satisfactoriamente a cualquier tipo de entrada.

### 2.2.6.3. Método de aprendizaje del perceptrón multicapa

Para explicar cual es el método de aprendizaje del perceptrón multicapa se utiliza una red  $N-L-M$  (ver Imagen 4.2), es decir, una red con una sola capa intermedia, sin que ello suponga pérdida de generalidad, ya que el teorema de Aproximación Universal establece que todos los perceptrones tienen igual capacidad representativa independientemente del número de capas intermedias presentes.

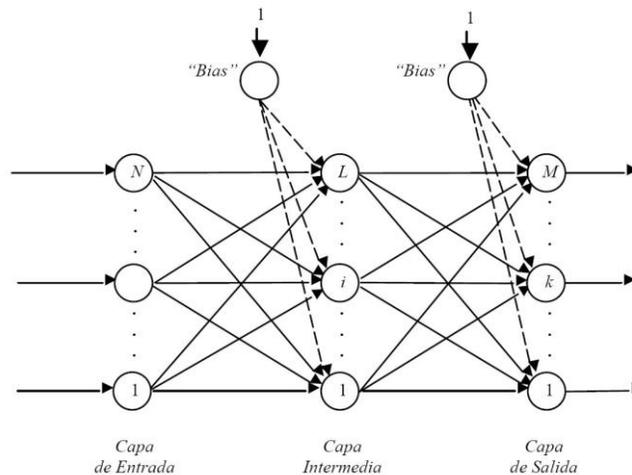


Imagen 4.2. Topología de un Perceptrón N-L-M

No existen reglas que indiquen en qué orden se deben presentar los pares de vectores de entrenamiento a la red. En general, es conveniente no entrenar la red de forma consecutiva con grupos de vectores que proporcionen la misma salida.

En términos de notación,

- Se denotará por  $x^p = (x_1^p, x_2^p, \dots, x_n^p)^T$  al  $p$ -ésimo vector de entrada del conjunto de entrenamiento.
- Se denotará por  $y^p = (y_1^p, y_2^p, \dots, y_n^p)^T$  al  $p$ -ésimo vector de salida esperado.
- Se utilizará el subíndice  $i$  para designar a las neuronas de la capa intermedia y el subíndice  $k$  para designar a las neuronas de la capa de salida. Así, por ejemplo, se denotará por  $s_i$ ,  $b_i$ , y  $d_i$  al nivel de excitación, "bias" y estado de activación de la neurona  $i$  de la capa intermedia, respectivamente. Análogamente para las neuronas de la capa de salida.

En primer lugar, el vector de entrada se propaga hacia la capa intermedia como se muestra en (4.2), por tanto para la neurona oculta  $i$ , el nivel de excitación  $s_i$ , ante el vector de entrada  $x^p$ , se determina mediante la siguiente regla de propagación:

$$s_i^p = \sum_{j=1}^n w_{ji} x_j^p + b_i \quad (4.1)$$

donde  $n$  es el número de entradas de la red,  $w_{ji}$  es el valor del peso de la conexión entre la neurona  $j$  de la capa de entrada y la neurona  $i$  de la capa intermedia y  $b_i$ , es el valor de las “bias” asociadas a la neurona  $i$ .

Si se asume que el estado de activación de la neurona  $i$ , es función del vector de entrada a la red, entonces la salida de las mismas viene dada por:

$$y_i^p = F_i(s_i^p) \quad (4.2)$$

Análogamente, para una neurona cualquiera  $k$  de la capa de salida, las ecuaciones que determinan su estado de activación son:

$$s_i^p = \sum_{i=1}^L w_{ik} y_j^p + b_k \quad (4.3)$$

$$y_k^p = F_k(s_k^p) \quad (4.4)$$

donde  $L$  es el número de neuronas de la capa intermedia,  $w_{ik}$  es el valor del peso de la conexión entre la neurona  $i$  de la capa intermedia y la neurona  $k$  de la capa de salida y  $b_k$  es el valor de las “bias” asociada a la neurona  $k$ .

El valor inicial de los pesos se genera de forma aleatoria, por ejemplo con valores en el intervalo  $[-0.1, 0.1]$ .

## **2.2.7. Redes Neuronales de Funciones de Base Radial RBF (*Radial Basis Function*)**

### **2.2.7.1. Introducción**

Las redes de neuronas de base radial son redes multicapa con conexiones hacia adelante (*feedforward*), al igual que el Perceptrón Multicapa antes visto. Las redes de base radial se caracterizan porque están formadas por una única capa oculta y cada neurona de esta capa posee un carácter local, en el sentido de que cada neurona oculta de la red se activa en una región diferente del espacio de patrones de entrada. Este carácter local viene dado por el uso de las llamadas funciones de base radial, generalmente la función Gaussiana, como funciones de activación. Las neuronas de la capa de salida de las redes de base radial simplemente realizan una combinación lineal de las activaciones de las neuronas ocultas.

La mayor contribución a la teoría, diseño y aplicaciones de las redes de neuronas de base radial se debe a Moody y Darken, Renals y a Poggio y Girossi [73, 74, 75]. Uno de los objetivos iniciales de los autores era construir una red de neuronas que requiriese un menor tiempo de aprendizaje que el que necesitaba el Perceptrón multicapa y, de este modo, disponer de una red de neuronas que pudiera ser apropiada para aplicaciones en tiempo real. Esto se consiguió incorporando funciones de activaciones locales en las neuronas ocultas de la red, lo cual permitía que solo unas pocas neuronas ocultas tuvieran que ser procesadas para nuevos patrones de entrada.

Al igual que el Perceptrón multicapa, las redes de neuronas de base radial son aproximadores universales, en el sentido de que puede aproximar cualquier función continua sobre un compacto de  $R^n$ . Una demostración formal de este resultado fue realizada por Park y Sandberg [76].

Las funciones de base radial definen hiperesferas o hiperelipses que dividen el espacio de entrada. Por tanto, cada neurona oculta de la red de base radial construye una aproximación local y no lineal en una determinada región de dicho espacio. Puesto que la salida de la red es combinación lineal de las funciones de base radial, las aproximaciones que construyen las redes de base radial son combinaciones lineales de múltiples funciones locales y no lineales. De este modo, se suele decir que las redes de base radial aproximan relaciones complejas mediante una colección de aproximaciones locales menos complejas, dividiendo el problema en subproblemas menos complejos. Esto hace que las aproximaciones construidas por las redes de base radial sean de naturaleza diferente a las aproximaciones globales y basadas en hiperplanos que construye el Perceptrón Multicapa.

Las redes de neuronas de base radial han sido aplicadas a una gran variedad de problemas, aunque es necesario señalar que su aplicación no ha sido tan extendida como en el caso del Perceptrón Multicapa. Sin embargo, se han utilizado en diferentes campos, como análisis de series temporales [77],

procesamiento de imágenes [78], diagnósticos médicos [79], reconocimiento automático del habla [79], etc.

### **2.2.7.2 Fritzke: aprendizaje rápido adaptativo del modelo RBF**

El entrenamiento de una RBF, requiere situar los centros en el lugar adecuado en función de la posición de los vectores del espacio de entrada. Esta operación normalmente se realiza utilizando alguno de los métodos expuesto anteriormente. Posteriormente, los pesos que comunican las neuronas del nivel medio con las del nivel de salida, se ajustan para reducir el error durante el entrenamiento utilizando la regla Delta. La complejidad de las redes RBF, depende de la dificultad para determinar qué centros se deben utilizar y dónde situarlos. La arquitectura propuesta por Fritzke [85], automatiza este proceso y garantiza un número de centros próximos al ideal.

#### **2.2.7.2.1 Iniciación de la red**

Inicialmente, se eligen dos vectores de forma aleatoria de entre los que componen el espacio de entrada. Ambos vectores se utilizan como centros en el nivel intermedio, asociando a cada centro una función gaussiana. La varianza de la función Gaussiana viene dada por la distancia euclídea media entre cada centro y los dos más próximos a él (inicialmente sólo hay dos centros y se utiliza la distancia entre ambos). El objeto de esta operación es superponer, en cierta medida, el conjunto de vectores del espacio de entrada asociados a cada centro, y así mejorar el proceso de generalización. El valor de los pesos iniciales se calcula aleatoriamente asignándoles un valor en el intervalo  $[-0.1, 0.1]$ .

#### **2.2.7.2.2 Adaptación de los centros y los pesos**

Una vez que un vector de entrada activa cada centro, esta activación se propaga hacia adelante a través de las conexiones ponderadas, y las neuronas del nivel de salida realizan la suma de los valores que provienen de dichas conexiones. La comparación entre el resultado obtenido y el deseado, proporciona el valor del error para este vector de entrada en particular. Dicho error es utilizado por la regla de los Mínimos Cuadrados Medios.

El centro más cercano a cada vector de entrada, se mueve hacia dicho vector un porcentaje  $\alpha$  de la distancia entre ambos. Esta técnica, hace que los centros se posicionen cerca de las zonas del espacio de entrada de mayor densidad. El objetivo de la adaptación, es forzar a que los centros estén tan cerca como sea posible del mayor número de vectores del espacio de entrada.

La regla Delta se utiliza para adaptar los pesos de las conexiones entre el nivel intermedio y el de salida. En particular, por cada par formado por un vector de entrada y uno de salida, se da un paso en la adaptación de los pesos de acuerdo con dicha regla.

### 2.2.7.2.3 Inserción de nuevos centros

Tras un cierto número de adaptaciones se introduce un nuevo centro, lo cual es necesario cuando el error  $E$ , es demasiado alto y no disminuye lo suficiente durante un intervalo predefinido de iteraciones.  $E$  es el error medio generado por todos los pares de vectores utilizados para entrenar la red, sin realizar ninguna modificación en los pesos y centros.

La introducción de un nuevo centro se realiza si el error no disminuye durante un cierto número de iteraciones (en el que no aumenta el número de centros), y además la regla delta y la adaptación de centros son incapaces de mejorar los resultados. Para determinar qué valor debe de tener este centro, se calcula la distancia euclídea entre cada centro y cada vector de entrada. Se selecciona aquel centro cuya distancia con todos los vectores del espacio de entrada sea mayor, y se le denomina  $G$ . El nuevo centro se introduce entre  $G$  y el centro más cercano a él.

### 2.2.7.2.4 Finalización del entrenamiento

Antes de empezar el entrenamiento, hay que determinar el número máximo de centros permitidos. Sin embargo, el error en las salidas puede ser suficientemente pequeño como para parar el proceso de entrenamiento de la RBF, sin que se llegue a alcanzar el número máximo de centros previamente definido.

Tanto el entrenamiento como la monitorización del mismo son continuos hasta que el error  $E$  es más pequeño que un valor preestablecido, o hasta que el número de centros es igual al máximo establecido y el error no disminuye. Este criterio permite que el tamaño de la red pueda ser menor que el de las RBF entrenadas con otras reglas.

## 2.2.8. Mapas Auto-Organizados De Kohonen – Som (*Self-Organizing Maps*)

### 2.2.8.1. Introducción

Existe un tipo de redes que se crean para realizar una proyección  $f: R^n \rightarrow R^m$ , presentando a la red un número de “ejemplos”  $(x^p, d^p)$  donde  $d^p = f(x^p)$ . Sin embargo, cuando se realizan operaciones de este tipo, pueden aparecer problemas si los patrones de entrenamiento no están disponibles, y la única información que se proporciona corresponde a vectores de entrada  $x^p$ . En estos casos la información relevante hay que buscarla dentro de los patrones de entrenamiento  $x^p$ . Este tipo de aprendizaje se denomina aprendizaje no supervisado. Los algoritmos no supervisados están normalmente basados en algún método de competición entre las neuronas.

Estos problemas se pueden resolver con los mapas autoorganizados, también llamados redes de Kohonen, donde el entrenamiento se lleva a cabo sin la presencia de un profesor externo.

Las características principales de este tipo de redes son:

- Poseen un solo nivel con muchas conexiones.
- Se necesita iniciar los pesos de estas conexiones.
- Las neuronas compiten de forma que las triunfadoras son las únicas que modifican los pesos asociados a sus conexiones.
- Estas redes agrupan las neuronas en diferentes clases, de tal forma que la topología de la red define cómo proyectar un espacio de entrada en otro, sin cambiar la configuración geométrica relativa del espacio de entrada.
- El aprendizaje es posible con la identificación de la estructura de los datos: correlaciones, agrupaciones, redundancia, etc.

En los mapas autoorganizados, las neuronas de salida se disponen habitualmente de forma matricial, aunque esto depende de la aplicación concreta. Esta disposición realizada por el usuario, determina qué neuronas de salida son vecinas.

Cuando los patrones de entrenamiento se presentan a la red, los pesos de las neuronas de salida se adaptan de forma que la clasificación presente en el espacio de entrada  $RN$ , se preserve en la salida. Esto quiere decir, que los patrones de entrenamiento que están próximos entre sí en el espacio de entrada (donde la proximidad se determina por alguna métrica, normalmente la distancia euclídea), deben proyectarse en las neuronas de salida que están también próximas unas a otras. Así, si las entradas están uniformemente distribuidas en  $RN$ , y se mantiene el orden, la dimensión de  $S$  (espacio de salida) debe ser por lo menos  $N$ . Sin embargo, si las entradas están restringidas a un sub-espacio de  $RN$ , se pueden usar mapas autoorganizados de menor dimensión.

Normalmente, los patrones de entrenamiento son muestras aleatorias de  $RN$ . En un tiempo  $t$ , se genera una muestra  $\mathbf{x}(t)$  que se presenta a la red. Usando el algoritmo que se explica a continuación, se determina la neurona ganadora y se modifican sus pesos y los de sus vecinas.

Con este tipo de entrenamiento se consigue que, sin necesidad de conocer la salida deseada, la respuesta de la red sea similar para vectores similares en el espacio de entrada. Este tipo de aprendizaje se denomina aprendizaje no supervisado (al no requerir un supervisor externo), para distinguirlo del aprendizaje supervisado de las redes Perceptrón, ADALINE o Backpropagation, donde se realiza la comparación entre las salidas actuales y las salidas deseadas suministradas por un supervisor externo.

Algunas de las aplicaciones de este tipo de red son:

- Agrupamiento (*Clustering*): los datos de entrada deben ser agrupados en “clusters” (grupos), y el sistema de procesamiento de datos tiene que encontrar esos grupos inherentes en los datos de entrada.
- La salida del sistema debería proporcionar el grupo al que pertenece el patrón de entrada.
- Reducción de la dimensión: los datos de entrada se agrupan en un subespacio el cual tiene una dimensión más baja que la de los datos. El sistema tiene que aprender a realizar una proyección óptima, tal que la mayoría de las variaciones (diferencias) en los datos de entrada, se mantengan en los datos de salida.
- Extracción de características: el sistema tiene que extraer las características útiles de la señal de entrada.

Un ejemplo son las aplicaciones de reconocimiento de voz donde se trabaja con un número infinito de sonidos, que aunque diferentes en detalle, pueden agruparse en un número finito de categorías bien definidas. Durante el entrenamiento se presentan a la red ejemplos de varios sonidos. Repitiendo este proceso un gran número de veces y ajustando, la red se entrena para asignar toda clase de sonidos a aquellos con los que tengan un mayor parecido.

### 2.2.8.2. Un Ejemplo Gráfico

Es posible ilustrar la autoorganización de una red de Kohonen gráficamente usando una red donde el espacio de entrada tiene sólo dos neuronas y la capa de salida consta de 6 neuronas o nodos formando una red rectangular, tal y como se muestra en la Imagen 6.4.

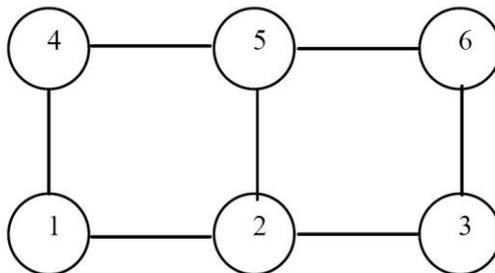


Imagen 6.4 Organización de la Capa de Salida

Otra representación posible dado que sólo hay 2 pesos, es el espacio de pesos, donde cada uno de ellos se dispone en cada uno de los ejes, siendo  $w_k$  el vector de pesos de la neurona de salida  $k$ -ésima. Inicialmente los pesos son aleatorios, pudiendo dar lugar a la Imagen 6.5.

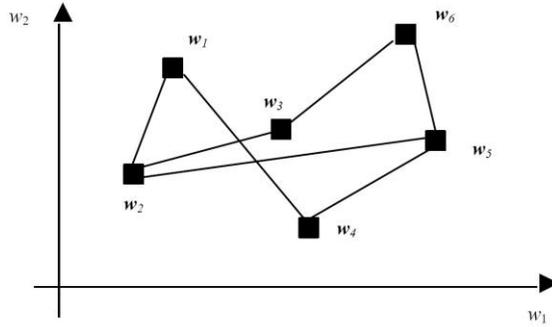


Imagen 6.6 Espacio de pesos inicial

En la Imagen 6.5, las líneas conectan nodos o neuronas que están físicamente adyacentes. Se considera que hay seis vectores de entrada representados en la Imagen 6.6.

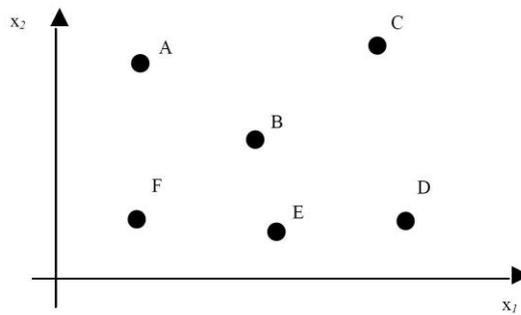


Imagen 6.6 Espacio de Entrada

En una red bien entrenada que ha desarrollado un mapa topológico, el diagrama de pesos deberá tener la misma topología que en el espacio de entrada, y reflejará las propiedades del conjunto de entrenamiento, tal y como se muestra en la Imagen 6.7.

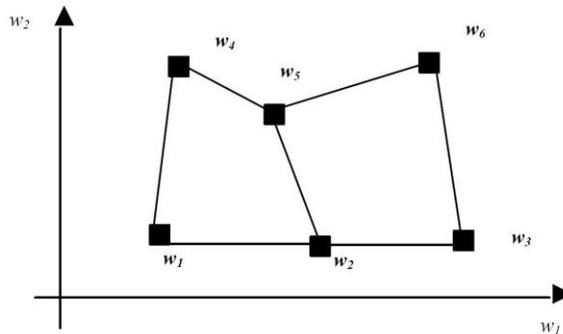


Imagen 6.7 Espacio de pesos final

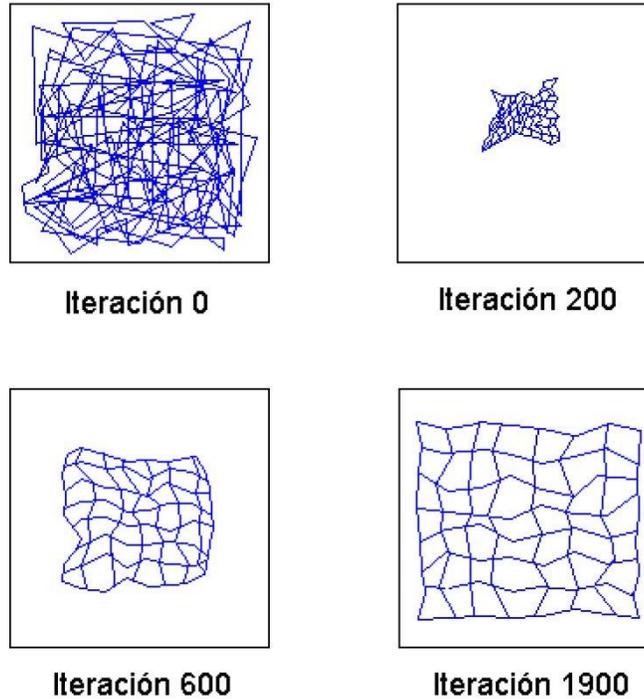


Imagen 6.8 Representación de los pesos de una red con 2 entradas y 64 Neuronas de salida [88]

Debido a este esquema de aprendizaje colectivo, las señales de entrada que están próximas unas a otras se proyectarán en neuronas vecinas. Así, la topología inherente en los vectores de entrada, se preservará en la proyección, tal como se muestra en la Imagen 6.8. Dicha Imagen, representa los vectores de peso de una red con dos entradas y  $8 \times 8$  neuronas de salida, colocadas en una matriz bidimensional. Una línea en cada figura conecta los pesos  $w_k$  con  $w_{k'}$  y  $w_{k''}$ , donde  $k = (k_1, k_2)$ ,  $k' = (k_1 + 1, k_2)$  y  $k'' = (k_1, k_2 + 1)$ . La imagen de la esquina superior izquierda muestra los pesos iniciales, mientras que la imagen inferior derecha representa la proyección casi completamente formada. En esta serie de figuras, se desarrolla una red SOM bidimensional partiendo de una serie de puntos de entrada, seleccionados aleatoriamente a partir de una distribución rectangular uniforme. En el mapa inicial, los vectores de pesos tienen valores aleatorios próximos al centro de coordenadas del mapa. A medida que empieza a evolucionar la correspondencia, los pesos se van alejando del centro como se puede observar en la figura correspondiente a la iteración 200. En la iteración 600, se observa cómo la estructura final del mapa empieza a surgir para, al final, terminar imitando la relación entre vectores de peso a la relación entre las coordenadas físicas de los elementos de proceso (iteración 1900).

Este tipo de red, puede determinar qué clase de estímulos coinciden más estrechamente, pero no es capaz de dar valores de miembros de una clase, en el mismo sentido que las redes Hopfield o BAM. Fundamentalmente, los mapas autoorganizados, deberían ser considerados como productores de una señal de salida activa-desactiva en cada una de sus neuronas. En este sentido, la

proyección es más difícil. En muchas circunstancias, esto puede ser más que compensado por su habilidad para aprender sin supervisión.

Como en la mayoría de las topologías de redes neuronales artificiales, existen muchas variantes. Por ejemplo, se pueden seleccionar las neuronas más cercanas a la señal, e interpolar entre ellas para incrementar el detalle de información devuelto (así no sólo se aprende a qué clase corresponde el estímulo, sino además, si el estímulo está cercano o alejado de la clase representada).

Cuando crece el número de clases, el número de estímulos o el número de vectores de entrada distintos que corresponden a cada clase, la cantidad de cálculos que se deben realizar en la red SOM puede llegar a ser un problema.

Una propiedad que poseen las redes SOM, es que pueden organizar con facilidad grandes cantidades de datos sin clasificar, descubriendo la estructura que subyace a ellos.

## ANEXO 3

### 3.1. Diagramas de Secuencia

Los diagramas de secuencia permiten exponer el comportamiento dinámico de cada uno de los componentes de prototipo de reconocimiento de rostros.

Se enfoca en los mensajes que van de un objeto a otro y el orden en que se fueron generando.

Los diagramas de secuencia presentados a continuación están ligados a los casos de uso expuestos en la sección 4.3 de la Monografía de la Descripción funcional de sistema.

#### 3.1.1. Análisis de Principales Componentes (PCA)

Este caso de uso fue presentado en el la sección 4.3.1 de la Mongrafía. El diagrama de secuencia de este caso de uso esta dividido en tres partes. La imagen 3.3 muestra la secuencia de análisis de principales componentes para el entrenamiento general de la Red Neuronal Artificial, la imagen 3.4 muestra la secuencia de análisis de principales componentes para un individuo en particular para un individuo en particular con imágenes capturadas desde la cámara digital, la imagen 3.5 muestra la secuencia de análisis de principales componentes para un individuo en particular con imágenes levantadas de la base de datos.

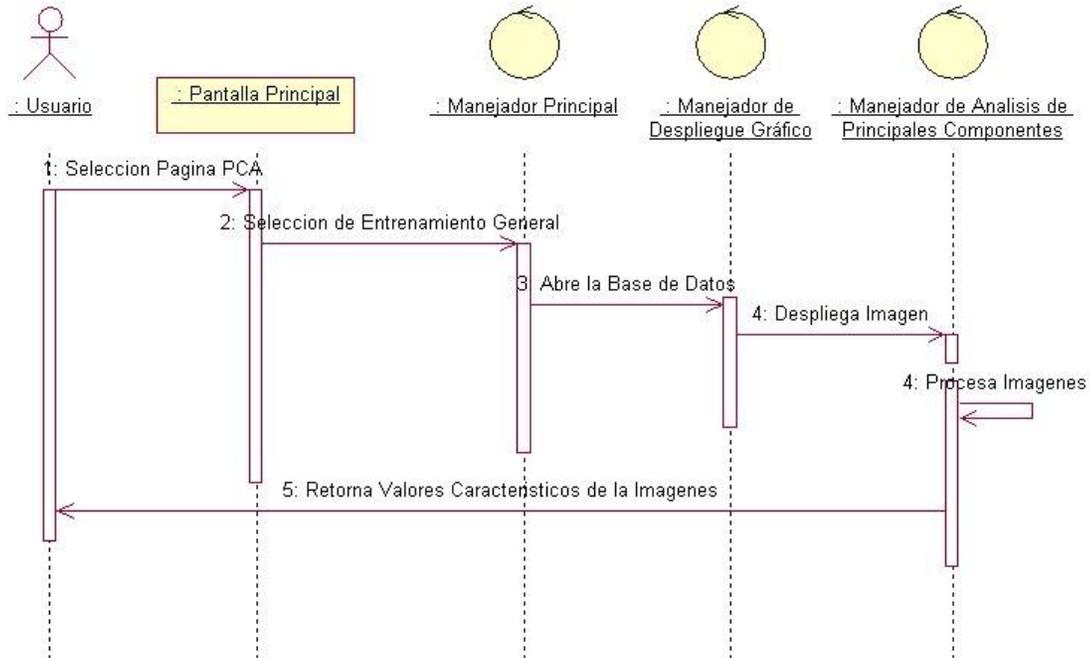


Imagen 3.3 Análisis de Principales Componentes: Entrenamiento General

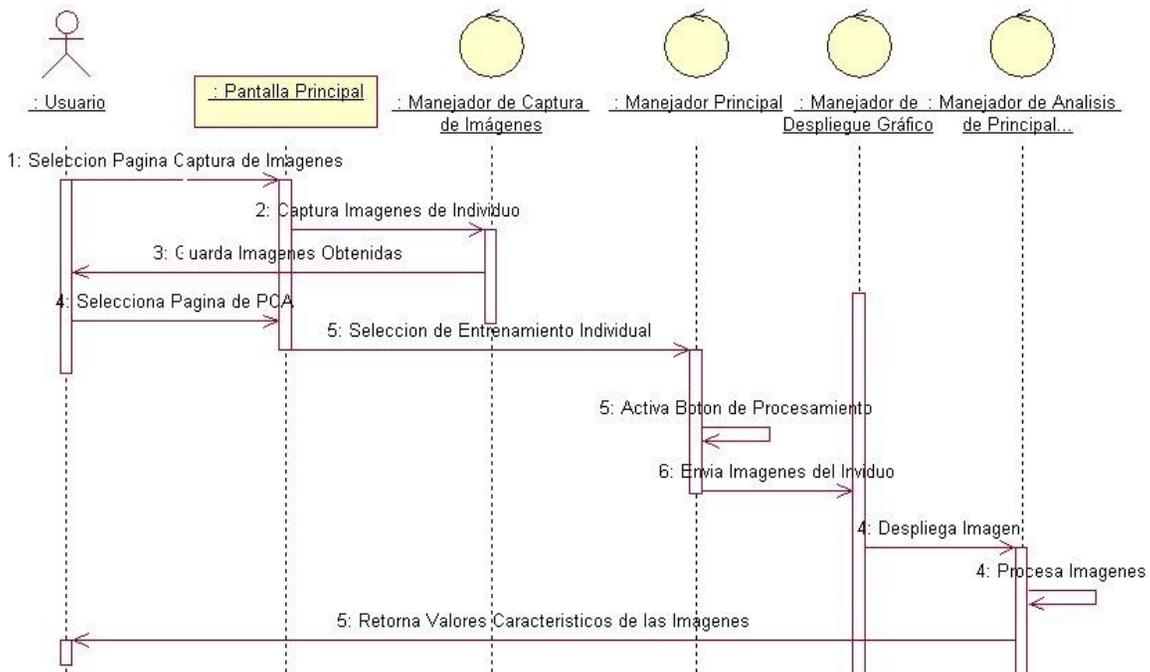


Imagen 3.4 Análisis de Principales Componentes: Entrenamiento Individual con Imágenes obtenidas desde la cámara digital.

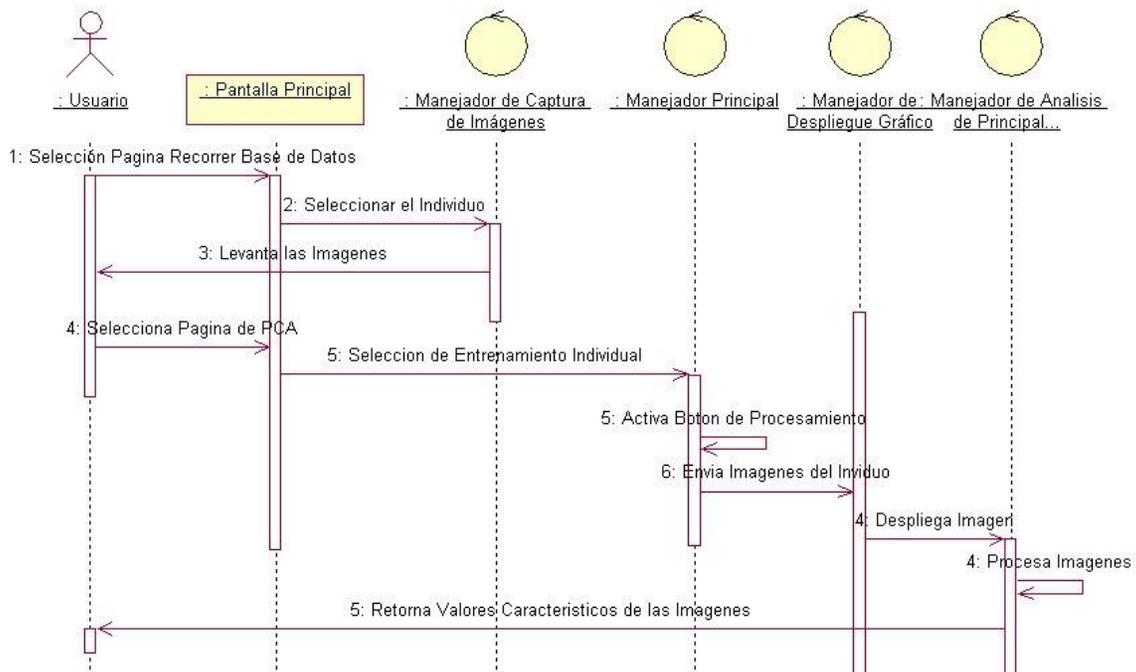


Imagen 3.5 Análisis de Principales Componentes: Entrenamiento Individual con Imágenes obtenidas desde la base de datos.

### 3.1.2. Captura de Imágenes

Este caso de uso fue presentado en la sección 4.3.2 de la monografía. El diagrama de secuencias de este caso de uso está dividido en dos partes. La imagen 3.6 muestra la secuencia de captura de imágenes para ampliar la base de datos de los rostros con los cuales se entrena la Red Neuronal Artificial. La imagen 3.7 muestra la secuencia de captura de imágenes para realizar la clasificación del rostro que en dicha imagen se encuentre.

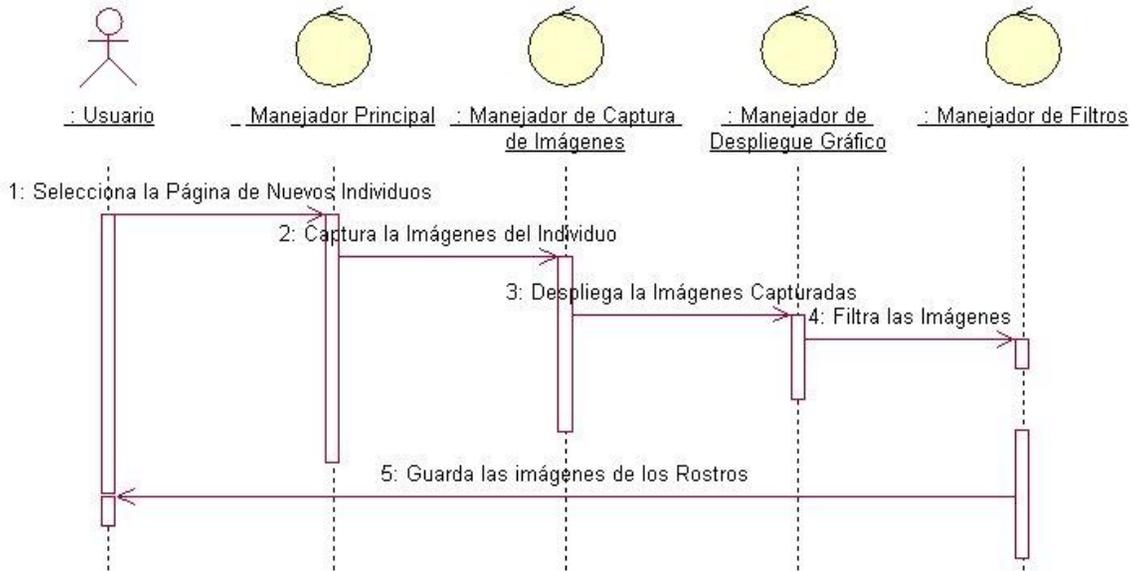


Imagen 3.6 Captura de Imágenes: Ampliación Base de Datos

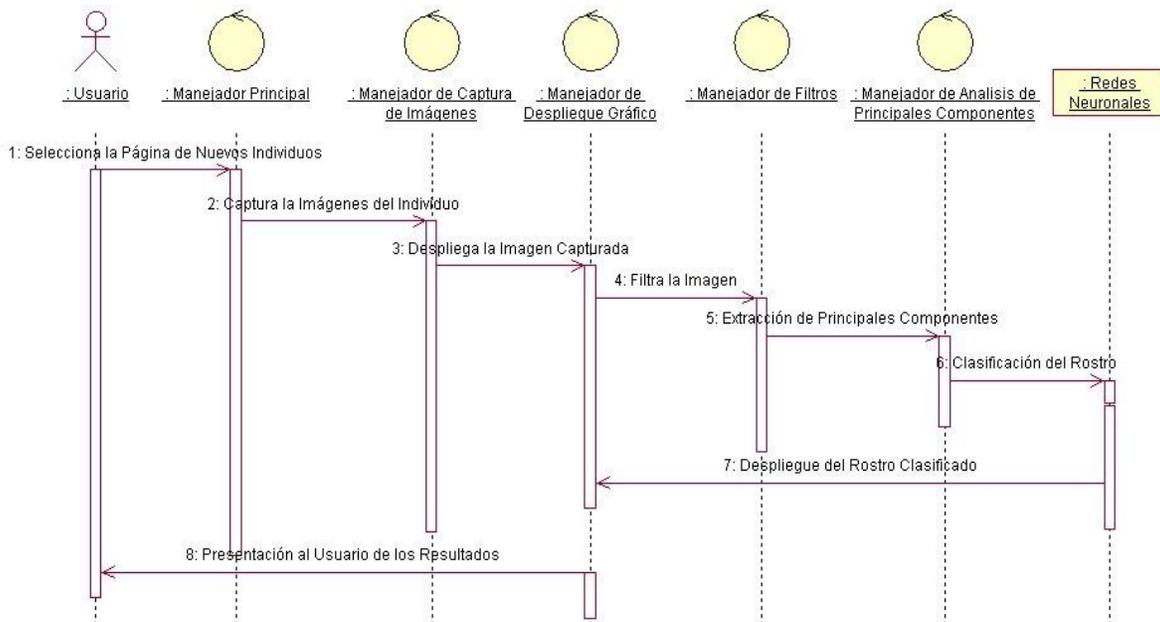


Imagen 3.7 Captura de Imágenes: Clasificación de Rostros

### 3.1.3. Identificación de Rostros

Este caso de uso fue presentado en la sección 4.3.3 de la monografía. En el diagrama de secuencias se ejemplifican los pasos necesarios para la identificación de un rostro haciendo uso de las RNA. La imagen 3.8 muestra el diagrama de secuencias para este caso de uso.

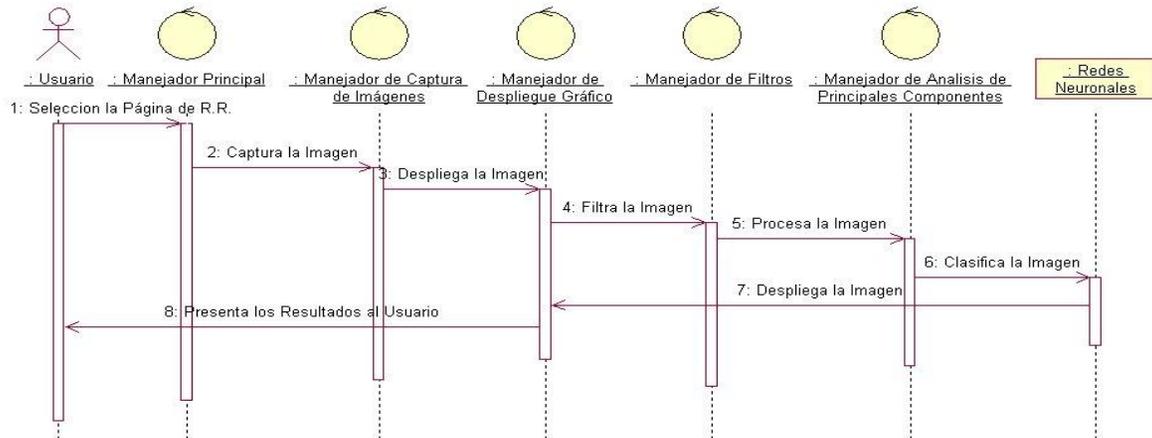


Imagen 3.8 Identificación de Rostros.

### 3.1.4. Filtros

Este caso de uso fue presentado en la sección 4.3.4 de la monografía. El diagrama de secuencias de este caso de uso esta dividido en dos partes. En la imagen 3.9 se muestra el diagrama de secuencias del filtrado de una imagen que es capturada con el fin de ampliar la base de datos. En la imagen 3.10 se presente el diagrama de secuencias del filtrado de una imagen con el fin de realizar su clasificación.

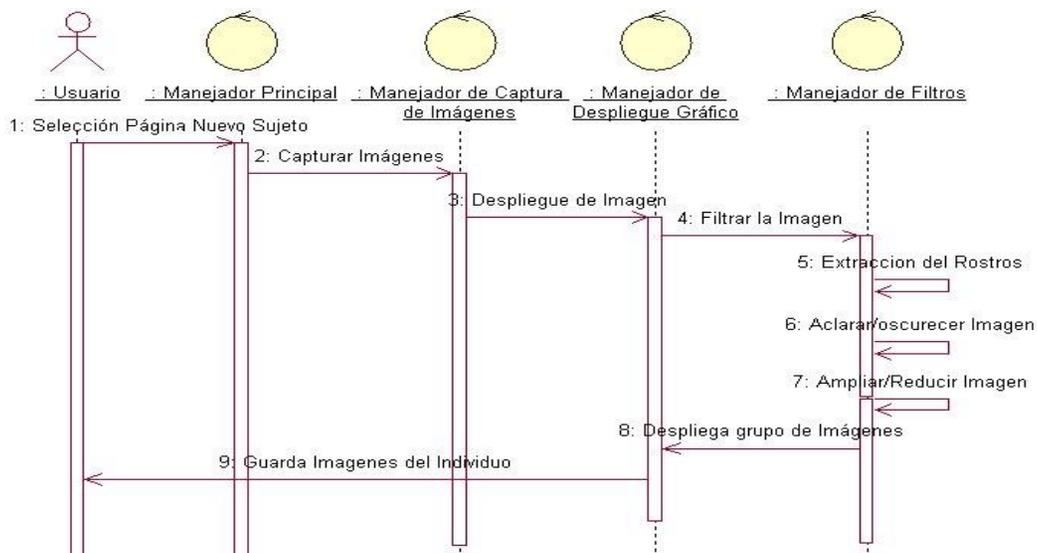


Imagen 3.9 Filtros: Ampliación de la Base de Datos

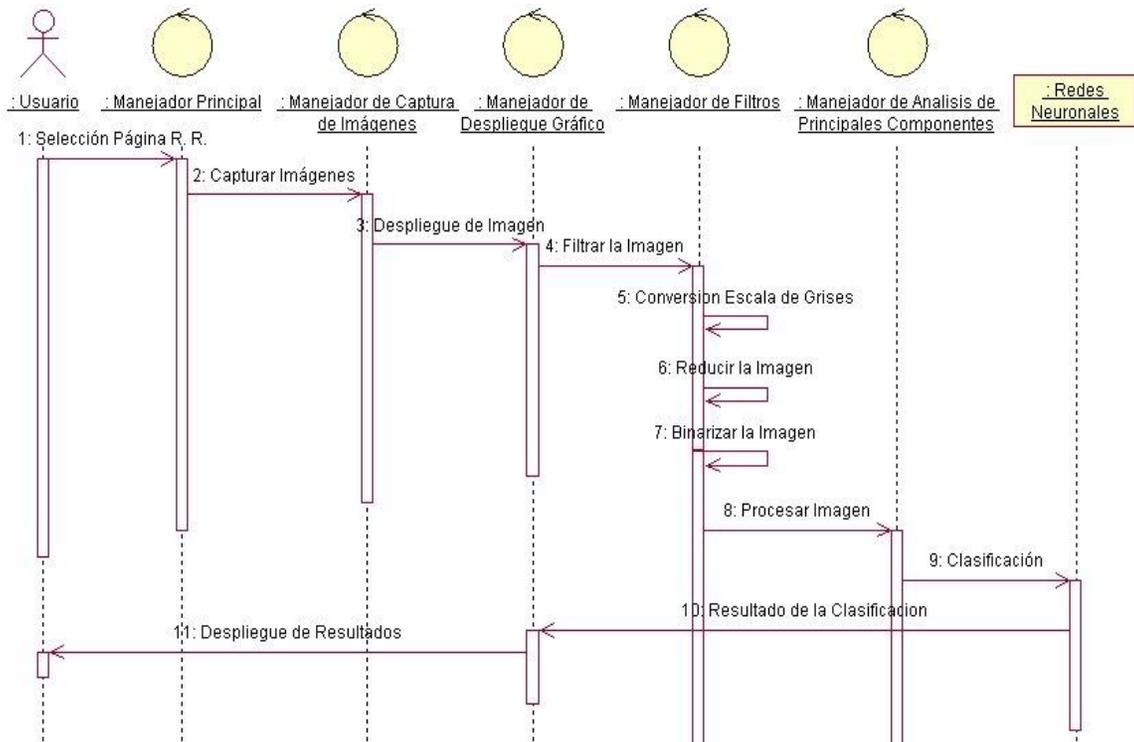


Imagen 3.10 Filtros: Clasificación de rostros

## ANEXO 4

### 4.1. Detalles de Implementación de las Redes Neuronales Artificiales

Las siguientes imágenes fueron obtenidas durante el proceso de entrenamiento de las redes neuronales, en busca de la mejor configuración para obtener un error mínimo establecido en 0.01.

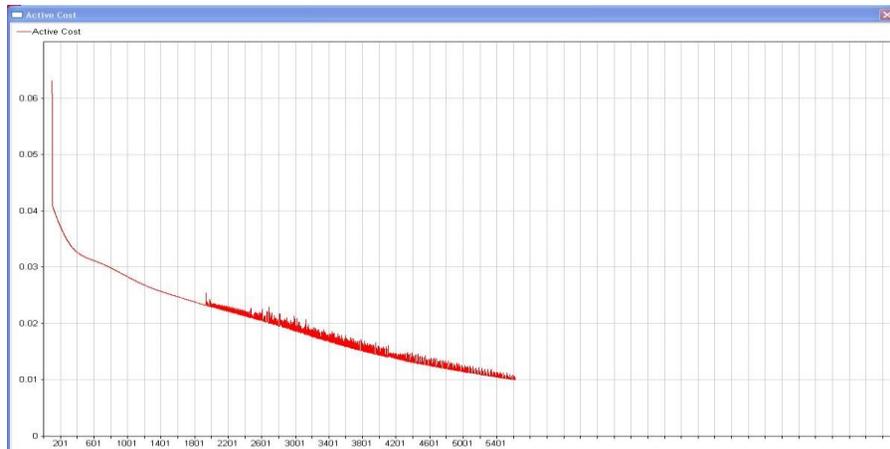


Imagen 3.1 SOM FT\_TanhAxon RL\_Momentum 10000\_Ephocs Hiden Layer 2 10 filas 10 col. (Tomada del Autor)

La imagen 4.1 presenta el comportamiento del entrenamiento de la Red neuronal de Mapas Auto organizado con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de dos capas ocultas y con veinticinco neuronas en la primera capa oculta y 75 neuronas en la segunda capa oculta, este entrenamiento se realizó con un máximo de 10.000 iteraciones.



Imagen 4.2 SOM FT\_TanhAxon RL\_Momentum 5000\_Ephocs 10 filas 10 col. (Tomada del Autor)

La imagen 4.2 presenta el comportamiento del entrenamiento de la Red neuronal de Mapas Auto organizado con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de dos capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 5.000 iteraciones.

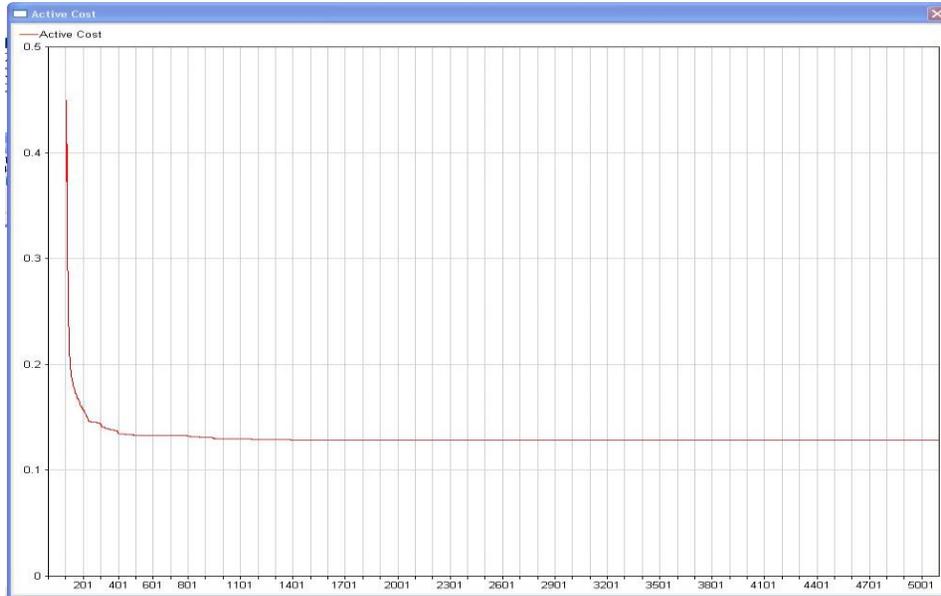


Imagen 4.3 SOM FT\_TanhAxon RL\_Momentum 5000\_Ephocs. (Tomada del Autor)

La imagen 4.3 presenta el comportamiento del entrenamiento de la Red neuronal de Mapas Auto organizado con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de tres capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 5.000 iteraciones.

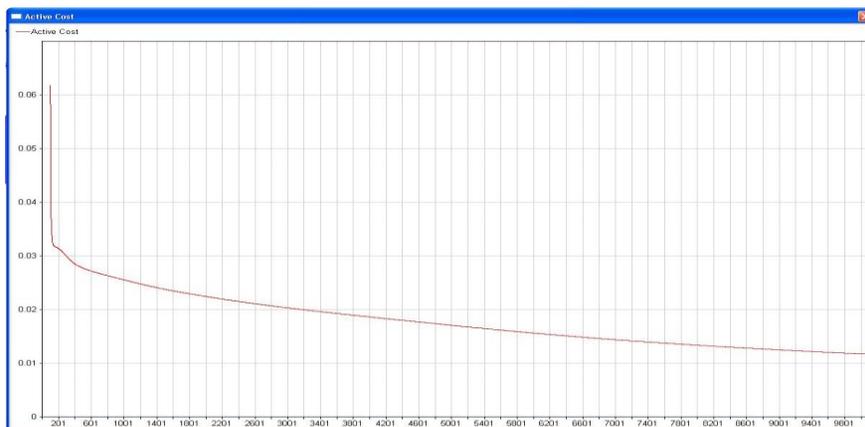


Imagen 4.4 SOM FT\_SigmoidAxon RL\_Momentum 10000\_Ephocs Hiden Layer 1 5 filas 5 col. (Tomada del Autor)

La imagen 4.4 presenta el comportamiento del entrenamiento de la Red neuronal de Mapas Auto organizado con una función de transferencia Sinusoidal, con una regla de entrenamiento *Momentum*, con una arquitectura de una capa oculta y con 50 neuronas en la capa oculta, este entrenamiento se realizó con un máximo de 10.000 iteraciones.

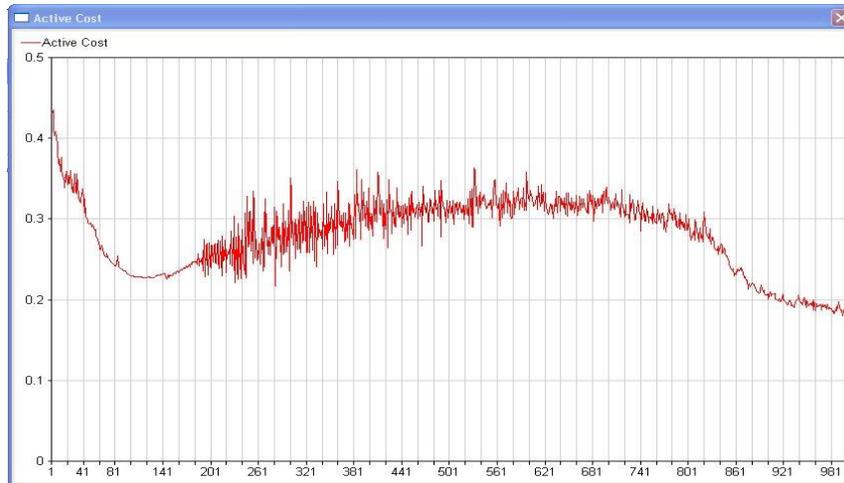


Imagen 4.5 RedNeuronal\_Perceptron Multicapa 5 Layers FT\_TanhAxon RL\_DeltaBarDelta 1000\_Ephocs. (Tomada del Autor)

La imagen 4.5 presenta el comportamiento del entrenamiento de la Red neuronal de Perceptrón Multicapa con una función de transferencia Tangente, con una regla de entrenamiento Delta, con una arquitectura de dos capas ocultas y con 100 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

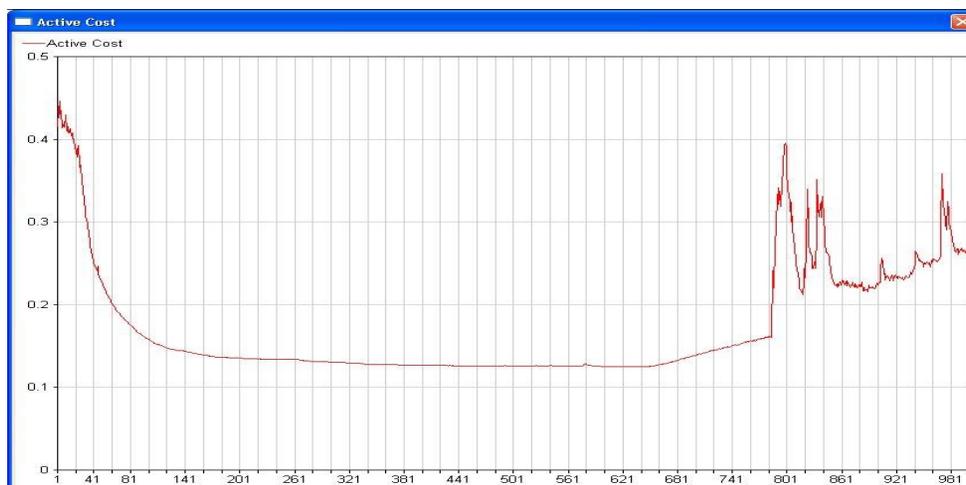


Imagen 4.6 RedNeuronal\_Perceptron Multicapa 3 Layers FT\_TanhAxon RL\_DeltaBarDelta 1000\_Ephocs. (Tomada del Autor)

La imagen 4.6 presenta el comportamiento del entrenamiento de la Red neuronal de Perceptrón multicapa con una función de transferencia Tangente, con una regla de entrenamiento Delta, con una arquitectura de tres capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

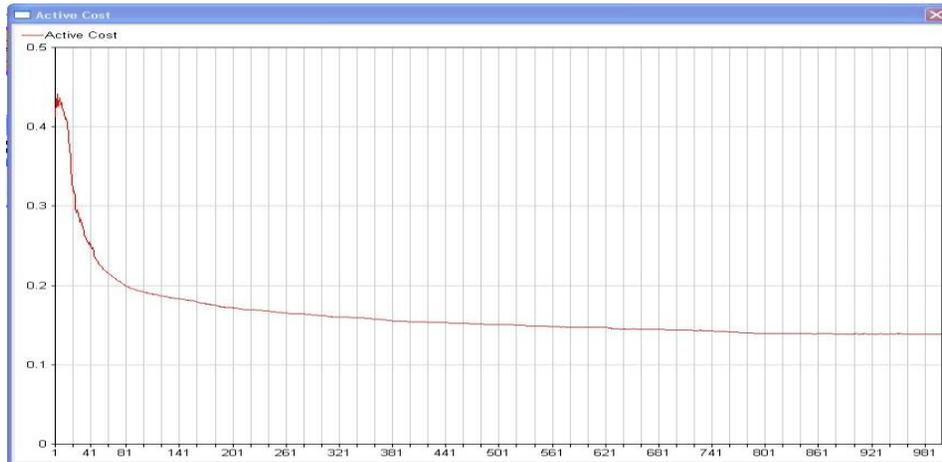


Imagen 4.7 RedNeuronal\_Perceptron Multicapa 2 Layers FT\_TanhAxonRL\_QuickProp 1000\_Ephocs. (Tomada del Autor)

La imagen 4.7 presenta el comportamiento del entrenamiento de la Red neuronal de perceptrón multicapa con una función de transferencia Tangente, con una regla de entrenamiento *QuickProp*, con una arquitectura de dos capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.



Imagen 4.8 RedNeuronal\_Perceptron Multicapa 2 Layers FT\_TanhAxon and LinealAxon RL\_Momentum 1000\_Ephocs. (Tomada del Autor)

La imagen 4.8 presenta el comportamiento del entrenamiento de la Red neuronal de perceptrón multicapa con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de dos capas ocultas y con 50 neuronas en la primera capa oculta y 100 neuronas en la segunda capa oculta, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

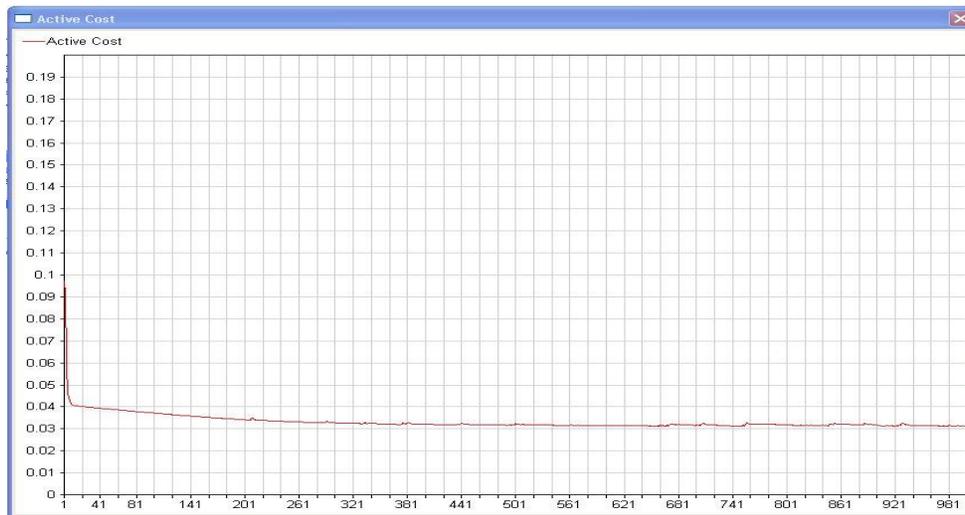


Imagen 4.9 RedNeuronal\_Perceptron Multicapa 2 Layers FT\_TanhAxon and LinealAxon RL\_DeltaBarDelta 1000\_Ephocs. (Tomada del Autor)

La imagen 4.9 presenta el comportamiento del entrenamiento de la Red neuronal de perceptrón multicapa con una función de transferencia Tangente, con una regla de entrenamiento *Delta*, con una arquitectura de dos capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

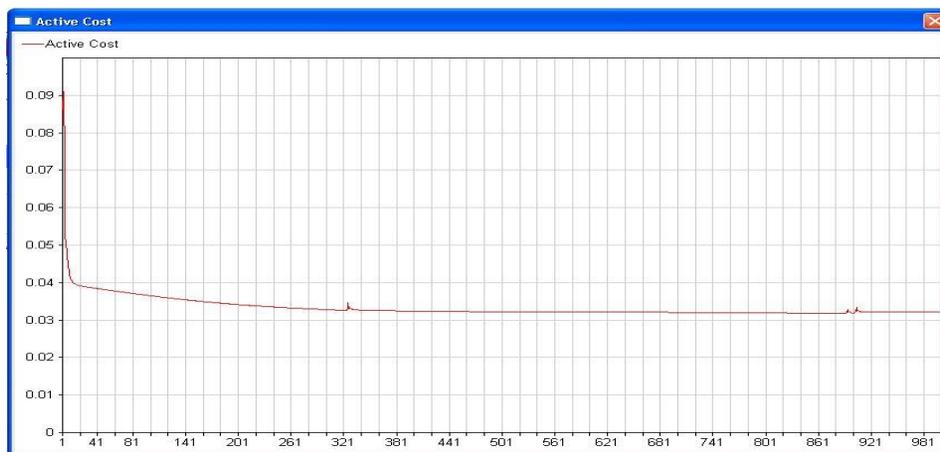


Imagen 4.10 RedNeuronal\_Perceptron Multicapa 2 Layers FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs. (Tomada del Autor)

La imagen 4.10 presenta el comportamiento del entrenamiento de la Red neuronal de perceptrón multicapa con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de dos capas ocultas y con 100 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

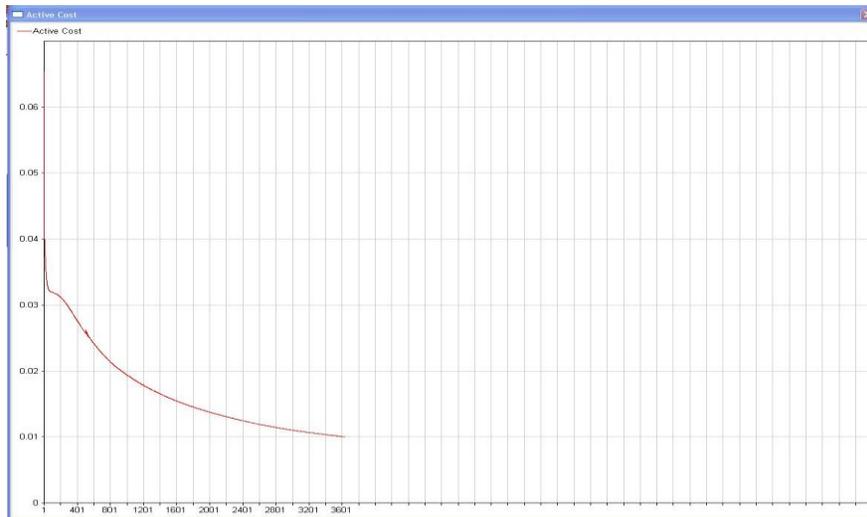


Imagen 4.11 RedNeuronal\_Perceptron Multicapa 1 Layers FT\_SigmoidAxon RL\_Momentum 10000\_Ephocs. (Tomada del Autor)

La imagen 4.11 presenta el comportamiento del entrenamiento de la Red neuronal de perceptrón multicapa con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de una capa ocultas con 50 neuronas en la capa oculta, este entrenamiento se realizó con un máximo de 10.000 iteraciones.

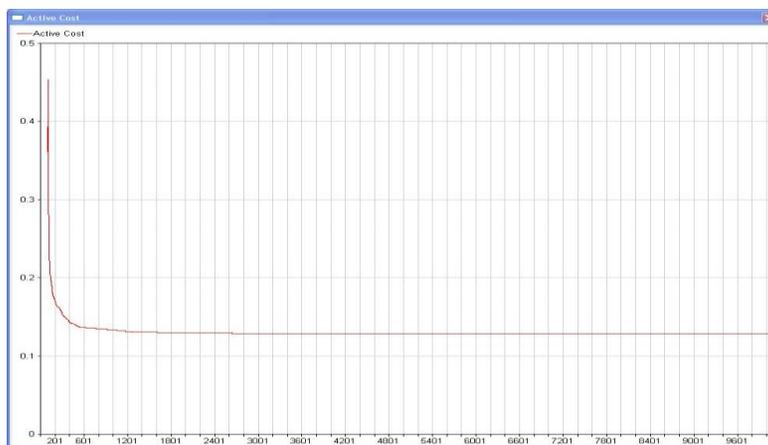


Imagen 4.12 RBF FT\_TanhAxon RL\_Momentum 10000\_Ephocs Hiden Layer 2. (Tomada del Autor)

La imagen 4.12 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de dos capas ocultas y con 100 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 10.000 iteraciones.

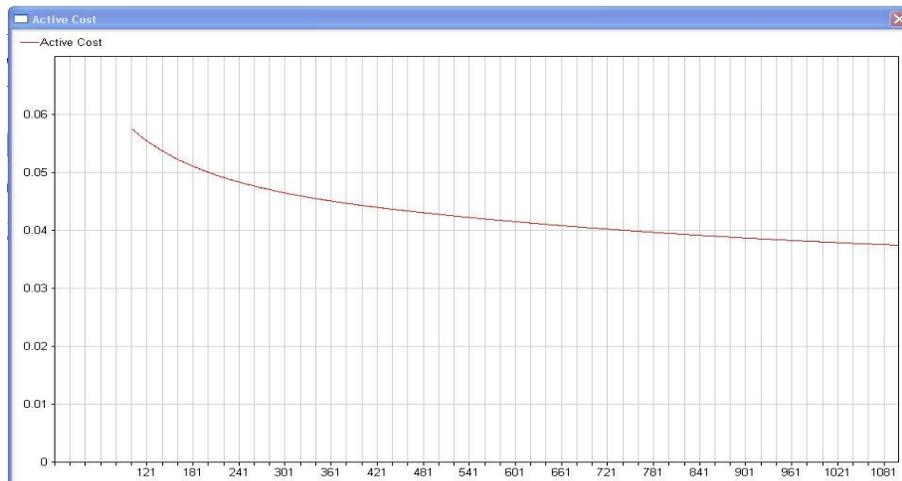


Imagen 4.13 RBF FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs Hiden Layer 5.  
(Tomada del Autor)

La imagen 4.13 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Sigmoidea, con una regla de entrenamiento *momentum*, con una arquitectura de cinco capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.



Imagen 4.14 RBF FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs Hiden Layer 4.  
(Tomada del Autor)

La imagen 4.14 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de cuatro capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

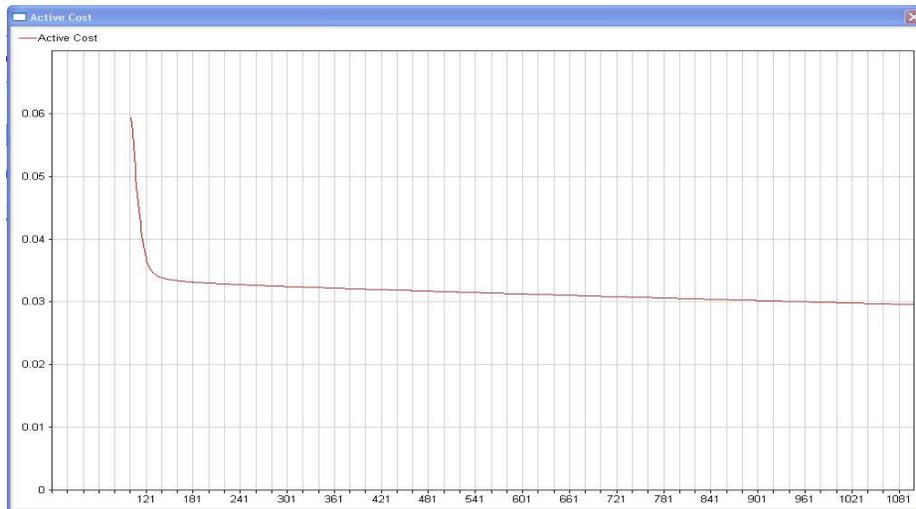


Imagen 4.15 RBF FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs Hiden Layer 3  
Regla de competitividad Standar Full. (Tomada del Autor)

La imagen 4.15 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de tres capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

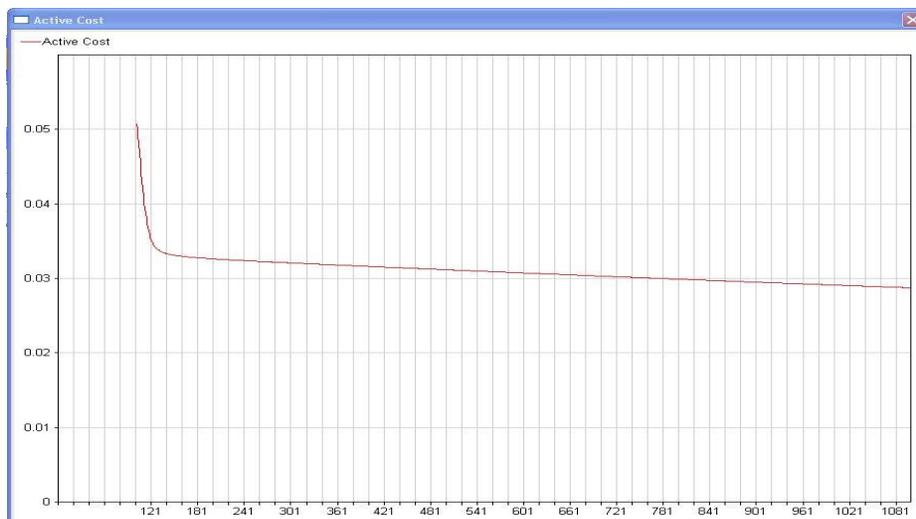


Imagen 4.16 RBF FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs Hiden Layer 3.  
(Tomada del Autor)

La imagen 4.16 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de tres capas ocultas y con 100 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.



Imagen 4.17 RBF FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs Hiden Layer 1.  
(Tomada del Autor)

La imagen 4.17 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de una capa oculta y con 50 neuronas en la capa oculta, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

La imagen 4.18 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura de dos capas ocultas y con 50 neuronas en la cada una de las capas ocultas, este entrenamiento se realizó con un máximo de 1.000 iteraciones.



Imagen 4.18 RBF FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs Hiden Layer 2.  
(Tomada del Autor)

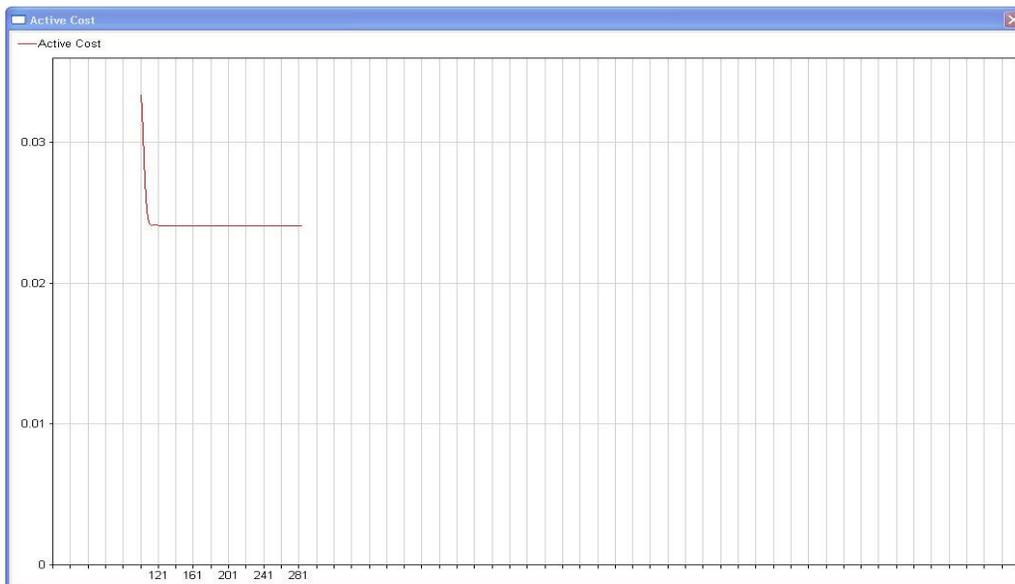


Imagen 4.19 RBF FT\_SigmoidAxon RL\_Momentum 1000\_Ephocs Hiden Layer 0.  
(Tomada del Autor)

La imagen 4.19 presenta el comportamiento del entrenamiento de la Red neuronal de funciones de base radial con una función de transferencia Tangente, con una regla de entrenamiento *momentum*, con una arquitectura sin capa oculta, este entrenamiento se realizó con un máximo de 1.000 iteraciones.

## ANEXO 5

### CASOS DE USO REALES

Los casos de uso reales son el fiel reflejo de los casos de uso propuestos en el diseño de la aplicación.

#### 5.1. Caso de Uso Análisis de Principales Componentes (PCA)



Imagen 5.1 Interfaz para Análisis de Principales Componentes para un Entrenamiento Individual de la RNA. (Tomada del Autor)

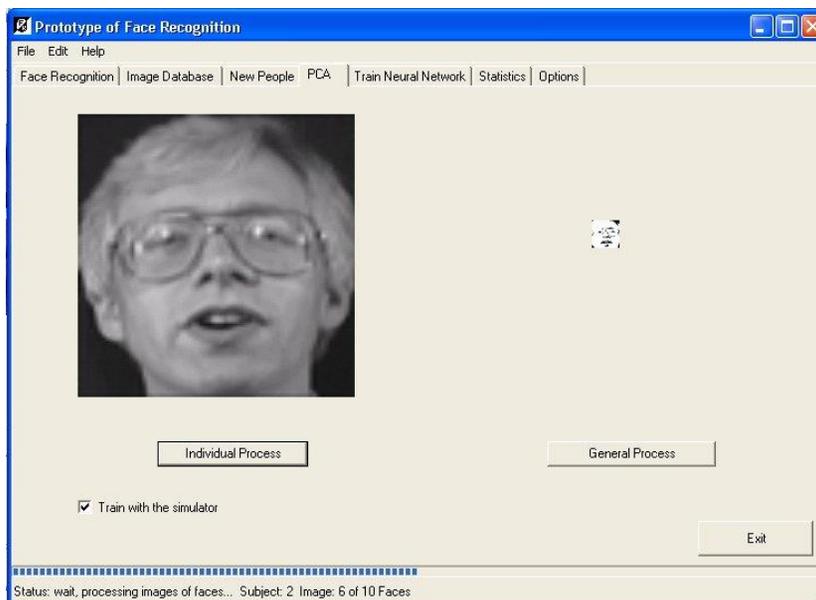


Imagen 5.2 Interfaz de Análisis de Principales Componentes para un Entrenamiento Individual de la RNA Iniciado. (Tomada del Autor)



Imagen 5.3 Interfaz para Análisis de Principales Componentes para un Entrenamiento General de la RNA

## 5.2. Caso de Uso Captura de Imágenes

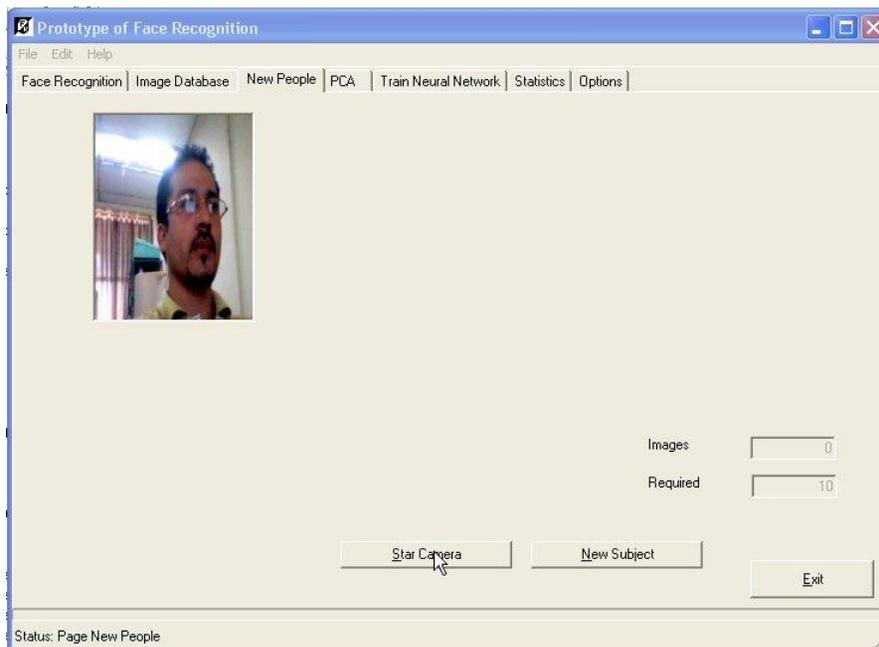


Imagen 5.4 Interfaz para captura de imágenes para ampliar la base de datos de entrenamiento

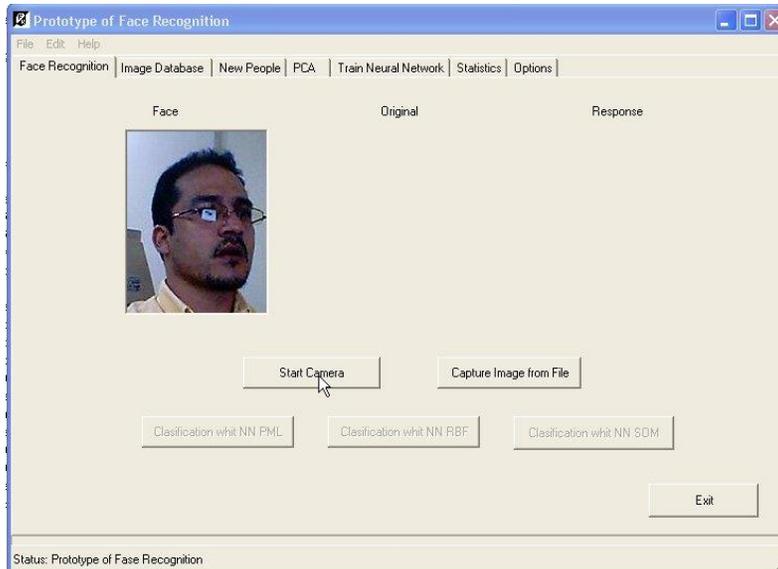


Imagen 5.5 Interfaz para captura de imágenes desde la cámara para la clasificación de rostros

### 5.3. Caso de Uso Identificación de Rostros



Imagen 5.6 Interfaz para captura de imágenes desde archivos para la clasificación de rostros

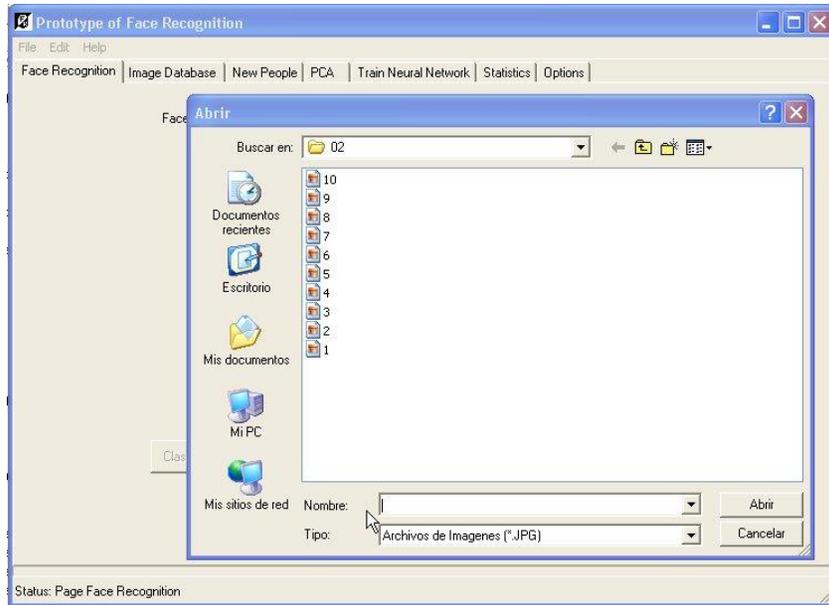


Imagen 5.7 Interfaz para abrir una archivo de imagen

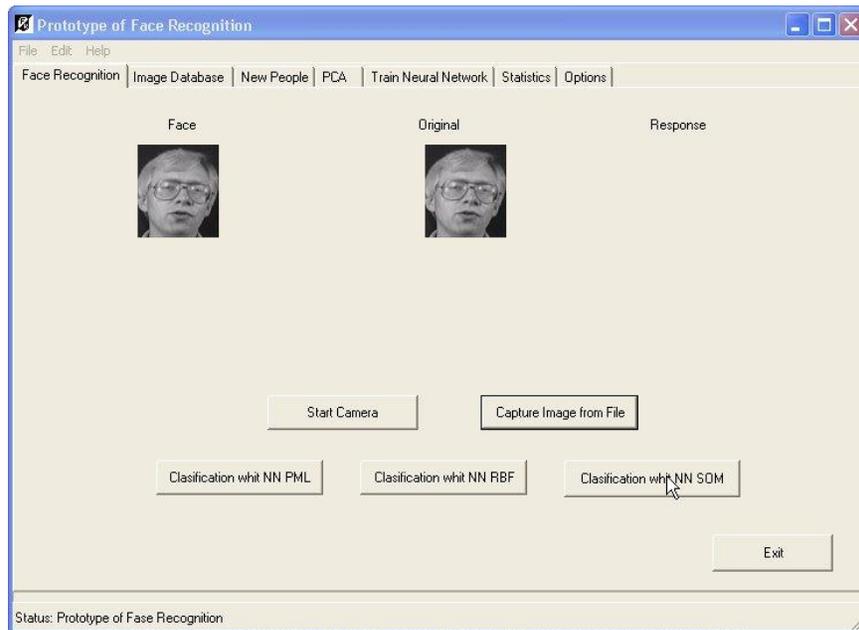


Imagen 5.8 Interfaz para clasificación de imágenes con las redes neuronales

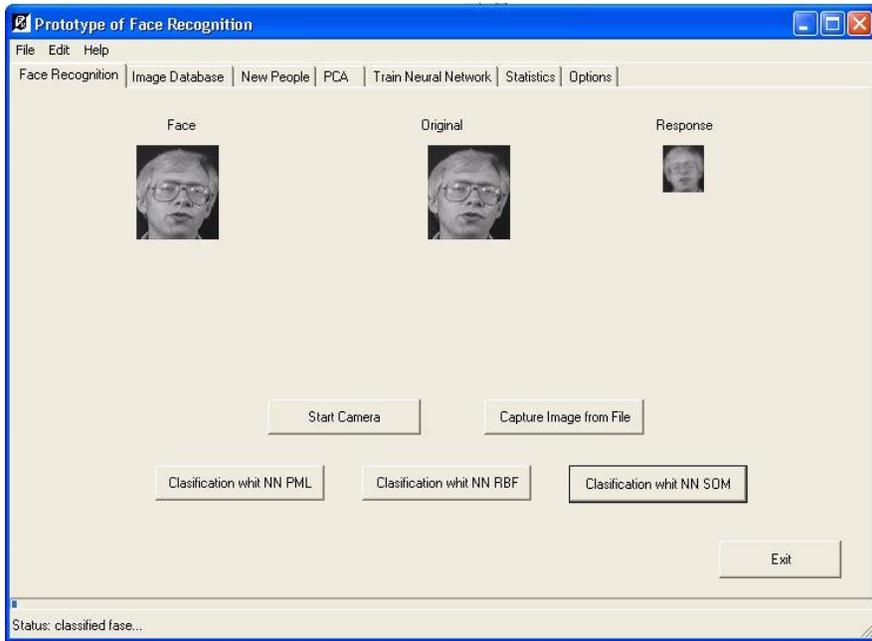


Imagen 5.9 Interfaz con la respuesta de clasificación del rostro