UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

"ANÁLISIS DE PATRONES DE EXPRESIÓN DE GENES UTILIZANDO REDES NEURONALES (por clustering)"

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRA EN CIENCIAS (COMPUTACIÓN)

P R E S E N T A:

MARÍA DE CARMEN EDNA MÁRQUEZ MÁRQUEZ

DIRECTOR DE TESIS:
DR. PEDRO PABLO GONZÁLEZ PÉREZ

MÉXICO D. F., 2004

Dedicada especialmente a mis Padres y Hermanos
quienes siempre han estado a mi lado para brindarme su amor
y por su apoyo para que siga adelante.

Gracias a todos los que colaboraron con sus conocimientos e ideas
en la realización de este proyecto de investigación.

Índice general

RESUMEN

El interés despertado por el estudio del genoma de diversas especies y del propio humano a partir de la década de los 90s se ve reflejado en los grandes bancos de información biológica con que se cuenta, tanto de dominio público como privado. Es ahora cuando se requiere el apoyo de las herramientas de la información que permitan extraer la mayor cantidad de conocimiento, que aplicado puede redundar en grandes beneficios para los seres vivos. Hasta ahora ha quedado al descubierto mucha información de diversos genomas, entre ellos el humano y el siguiente paso es explotar esa cantidad de información para lograr el conocimiento que allí se encuentra. Es así como una de las tareas de las ciencias de la computación es crear la tecnología de la información necesaria para el procesamiento de bancos de datos biológicos, y específicamente la computación inteligente enmarcada en la Inteligencia Artificial han dado importantes logros en este campo, como sucede con las redes neuronales artificiales.

Este trabajo se encuentra enmarcado en la nueva disciplina que utiliza las herramientas computacionales para el procesamiento de información biológica con el fin de extraer nuevos conocimientos con aplicaciones prácticas importantes, la Bioinformática. Para las ciencias biológicas, en especial para la medicina y la farmacéutica, es de gran importancia la identificación de los genes que se expresan en determinadas circunstancias, ya que la expresión de éstos, principalmente en grupos, interfiere directamente con el desarrollo de las enfermedades. Los datos correspondientes a miles de genes de un organismo pueden someterse a diversos experimentos conjuntamente mediante el uso de microarreglos, lo que vierte como resultado una matriz de miles de datos numéricos cuyo análisis y procesamiento requiere sistemas de computación no triviales.

El objetivo de la presente tesis de maestría es proporcionar una herramienta inteligente para colaborar en el análisis de la expresión de genes de cualquier especie y encontrar similitudes entre grupos de genes. La herramienta inteligente está basada en unidades de procesamiento distribuidas en tres capas interconectadas que evoluciona para realizar la tarea de agrupación de patrones de expresión de genes; con este modelo de red neuronal artificial pretendemos colaborar en el análisis de expresión de genes al encontrar similitudes entre grupos de genes. Esta agrupación de genes está enmarcada en lo que se denomina clustering supervisado y no supervisado, de acuerdo con el conocimiento o no de clases a que corresponden los grupos o clusters a que pertenecen los genes.

Una de las aplicaciones principales del análisis de expresión de genes se encuentra en la medicina, involucra la obtención de datos de expresión de genes de células afectadas por diferentes enfermedades, los cuales son comparados contra niveles de expresión normales provenientes de células sanas. La identificación de los genes que se expresan de modo diferente en estos casos otorga una base del perfil genético del individuo para la explicación de causas de enfermedades y permite establecer el blanco de acción de los medicamentos.

Esta tesis se encuentra organizada en 5 capítulos, de los cuales el primero, así como la introducción corresponden al marco teórico y estado del arte de la bioinformática aplicada para la agrupación de expresión de genes, y de las redes neuronales artificiales como parte de la inteligencia artificial. Los capítulos dos y tres presentan el modelo de la red neuronal dinámica que hemos desarrollado. En el segundo se hace un agrupamiento supervisado, en el cual se conocen las clases de los patrones de expresión de genes presentados a la red, y en el tercero se realiza un agrupamiento no supervisado, en el que se desconocen las clases a las que pertenecen los patrones de expresión de genes. El capítulo cuatro presenta el modelo de tecnología orientado a objetos que se utilizó para desarrollar la red neuronal con sus diversas clases de objetos y las distintas relaciones establecidas entre estas clases. Y en el capítulo cinco se muestran algunos experimentos realizados con la implementación de la herramienta; utilizando diversas bases de genes aquí se presentan también algunos aspectos de la interfaz del sistema con el usuario. Finalmente presentamos las conclusiones en las que revisamos los resultados obtenidos.

Con los experimentos que se realizarán como parte final de nuestro trabajo se pretende probar que el modelo de red neuronal propuesto es una buena alternativa para el análisis de expresión de genes, permitiendo utilizar datos binarios, ternarios o en números reales, así como diferentes medidas de distancia para generar los clusters y, además, permitir al usuario jugar con los parámetros de la red para que pueda seleccionar aquellos que le proporcionen mejores resultados.

Es importante aclarar que, dentro del documento se utilizará de manera indistinta el término cluster y cúmulo para referirnos al grupo de centroides, con los que se representará la clase a la que pertenece el patrón de expresión genética.

INTRODUCCIÓN

Como resultado de la principal tarea dada a la biología molecular, que es la búsqueda de las secuencias de los diversos genomas, la cantidad de información emergente ha generado una estrecha relación entre las ciencias genómicas y las ciencias de la computación, dando origen a la bioinformática. Mientras más datos experimentales se tienen, más útiles se vuelven los modelos y métodos computacionales. Utilizando el código genético y los métodos computacionales, la secuencia de todas y cada una de las proteínas se puede producir a partir de un genoma particular, lo que daría una relación de todos los genes involucrados en enfermedades genéticas y de las proteínas que pueden ser blanco de fármacos que curen enfermedades específicas.

El enorme tamaño del genoma de los organismos superiores, más de 20000 genes y en el hombre entre 30000 y 40000 genes [13], hace su análisis extraordinariamente difícil. Actualmente no se tiene idea de la función de más del 40% de los genes identificados en el genoma humano, lo que significa que se desconoce para qué sirven alrededor de 12000 proteínas que se sintetizan diariamente en nuestras células [26]. A muchos genes sólo se les puede asignar una función de manera indirecta, al comparar su secuencia y encontrar semejanzas con la de otros genes de otros organismos que ya se conocen. La liberalización del genoma completo del *Homo sapiens* ha dado paso a una fase post-genómica, donde la tarea consiste principalmente en el análisis de la información.

Una de las primeras tareas de la nueva disciplina denominada bioinformática es un mejor manejo de las secuencias de ADN y la localización de genes. Un subcampo importante de la bioinformática, el cual llegará a ser más y más relevante en la medida en que más genomas hayan sido secuenciados, es la genómica comparativa donde la información de los diferentes genomas es relacionada para obtener más información acerca de la funcionalidad de los genes (genómica funcional).

1. La genómica

A partir del estudio del genoma han surgido nuevas disciplinas entre las que encontramos la genómica y la bioinformática, las cuales cubren un amplio rango de materias de biología, medicina, matemáticas, ciencias de la computación, teoría de la información y física. El estudio del genoma está relacionado con la generación, manipulación, análisis e interpretación de una amplia gama de datos biológicos, por ejemplo secuencias de datos de ADN o análisis global del contenido total de lo transcrito (transcriptoma) o de proteínas (proteoma).

El conocimiento del genoma es uno de los acontecimientos más relevantes del siglo XX y que sin duda revolucionará la ciencia médica del siglo XXI. Así, los conocimientos que de ello se deriven serán de gran trascendencia para la vida del hombre, principalmente si se trata del genoma humano, puesto que la información genómica aporta más conocimientos del ser humano.

El genoma de todo organismo está constituido por ácido desoxirribonucleico (ADN), una molécula con forma de doble hélice formada por subunidades denominadas nucleótidos. Cada nucleótido se compone de bases, sustancias que contienen nitrógeno, ligadas a un azúcar y un grupo fosfato. Las bases que describen las instrucciones para ensamblar las proteínas son: A-adenina, G-guanina, C-citocina y T-timina. El orden en que se encuentran dichos nucleótidos y su frecuencia de aparición da como resultado el lenguaje genético. Determinar ese orden es lo que se conoce como secuencia del genoma.

Aunque el genoma de cada individuo es único, se puede hacer una secuencia estereotípica, una generalización o idealización del genoma de los organismos de la misma especie. Por lo anterior tenemos que el genoma en los humanos no es exactamente el mismo en todos, se estima una similitud del 99% entre dos individuos al azar. Con el conocimiento del genoma humano, en un futuro será posible elaborar y utilizar fármacos personalizados, que serán mucho más eficientes que los actuales ya que se diseñarán para cada enfermo con base en el conocimiento preciso de las características fisiológicas que determinen una sintomatología particular (farmacogenética o farmacogenómica). Entre los candidatos para este nuevo diseño de fármacos se encuentran aquellos para desórdenes de la conducta y esquizofrenia, los que sirven para combatir enfermedades alérgicas y los que se contrapropagan a la actividad de numerosas enzimas que intervienen en el desarrollo de varios padecimientos [26].

Actualmente se encuentra en pleno desarrollo la caracterización de genes involucrados en varias enfermedades. Virtualmente, si todos los genes tuvieran una actividad diferente a la normal podrían provocar una enfermedad. Además, el diagnóstico molecular podría extenderse en su aplicación a otras esferas de la naturaleza humana, como el talento o la inteligencia, así como la inclinación hacia un comportamiento en especial. El auge de las ciencias del genoma también está proporcionando el conocimiento del acervo genético de diversos microorganismos en beneficio para los seres vivos, y con este nuevo bagaje científico será posible ahondar en el origen de las enfermedades infecciosas.

2. La expresión genética

Todas las vías metabólicas, la expresión de los anticuerpos, la actividad de los virus, todas las manifestaciones del fenómeno viviente, están, en última instancia, orquestadas por la expresión ordenada y regulada de los genes, por medio de la producción de proteínas específicas [32].

En un organismo pluricelular, cada tipo de célula tiene una forma característica, realiza actividades muy específicas y produce un conjunto de proteínas distinto. Aún así, con pocas excepciones, todas las células contienen la misma información genética. Los genes son regulados, y sólo determinados subconjuntos de la información genética total se expresan en cualquier célula dada, esto trae como resultado que no todas las células sean idénticas en composición y en estructura.

La expresión genética es resultado de una serie de procesos, cada uno de los cuales puede ser controlado en diversas formas. Algunas de las principales estrategias utilizadas para regular la expresión genética comprende el control de: 1) la cantidad de mARN disponible, 2) la rapidez de traducción del mARN, y 3) la actividad de la proteína producida.

Todo organismo, aun el más simple, contiene una enorme cantidad de información en la forma de ADN. En cada célula, el ADN se organiza en unidades llamadas genes, que en última instancia controlan todos los aspectos de la vida del organismo [32]. La expresión de los genes consiste en una compleja serie de procesos, entre los que se incluye la síntesis de moléculas de ARN complementarias al ADN (transcripción), así como la síntesis de proteínas (traducción). A esto último se le denomina el "dogma de la biología molecular". La idea de este dogma se esquematiza en la figura 1.

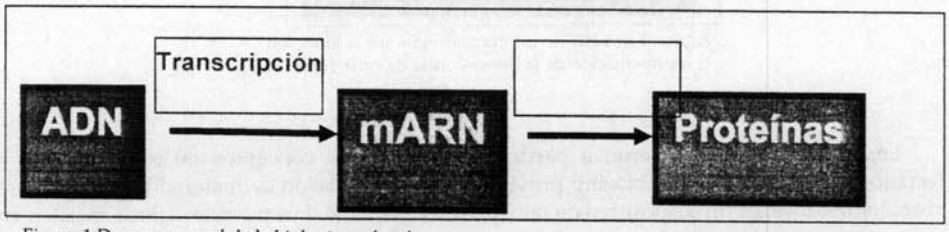


Figura 1 Dogma general de la biología molecular.

Los datos correspondientes a la expresión de genes pueden provenir de técnicas como los Microarreglos[1], los ESTs[2] [1] y SAGE[3] [17]. El SAGE trabaja sobre una secuencia de genes, es un método cuantitativo para la comprensión del análisis de patrones de expresión de genes, permite identificar nuevos genes a partir del conocimiento de otros; el EST se trata de pequeñas partes de regiones activas del gene y se utiliza para localizar el resto de un gene fuera del cromosoma. Sin embargo, nosotros hacemos incapie en los microarreglos, pues a partir de su aparición ha crecido exponencialmente el interés para realizar análisis de la expresión genética por su capacidad para presentar datos confiables de más de 6000 genes a la vez y por su base tecnológica [38]. Los microarreglos constituyen la técnica post-genómica más desarrollada y exitosa, permiten la búsqueda de genes involucrados en enfermedades de origen genético. Proporcionan la expresión de hasta 10000 genes a la vez, en la dimensión de un porta-objetos, en la figura 2 puede verse un ejemplo de microarreglo.

3. Los microarreglos

Los microarreglos o biochips [25] representan una de las herramientas más importantes en el ámbito de la investigación genética y genómica para la obtención de información. Los microarreglos corresponden a una nueva tecnología que contiene conjuntos ordenados de moléculas de ADN de una estructura conocida y que corresponden a los niveles de expresión de miles de genes. A través del análisis de microarreglos es posible determinar la expresión de múltiples genes en forma simultánea. Un conjunto de datos de microarreglo típico incluye los niveles de expresión de miles de genes en cientos de condiciones, representadas en experimentos. Estas condiciones pueden ser debidas a: 1) la fisiología celular interna desde líneas de tejidos diferentes, 2) diversas condiciones fisiológicas en un organismo intacto, o 3) tejidos patológicos diferentes [23] .

[1] Microarrays o microchips
[2] Expressed Sequence Tag
[3] Serial Analysis of Gene Expression

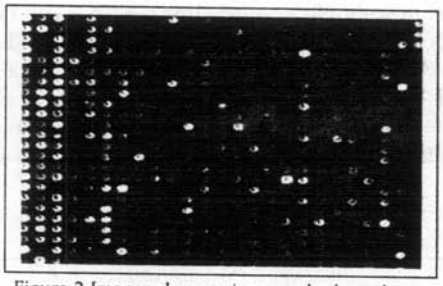


Figura 2 Imagen de un microarreglo, los colores son la representación de la fluorescencia de cada gene[4].

Los microarreglos surgieron a partir de la mezcla de conceptos microelectrónicos y biotecnológicos pues el término biochip proviene de la combinación de material biológico sobre un chip, lográndose así un dispositivo de tamaño muy pequeño. Los microarreglos consisten en la inmovilización del material biológico[5] sobre una superficie sólida en la que alcanza una elevada densidad de integración. La inmovilización se realiza generando una matriz de dimensión *MxN*, formada por *N* filas que representan los genes y *M* columnas que corresponden a los experimentos aplicados sobre todos los genes. La generación de esta forma de matriz o gradilla les proporciona el término por el que son más conocidos, microarreglos. La utilidad de los microarreglos radica no solo en contener gran parte del genoma sino en la posibilidad de proporcionar el transcriptoma de ese grupo de genes representados e identificados en la placa de silicio. Como se menciona en la sección 2 de este capítulo, las moléculas de ADN que representan a los genes se expresan por medio de ARN mensajero (mARN) para dar paso a la formación de proteínas, y lo que interesa en los experimentos de microarreglos es conocer la cantidad de mARN que expresa cada gene para identificar si éste reacciona o no ante ciertas condiciones dadas en los experimentos. Por lo anterior puede decirse que lo que se tiene con un microarreglo es un transcriptoma.

Al nivel de transcripción alcanzado por cada gene se le asigna una coloración derivada de la presencia de tintas fluorescentes (roja y verde) con las que se marca y al entrar en contacto físicamente ese mARN con el gene en el microarreglo se pega para fijarse. De esta forma se obtiene en el microarreglo un conjunto de genes con sus niveles de expresión que podrán utilizarse posteriormente bajo luz que excite la fluorescencia de las tintas marcadoras para conocer la intensidad que corresponde a la reacción de los genes ante determinadas circunstancias.

3.1 Obtención de información de los microarreglos

Como hemos mencionado, en los microarreglos la información acerca de la expresión de los genes se obtiene mediante la intensidad de los colores (rojo y verde) de las tintas fluorescentes con que se representó el mARN sobre cada gene. Al hacer reaccionar la

[4] Fuente de la imagen: http://www.gene-chips.com

[5] Este material biológico no sólo se trata de cADN o mARN que corresponden al transcriptoma, sino que actualmente también puede tratarse proteínas, lo que representa al proteoma.

fluorescencia en cada gene se mostrará gradualmente el rojo y el verde, para indicar la reacción del gene ante cierto agente promotor, en esta parte se aplica procesamiento de imágenes para obtener los valores numéricos que representen la concentración de mRNA transcrito. A la razón obtenida de esos valores de intensidad se aplica una transformación para su reducción como puede ser el cuadrado del valor, que comprime la escala para los pequeños pero que la expande para valores grandes o la raíz cuadrada que actúa de manera inversa a la anterior, o la transformación logarítmica, siendo ésta la más utilizada ya que permite contar con un rango de valores cuya media es el 0. De acuerdo a esta última transformación podemos presentar la expresión de un gene, *Y*, con la siguiente fórmula:

$$Y = \log \frac{I_{rojo}}{I_{verde}} = \log I_{rojo} - \log I_{verde}$$ donde I_{verde} e I_{rojo} se refiere a la intensidad de cada color medida en el gene

Los resultados obtenidos generalmente se encuentran en el rango de $-4 \leq Y \leq 4$. A partir de estos resultados se hace la representación numérica de la expresión de genes proveniente de un microarreglo, como se muestra en la figura 3.

Los resultados obtenidos de múltiples microarreglos forman las bases de datos de la expresión de genes, que pueden comprarse o en algunos casos encontrarse libres en la internet como en el caso de los sitios:
Expression Profiler - European Bioinformatics Institute (EBI)
http://ep.ebi.ac.uk/
Stanford Microarray Database (SMD)
Http://genome-www.stanford.edu/microarray

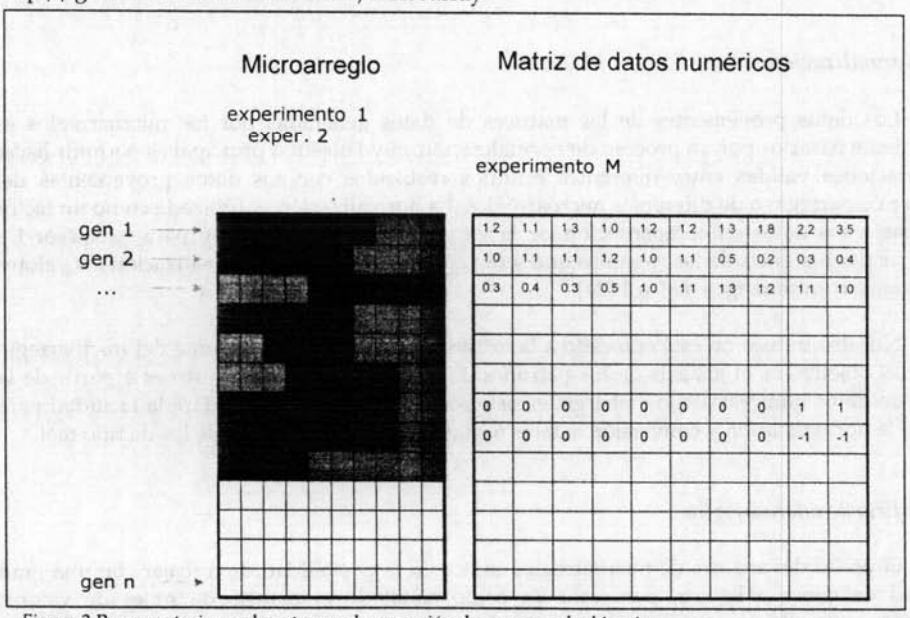


Figura 3 Representaciones de patrones de expresión de genes, real y binario.

Con cada renglón obtenido de los experimentos, se forma un vector de valores de expresión de cada gene y a partir de la matriz de datos numéricos se realiza el análisis de los patrones de expresión. Debido a la cantidad de datos que se generan en la mayoría de los casos se aplican técnicas para su reducción y facilitar su manipulación o para estandarizar y poder hacer comparaciones. Entre los métodos más utilizados para estandarizar se encuentra la normalización, que deja el rango de los valores entre 0 y 1 y permite hacer comparaciones entre diferentes experimentos o análisis realizados, en [2] se presentan diferentes formas de normalización. Nosotros aplicaremos una normalización por experimento. Entre las técnicas de reducción la más utilizada es el análisis de componentes principales donde los ejemplos que presenten genes con idéntico nivel de expresión se eliminan con el fin de desechar información redundante. La complejidad de los datos del arreglo puede verse reducida utilizando datos ternarios o hasta datos binarios, en lugar de los datos reales obtenidos, así en lugar de los valores continuos obtenidos inicialmente se trabaja con valores discretos. Mediante una función de umbral que cambie el nivel de transcripción en una expresión de datos ternaria (-1 bajo regulado, +1 alto regulado y 0 invariante) [15] o en la representación binaria, considerando únicamente los valores de 1 (alto regulado) y 0 (bajo regulado), con esta simplificación se permite asegurar un nivel aceptable de certeza en la especificación de cuáles genes son significativos para cambio de niveles de la expresión de mensajes en diversos microarreglos individuales. En [15], se propone el uso de datos ternarios para ayudar a un manejo más sencillo de la información. Sin embargo, con la representación mediante números reales disminuye la cantidad de cálculos realizados, debido a que los datos se quedan en sus valores originales, pero la variabilidad entre los datos es por mucho mayor que en los otros por lo que detectar los cambios de expresión significativos puede resultar más difícil. En el modelo neuronal para el análisis de patrones de expresión de genes que proponemos más adelante es posible trabajar con cualquiera de los tres tipos de datos para lo cual se incluyen procedimientos que permitan la manipulación de valores continuos o discretos.

3.1.1 Normalización

Los datos provenientes de las matrices de datos generadas por los microarreglos es conveniente pasarlos por un proceso de normalización, cuyo objetivo principal es permitir hacer comparaciones válidas entre diferentes estudios realizados con los datos provenientes del mismo microarreglo o de diferentes microarreglos. La normalización es utilizada como un factor de ajuste a los datos en compensación de la variabilidad experimental y para balancear los signos de fluorescencia de los ejemplos que serán comparados. Con la normalización los valores se encuentran entre rangos de 0 a 1 [24].

Nuestro trabajo no está enfocado a la extracción de datos directamente del microarreglo ya que el objetivo es el análisis de los patrones de expresión de genes y esto es a partir de la matriz de datos numéricos. Sin embargo, el sistema que proponemos brindará la facilidad para realizar la normalización y conversión a datos ternarios y binarios además de los de tipo real.

3.2 Análisis de microarreglos

El poder del análisis de microarreglos radica en la posibilidad de trabajar con una gran cantidad de genes a la vez, pero esto da como resultado el manejo de miles de valores numéricos, por ello se requieren herramientas poderosas que permitan llegar a resultados

significativos y confiables en un tiempo razonable. Dentro de las principales técnicas aplicadas en los últimos años al análisis de los microarreglos se encuentran [14, 12, 39, 22]:

- Los métodos estadísticos
- Los algoritmos para el análisis de clustering (agrupamientos)
- Aprendizaje automatizado (redes neuronales)
- Ingeniería en reversa
- Minería de datos

3.2.1 Métodos estadísticos

El análisis estadístico de datos de microarreglos es probablemente el problema más difícil. La estadística por lo general trabaja con pocas variables, mientras que los microarreglos producen miles de variables desde pequeñas cantidades de ejemplos. El objetivo es aplicar una aproximación estadística estándar para determinar la expresión genética y la alteración significante de la expresión de genes y extraer la información biológica significativa [22].

Más allá de considerar que la mayoría de los métodos para el análisis de microarreglos utilizan alguna forma o función estadística para simplificar o mejorar el aprovechamiento que puede hacerse de los miles de datos proporcionados por los microarreglos, existen métodos que son puramente estadísticos para guiar el análisis y la interpretación de datos producidos por los microarreglos.

Una de las principales aplicaciones estadísticas en este último sentido es el análisis de componentes principales (PCA), con lo que se hacen estudios de los cambios en la expresión de genes bajo ciertas condiciones, como puede ser en su resistencia a un fármaco, como es el caso del tamoxifén para cáncer de mama, presentado en [12]. El PCA es un método práctico y útil para el análisis de datos de expresión de genes provenientes de un microarreglo de ADN, con el objetivo principal de reducir la cantidad de información proporcionada. El análisis de componentes principales es considerado práctico y estadísticamente válido como una aproximación a examinar simultáneamente un arreglo de datos en varios puntos de tiempo en un modelo *in vivo*, lo que permite detectar alteraciones medidas moderadamente en la expresión de genes. Sin embargo, la principal utilidad dada al PCA es previa al análisis de datos de expresión de genes hecha por k-means o SOMs pues ayuda al usuario a especificar el número de cúmulos (clusters), proporciona una aproximación.

Otras de las técnicas estadísticas más utilizadas son la varianza y covarianza, para encontrar el grado de variabilidad entre los genes expresados para distinguir verdaderas diferencias de expresión de las derivadas del experimento mismo.

Otro de los métodos para crear redes genéticas a partir de la expresión de genes es el correspondiente a los modelos de redes bayesianas dinámicas, presentado en [14], donde se obtienen redes genéticas con datos en series de tiempo. Estas redes bayesianas pueden modelar estadísticamente los datos, incorporar conocimiento *a priori*, y tener variables ocultas, con lo que compensan la falta de datos iniciales. Los algoritmos son capaces de considerar el ruido de los datos. Los autores de [14] consideran apto su método probabilístico ya que es un modelo

estocástico y la expresión de genes es un fenómeno inherentemente estocástico; por otro lado, eligieron las redes bayesianas por ser modelos que representan la incertidumbre, además son dinámicas ya que representan cómo una variable aleatoria evoluciona en el tiempo. Las redes bayesianas con arcos dirigidos permiten conocer causas y también pueden ser utilizadas para efectos desconocidos, pero también con efectos conocidos y causas desconocidas o en cualquier combinación. Con una gráfica dirigida las relaciones son deterministicas y fácilmente se puede aprender; por otro lado los modelos creados por la red son separables. Sin embargo, en la mayoría de las ocasiones no se cuenta con información *a priori* confiable para utilizar este método.

3.2.2 *Clustering o generación de cúmulos*

Uno de los métodos más utilizados actualmente para el análisis de expresión de genes basado en el estudio de microarreglos es el clustering. Esta es una herramienta capaz de formar grupos de genes con perfiles de expresión similares para obtener patrones de expresión, los genes con estos perfiles presentan funciones similares o su regulación es de manera común. Existen diversos tipos de algoritmos que tienen la capacidad de identificar patrones y también de relacionar las secuencias de ADN con otros patrones ya conocidos [31]. Los algoritmos difieren por la forma en que realizan la transformación de datos, en la obtención de las distancias establecidas entre los genes y en el establecimiento del número de clusters [6].

La meta de métodos como el análisis de clustering es identificar los genes que muestran patrones similares de expresión. Se comienza por definir un vector de expresión para cada gene, el cual representa su localización en un espacio de expresión. La dimensión del espacio de expresión crece con el número de experimentos, donde cada vector de expresión de gene es representado como un simple punto en el espacio. Debido a que las actividades de los genes son frecuentemente relacionadas con las de otros, los patrones de expresión de genes bajo diferentes condiciones pueden ser similares, el clustering es necesario para identificar los patrones. La funcionalidad del clustering aplicada a la expresión genética radica en: a) permite extraer co-regulación a partir de co-expresión, b) permite la inferencia funcional y c) representa la firma molecular para distinguir entre diferentes tipos de células o tejidos [4].

Existen tres diferentes clasificaciones para los métodos de clustering (ver la figura 4.):

- Supervisado y no supervisado
- Jerárquicos y los no jerárquicos
- Aglomerativos y de división

El resultado del clustering jerárquico es un número de clases anidadas que asemeja una clasificación filogenética, mientras que el clustering no jerárquico muestra una serie de particiones de objetos que representan los cúmulos y sin tratar de mostrar relaciones entre los elementos individuales. En cuanto a su clasificación como aglomerativo o divisible podemos decir que, el clustering aglomerativo es aquel que comienza generalmente con un solo miembro por cúmulo y gradualmente se van fusionando más elementos. Mientras que en el cúmulo divisible se inicia con todos los elementos en un solo cúmulo y conforme avanza el algoritmo se va dividiendo en cada vez más pequeños cúmulos. En el clustering supervisado los vectores son

clasificados con respecto a vectores de referencia, ya conocidos, mientras que en el no supervisado no existen vectores de referencia. Ya que nuestro modelo propuesto para reconocer patrones de expresión de genes pertenece a los métodos de clustering, en el punto 4 se tratan con más profundidad algunos algoritmos de clustering.

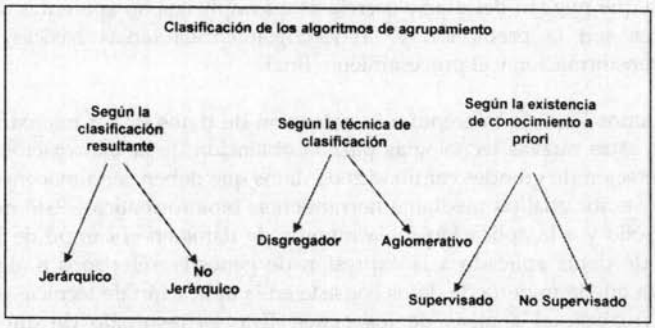


Figura 4 Clasificación de los algoritmos de clustering.

3.2.3 *Aprendizaje automatizado (redes neuronales)*

Son múltiples las aplicaciones que se han hecho de redes neuronales para el análisis de expresión de genes y en la sección 5 de este capítulo presentamos algunos ejemplos de este tipo. Nuestro modelo propuesto para el análisis de expresión de genes pertenece a esta clase de métodos de análisis por medio de clustering.

3.2.4 *Ingeniería en reversa*

La expresión de los genes permite conocer relaciones de asociación entre los mismos y ésta se puede representar por medio de redes genéticas [33]. La ingeniería en reversa es uno de los métodos para encontrar la influencia que ejercen unos genes sobre otros, es decir, para la creación de una red genética con datos multivariantes. La red genética toma como información inicial el patrón de expresión de diversos genes en una variedad de circunstancias y contextos celulares, considerando el patrón como respuesta a su ambiente y a la acción de otros genes, por lo que considera que los genes están influenciados por otros. Este método fue creado por ingenieros que han desarrollado diversas aproximaciones matemáticas para modelar las relaciones que gobiernan la función de un componente de un sistema complejo por interfaces derivadas de las comparaciones de estados de otro sistema. Esto fue aplicado a la biología, en donde lo primero que se debe hacer es modelar el sistema con características básicas y obtener un número extenso de mediciones del ambiente y de los componentes como operadores del sistema y todo esto debe ser reconstruido, como un estudio *in vitro*. Este tipo de modelado requiere tener bien claro cuales son los genes que toman parte en el proceso a ser estudiado, para medir solo los componentes relevantes y esto último dificulta mucho su uso.

3.2.5 *Minería de datos*

La minería de datos consiste en la extracción de información predictiva que se encuentra oculta en las grandes bases de datos [19]. Consiste en el análisis metódico y la organización de los datos de manera que puedan detectarse correlaciones implícitas no aparentes. Las metas de la minería de datos son la predicción y la descripción. Sus tareas básicas son el pre-procesamiento, la transformación y el procesamiento final.

Cuando tratamos la parte referente a la obtención de datos de un microarreglo se dijo que la aparición de estas nuevas tecnologías para la obtención de la información genética ha conducido a la generación de grandes cantidades de datos que deben ser almacenados en bases de datos para su posterior análisis mediante herramientas bioinformáticas. Este nuevo reto ha conducido al desarrollo y a la aplicación de la minería de datos en el campo de la genómica. Hablar de minería de datos aplicado a la expresión de genes es referirnos a algunos de los puntos anteriores, ya que la minería de datos consiste en la aplicación de técnicas estadísticas y de clustering para realizar el análisis de los genes. Esto es resultado de que los valores correspondientes a la expresión de genes, obtenida de los microarreglos principalmente, llegan a conformar grandes bases de datos. A estas bases de datos biológicas en esencia la minería de datos les ofrece [39]:

- Limpieza de datos, procesamiento de datos, integración de datos heterogéneos
- Bases de datos biomédicos distribuidas
- Exploración de datos biológicos
- Búsquedas y comparación entre datos
- Análisis de asociaciones: identificación de co-ocurrencias de secuencias o de patrones correlacionados
- Análisis de tipo clustering basado en patrones
- Visualización de los datos
- Privacidad de datos

4. La agrupación de datos por cúmulos (clustering)

El análisis por medio de cúmulos es un término genérico utilizado para hacer referencia a los métodos numéricos que examinan datos multivariantes, es una forma de descubrir grupos (clusters) de observaciones homogéneas. El análisis por medio de cúmulos está orientado esencialmente al descubrimiento de grupos de datos, por lo que integra una de las formas de minería de datos. El cúmulo es visto como un grupo, una clase, un cluster de datos. En la agrupación por cúmulos son utilizados diversos algoritmos de clasificación que pueden proveer de algunas taxonomías.

Los algoritmos de clustering permiten ordenar los datos y agrupar los genes con base en su separación en el espacio de expresión. Para realizar el clustering es indispensable representar

por medio de un vector los valores de expresión de cada gene, esto se define como un espacio cuya dimensionalidad es igual al número de experimentos aplicados a los genes.

4.1 Medidas de distancia

Uno de los aspectos más importantes dentro del análisis de cúmulos es la identificación de qué tan próximos entre sí o, qué tan lejanos, están los elementos. Para esto nos servimos del uso de medidas de distancias que se basan en que dos individuos están próximos cuando su disimilaridad o distancia que los separa es pequeña, es decir, su similitud es grande. Las medidas de proximidad, similitud o semejanza miden el grado de semejanza entre dos elementos de forma que, cuanto mayor es su valor, mayor es el grado de similitud existente entre ellos y con más probabilidad los métodos de clasificación tenderán a ponerlos en el mismo grupo. Mientras que las medidas de disimilitud miden la distancia entre dos elementos de forma que, cuanto mayor sea su valor, más diferentes son los elementos y menor la probabilidad de que los métodos de clasificación los pongan en el mismo grupo.

Existe una gran variedad de medidas de semejanza y de distancia dependiendo del tipo de variables y datos considerados. La medida de distancia se elige de acuerdo al objetivo para el cual se hará el análisis de clustering. La distancia euclidiana y la correlación de Pearson son especialmente las más utilizadas, la primera es conveniente cuando el clustering se hace para conocer los niveles de expresión similares, es decir, patrones de expresión idénticos, y la segunda para distinguir la tendencia de los patrones de expresión, pues utiliza la desviación estándar y da valores entre 1 y -1. Nuestro modelo permitirá elegir entre estas dos medidas principales además de la de Manhattan, Mahalanobis y Hamming.

Entre los algoritmos para clasificación no supervisada con análisis de cúmulos los más utilizados son *k-medias* y los mapas auto-organizados (*SOM*), pero también son muy utilizados para el análisis de la expresión genética los algoritmos de clustering jerárquico.

4.2 Clustering jerárquico

Este tipo de clustering se reconoce como el más simple y fácil de visualizar ya que nos da como resultado un dendograma que se forma de manera aglomerativa, donde inicialmente cada gene es un cluster, sus pasos en general son:

1) Calcular la distancia entre pares con todos los genes considerados y se crea una matriz de distancias;
2) Buscar el par de genes más semejante en la matriz de distancias y se representan como dos clusters;
3) Los dos cúmulos seleccionados son mezclados para producir un nuevo cúmulo;
4) Las distancias son calculadas entre el nuevo cúmulo y los demás, y
5) Los pasos del 2 al 4 se repiten hasta que todos los elementos están en un solo cúmulo.

Existen diferentes variaciones del clustering jerárquico, como puede verse en el artículo de Quackenbush [24].

En esta técnica de clustering, conforme avanza el algoritmo el patrón de genes encontrado es cada vez menos significativo por englobar a todos los perfiles, pero lo mismo sucede en una etapa temprana, pues casi todos los genes están dispersos de manera individual. Este algoritmo varía de acuerdo a la distancia utilizada como medición entre los cúmulos construidos, y al final se presentan todos los cúmulos formando un dendograma, como lo muestra la figura 5.

Un problema que se presenta con los métodos de clustering es debido a que los cúmulos crecen en tamaño y los vectores de expresión que representa el cúmulo no podrán crecer para representar cualquiera de los genes de los clusters. En la medida en que progresa el algoritmo de este tipo de clustering los patrones de expresión de los genes encontrados llegan a ser menos relevantes.

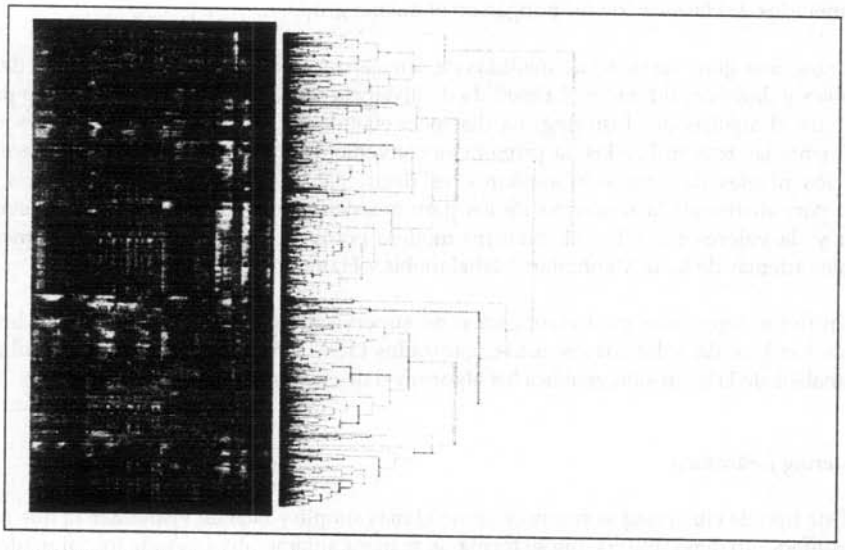


Figura 5 Representación de clustering jerárquico, con su dendograma[6].

4.3 Clustering con el algoritmo de k-medias

Clustering por *k-medias* es del tipo de cúmulo no jerárquico y divisible, los elementos son particionados en un número fijo de cúmulos (k cúmulos), de tal manera que dentro de cada cúmulo los elementos son muy semejantes, pero externamente, entre los cúmulos, son muy distintos. En un inicio todos los elementos se encuentran asignados aleatoriamente en los k cúmulos, el número k lo proporciona el usuario y se obtiene comúnmente por la técnica

[6] Fuente: Eisen et al., cluster analysis and display of genome-wide expression patterns, 998, Proc. Nat. Acad. Vol. 95.

estadística de análisis de componentes principales, en la figura 6 se representa este método. Los pasos a seguir para la realización de este tipo de clustering son:

Dado un conjunto no etiquetado de perfiles de expresión,

1) Definir un número *k* de cúmulos, dividir aleatoriamente todos los perfiles de expresión en k cúmulos.
2) Obtener un vector de expresión promedio (la media) para cada cúmulo.
3) Calcular las distancias entre los elementos y cada vector promedio, tanto de manera interna como externa, de cada cúmulo.
4) De manera iterativa, los elementos o vectores de expresión se mueven entre los cúmulos de acuerdo a las distancias que se van obteniendo, los elementos pueden cambiar de cúmulo si está más cercano a otro distinto al que se encontraban.
5) El procedimiento continúa (pasos 2, 3 y 4) hasta que los cúmulos se estabilizan, es decir, hasta que ya no hay movimientos, y las distancias entre los cúmulos ya no aumentan ni disminuyen en el interior de cada uno.

El comportamiento del algoritmo depende en gran medida del número *k* de cúmulos especificado, de la selección inicial de centros y del orden en el que los ejemplos se presentan.

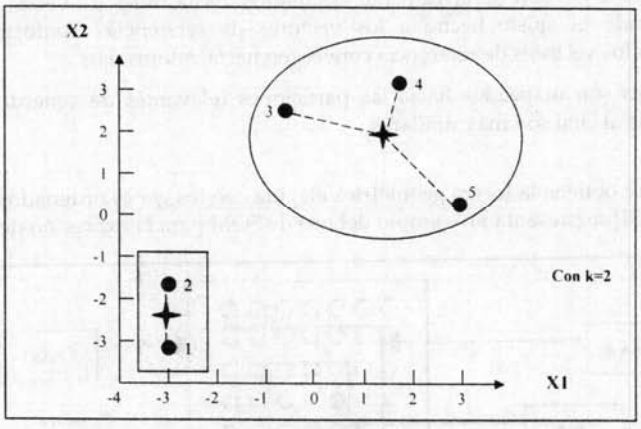


Figura 6 Formación de k cúmulos, con K=2.

En la figura 6 se presentan dos clusters de acuerdo al valor dado a *k*, en el primer cluster se encuentran los puntos 1 y 2 unidos por su media, y en el segundo cluster se unen los tres puntos 3, 4 y 5 unidos por la media de su cluster. Así, cada cluster agrupa los puntos más cercanos a un punto medio para formar un número ya determinado de clusters.

4.4 Los mapas auto-organizados (SOM)

Son conocidos como SOM por sus siglas en inglés (Self-Organizing Map), este tipo de cúmulo es desarrollado con base en una red neuronal, la de Kohonen [16]. También corresponde al clustering no jerárquico y divisible. Su forma de red neuronal artificial la presentaremos detalladamente en el capítulo 1, por ahora sólo mencionaremos de manera general en qué consiste el algoritmo. En los mapas auto-organizados se asignan los genes a un grupo de particiones de acuerdo con su similitud, las particiones son definidas de acuerdo a una forma geométrica establecida al inicio, generalmente se trata de un rectángulo o de un hexágono de dos dimensiones, en la figura 7 puede verse un ejemplo de las formas geométricas. En este caso, como en el de *k-medias*, también es posible utilizar el PCA como apoyo para la elección de la forma y dimensiones de la forma geométrica. El algoritmo se desarrolla siguiendo los siguientes pasos:

Una vez definida la forma geométrica,

1) Los vectores para cada cúmulo son construidos aleatoriamente.
2) Un gene es elegido de manera aleatoria, y por medio de la distancia se identifica el vector de referencia que es más cercano a ese gene.
3) El vector de referencia es ajustado para que sea más similar al vector del gene asignado. Los vectores de referencia más cercanos (los vecinos) en el espacio de dos dimensiones son también ajustados para hacerlos más similares.
4) Los pasos 2 y 3 son iterativamente ejecutados, hasta miles de veces, en las que va decreciendo el ajuste hecho a los vectores de referencia. Conforme el proceso continúa los vectores de referencia convergen hacia valores fijos.
5) Los genes son mapeados hacia las particiones relevantes de acuerdo al vector de referencia al cual son más similares.

Finalmente se obtiene la forma geométrica elegida con los genes ordenados de acuerdo a su expresión. En [34] se presenta un ejemplo del uso de SOM para la expresión de genes.

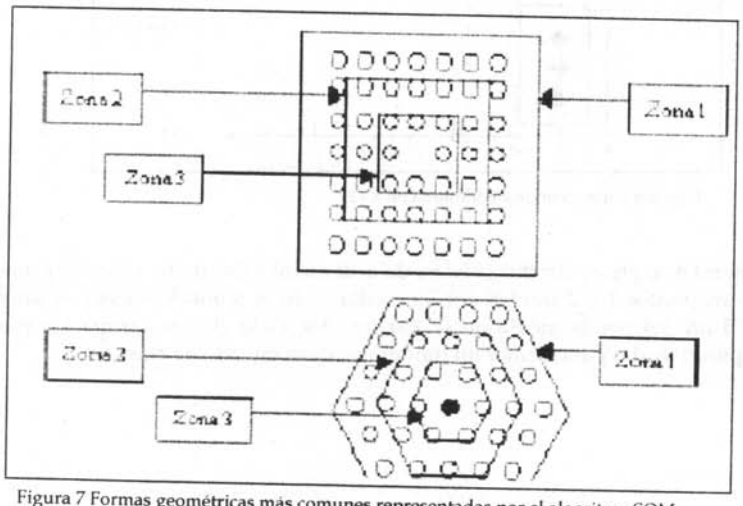


Figura 7 Formas geométricas más comunes representadas por el algoritmo SOM.

5. Aproximación a los estudios de patrones de expresión de genes con redes neuronales artificiales

En la última década del siglo XX, las redes neuronales artificiales (RNA) han sido utilizadas ampliamente para el análisis de información de expresión de genes, además de su aplicación para el estudio de secuencias de ADN. Esto se debe principalmente a la habilidad que presentan estos métodos para el reconocimiento y clasificación de patrones, no sólo de datos de tipo cuantitativo sino también de tipo cualitativo, como las secuencias de ADN [23].

Entre los modelos de RNA más utilizados encontramos a las redes multicapa y las de tipo Kohonen [16]. Cada perfil de expresión de genes es presentado como un vector de entrada a la RNA, en donde la dimensión de ese vector indica la cantidad de experimentos realizados definiendo así el número de entradas a la red. Las salidas proporcionadas por estas redes comúnmente representan las diferentes categorías que agrupan patrones de expresión de genes similares a las asociaciones entre los genes.

A continuación revisaremos algunos de los trabajos desarrollados en los últimos años, los cuales involucran información proveniente de los microarreglos y técnicas de análisis usando RNA.

Herrero et al. crearon un modelo neuronal no supervisado denominado SOTA [10]: SOTA es un método de análisis de expresión de patrones de genes; éste corresponde a los métodos de aprendizaje de RNA, pues su funcionamiento se basa en el algoritmo de aprendizaje no supervisado de Kohonen [16]. Esta RNA crece adoptando una topología de árbol binario del cual se deriva el nombre SOTA (self-organizing tree algorithm) [10]. El resultado final del método es dado por medio de un dendograma, donde los niveles jerárquicos están definidos por neuronas que representan el promedio de los patrones de expresión contenidos en los clusters. El espacio de salida puede crecer tanto como se requiera para mostrar la variabilidad de los datos de entrada. En la operación de la RNA, como en la red de Kohonen, se obtiene un nodo ganador y se modifica a sus vecinos, el entrenamiento se realiza mientras las neuronas sean heterogéneas ya que en este caso en cada ciclo se divide a la neurona en dos neuronas hermanas. La estructura del árbol jerárquico final (ver figura 8) depende de la heterogeneidad de los perfiles presentados, pues ello determina cuantos cúmulos fueron necesarios para representar esa heterogeneidad, aunque puede incluso quedar el caso de un solo cluster.

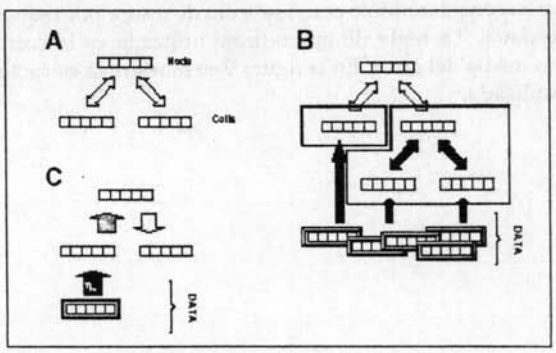


Figura 8 Evolución del modelo SOTA.

En la figura 8 se presentan las neuronas como vectores (células y nodos). Inicialmente (en A) la red cuenta con dos células hermanas conectadas a la neurona interna (nodo), posteriormente la red va agregando células hermanas; si existe heterogeneidad en los datos presentados, las células hermanas se actualizan de acuerdo a los datos que ingresan a la red. Las flechas negras indican actualizaciones y las grises donde se ha estabilizado la red; esto puede apreciarse en B. Los flujos de las interacciones de la red se presentan en C.

En este método de análisis la estructura final no se encuentra determinada en forma geométrica como sucede con la red de Kohonen, lo cual le da más flexibilidad al formar clusters y para obtener los valores de las distancias permite elegir entre dos medidas, la euclidiana y la correlación de Pearson. Este método tiene semejanza con el que nosotros proponemos en cuanto a que la estructura final será dada de acuerdo a los datos de entrada, es decir, a los perfiles de expresión de genes que se presentan y no estará determinada *a priori*, así como en la posibilidad de utilizar diferentes medidas de similitud. Sin embargo, nosotros damos la posibilidad de utilizar más medidas de similitud, las cuales se eligen de acuerdo a la finalidad para la que se hace el estudio biológico, además nuestra estructura final no es un dendograma, y más allá de indicar similitudes entre patrones, se proporciona además la clase o categoría a la que pertenece cada grupo.

El equipo de Kim et al., muestra en [15] un reporte de su perceptrón simple para el análisis de patrones de expresión de genes. Este método se encarga principalmente de realizar asociación de genes, utilizando la expresión de genes; a partir de la transcripción de dos genes dados se puede predecir el estado transcripcional de otro gene, y de esta manera puede armarse una red genética, la cual indica las relaciones dadas entre diversos genes. Considerando que los datos que se utilizarán son multivariantes (por provenir de diversos genes y no de uno solo) y que pueden existir errores en los valores de entrada, se utiliza una red neuronal. En este caso se trata de un perceptrón simple con el algoritmo de *backpropagation*. Con este método se pretende saber si los genes son más o menos interactivos por sus asociaciones establecidas en la red genética. El uso de una red neuronal permite tratar la información inexacta y es un método rápido de identificar nuevos componentes inesperados de procesos ya identificados, al igual que permite encontrar ligas no esperadas entre procesos no conocidos previamente. También permite incorporar conocimiento de condiciones incluidas como elementos productivos que afectan el nivel de expresión de un gene. El objetivo principal es encontrar conjuntos pequeños de genes involucrados en un proceso particular. Este método propone reducir los datos a una representación ternaria, -1, 0 y 1 para reducir el rango de representación de valores, lo que les permitió utilizar el perceptrón considerándolo como sencillo de usar y por requerir una cantidad relativamente pequeña de datos. La regla de aprendizaje utilizada es la corrección del error basado en el cuadrado de la media del error. En la figura 9 se muestra la estructura general de la red de perceptrón simple utilizada.

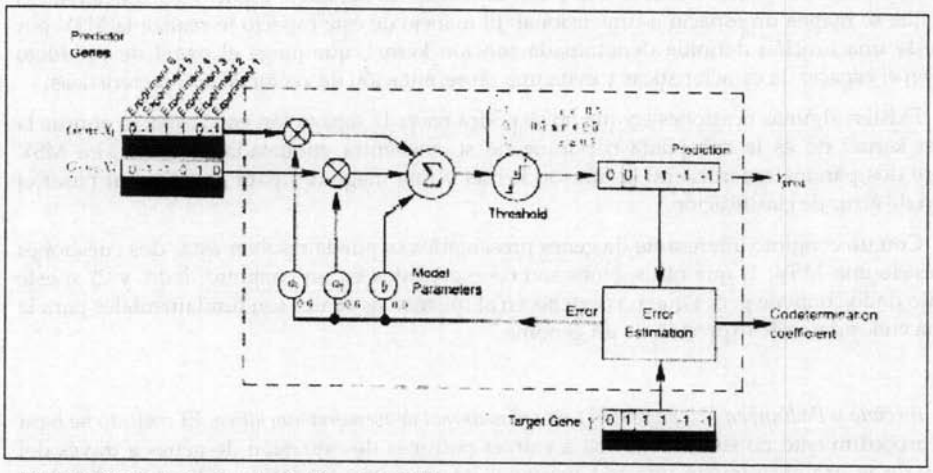


Figura 9 Perceptrón simple de Kim *et al.* [15]

Por los resultados mostrados por Kim [15], este método ayuda a la creación de una red genética; sin embargo esa red no expresa más información de los genes ya que no puede conocerse la funcionalidad correspondiente a los genes y con ello crear cúmulos para finalmente advertir alguna clasificación. La representación ternaria dada a los datos se obtiene a partir de la aplicación de un umbral, con lo que parece no perderse información por esta reducción a valores discretos. Por ello en nuestro modelo consideramos también este método de representación de la información.

Brown et al., proponen en [3] una máquina de soporte vectorial (MSV) para el análisis de patrones de expresión de genes. La máquina de soporte vectorial representa un método de clasificación funcional de genes basado en los datos de expresión de genes de experimentos en microarreglos. Es del tipo de aprendizaje supervisado por computadora porque explora el conocimiento *a priori* de la función de los genes ya identificados para encontrar los genes desconocidos con función similar de los datos expresados. Este modelo de MSV tiene muchas características matemáticas que lo hacen muy atractivo para el análisis de genes, trata de evitar la dispersión en la solución dada cuando son grandes conjuntos de datos. Las MSV [8] son aptas para la clasificación basándose en la discriminación entre miembros y no miembros de una clase, la separación de ello se realiza por medio de un hiperplano.

En [3] se da una clase funcional y se dice cuáles genes pertenecen a ella y cuáles no. Durante el entrenamiento, la MSV trata de localizar las características de expresión que diferencian a un grupo funcional y con esta información decide cuál gene puede ser miembro del grupo. Con la información *a priori* de este método puede darse no sólo su pertenencia a un grupo o no sino que además se puede decir si están correlacionados los genes, lo que no logran todos los métodos.

Se crea un hiperplano para separar los miembros y no miembros, expresados como positivos y negativos respectivamente. En el problema de clasificación de genes se encuentra

que el hiperplano no puede referirse solo a dos dimensiones debido al espacio de características por lo que se mapea un espacio n-dimensional. El manejo de este espacio lo realiza la MSV por medio de una función definida denominada función kernel, que juega el papel de producto punto en el espacio de características y evita una representación de vectores de características.

Existen algunas ocasiones en que no se podrá hacer la separación en el espacio porque la función kernel no es la apropiada o porque no se encuentra etiquetada esa clase. La MSV requiere dos parámetros iniciales: la función kernel y una magnitud para penalizar al pasar el margen de error de clasificación.

Con un conjunto interesante de genes presentados se puede resolver estas dos cuestiones a través de una MSV: 1) que otros genes son co-expresados en un conjunto dado, y 2) si este conjunto dado contiene genes que no pertenecen al mismo, las cuales son fundamentales para la biología en los datos de expresión de un genoma.

Bicciato y Padinni en [2], presentan una red neuronal de memoria asociativa. El método se basa en un procedimiento no supervisado para extraer patrones de expresión de genes a través del análisis de la arquitectura de una red neuronal de memoria asociativa aplicada a datos de leucemia en humanos y de cultivos de levadura. El propósito del trabajo realizado en [2] es proporcionar un procedimiento para determinar todas las relaciones posibles y asociaciones dadas entre diferentes genes y destacar la dinámica de patrones de expresión correlacionados. Para lograr lo anterior se utiliza un modelo basado en el esquema de las redes de memoria asociativa [2] y de redes multicapa [8]. Por esta última característica, la red tendrá la habilidad de realizar asociaciones no lineales, usando la regla de aprendizaje Hebbiana para el entrenamiento de la red. Asimismo se le incorpora el algoritmo de *backpropagation*. En el capítulo 1 se tratan con mayor amplitud los modelos de redes neuronales artificiales.

La arquitectura del modelo presentado se puede observar en la figura 10 con tres capas, una capa de entrada, con conexiones uno-a-uno a una segunda capa (la capa oculta) y una capa de salida fuertemente conectada a la oculta. La red detiene su entrenamiento de acuerdo a un criterio de minimización de error y de la derivada de la función de error respecto al conjunto de entrenamiento. La conexión de tipo uno a uno (monoconexión) entre las capas de entrada y la oculta es sólo para preservar la característica de perceptrón multicapa.

Los valores utilizados, como en casi todos los experimentos, son normalizados entre 0 y 1 para su procesamiento en la red. Los pesos iniciales de la red son dados en un rango de 0.01 a 0.8. El funcionamiento de la red ha sido comparado utilizando ejemplos etiquetados y en el 88% de los casos se encontró correcto [2].

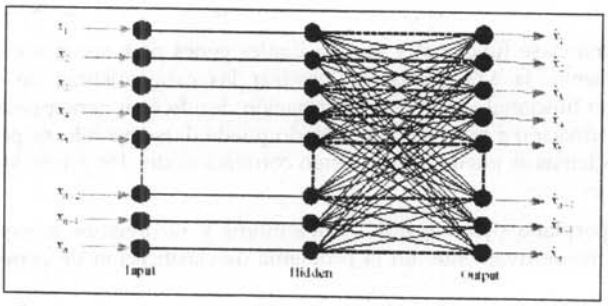


Figura 10 Representación de la estructura del modelo de RNA. [2]

El modelo de red de memoria asociativa puede aplicarse directamente para la obtención de asociaciones entre los patrones de expresión de genes; no obstante, esta red al tratar de clasificar también a los genes de manera no lineal, incorpora una forma de perceptrón multicapa con *backpropagation* solo entre las capas de entrada y oculta con monoconexión tratando de evitar mayores complicaciones al modelo, y en esta parte se realiza un tipo de aprendizaje supervisado, mientras que entre la capa oculta y la de salida es un aprendizaje no supervisado.

El grupo de investigación del Departamento de Informática, en Helsinki, UoB by Bjarte Dysvik e Inge Jonassen desarrollaron un software denominado J-express (http://www.molmine.com). El principal objetivo de este trabajo es el análisis de material genético correspondiente a microarreglos de datos de expresión de genes provenientes de microarreglos, es decir, de miles de genes a la vez. Su objetivo es detectar patrones de expresión de genes utilizando métodos de clustering como: clustering jerárquico, mapas auto-organizados y *k-medias*. Es un software muy versátil a este respecto. Se programó en Java 1.3 y permite visualizar los clusters por medio de gráficos y listas de genes. Una versión de demostración se encuentra libre en el sitio de internet y la utilizaremos en el capítulo 5 de este trabajo para realizar comparaciones con nuestros experimentos.

El modelo de red neuronal que nosotros proponemos integra tanto técnicas del clustering supervisado como del no supervisado, pues realizará la agrupación en cúmulos de los vectores de expresión de genes presentados ya sea que tengan o no una clase asociada. Como ya se mencionó es importante la medida de distancia utilizada pues los resultados pueden alterarse, por lo tanto permitiremos que el usuario elija entre las medidas de similitud y distancia más importantes para este tipo de datos. Esta flexibilidad la encontramos en casi todos los modelos anteriormente citados. En cuanto a la estructura general de la red, estará compuesta de tres capas, la de entrada cuyo número de neuronas será de acuerdo a la dimensión de los vectores de expresión de genes (es decir, número de experimentos realizados), la oculta en donde se formarán los cúmulos asociados a cada clase, y la capa de salida, en la cual se indicarán las clases encontradas. La clasificación no es lineal y esto se denota inmediatamente porque es una red multicapa, con una capa oculta. Una de las mayores ventajas del modelo que proponemos es que la RNA no tiene una estructura predeterminada, es decir, que la RNA tendrá al final la estructura que necesita de acuerdo con los vectores presentados, al contrario de lo que pasa en los modelos anteriores en los cuales se anticipa en cierta forma la estructura de la red. Este modelo propuesto también permite obtener más conocimiento de los genes presentados, pues al crear diversos cúmulos para la clasificación, será posible establecer relaciones entre los genes expresados o encontrar diversos grados de similitud. Los vectores de entrada podrán presentarse con datos binarios, ternarios o reales, con el objeto de facilitar su representación, pero sin perder información. En el capítulo 2 explicamos con más profundidad el modelo que proponemos.

Tamayo et al. en [34] proponen una RNA basada en SOM. Este método se ha implementado en un paquete denominado Genecluster, el cual realiza diversos cálculos y se encarga de la presentación visual de los datos. Es un modelo basado en los mapas auto-organizados (SOMs), por lo que pertenece al clustering no supervisado. Su análisis de los patrones de expresión de genes da una estructura parcial de los clusters, propuesta en contraste con la estructura rígida del clustering jerárquico y la hipótesis fuertemente *a priori* del de Bayes y la no estructura de *k-*

medias, y que facilita además la visualización e interpretación. Consideran que SOM tiene propiedades computacionales buenas e implementación razonablemente rápida y escalable a conjuntos de datos grandes. El funcionamiento de la RNA es basado en la.de Kohonen, con una función de aprendizaje que incluye constantes para llevarla a que converja. El número de nodos dado a la geometría de SOM es básico ya que pocos nodos no permiten distinguir diversos patrones, muchos hacen una gran dispersión, por lo que es necesario identificar un número apropiado. SOM es muy utilizado en minería de datos por sus bases matemáticas. Presenta ejemplos realizados con datos de células de levadura y de humano para estudiar células cancerígenas. En la figura 11 se presenta un ejemplo de su evolución para una estructura rectangular de 3x2. Podemos ver la movilidad que tienen los centroides en las iteraciones para formar los 6 clusters indicados mediante la forma geométrica sugerida.

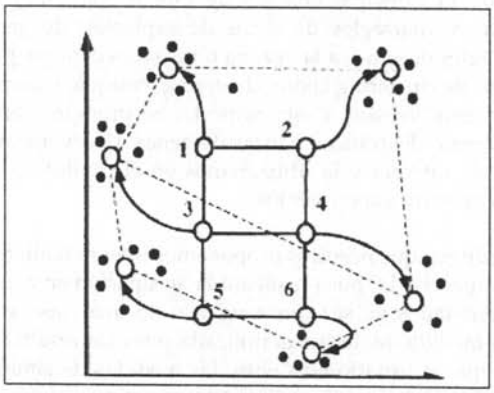


Figura 11 Evolución de la red de Tamayo et al. [34]

La debilidad presentada en este modelo, como en todos los basados en SOM, es que requiere darse inicialmente la estructura geométrica y con ello también el número de cúmulos que se esperan formar, lo cual puede generar tendencias en el resultado final, así como pérdidas de nueva información.

6. Conclusiones

La información provista por el amplio estudio del genoma de los modelos de sistemas, no sólo transcriptoma (lo transcrito en una célula) sino además de todas las proteínas (el proteoma) y todos los componentes del metabolismo (el metaboloma) en un organismo, representa un importante reto para mejorar la automatización hacia un alto desempeño de procesamiento en el análisis. Como respuesta a ello se ha incrementado la necesidad de utilización de procedimientos de minería de datos, entre los que destaca el clustering, que permiten obtener conocimiento más significativo.

Existe una fuerte necesidad de ideas novedosas sobre cómo tratar y evaluar esa complejidad en las aproximaciones de sistemas biológicos. En adición, toda la información desconsiderada de la secuencia de ADN, el análisis de expresión del amplio genoma o la alta escala de investigación del comportamiento mutante, tiene que ser tratada, analizada y

presentada a la comunidad. Se requiere de la colaboración de diversas competencias, englobadas en genómicas y bioinformáticas, esta multidisciplinaridad será la clave de los sucesos científicos en el área biológica. En este sentido las ciencias de la computación tiene mucho que aportar a los métodos de análisis de información pues la automatización de un método de clustering mediante una red neuronal permitiría inicialmente reducir la complejidad en el manejo y exploración de los datos y su resultado ordenado, que en diversas formas facilitaría la interpretación. Permitirían descubrir nuevas funcionalidades para los genes además de las ya conocidas, indicando quiénes son los genes relacionados en un mismo proceso celular y permitiría la eliminación de redundancias encontradas. Una de las mejores formas de hacer clustering de manera automática es por medio de las RNA, las cuales representan modelos computacionales para el procesamiento de grandes volúmenes de información de gran complejidad y con datos erróneos o confusos.

Considerando todas estas características deseadas en los métodos de clustering hacemos nuestra propuesta de modelo de RNA para creación de cúmulos de patrones de expresión de genes, la cual presentamos en los capítulos 2 y 3, después de mostrar y discutir sus principales características estructurales y funcionales de las redes neuronales artificiales.

Todo el software encontrado presenta dificultad para ser utilizado en cuanto a licencias, así como en su operación, cualquier mejora o adaptación depende de sus creadores. Es por ello que consideramos importante realizar uno nacional.

Capítulo 1.
Panorama de las redes neuronales

Existen diversos modelos computacionales para el procesamiento de información que han sido inspirados en el funcionamiento de sistemas biológicos. Asimismo, estos modelos computacionales han constituido diferentes aproximaciones para modelar el comportamiento de diversos sistemas biológicos [7]. Dentro de las ciencias de la computación, el área de la inteligencia artificial constituye uno de los principales escenarios para el modelado de los sistemas biológicos. Este hecho responde a la gran variedad de modelos, técnicas y métodos que soportan esta área de investigación, heredados de disciplinas como la biología, la psicología y las ciencias cognitivas. Como ejemplo de los sistemas de procesamiento de información inspirados en los sistemas biológicos se encuentran las redes neuronales artificiales (RNA). Las RNA están basadas en el procesamiento de información que toma lugar en el cerebro humano, simulando ciertos rasgos propios de la inteligencia, tales como las facultades de aprendizaje, memoria y adaptación.

Las RNAs, como modelos de procesamiento de información basados en el funcionamiento del cerebro humano, están formadas por una gran cantidad de unidades o elementos de procesamiento que representan a las neuronas. Una RNA es un sistema de procesamiento distribuido y paralelo, hecho de simples unidades de procesamiento que tienen una natural propensión para almacenar conocimiento experimental y habilitarlo para su uso [8]. Los problemas a los que se aplican las RNAs son aquellos que resultan difíciles de describir cuantitativamente y de los que sólo se dispone de información parcial sobre ellos [16]. Entre las principales áreas de aplicación se encuentran:

- La clasificación. El objetivo es encontrar la clase a la que pertenece cada elemento representado como un patrón para la RNA, consiste en la tarea de asignar cada objeto o elemento a una clase específica. Para cumplir esta tarea se cuenta con un grupo de entrenamiento que consta de patrones de ejemplo que representan todas las clases. Con el conjunto de entrenamiento se deducen las reglas de pertenencia a cada clase y ello crea un clasificador. Las salidas de la red corresponden a las clases y los patrones de entrada se mapearán hacia esas clases, a la salida de la red se muestra a cuál clase pertenece el patrón de entrada. Es importante recalcar que las clases de salida son conocidas previamente y lo importante de este proceso es colocar los patrones de entrada en su clase correspondiente.

  Este tipo de RNA son utilizados en:
  Reconocimiento de caracteres impresos o escritos a mano o de cualquier tipo de imágenes.
  Reconocimiento de voz.
  Clasificación de aplicaciones de préstamo, créditos.
  Análisis de sonares y datos de radar para determinar la naturaleza del origen de la señal.
  Reconocimiento de secuencias de ADN
  Clasificación de patrones de expresión de genes

- Aglomeramiento (clustering). El objetivo es agrupar los objetos o patrones, que resultan ser similares entre sí, en clusters o cúmulos. La RNA se encarga de crear grupos o cúmulos a

partir de un parámetro de similitud, el número de grupos identificados es variable de acuerdo con el grado de sensibilidad del parámetro de similitud dado por una medida de distancia, por lo que los algoritmos de clustering están basados en alguna medida de distancia. Los patrones de entrada son agrupados de manera que las distancias dentro de los cúmulos sean minimizadas y por fuera, entre los cúmulos maximizadas. El número de cúmulos depende del problema, pero debe tender a un óptimo. A diferencia de la clasificación, en el aglomeramiento es común no conocer las clases a las que pertenecen los cúmulos, pues lo importante es el conocimiento de que ciertos patrones son más similares entre sí que con respecto a otros por lo que se agrupan en los cúmulos.

Se utiliza en la comprensión de datos.
Para la realización de mapas fonéticos.
En la monitorización de procesos
Reconocimiento de patrones

- Aproximación de funciones. Estos modelos de RNAs pueden describirse como un mapeo de ciertas funciones de algunos vectores de entrada en salidas numéricas. Es una tarea de aprendizaje mediante la construcción de funciones que generan aproximadamente la misma salida a partir del vector de entrada como un proceso modelado, basado en los datos de entrenamiento. Existe un número infinito de funciones para un conjunto finito de puntos, por lo que es necesario un criterio que decida cuál de las funciones es la más adecuada.

  Algunos problemas industriales y de manufactura involucran estabilizar el comportamiento de un objeto o rastrear el comportamiento de su movimiento, lo que también puede verse como un problema de aproximación de funciones en el cual se desea una función que describa el comportamiento variante en el tiempo del objeto en cuestión.

- Predicción y pronóstico. La predicción puede darse por el conocimiento de leyes y el descubrimiento de regularidades en observaciones al sistema; sin embargo, las leyes son difíciles de descubrir y las regularidades empíricas o periódicas no son evidentes y son frecuentemente enmascaradas por el ruido. Por lo anterior, las RNAs son usadas para obtener razonablemente buenas predicciones en un número de casos. En un alto nivel, el problema de predicción es un caso especial de los problemas de aproximación de funciones, en el que los valores de la función son representados mediante las series de tiempo.
  Son de gran utilidad en la predicción de situaciones en la bolsa de valores y en el pronóstico del clima.

- Sistemas de control. Un control se utiliza para modificar a discreción la relación entrada-salida de un proceso. Cuando una relación es difícil de identificar, una RNA puede aprender a controlar un proceso a partir del conjunto de datos de entrada-salida. Estos sistemas se utilizan en procesos de manufactura.

1.1 La neurona artificial

La neurona artificial es el elemento de procesamiento de una RNA, captura las funciones elementales de una neurona biológica, se representa por medio de un círculo, denominado nodo o celda, y la sinapsis de la neurona biológica mediante una conexión (una flecha) que va o viene hacia o desde otro nodo de la red. La eficiencia de la sinapsis se modela otorgándole una

ponderación a través de un número (positivo, negativo o cero), que recibe el nombre de factor de conexión o peso sináptico, y el axón se representa con una flecha que sale del nodo. El número que acompaña a la flecha modela la salida producida durante el disparo o activación de la neurona; en la figura 1 pueden verse esos elementos de la neurona artificial El modelo de la neurona artificial fue propuesto por McCulloch y Pitts en 1943, ellos propusieron un modelo simple de una neurona como una unidad de umbral binario, [18].

En general la RNA se asemeja al cerebro en dos aspectos básicos [8]:

1) En una RNA el conocimiento es adquirido desde el ambiente externo a través del proceso de aprendizaje, el cual inicia con la presentación de los patrones a la red.
2) Las conexiones interneuronales incluyen factores numéricos conocidos como pesos sinápticos, los cuales son usados para almacenar el conocimiento adquirido.

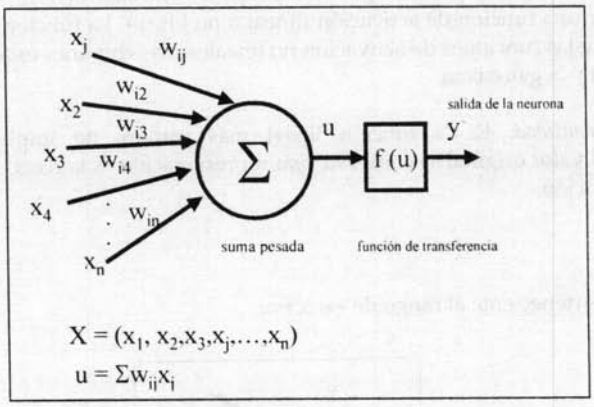


Figura 1 Modelo de una neurona artificial.

La neurona artificial representada en la figura 1 cuenta con una cantidad variable de entradas $(x_1, x_2, \ldots x_n)$ provenientes del exterior y las que componen el vector de entrada X. A su vez, dispone de una única salida y, para transmitir la información hacia el exterior. La señal de salida se calcula en función de las entradas, para lo cual cada una de ellas es afectada por un peso determinado $(w_{i1}, w_{i2}, \ldots, w_{in})$, en dichos pesos se encuentra el conocimiento de la red. Todas las señales de entrada se combinan comúnmente mediante la denominada función sumatoria: $u_i = \Sigma w_{ij} x_i$, para pasar posteriormente por otra función $f(u)$, llamada función de activación o de transferencia que genera el valor final de salida y.

1.1.1 Funciones para el procesamiento en la neurona artificial

El procesamiento de información en la neurona artificial se ejecuta a través de dos funciones: 1) la función red que en la mayor parte de los casos es la sumatoria de los productos

de la matriz de pesos por el vector de entrada, y 2) la función de activación , para la que existen múltiples tipos.

Función base (red).

La función lineal de base (LBF) es una función de tipo hiperplano, de primer orden. El valor de la función es una combinación lineal de las entradas.

$$u_i(\mathbf{w},\mathbf{x}) = \sum_{j=1}^{n} w_{ij} x_j$$

Si consideramos a w como la matriz de pesos, y X el vector de entrada de n dimensión.

Función de salida.

El valor de la red, expresado por la función de entrada, u(w,x), será inmediatamente transformada por una función de activación (lineal o no lineal). La función lineal más simple es la identidad, entre las funciones de activación no lineales más comunes están la función escalón, rampa, sigmoidal y la gaussiana.

Función identidad. Es la función lineal más común, no implica ningún tipo de procesamiento, el valor original se conserva y su representación es la recta. En figura 2 se puede apreciar dicha función.

$$F(x) = x$$

Su valor perteneciente al rango de $-\infty$ a $+\infty$.

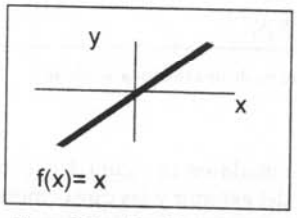


Figura 2 Función identidad.

Función escalón. Esta es la forma más fácil de definir la activación, como se ve en la Figura 3. La función escalón se asocia a neuronas binarias en las cuales cuando la suma de las entradas es mayor o igual que un cierto valor (umbral), la activación es 1; si es menor, la activación es 0 o -1.

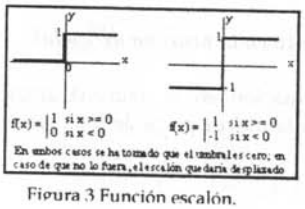


Figura 3 Función escalón.

Función mixta. En las neuronas de función mixta, si la suma de las señales de entrada es menor que un límite inferior, la activación se define como 0 ó - 1, lo que le da la forma que puede distinguirse en la Figura 4. Si dicha suma es mayor o igual que el límite superior, entonces la activación es 1. Si la suma de entrada está comprendida entre ambos límites, entonces la activación se define como una función de interpolación lineal que depende de la suma de las señales de entrada.

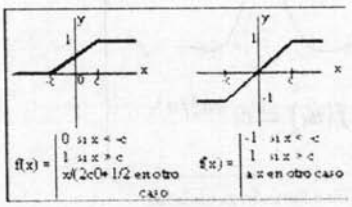


Figura 4 Función mixta.

Función sigmoidal. Cualquier función definida en un intervalo, con un incremento monotónico y que tenga ambos límites superiores e inferiores, puede emular la función de activación de forma satisfactoria. Con la función sigmoidal, presentada en la figura 5, para la mayoría de los valores de entrada, el valor dado por la función es cercano a uno de los valores asintóticos.

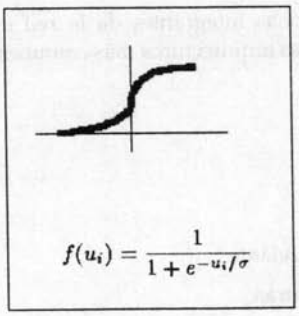


Figura 5 Función sigmoide.

Función gaussiana. Los centros y anchura de estas funciones pueden ser adaptados, mapeos que suelen requerir dos niveles ocultos utilizando funciones de transferencia sigmoidales algunas veces se pueden realizar con un sólo nivel en redes con neuronas de función gaussiana, vista en la figura 6.

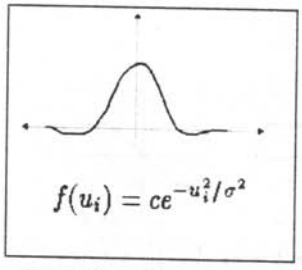


Figura 6 Función gaussiana.

1.2 Arquitecturas de las RNA

Las diversas formas de conexión entre los nodos de una RNA y la organización dada entre sus conjuntos de neuronas dan origen a la arquitectura de la red, también denominada topología o estructura. Este es uno de los factores determinantes para el procesamiento dentro de la red y que está estrechamente relacionado con el número de nodos o neuronas integrantes de la estructura. A la estructura de la RNA se le aplica un algoritmo de aprendizaje para el procesamiento de la información.

Si consideramos que las neuronas integrantes de la red están organizadas formando capas dentro de la red, tenemos entre las arquitecturas más comunes:

- RNA de una sola capa
- RNA multicapa
- RNA recurrentes
- RNA de propagación hacia delante
- RNA de propagación hacia atrás

1.2.1 RNA de una sola capa

Las RNA que presentan esta arquitectura tienen como característica principal que todas sus neuronas se encuentran en una sola capa. A estas neuronas se conectan todas las entradas, y también se encargan de emitir las salidas de la red. En la figura 7 se presenta un ejemplo, en ella los nodos de entrada no son considerados neuronas ya que no realizan ningún tipo de procesamiento con los datos.

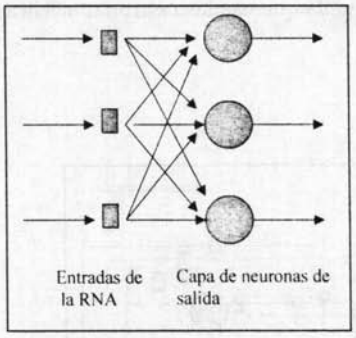


Figura 7 RNA de una sola capa.

1.2.2 RNA multicapa

Estas RNA presentan sus neuronas organizados en más de una capa, una o más de estas capas son denominadas capas ocultas o de neuronas ocultas. Las neuronas ocultas se encuentran entre las capas de entrada y de salida de la RNA. En la figura 8, se presenta una RNA multicapa fuertemente o completamente conectada, pues las neuronas de cada capa están conectadas a todas las neuronas de la capa que le sigue, como sucede entre las neuronas de la capa oculta y de salida.

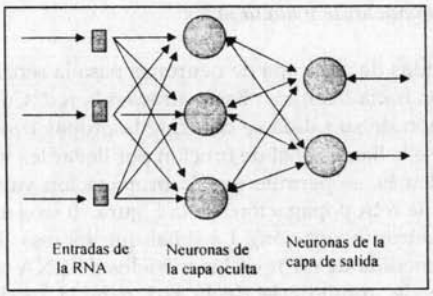


Figura 8 RNA multicapa (dos capas).

1.2.3 RNA recurrentes

Este tipo de RNA se distingue por presentar entre sus conexiones por lo menos un ciclo formado por la salida de una neurona que se dirige como entrada hacia otra neurona de una capa anterior o de su misma capa, o inclusive a la neurona misma. En la figura 9, se presenta un ejemplo de RNA recurrente en la cual se muestra como las salidas de la red se convierten en sus propias entradas; esto implica manejo de tiempos de retraso entre la salida y la entrada actuales de la RNA. La RNA opuesta es aquella que usa las neuronas acíclicas, la cual puede verse tanto en la figura 7 como en la 8.

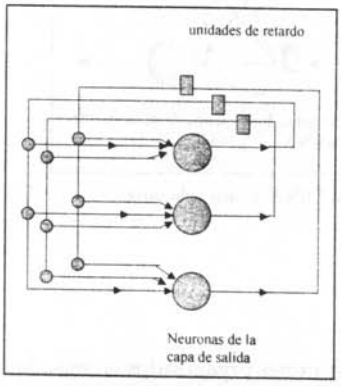


Figura 9 RNA recurrente.

1.2.4 RNA de propagación hacia adelante y hacia atrás

La forma en que las salidas de una capa de neuronas pasa la señal a las neuronas de otra capa, determina la propagación hacia delante o hacia atrás en la red. Cuando la señal va de las entradas de la RNA en dirección de su salida se dice que la propagación es hacia delante; a la señal que va en esta dirección se le llama señal de función por llevar los valores generados en las funciones de activación. Si además, se permite que la propagación vaya de la capa de salida hacia la capa entrada se habla de retropopagación. En la Figura 10 se ejemplifica la propagación hacia adelante y hacia atrás (retropropagación). La señal que regresa desde la salida hacia la capa de entrada representa la medida de error en los modelos de RNA supervisadas. Este error es utilizado por el algoritmo de aprendizaje de la red para la modificación de los pesos sinápticos. Todas las RNAs permiten el paso de la señal de función, pero sólo algunas utilizan el paso de la señal de error. Ambas señales son ejemplificadas especialmente en el perceptrón

multicapa al hacer uso del algoritmo de aprendizaje *backpropagation*, del que hablaremos más adelante.

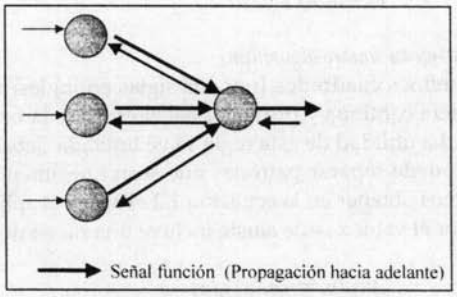


Figura 10 Propagación en la RNA.

1.3 El aprendizaje

El aprendizaje en las RNAs se refiere al proceso por el cual se produce el ajuste de los pesos sinápticos de la red a partir de un proceso de estimulación por el entorno que la rodea. La adaptación de los pesos se realiza siguiendo una regla de aprendizaje. Estas reglas de aprendizaje son clasificadas en dos categorías de aprendizaje según sea la forma en que adquieren el conocimiento del medio: supervisada y no supervisada.

El aprendizaje está definido por tres pasos fundamentales: [8]

1. La RNA es estimulada por un medio ambiente.
2. La RNA experimenta cambios en los pesos como un resultado de esta estimulación.
3. La RNA responde en una forma distinta al medio ambiente por los cambios que ha sufrido.

1.3.1 Reglas de aprendizaje

Las reglas de aprendizaje utilizadas por las RNAs son procedimientos que les permiten realizar cambios en los pesos con el objetivo de lograr que la RNA ejecute la tarea asignada de manera correcta. A este procedimiento también se le denomina algoritmo de entrenamiento. Estas reglas se dividen en dos categorías: de aprendizaje supervisado y de aprendizaje no supervisado, como veremos en 3.2.

1.3.1.1 Aprendizaje por corrección de error

La RNA es entrenada hasta producir los valores de salida que corresponden a los patrones de entrada. El error o la diferencia entre el valor esperado y el producido por la red se utiliza para modificar los pesos con la finalidad de ir disminuyendo gradualmente ese error. En la ecuación 1.1 se muestra la obtención de la señal de error e_k, de la neurona k que produce la salida y_k, en el instante de tiempo n. Esa salida obtenida es comparada con la salida esperada d_k.

El algoritmo de retropropagación del perceptrón multicapa utiliza esta regla. Widrow y Hoff desarrollaron una regla denominada Delta para aplicar la regla en los modelos neuronales Adaline y Madaline [20].

$$e_k = d_k(n) - y_k(n) \tag{1.1}$$

Regla Delta o LMS (least-mean square algorithm)

La regla delta o de mínimos cuadrados (por sus siglas en inglés) realiza actualizaciones en los pesos de la red de manera continua y proporcional al error de la neurona con la finalidad de ir disminuyendo ese error. La utilidad de esta regla se ve limitada debido a que se trata de un proceso lineal, es decir, solo puede separar patrones que sean linealmente independientes. Por medio de la regla delta podemos obtener en la ecuación 1.2 el ajuste Δ aplicado al peso sináptico w_{kj} de la neurona k excitada por el valor x_j, este ajuste incluye una razón de aprendizaje η.

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \tag{1.2}$$

El aprendizaje por error finaliza cuando se ha obtenido el resultado de salida esperado o hasta que los valores de los pesos se estabilizan. Esta regla, debido a que se conoce con anterioridad el resultado esperado como salida del procesamiento neuronal, se encuentra en la categoría de aprendizaje supervisado, del que hablaremos más adelante.

1.3.1.2 Aprendizaje de Hebb o Hebbiano

Se deriva del postulado acerca del aprendizaje de D. Hebb, quien sostuvo que las conexiones entre las neuronas cambian conforme el cerebro aprende nuevas actividades. Su postulado consiste en *si un axón pre-sináptico causa la activación de cierta neurona post-sináptica, la eficacia de la sinapsis que las relaciona se refuerza* [18], es decir, si dos neuronas biológicas participan en cierta actividad cerebral, su interconexión se fortalece. En este tipo de aprendizaje el ajuste de los pesos sinápticos es en función de las respuestas pre-sinápticas y post-sinápticas, es decir, de la entrada y la salida de la neurona. En la ecuación 1.3, puede apreciarse el ajuste Δ a realizar al peso w_{kj} a partir de las señales pre-sináptica x_j y post-sinápticas y_k de la neurona k.

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) \tag{1.3}$$

Esta regla pertenece a la categoría no supervisada porque no requiere ninguna información previa acerca de la salida esperada.

1.3.1.3 Regla de aprendizaje competitivo

En este tipo de aprendizaje las neuronas de salida compiten entre ellas por ganar y activarse ya que sólo una neurona será activada a la vez, compiten por la oportunidad de aprender.

Este aprendizaje se basa en los siguientes postulados [8]:

1) Existe un conjunto de neuronas casi idénticas, su diferencia radica en que las conexiones distribuidas aleatoriamente responden de forma distinta al vector de entradas.
2) Hay un mecanismo para limitar el refuerzo de los pesos sinápticos relacionados con cada neurona.
3) Las neuronas compiten para dar una respuesta adecuada a las entradas dadas, de forma que sólo una neurona de salida puede activarse a la vez. La neurona que gana la competencia se conoce como "la neurona que se lleva todo" (*the winner takes all*).

Las RNAs que adoptan este tipo de aprendizaje cuentan sólo con la señal de propagación hacia adelante al ser excitadas las neuronas por las entradas y se incluye también la conexión de forma colateral a las neuronas de su misma capa para inhibirlas y limitarlas. El cambio de peso sináptico en esta regla se muestra en 1.4, si una neurona no responde adecuadamente a una entrada en particular no hay aprendizaje en esa neurona (es decir, no se modifica su vector de pesos).

$$\Delta w_{kj}(n) = \begin{cases} \eta(x_j(n) - w_{kj}(n)) & \text{Si } k \text{ es la ganadora} \\ 0 & \text{Si } k \text{ no es la ganadora} \end{cases} \qquad (1.4)$$

Con η como razón de aprendizaje.

Esta regla de aprendizaje es aplicada en RNAs de aprendizaje no supervisado.

1.3.2 Categorías de aprendizaje

La forma en que la red neuronal adquiere la información proveniente de su medio ambiente, es decir, si existe o no un mecanismo que proporcione a la red el conocimiento del medio, distingue a las RNA en dos categorías de aprendizaje:

- Aprendizaje supervisado
- Aprendizaje no supervisado

1.3.2.1 Aprendizaje supervisado

Este aprendizaje es conocido también como aprendizaje con maestro, el maestro es quien tiene el conocimiento del ambiente y que es desconocido por la RNA, el maestro trata de enseñarlo a la red por medio de un conjunto de ejemplos de entrada-salida. En la figura 11, se ilustra los actores que intervienen en el aprendizaje supervisado, así como su procedimiento. La RNA como un sistema de aprendizaje obtiene las entradas del medio para procesarlas y dar una respuesta que será comparada con la respuesta esperada proporcionada por el maestro, para de allí obtener una señal de error que tiende a disminuir, como en la ecuación (1.1). La RNA se

retroalimenta con la señal de error para lograr la salida esperada dada por el maestro para el patrón de entrada.

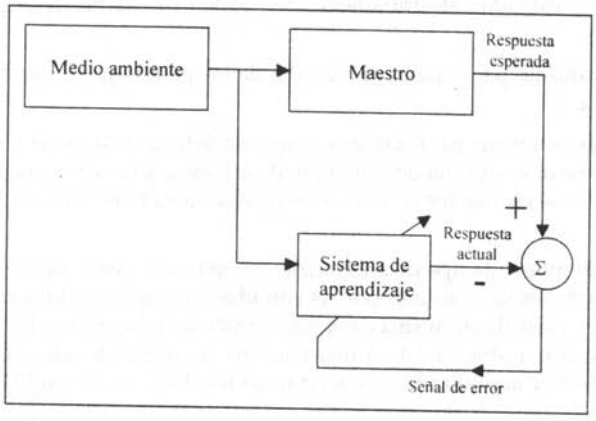


Figura 11 Aprendizaje supervisado. [8]

Con su conocimiento, el maestro, es capaz de proveer a la red de la respuesta deseada para el entrenamiento de un vector de entrada dado. Esta respuesta deseada representa la acción óptima que ejecutará la RNA. Durante el entrenamiento, el conocimiento del medio ambiente con que cuenta el maestro se tranfiere a la red. Cuando esto se logra, el maestro deja a la red para que actúe con el ambiente completamente sola. A este esquema de aprendizaje pertenecen la regla de aprendizaje de corrección de error y la regla delta, mencionadas anteriormente.

Los modelos de RNAs que utilizan estas reglas de aprendizaje y que por lo tanto son de tipo supervisado son el perceptrón simple y multicapa, la ADALINE y MADELINE [20]. A estos modelos neuronales es necesario entrenarlos antes de su funcionamiento, mediante un conjunto de vectores elegidos para ello. Todos los patrones de entrada están acompañados por la salida esperada, el entrenamiento finaliza cuando se ha obtenido la salida esperada para las entradas presentadas, entonces se dice que la red ha aprendido.

1.3.2.1.1 El perceptrón simple

El perceptrón es la forma más simple de RNA utilizada para la clasificación de patrones que son linealmente separables [8]. El perceptrón puede consistir en una simple neurona con sus pesos sinápticos que deberán ajustarse, además cuenta con otro elemento denominado bias. Como puede verse en la figura 12 la salida será 1 ó 0 para indicar si el patrón de entrada pertenece o no a la clase que representa la neurona. El bias es una entrada constante utilizada en muchos modelos neuronales para llevar a la red a que converja. En su procesamiento utiliza un algoritmo de aprendizaje desarrollado por Rosenblatt [21]. El perceptrón de capa simple no es

adecuado para situaciones en las que aparecen más de dos clases y/o fronteras de decisión que no son lineales. La regla de aprendizaje delta es aplicada al perceptrón simple para implementarlo como un filtro adaptativo utilizado en los campos de sistemas de comunicación, sistemas de control, radares, sismología y sistemas de ingeniería biomédica [8].

El perceptrón presentado por Rosenblatt recibió fuertes críticas por parte de Minsky y Papert, quienes indicaron que tenía grandes limitaciones, probaron que este perceptrón era incapaz de generalizar con sus ejemplos aprendidos localmente. Por generalizar se entiende la capacidad de la RNA de producir salidas correctas (o al menos aceptables) ante patrones no incluidos en los patrones de entrada usados en la fase de entrenamiento.

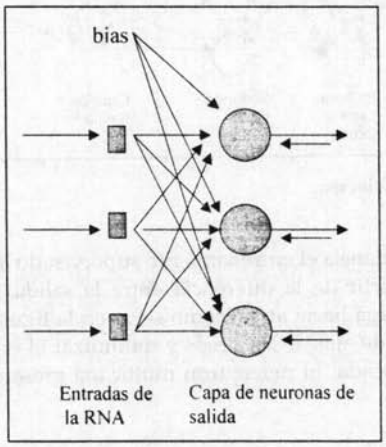


Figura 12 Red de perceptrones simples.

El modelo de Kim [15], presentado en la introducción, se basa en esta forma de perceptrón. Es una red muy básica para encontrar si un gene está o no influenciado por otro.En este caso también encontramos una separación lineal con dos clases de salidas, lo que limita el alcance de la propuesta.

1.3.2.1.2 Perceptrón multicapa

El perceptrón multicapa es una respuesta a las limitaciones encontradas en el perceptrón simple. Consiste de múltiples capas de neuronas que presentan entre ellas conectividad completa, es decir, las neuronas de una capa están conectadas con todas las neuronas de la capa anterior [8]. En la Figura 13 se presenta un modelo de perceptrón multicapa, con dos capas ocultas. El algoritmo de aprendizaje que utiliza es conocido como b*ackpropagation*, de

retropropagación o propagación hacia atrás, basado también en la regla delta. Este modelo de RNA también se conoce con el nombre de *backpropagation* [18].

Las características básicas de RNA *backpropagation* son: [8]

1) Utiliza una función de activación no lineal.
2) La red está integrada por una o más capas ocultas, cuyas neuronas no están en contacto con el exterior ni por las entradas ni por las salidas.
3) La red neuronal exhibe un alto grado de conectividad a través de sus sinapsis.

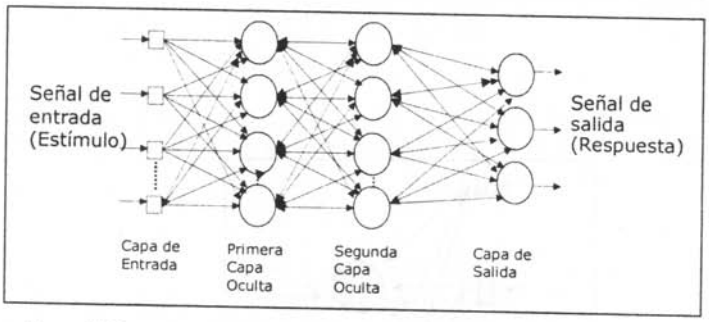


Figura 13 Perceptrón multicapa.

El perceptrón multicapa maneja el entrenamiento supervisado al utilizar la señal de error que se origina en la salida a partir de la diferencia entre la salida esperada y la salida real obtenida. La señal de error propaga hacia atrás, como se ve en la figura 8, pasa capa por capa a través de la red con la finalidad de ajustar los pesos y minimizar el error cometido en la salida obtenida respecto a la salida deseada. El perceptrón multicapa presenta una alta capacidad de generalización.

Las neuronas ocultas dentro de un perceptrón multicapa tienen la función de:

- Procesar la función de activación de tipo no lineal, mandando la señal de función hacia la capa de salida de la red.
- Calcular un estimado del gradiente necesario para regresar la señal a través de la red.

El Perceptrón multicapa ha sido aplicado en procesos de clasificación de patrones, aproximación de funciones, análisis de escenas. Una sola capa oculta en los perceptrones es suficiente para lograr un alto grado de generalización. La desventaja de estos modelos es que el tiempo de computación puede ser muy alto.

1.3.2.2 *Aprendizaje no supervisado*

En este aprendizaje, a diferencia del anterior, no participa un maestro que supervise el proceso de aprendizaje, no hay ejemplos conocidos de los patrones de entrada presentados a la

red para que exista un maestro que se los muestre. En la figura 14 se puede ver como el medio ambiente es el único que participa al proporcionar entradas a la RNA y no hay un maestro que indique a la red si está operando de manera correcta o incorrecta ya que no hay salidas conocidas, la RNA tiene que descubrir por sí misma regularidades, correlaciones o semejanzas entre los datos de entrada e incorporarlos a su estructura interna de conexiones [18].

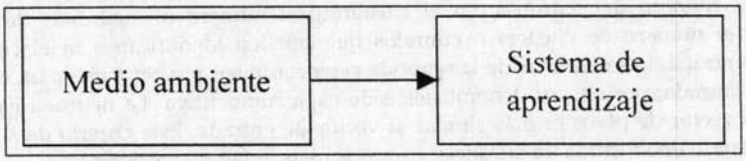


Figura 14 Aprendizaje no supervisado. [8]

Generalmente los modelos de RNA no supervisados son de una sola capa y con algoritmos sencillos y más rápidos que el de *backpropagation*. Entre las tareas que puede realizar una RNA no supervisada, de acuerdo a su arquitectura son: reconocimiento de patrones, mapas de rasgos, análisis de componentes especiales, codificación, cuantificación vectorial, clustering o agrupamientos, etc. [18]

1.3.2.2.1 Mapas auto-organizados (SOM)

Las RNAs de mapa auto-organizado son las más representativas del aprendizaje no supervisado, este modelo fue propuesto por Kohonen [16]. Hace uso de los principios de aprendizaje competitivo puesto que se basan en el principio de que solamente una neurona (o un grupo de neuronas vecinas) gane y pueda ser activada. El aprendizaje competitivo requiere conexiones inhibitorias entre los nodos laterales, la neurona más activa consigue inhibir a las demás lo que hace que sea la única en permanecer activa y con ello se convierte en la ganadora. Lo anterior implica también que cada nodo tenga conexiones excitatorias hacia un pequeño número de nodos en la red, sus vecinas. La topología se especifica en términos de la vecindad entre los nodos.

Esta red es auto-organizada en el sentido de que es entrenada para representar en sus vectores de pesos las características del vector de entrada, con la relación de vecindad establecida por la proximidad en un espacio euclidiano [21].

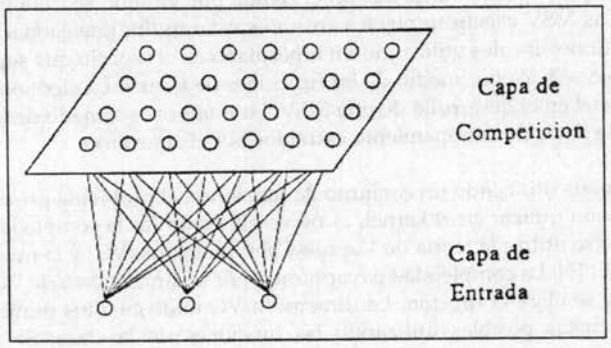


Figura 15 Mapa auto-organizado, bidimensional.

La arquitectura de la red, como puede verse en la figura 15, consiste de dos capas interconectadas de modo unidireccional con un vector de pesos que va de la capa de entrada a la capa de competencia. La primera capa con m neuronas de entrada y la segunda capa que contiene n neuronas de salida, se encuentran unidas por los vectores de pesos. Esta segunda capa toma una forma geométrica, un rectángulo o un hexágono generalmente, la dimensión determina su número de neuronas, en el clustering el número de neuronas de esta capa corresponde al número de clusters o cúmulos que pueden identificarse en el conjunto de patrones de entrada. Las neuronas de la segunda capa compiten por activarse para representar el vector de entrada, por ello su denominación de capa competitiva. La neurona ganadora es aquella cuyo vector de pesos es más similar al vector de entrada. Este criterio de similitud se obtiene utilizando una medida de distancia, como las mencionadas anteriormente.

Cada vez que se presenta un vector de entrada se realiza la actualización de los pesos de la neurona ganadora y de las vecinas y el nuevo peso se obtiene por la ecuación (1.5):

$$w_j(n+1) = w_j(n) + \eta(n)\, h_{j,i(x)}(n)(x(n)-w_j(n)) \qquad (1.5)$$

donde $w_j(n+1)$ es el peso w de la neurona j en el tiempo $n+1$,
$\eta(n)$ es la razón de aprendizaje en el tiempo n
$h_{j,i(x)}(n)$ es la función del vecindario alrededor de la neurona ganadora $i(x)$

Los parámetros $\eta(n)$ y $h_{j,i(x)}(n)$ van decreciendo dinámicamente durante la ejecución del algoritmo. Después de miles de iteraciones, que generalmente se hacen con este algoritmo, se espera que las neuronas vecinas a la neurona representativa del patrón de entrada presentado sean cada vez menos hasta llegar a uno o cero.

El clustering realizado a través de RNA se hace utilizando principalmente este modelo como en el caso del reconocimiento de patrones de expresión de genes. También existen otras aplicaciones como clasificación de patrones, cuantificación vectorial, reducción de dimensiones, extracción de rasgos y monitorización de procesos.

1.3.2.2.2 Máquina de soporte vectorial (MSV)

Es una RNA de propagación hacia adelante, creada por Vapnik, se trata básicamente de una máquina lineal. Las MSV clasifican objetos en un espacio multidimensional en dos clases. Extienden los clasificadores lineales utilizando un hiperplano en el espacio que separa el espacio en dos partes. El mapeo se hace por medio de las funciones de kernel. La elección de la función kernel es la parte esencial en el desarrollo de una MSV. Su nombre es tomado de los vectores de soporte, un conjunto de datos de entrenamiento extraídos por el algoritmo.

La red es entrenada utilizando un conjunto de funciones, el problema principal radica en la elección de qué función utilizar en el kernel, es necesario controlar la complejidad de la clase de funciones y para eso se utiliza la teoría de Vapnik-Chervonenkins (VC) y la minimización del riesgo estructural (MRE) [9]. La complejidad es capturada por la dimensión h de VC de una clase de funciones de la que se elige la función. La dimensión VC mide cuantos puntos pueden ser separados por las etiquetas posibles utilizando las funciones de la clase. Se construye un

conjunto de todas las clases de funciones que pueden hacerse $F_1 \subset F_2, \ldots \subset F_k$, sin decrecer la dimensión VC, $(f_1, f_2, \ldots f_k)$ es la MRE en la función de clase F_i. MRE elige la clase de funciones F_i y la función f_i, la mejor cuya generalización de error es minimizada. Cuando se utiliza el hiperplano para separar el ejemplo de entrenamiento, se puede pensar que VC es un separador de clases de hiperplanos, VC es el margen. El margen (ρ) es la distancia mínima de un ejemplo hacia la superficie de decisión. El margen es dado por la distancia de separación de dos puntos de diferentes clases hacia la superficie. Como vemos en la figura 16, el objetivo de la MSV es encontrar el hiperplano cuyo margen de separación (ρ) sea maximizado, es decir, hallar el hiperplano óptimo [8].

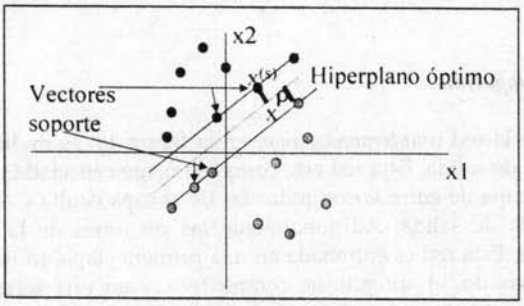


Figura 16 Hiperplano para separación lineal.

Las funciones más utilizadas para los kernel son la gaussina y la sigmoide (presentadas en 1.1.1 de este capítulo). Por medio de los pesos se busca el hiperplano óptimo, definido por la función:

$$w_o x + b_o = 0 \qquad (1.6)$$

donde x es el vector de entrada, w_o y b_o son los valores óptimos del vector de pesos y del bías, En la figura 17 podemos ver la arquitectura de este tipo de red, donde las neuronas de la capa oculta representan los kernel.

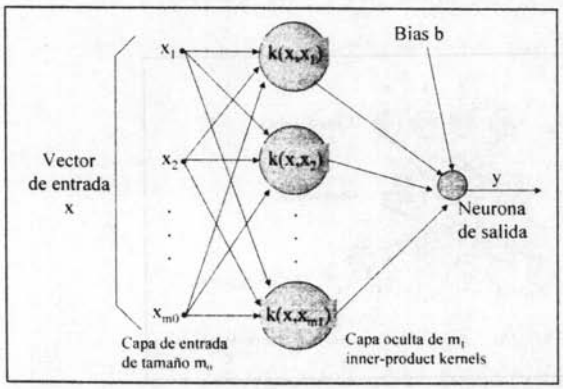


Figura 17 Arquitectura de una MSV.

Con la MSV se pueden construir otros tipos de RNAs, como redes de función de base-radial y perceptrones con una capa oculta. La MSV ayuda a determinar automáticamente el número de neuronas ocultas requeridas para esos modelos neuronales de acuerdo con el conjunto de entrenamiento dado, ese número es dado por el número de vectores de soporte encontrados. Puede utilizarse tanto para aprendizaje supervisado como no supervisado. Este modelo es aplicado principalmente en clasificación de patrones y problemas de regresión lineal. Existen algunas aplicaciones de este modelo neuronal en la clasificación de patrones de expresión de genes.

1.3.2.2.3 RNA counterpropagation

La arquitectura de la red *counterpropagation*, en la figura 18, es multicapa: una capa de entrada, una oculta y una de salida. Esta red está completamente conectada entre sus neuronas, todas las neuronas de la capa de entrada con todas las de la capa oculta y todas las de la capa oculta con las de la capa de salida. Adicionalmente las neuronas de la capa oculta están interconectadas entre ellas. Esta red es entrenada en una primera etapa utilizando un algoritmo de aprendizaje no supervisado, el aprendizaje competitivo, y en una segunda etapa con un aprendizaje supervisado mediante una variante de la regla delta propuesta por Grossberg [21]. La capa oculta es utilizada con la finalidad de dividir el conjunto de entrenamiento en cúmulos de patrones de entrada similares. La combinación de estos dos tipos de aprendizaje hace que su convergencia sea más rápida que en el caso de las *backpropagation*; sin embargo su generalización no es buena [18].

Existen otros modelos de RNA, tanto supervisados como no supervisado, de los que listamos algunos en la tabla 1, donde se muestra que una de las tareas más tratadas por todos los modelos es la clasificación de patrones. No obstante, los más utilizados en el problema de clasificación de patrones de expresión de genes son los descritos en los puntos anteriores. De la red de mapa auto-organizado destaca el trabajo de Tamayo et al. [34], del perceptrón el de Kim et al.[15] y de la MSV el de Brown et al. [3] (expuestos en la introducción), entre otros. Nosotros presentamos otro modelo de RNA basado en el trabajo de Sánchez et al. [29]. Este modelo combina las principales características del clustering no jerárquico y aglomerativo, redes neurales SOM con un esquema de aprendizaje no supervisado.

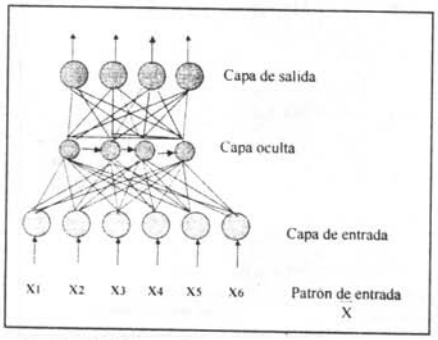


Figura 18 RNA counterpropagation.

El siguiente modelo a presentar es el de Sánchez et al. [29] en donde existe una clasificación no lineal, y una libre dinámica de la red. Su estructura es la necesaria para la clasificación de los patrones presentados, ya que crea los cúmulos necesarios compuestos de los centroides también necesarios (las neuronas que integran los cúmulos). Esta característica es semejante a la de la MSV que cuenta con la estructura de red necesaria para su funcionamiento. En cuanto a la red de *counterpropagation*, el modelo neuronal de Sánchez et al. es semejante en la idea de una estructura de tres capas, una de entrada, una oculta utilizada para crear los cúmulos y una de salida, como puede verse en las figuras 18 y 19. Obsérvese que las conexiones entre las neuronas no son iguales, ya que para nuestro modelo de red no es necesario que todas las neuronas de la capa oculta se conecten con todas las neuronas de la capa de salida, sino sólo entre aquellas neuronas ocultas y su clase de salida a la que pertenecen, por lo que no existe competitividad entre las neuronas de la capa de salida.

Es importante mencionar que en el modelo de Sánchez et al. no se utiliza ninguno de los algoritmos de aprendizaje utilizados por alguna de las RNAs mencionadas sino uno nuevo que requiere de la existencia de ciertos parámetros que proporcionen información acerca de las neuronas ocultas, a las que denominaremos centroides y que en conjunto integran los cúmulos o clusters. Esos parámetros son el radio del centroide y el coeficiente de vitalidad. En la siguiente sección expondremos el trabajo de Sánchez et al. [29] para presentar a partir del siguiente capítulo las características específicas del modelo propuesto.

Categoría	*Regla de aprendizaje*	*Red neuronal*	*Tarea*
Supervisado	Corrección de error	Perceptrón simple y multicapa	Clasificación de patrones Aproximación de funciones Predicción Control
	Boltzmann	Recurrente	Clasificación de patrones
	Hebbian	Feedforward	Análisis de datos Clasificación de patrones
	Competitivo	ART	Clasificación de patrones Categorización
No supervisado	Corrección de error	Feedforward	Análisis de datos
	Hebbian	Feedforward	Análisis de datos Comprensión de datos
		Red de Hopfield	Memoria asociativa
	Competitivo	Vector de cuantización	Categorización Análisis de datos
		SOM de Kohonen	Categorización Análisis de datos
		ART	Categorización

Tabla 1. RNA supervisada y no supervisadas.

1.4 El modelo de Sánchez *et al.*

El modelo neuronal de Sánchez et. al. [29], ha sido desarrollado para abordar el problema del aprendizaje de patrones pertenecientes a clases que no son linealmente separables y cuya distribución varía en el tiempo, y para patrones pertenecientes a medio ambientes estacionarios y no estacionarios. La RNA aprende a clasificar patrones de una forma supervisada utilizando para ello un conjunto de neuronas ocultas denominadas centroides que van adaptándose. El aprendizaje envuelve la creación de nuevos centroides, la actualización de centroides existentes y sus parámetros. Entre los parámetros de los centroides se encuentran el coeficiente de vitalidad y el radio de cada centroide, los cuales dan información acerca del centroide para tomar decisiones acerca de los patrones que logra clasificar, así como de la permanencia del propio centroide.

El modelo neuronal se caracteriza por las siguientes propiedades:

- Aprendizaje supervisado
- Aprendizaje en tiempo real
- Existencia de un radio para definir las regiones de clasificación de los centroides
- Adaptación de los centroides a través de sus parámetros
- Incremento controlado de la cantidad de centroides (utilizando centroides y fusión de centroides)
- Garantía de la convergencia
- Aprendizaje de patrones pertenecientes a medios estacionarios y no estacionarios

1.4.1 Los parámetros de la RNA

Esta red utiliza diversos parámetros para controlar su funcionamiento, algunos son dados por el usuario, otros son derivados de los patrones de entrada. De acuerdo a la evolución de la RNA algunos parámetros se modifican en el tiempo:

X: patrón de entrada
R_p: radio del centroide
P: centroide
V_p: coeficiente de vitalidad del centroide p
β_v: constante de decisión de eliminación para el mecanismo de olvido
g_v: constante de decisión de eliminación considerando el radio
M: total de centroides
PC: clase del patrón de entrada
WC: clase del centroide ganador
R_o: radio inicial
d(X,P): distancia euclidiana entre el patrón X y el centroide P

1.4.2 Algoritmo de la RNA

Inicialmente, esta RNA no tiene memoria, es decir, los cúmulos no se encuentran formados por lo que la capa oculta de la red no existe sino que con la entrada de los patrones se produce la creación dinámica de centroides y con ello de cúmulos o clusters.

1) Calcular la distancia $d(X,P_i)$ y $V_{pi} = V_{pi}/\alpha$, para $1<i<M$. Determinar el centroide ganador P con radio R_p. Denotar WC como la clase del centroide P.
2) Si PC = "No clasificado" entonces el patrón X es asignado a la clase WC y el proceso finaliza.
3) Si WC = "No clasificado" entonces un nuevo centroide es creado asociado al patrón X. Ir al paso 5.
4) Si WC ≠ "No clasificado" entonces:

   4.1) Si $d(X,P) < R_p$ y WC = PC, entonces actualizar el centroide P haciendo $V_p=V_p*a^2$.Ir a paso 5.

   4.2) Si $d(X,P) < R_p$ y WC ≠ PC, entonces reducir R_p. Ir al paso 5.

   4.3) Si $d(X,P) \geq R_p$ entonces:

   Si es posible, fusionar P y X, actualizar $V_p = V_p*\alpha^2$. Ir al paso 5.

   Si no es posible fusionar P y X, entonces crear un nuevo centroide asociado a X. Ir al paso 5.

5) Eliminar centroides satisfaciendo la condición: $V_p \leq \beta_v$ o $R_p \leq g_v$.

El patrón X es asignado a la clase WC.

Procesos básicos asociados a la creación y actualización de centroides:

- Creación de un nuevo centroide
- Actualización de un centroide existente (incluyendo su radio)
- Verificación de la intersección de las regiones de clasificación
- Fusión de un patrón con un centroide
- Reducción del radio
- Selección del ganador
- Principios generales que aseguren la convergencia de la red

El modelo neuronal está compuesto por tres capas, figura 19. La cantidad de neuronas en la capa de entrada es determinada por la dimensión del vector de cada patrón de entrada. La capa oculta contiene los centroides que han sido creados durante el proceso de evolución de la red para lograr la clasificación de los patrones de entrada presentados.

Inicialmente la capa oculta no contiene neuronas y cada neurona adicionada a esta capa representa un nuevo centroide para la existencia de una clase y la capa de salida contiene tantas neuronas como clases existentes. Cada neurona de la capa oculta recibe entradas desde todas las neuronas de la capa de entrada. Cada neurona de la capa de salida recibe entradas desde un subconjunto de neuronas de la capa oculta. Cada uno de estos subconjuntos representa un grupo de centroides para una clase y definen a un cúmulo o cluster. Todos los centroides del cúmulo se conectan con la única neurona de su clase localizada en la capa de salida.

La RNA construye los centroides, los cuales son combinaciones no lineales de los patrones de entrada. Por otra parte, cada centroide tiene asociado un radio (o umbral) el cual determina el conjunto de patrones que el centroide puede clasificar. Las neuronas de la capa oculta también se caracterizan por un coeficiente de vitalidad. Este coeficiente representa la probabilidad de que la neurona se encuentre activa un número mínimo de veces en un número dado de presentaciones de los patrones.

Nuestro trabajo de clasificación de patrones de expresión de genes se basará en el modelo de red neuronal propuesto para la clasificación en tiempo real. Este modelo será modificado para lograr tanto aprendizaje supervisado como no supervisado, clasificación y clustering.

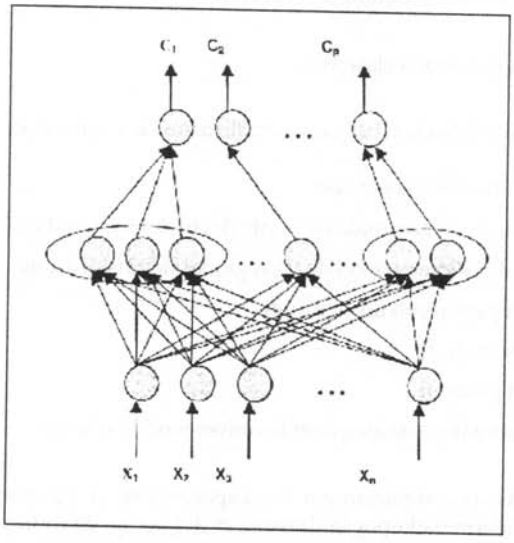


Figura 19 Modelo de la red neuronal de Sánchez *et al.* [29]

1.5. Adaptación del modelo de Sánchez et. al. para la clasificación de patrones de expresión de genes

Después de lo que hemos revisado en este capítulo podemos decir que para resolver el problema de clasificación de patrones de expresión la aplicación de un perceptrón multicapa sería complejo pues la clasificación de genes no lineal requiere un tamaño del vector de entrada muy grande, lo cual complicaría mucho los cálculos realizados dentro por la red, aún más, el problema de clustering no puede ser resuelto por un perceptrón y una de las técnicas más utilizadas con datos de expresión de genes es el clustering por el desconocimiento de los mismos.

El modelo SOM realiza clustering, pero su inconveniente es que al inicio del algoritmo debemos proporcionar el número de cúmulos o clusters esperado ya que con éste se formará la figura geométrica de la capa superior o de salida, lo que hace necesario aplicar antes algún método como el análisis de componentes principales para obtener una aproximación de las clases y cúmulos, y la estructura de la RNA queda limitada para ser utilizada posteriormente con nuevos patrones desconocidos. Con la RNA de MSV el modelo de Sánchez et al. se asemeja en que la red es capaz de crear la estructura que necesita para su funcionamiento adecuado, la complicación está principalmente en encontrar una función para el *kernel* que permita realizar el clustering deseado. En cuanto a la red de *counterpropagation* es semejante en la idea de una estructura de tres capas, una de entrada, una oculta utilizada para crear los cúmulos y una de salida, aunque las conexiones y algoritmo de entrenamiento es distinto pues la generalización alcanzada con el *counterpropagation* no es exitosa. En este sentido, el modelo de Sánchez et al. puede dar más información ya que la capa oculta cuenta con información obtenida directamente de las características propias de los patrones presentados.

Existen algunas propiedades consideradas en el modelo de Sánchez et al. [29] que difieren con respecto a nuestro problema a resolver de clasificación de genes. De acuerdo a lo anterior, el modelo base será adaptado para que responda a las necesidades planteadas por los patrones de expresión de genes. Entre las diferencias que debemos considerar es que los patrones de expresión de genes no se modifican durante el entrenamiento de la red sino que el conjunto de genes a clasificar presentan los mismos valores, son estáticos, una vez tomados de los microarreglos ya no se modifican, por lo que no se aplica la característica inicial de los patrones a introducir en el modelo de Sánchez et al. [29] en cuanto a que la distribución de patrones varía en tiempo real. Esto sugiere una modificación para la eliminación de los centroides o neuronas ocultas. La red permitirá el uso de diversas medidas de distancia: distancia euclidiana, distancia de Hamilton y Correlación de Pearson (expuestas previamente en la introducción) para dar flexibilidad al usuario a que elija la medida de distancia que sea más adecuada para el objetivo de su estudio biológico.

La función de transferencia de las neuronas de salida permitirá que sólo aquella que clasifica el patrón emita como salida 1, mientras que las demás 0. Además los patrones de entrada pueden ser transformados en binarios o ternarios, para lo cual la función de activación de las neuronas de entrada tendrán la función adecuada que convierta del valor en números reales al nuevo valor. Con las modificaciones hechas, la red podrá dar más información ya que en la capa oculta cuenta con información obtenida directamente de las características propias de los patrones presentados que el usuario podrá utilizar para el análisis de información de genes, proveniente principalmente de microarreglos.

En los siguientes capítulos se presenta de forma detallada la propuesta de modelo neuronal que desarrollamos para la clasificación de expresión de genes por medio de clustering supervisado y no supervisado. En primer lugar presentamos el modelo supervisado con el que pretendemos verificar y adaptar el modelo de red ante los patrones de expresión de genes que están acompañados por sus clases para posteriormente resolver los casos de clasificación en que no se conocen a priori las clases de los patrones, lo que representa la finalidad del modelo propuesto.

Capítulo 2.
Modelo supervisado para clasificación de patrones de expresión de genes

Las RNAs representan una de las opciones más utilizadas recientemente para el reconocimiento de patrones. Esto se debe a que pueden proporcionar una excelente aproximación a la clasificación real obtenida mediante métodos estadísticos. La ventaja sobre los métodos estadísticos es que no requieren de datos exactos o completos, ya que la red puede completar los datos faltantes o incompletos. Cuando se usa una RNA es posible combinar diversos métodos estadísticos. Por lo general las RNAs toman como base un método estadístico y lo combinan con otras técnicas computacionales por lo que mejoran el desempeño de la red con respecto a los métodos estadísticos. Ejemplo de lo anterior son los modelos de ajuste integrados por una RNA tipo *perceptrón* combinados con métodos de regresión, así como el uso del análisis de componentes principales para la reducción de dimensionalidad requerida en las redes de Kohonen [16].

Como se mencionó en la introducción que los genes forman en su conjunto el genoma de un ser vivo y por medio de las diferentes expresiones de los genes en un momento determinado se llevan a cabo todos los procesos necesarios para el funcionamiento del organismo de cada individuo, de acuerdo a las proteínas correspondientes para cada gene. Para las ciencias biológicas es muy importante la identificación de los genes que se expresan en determinadas circunstancias, en especial para la medicina y la farmacéutica, pues la expresión de genes, principalmente en grupos, interfiere directamente con el desarrollo de las enfermedades humanas que son la materia de estudio de estas dos áreas de la biología.

En este capítulo, presentamos el modelo de RNA que proponemos para el reconocimiento de patrones, específicamente para la clasificación y reconocimiento de patrones de expresión de genes de manera supervisada. Partiendo del conocimiento previo de las diferentes clases o categorías a las que pertenecen conjuntos de genes ya analizados, la supervisión como característica de una primera versión del modelo neuronal nos permitirá estimar los valores de los parámetros con los cuales deberá funcionar adecuadamente la red. Es decir, dotar al modelo neuronal no supervisado de cierto conocimiento biológico relevante para efectuar la tarea de categorización de patrones de expresión de genes.

El modelo neuronal supervisado constituye una adaptación del modelo propuesto por Sánchez et al. [29]. El modelo está basado en la formación de clusters de forma supervisada, para lo cual se requiere a priori de muestras integradas por los pares (patrón de expresión de genes, categoría a la que pertenece) [21]. Con los clusters iniciales formados por esta red supervisada más adelante podrá hacerse la clasificación de patrones que arriban a la red sin indicar la clase a la que pertenecen. Por lo tanto, el objetivo de este capítulo es obtener una red neuronal inicial supervisada que clasifique correctamente patrones de expresión de genes presentados a la red y que, además, muestre las características topológicas de los clusters formados, así como los valores adecuados de los parámetros de la red. Una vez logrado este objetivo, el próximo paso será modificar el modelo neuronal para permitir el arribo de patrones de entrada que no especifican la clase a la que pertenecen. La red neuronal recibirá como entrada los patrones de expresión de genes, representados por vectores cuya dimensión corresponderá al número de

experimentos realizados sobre esos genes. En la introducción se presentó de manera más amplia la integración de un vector de expresión de gene tomado de un microarreglo.

Recordemos que una RNA para clasificación por medio de clustering puede funcionar de forma supervisada cuando todos los patrones presentados como entradas a la red cuentan con la información referente a qué clase pertenecen. Esta idea se puede apreciar en la Figura 1, donde se muestran 4 patrones de entrada que conforman un vector de entrada X de dimensión 6, (x_1, x_2, x_3, x_4, x_5, x_6), acompañados de su clase.

	X1	X2	X3	X4	X5	X6	Clase
Gen1	0.890	0.870	0.880	-0.250	0.460	-0.420	A
Gen2	0.760	0.660	0.790	-0.450	0.260	-0.600	A
Gen3	-0.300	-0.250	-0.340	-0.120	-0.090	-0.400	B
Gen4	-0.150	-0.060	0.180	-0.510	-0.040	0.010	C

Todos los patrones indican a que clase pertenecen

Figura 1 Patrones de expresión de genes indicando la clase a la que

2.1 Topología del modelo supervisado de la RNA

2.1.1 Arquitectura de la RNA

La red propuesta en su presentación supervisada requiere para su funcionamiento de tres capas: una primera capa correspondiente a las neuronas de entrada; la segunda, denominada oculta, la cual cuenta con las neuronas que integrarán los cúmulos o clusters para reconocer patrones. En esta capa las neuronas conocidas como centroides. La tercera capa, que incluye las neuronas de salida correspondientes a las clases a las que pertenecen los patrones de expresión de genes presentados. La arquitectura final de la RNA de este tipo puede ser apreciada en la figura 2, donde se muestran las tres capas que la integran. En el modelo neuronal supervisado, las capas de entrada y de salida son fijas, es decir, la cantidad de neuronas en cada una de estas capas se conoce desde el inicio del entrenamiento, mientras que la capa oculta se formará durante el proceso de aprendizaje a partir del arribo de los patrones de entrada a la red. En la capa oculta las neuronas se organizan dinámicamente formando cúmulos y vecindades de centroides como resultado de la aplicación de una medida de similitud. Por medio de las capas de salida y oculta se obtiene conocimiento de los patrones de expresión de genes, no solo al indicar su pertenencia a una misma clase sino que además por un grado mayor de relación entre ciertos grupos de genes representados por los centroides del cúmulo. Esta es una ventaja presente en nuestro modelo con respecto a otros existentes, los cuales sólo indican que ciertos genes pertenecen a una misma clase pero no muestran que dentro de cada clase existen otras relaciones entre los genes.

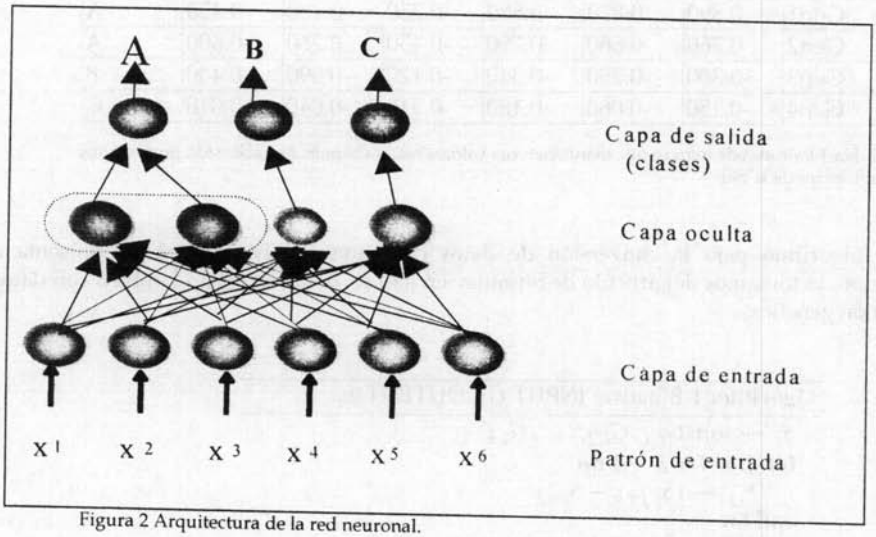


Figura 2 Arquitectura de la red neuronal.

2.1.1.1 *La capa de entrada*

La capa de entrada cuenta con un número de neuronas correspondiente a la dimensión de los vectores presentados como patrones de expresión de genes. La dimensión del vector de entrada corresponde al total de experimentos realizados con cada gene, en cada experimento se identifica el nivel de expresión del gene obtenido de un mismo microarreglo. Los valores numéricos para los experimentos en los vectores de entrada son números reales (figura 1). Con la presentación de patrones en esta capa inicia la dinámica de la red, recibiendo los datos de vectores provenientes del exterior para su propagación hacia el interior de la red. Los datos propagados pueden ser de tipo real, binarios o ternarios. En la capa de entrada los datos son procesados por dos funciones, la de activación y la de transferencia, que darán como resultado la salida de los datos de esa capa para propagar la información hacia el interior de la red como se aprecia en la figura 3. Estos datos de los vectores de entrada provienen de un tratamiento previo que los normalizó para su utilización en la red. La función de activación variará de acuerdo con el tipo de valores deseados, es decir, si son binarios, ternarios o reales.

En la capa de entrada la función de activación varia: para hacer la reducción de los valores continuos a valores discretos como son binarios o ternarios se utiliza una función umbral, mientras que en el caso que se desea mantener los mismos valores continuos viene utilizada la función identidad. El uso de la función identidad en esta capa tiene como objetivo que sean proyectados los datos hacia el interior de la red neuronal con los mismos valores recibidos del medio externo. En la tabla 1 se muestra un pequeño conjunto de datos reales o continuos, los cuales no son convertidos a través de su paso por la capa de entrada.

	X1	X2	X3	X4	X5	X6	Clase
Gen1	0.890	0.870	0.880	-0.250	0.460	-0.420	A
Gen2	0.760	0.660	0.790	-0.450	0.260	-0.600	A
Gen3	-0.300	-0.250	-0.340	-0.120	-0.090	-0.400	B
Gen4	-0.150	-0.060	0.180	-0.510	-0.040	0.010	C

Tabla 1 Patrones de ingreso que mantienen sus valores reales después de haber sido proyectados al interior de la red

El algoritmo para la conversión de datos continuos a binarios, que se presenta a continuación, lo tomamos del artículo de Shmulevich [30] en donde trabajan también con datos de expresión genética.

Algorithm 1 Binarize: INPUT G_i, OUTPUT B_i

$S_i \leftarrow \text{sort}(G_{i,1}, G_{i,2}, \ldots, G_{i,k})$
for $j = 1$ to $k - 1$ **do**
  $D_{i,j} \leftarrow (S_{i,j+1} - S_{i,j})$
end for
$t \leftarrow (S_{i,k} - S_{i,1})/(k - 1)$
$m = \min\{j : D_{i,j} > t\}$
for $j = 1$ to k **do**
  if $G_{i,j} \geq S_{i,m+1}$ **then**
    $B_{i,j} \leftarrow 1$
  else
    $B_{i,j} \leftarrow 0$
  end if
end for

Donde $G_i = (G_{i,1}, G_{i,2}, \ldots, G_{i,k})$ es el perfil de un gene
y $B_i = (B_{i,1}, B_{i,2}, \ldots, B_{i,k})$ es el perfil con valores binarios

En la tabla 2 se muestra la conversión a valores binarios correspondientes a los datos presentados en la tabla 1.

	X1	X2	X3	X4	X5	X6	Clase
Gen1	1	1	1	0	1	0	A
Gen2	1	1	1	0	1	0	A
Gen3	0	0	0	0	0	0	B
Gen4	0	0	1	0	0	1	C

Tabla 2 Patrones de ingreso binarios correspondientes a los datos originales mostrados en la tabla.

La función escalón utilizada para efectuar la conversión de los datos reales a ternarios es la siguiente[1]:

$f(x) = 0$ si $-0.5 < x < 0.5$
$f(x) = 1$ si $x >= 0.5$
$f(x) = -1$ si $x =< -0.5$

En la tabla 3 se muestra la conversión a valores ternarios correspondientes a los datos presentados en la tabla 1.

	X1	X2	X3	X4	X5	X6	Clase
Gen1	1	1	1	0	0	0	A
Gen2	1	1	1	0	0	-1	A
Gen3	0	0	0	0	0	0	B
Gen4	0	0	0	-1	0	0	C

Tabla 3 Patrones de ingreso ternarios correspondientes a los datos originales mostrados en la tabla.

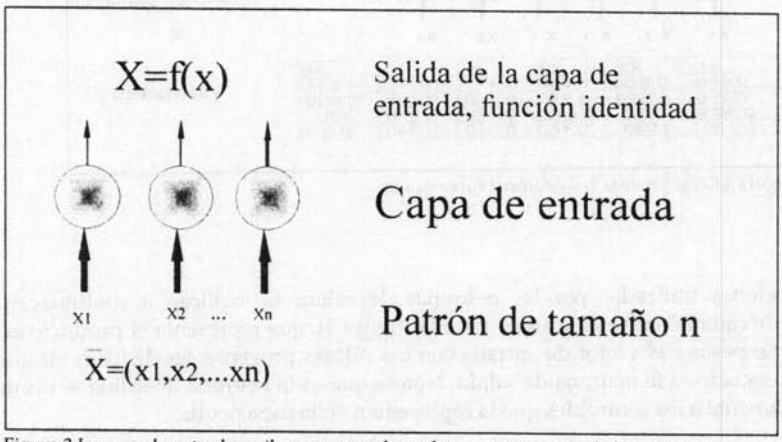


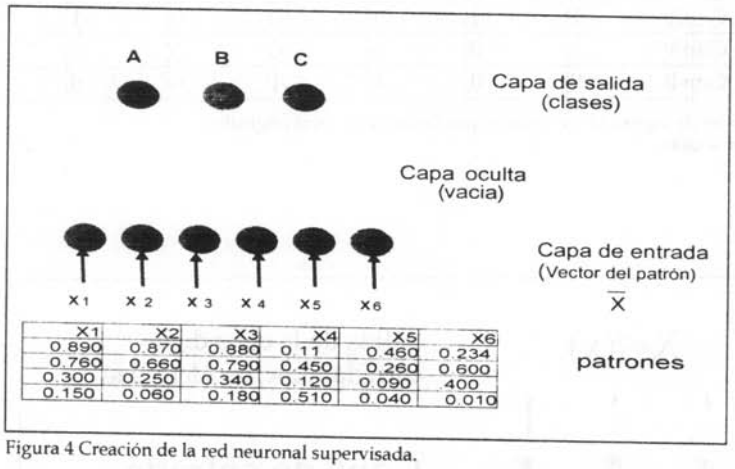
Figura 3 La capa de entrada recibe y propaga los valores provenientes de los patrones presentados.

[1] Esta función escalón fue tomada del artículo de Kim *et al.* 2000.

2.1.1.2 La *capa de salida*

Antes de describir la capa oculta de la red o la capa de los centroides, describiremos la capa de salida. El motivo de ello es que tanto la capa de entrada como la de salida son parte de la red neuronal supervisada al momento de su creación, como lo indica la figura 4, mientras que la capa oculta se formará dinámicamente, es decir, en la medida en que la red recibe los patrones de entrada.

La capa de salida contiene las diferentes clases a las que pueden pertenecer los patrones de expresión de genes que se presentan a la red. Debido a que en esta primera parte del modelo de red neuronal todos los patrones de expresión indican a qué clase están asociados. Al momento de crear la red neuronal se conoce cuántas neuronas forman la capa de salida. De esta forma, es posible definir también la capa de salida como parte de la topología inicial de la red.



X1	X2	X3	X4	X5	X6
0.890	0.870	0.880	0.11	0.460	0.234
0.760	0.660	0.790	0.450	0.260	0.600
0.300	0.250	0.340	0.120	0.090	400
0.150	0.060	0.180	0.510	0.040	0.010

Figura 4 Creación de la red neuronal supervisada.

Las funciones utilizadas por las neuronas de salida se explican a continuación. La función inicial de cada neurona es la suma pesada (figura 5), que representa el producto escalar entre el vector de pesos y el vector de entrada con los valores provenientes de los centroides de la capa oculta asociados a la neurona de salida. Nótese que cada neurona de salida se encuentra conectada únicamente a los centroides que la representan en la capa oculta.

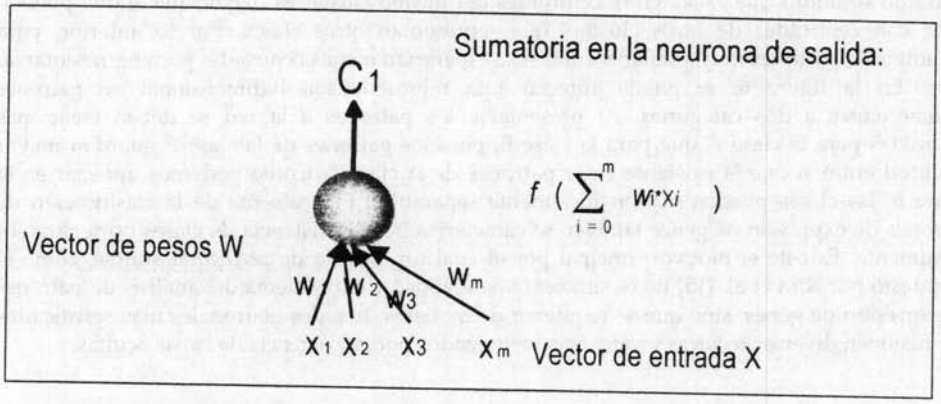


Figura 5 Función de activación en la capa de salida.

Suma pesada: $f(w_1x_1+w_2x_2+\ldots+w_mx_m)$ que es igual a $f(\sum_{i=1}^{m} w_ix_i)$

donde cada peso está representado por w y el valor de salida de cada neurona de la capa anterior es x con $1 \leq i \leq m$ [21]. La función de activación que obtiene el valor de salida de esta capa y por tanto de la red es la función identidad $x = f(x)$.

La neurona con el mayor valor de salida será la que clasifique el patrón de entrada proporcionado a la red. La discriminación de los patrones de entrada efectuada a nivel de los centroides o neuronas de la capa oculta se mantiene cuando la información es propagada de la capa oculta a la capa de salida. Las neuronas de la capa de salida solo se encargan de proyectar al exterior el resultado de la clasificación del patrón presentado a la red.

2.1.1.3 La capa oculta

Las neuronas de la capa oculta son también conocidas como centroides. Cuando es creada la topología inicial de la red, no existe ninguna neurona en esta capa y se desconoce cuántas neuronas ocultas serán necesarias para la clasificación de los patrones de entrada. Esto quiere decir que la red no cuenta con una memoria inicial. La primera neurona de la capa oculta se crea a partir de la presentación del primer patrón de entrada y la capa oculta de la RNA se modificará durante según las características de los patrones de entrada, siendo estos quienes definirán la topología de la red. Los centroides son agrupados formando cúmulos, cada cúmulo o cluster representa una clase o categoría. Los centroides creados en la capa oculta de la red se unen con las neuronas de la capa de la salida, las cuales representan las diferentes categorías en las que estos pueden ser clasificados. Un centroide de la capa oculta puede conectarse a una y solo una neurona de la capa de salida, ya que un centroide no puede caracterizar a más de una clase ni un cúmulo puede agrupar a centroides de distintas clases.

En el modelo de red neuronal que presentamos, los centroides se crearán, modificarán y eliminarán de acuerdo a las necesidades dictaminadas por los patrones presentados a la red. El

grado de similitud que existe entre centroides del mismo cluster es mayor que el que pueden tener con centroides de otros clusters que representan otras clases. Por lo anterior, para conjuntos de patrones muy similares entre sí se esperarían menos centroides para representar su clase. En la figura 6, se puede apreciar una representación bidimensional de patrones pertenecientes a dos categorías. Al presentarse los patrones a la red se deben crear más centroides para la clase A que para la clase B, pues los patrones de la clase B guardan mayor similitud entre sí que la existente entre patrones de la clase B. Como podemos apreciar en la figura 6, las clases pueden no ser linealmente separables. El problema de la clasificación de patrones de expresión de genes también se caracteriza por la existencia de clases no separables linealmente. Es este el motivo principal por el cual un modelo de *perceptrón* simple, como el propuesto por Kim et al. [15] no es suficiente para abordar el problema del análisis de patrones de expresión de genes, sino que se requieren de modelos de redes neuronales más sofisticados que fusionen diversas técnicas y vengan caracterizados por la existencia de capas ocultas.

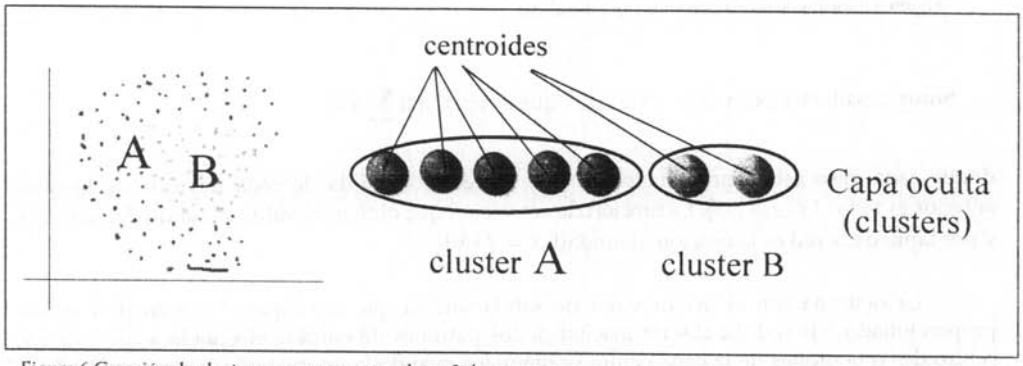


Figura 6 Creación de cluster para representar clases 2 de patrones.

La función de activación utilizada por las neuronas de la capa oculta es la distancia entre el vector de peso y el patrón de entrada. Las métricas a utilizar son la distancia de Manhatan, la distancia Euclidiana, el coeficiente de correlación de Pearson, la distancia de Mahalanobis y la distancia de Hamming las cuales son las funciones más encontradas en la literatura especializada en el análisis de patrones de expresión de genes [34, 7, 11, 21]. El modelo neuronal propuesto brinda la posibilidad de seleccionar la métrica a utilizar.

Distancia euclideana: $d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})_2}$

Distancia de Manhattan (o valor absoluto): $d(x_i, x_j) = \sum_{k=1}^{n} | x_{ik} - x_{jk} |$

Coeficiente de la correlación de Pearson: $d(x_i, x_j) = (1 - r) = 1 - \frac{\sum((x1i - dvx1)(x2i - dvx2))/n}{Sx1Sx2}$

la cual trabaja con la desviación estándar

Distancia de Hamming, utilizada para datos binarios:

Si *dx <> dy dist=dis+1, donde dx y dy tienen valores binarios*

La función de salida para las neuronas de la capa oculta está dada por la expresión (1.0) y cuya gráfica se aprecia en la figura 7, en ella se distingue como disminuye proporcionalmente el valor de la salida al aumentar la distancia.

$$\text{salida}=1/(\text{distancia}+1) \qquad (1.0)$$

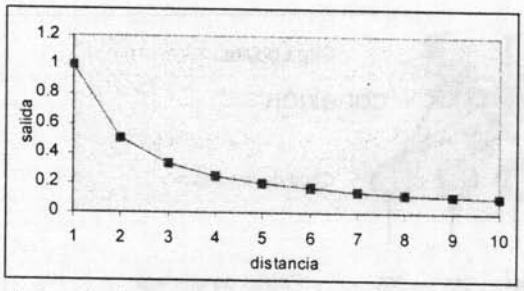


Figura 7 Gráfica de la función de salida de las neuronas ocultas.

A mayor distancia calculada entre el vector de pesos y el patrón de entrada, menor será el valor de salida de la neurona o centroide. Las neuronas de la capa oculta compiten entre si para decidir cual neurona clasificará el patrón de entrada. Sólo un centroide puede ganar la competencia, el cual recibirá una "recompensa", mientras que los restantes centroides recibirán una "penalización". Los mecanismos de "recompensa" y "penalización" envuelven la modificación de diversos parámetros, los cuales se introducirán a continuación.

Para soportar la dinámica de la red que se desarrolla en la capa oculta, donde las neuronas o centroides pueden ser creados, eliminados, modificados o fusionados, dos parámetros caracterizan las neuronas de esta capa: el radio y el coeficiente de vitalidad que vienen inicializados con valores elegidos a priori para la red. El mecanismo de recompensa se encarga del incremento del coeficiente de vitalidad. Por otra parte, el mecanismo de penalización provee una disminución del coeficiente de vitalidad y en casos particulares, discutidos mas adelante, una disminución del radio.

Radio del centroide. El radio de las neuronas ocultas interviene directamente en la decisión de si el patrón de entrada puede o no ser clasificado por una neurona ya que el radio es el valor comparado directamente con la medida de similitud.

Coeficiente de vitalidad. El coeficiente de vitalidad proporciona una medida de que tan "eficiente" es el centroide. Es decir, que tantas veces ha ganado el centroide en la competencia por clasificar patrones y por lo tanto que tan representativo es de toda una clase de patrones.

Tanto el radio como el coeficiente de vitalidad de los centroides son comparados con factores de decisión que envuelven valores críticos del radio y del coeficiente de vitalidad, respectivamente (introducidos más adelante). Esta comparación es realizada para determinar cuales centroides serán eliminados, aquellos más debilitados por su poca utilidad para la clasificación de patrones y que pueden ser absorbidos por otros. Esto hace que la capa oculta cuente con las neuronas necesarias de acuerdo a las características de los patrones de entrada presentados.

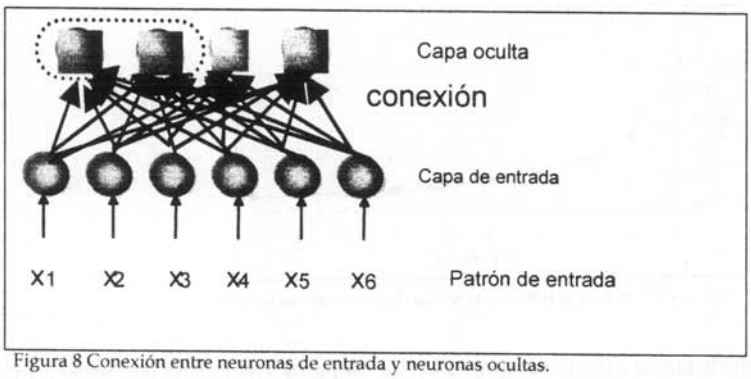


Figura 8 Conexión entre neuronas de entrada y neuronas ocultas.

2.1.2 *Conexiones entre las neuronas*

Las conexiones que se establecen en la red neuronal sólo son entre neuronas pertenecientes a diferentes capas. La capa de entrada se encuentra completamente conectada con la capa oculta, cada neurona de la capa de entrada se conecta con todas las neuronas de la capa oculta. En la figura 8 se puede apreciar este patrón de conexión. En tanto que, sólo ciertas neuronas de la capa oculta se conectarán con una neurona de la capa de salida, es decir, cuando un cluster en la capa oculta viene caracterizado por más de una neurona o centroide, entonces la neurona de la capa de salida que representa la clase de este cluster recibirá tantas conexiones de entrada como neuronas hayan en el cluster. El patrón de conexión entre las capas oculta y de salida se puede apreciar en la figura 9.

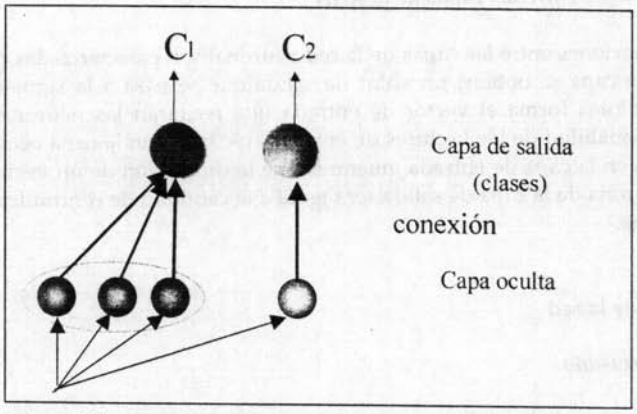


Figura 9 Cúmulos o clusters conectados a la clase de salida que

Es importante recordar que, debido a la ausencia de la capa oculta, en el inicio del proceso no se cuenta con centroides creados que formen cúmulos, por lo que no existe ninguna conexión en la red. Las primeras conexiones se crearán en cuanto las neuronas de la capa de entrada reciban información de un patrón de expresión de genes, se activen y obtengan su valor de salida con lo que inician la propagación. Con el primer patrón presentado dará inicio la creación del primer centroide, pues al no existir ningún centroide para clasificar el patrón, este debe crearse en ese momento, entonces la red mostrará la conexión de la capa de entrada con el centroide de la capa oculta y de ese centroide con la correspondiente neurona de la capa de salida, como se aprecia en la Figura 10, donde se ha dado la primera conexión con una de las tres capas de la red. La figura 4 muestra la ausencia de conexiones entre las capas de entrada y de salida antes del ingreso del primer patrón.

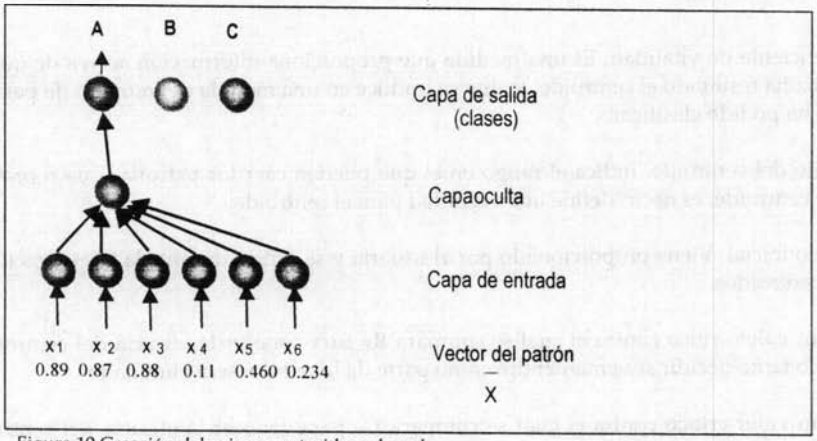


Figura 10 Creación del primer centroide en la red.

2.1.3 *Flujo de información entre las capas de la RNA*

Todas las conexiones entre las capas de la red neuronal son caracterizadas por pesos. De las neuronas de cada capa se obtiene un valor de salida que se pasa a la siguiente capa y el conjunto de estos valores forma el vector de entrada que recibirán las neuronas de la capa superior. La dimensionalidad de los vectores de entradas recibidos en la capa oculta es igual al número de neuronas en la capa de entrada, mientras que la dimensión de un vector de entrada recibido por una neurona de la capa de salida será igual a la cantidad de centroides en el cluster asociado a tal neurona.

2.2 Funcionamiento de la red

2.2.1 *Algoritmo supervisado*

Antes de presentar el algoritmo para el aprendizaje supervisado es necesario introducir la notación que describe los parámetros y los restantes elementos del modelo neuronal. Los parámetros iniciales que deben ser proporcionados antes de iniciar la fase de aprendizaje de la red son los siguientes:

C_V	Coeficiente de vitalidad
R_P	Radio del centroide
R_O	Radio inicial
g_v	Constante de decisión para eliminación por R_P
β_v	Constante de decisión para eliminación por C_V
X	Patrón de entrada (patrón de expresión de genes)
N	Dimensión del patrón de entrada X
m	Número de neuronas de la clase de salida
α	Factor para la modificación del coeficiente de vitalidad
wp	Vector de pesos

C_V Coeficiente de vitalidad. Es una medida que proporciona información acerca de que tan eficaz ha resultado el centroide, lo que se traduce en una medida del número de patrones que ha podido clasificar.

R_P Radio del centroide. Indica el rango en el que pueden caer los patrones que representa este centroide, es decir, define una vecindad para el centroide.

R_O Radio inicial. Viene proporcionado por el usuario y se utiliza durante la inicialización de los centroides.

g_v Es un valor crítico contra el cual se compara R_P para conocer la eficacia del centroide y por lo tanto decidir si se mantendrá como parte de la red o si será eliminado

β_v Es un valor crítico contra el cual se compara C_V para conocer la eficacia del centroide. Centroides con un coeficiente de vitalidad inferior a este valor vienen eliminados de la red.

X Patrón de entrada que contiene los valores del patrón de expresión de genes presentado, es decir, los valores de expresión del mismo gene en un experimento.

d(X,P) Distancia entre el patrón de entrada y el vector de pesos del centroide.

P_i Centroide que se está evaluando actualmente, con $1<i<n$

m Número de neuronas en la clase de salida, en este modelo de aprendizaje supervisado este valor es conocido a priori.

WC Clase que representa el centroide. Uno o más centroide pueden tener la misma clase.

CS Clase que representa la neurona de salida, cada neurona de salida tiene diferente clase

PC Clase del patrón presentado a la red. Cada patrón debe pertenecer a una clase y en este modelo supervisado todos los patrones traen indicado este valor.

α Factor para modificación del C_V, permite definir el grado en que se modificará el coeficiente de vitalidad de cada centroide.

wp Vector de pesos. La capa oculta y la capa de salida cuentan con su vector de pesos que se modifica durante el proceso evolutivo de la red para lograr la clasificación de los patrones presentados.

La red neuronal propuesta en esta primera aproximación al análisis de patrones de expresión de genes usa un modelo de aprendizaje supervisado y trabaja de acuerdo al siguiente algoritmo:

1. Presentar a la red neuronal el patrón de expresión de gene, este patrón se presenta como un vector (*X*), y viene acompañado por la clase o categoría a la cual pertenece el patrón (*PC*).

2. Verificar si la clase del patrón (*PC*) coincide con alguna de las clases de salida conocidas

   2.1 Si **no** coincide: poner la *PC=NoClasificado*

   Si la clase del patrón presentado no se encuentra entre las clases dadas inicialmente como parámetro a la red, se pone dentro de una clase que representa los patrones "NoClasificados"

3. Verificar si existe ya un centroide en la capa oculta.

   Al inicio cuando se crea la red no existe ningún centroide o neurona en la capa oculta, por lo cual esta pregunta tiene una respuesta negativa sólo para el primer patrón presentado a la red. En el momento de propagar la información del primer patrón desde la capa de entrada hacia las demás capas de la red se crea la primera neurona o centroide en la capa oculta.

3.1 Si **no** existen centroides:

Crear el primer centroide con los datos iniciales para el radio (R_P) y coeficiente de vitalidad (C_V) dados por el radio inicial (R_O) y el factor de vitalidad (α) respectivamente, y el vector de pesos correspondiente al patrón de entrada y con la clase *WC* misma que la clase del patrón de entrada (*PC*).

$wp = X$, $R_P = R_O$, $C_V = \alpha$ y $WC=PC$

El centroide creado se conecta a la neurona de la capa de salida que le corresponde de acuerdo con su clase.

3.2 Si existen centroides o neuronas en la capa oculta:

Calcular la distancia entre el patrón presentado actualmente a la red y cada centroide de la capa oculta,

$d(X,P_i)$, $1 \leq i \leq m$

La distancia nos indica la medida de similitud entre los dos vectores evaluados, el de pesos de cada centroide y el de entrada o del patrón. Estas distancias pueden ser: la euclidiana, la de Manhatan y la correlación de Pearson, que son las más utilizadas en la literatura especializada [32, 25, 15, 29].
P_i representa el *wp* del centroide o neurona oculta.

El centroide ganador es aquel para el cual se ha calculado una menor distancia al patrón de entrada.

3.2.1 Si $d(X,P) < R_P$ y
Si $WC=PC$

El patrón es perfectamente clasificado por el centroide ganador.
Esto quiere decir que la diferencia entre el patrón presentado y el vector de pesos del centroide es menor al radio del prototipo, por lo que logra clasificar al patrón y además las clases, tanto del patrón de entrada como del centroide ganador, son iguales. Este es el caso ideal para la clasificación de patrones.

Entonces, se actualiza el centroide ganador, reforzándolo para clasificar ese patrón: incrementa el coeficiente de vitalidad (C_V) : $C_V = C_V * \alpha^2$

Sino: $WC \neq PC$

Crear un centroide que lo clasifique

$wp = X$ $R_P = R_O$, $C_V = \alpha$ y $WC=PC$

3.2.2 Sino: $d(X,P) > R_P$ y
Si $WC=PC$

La menor distancia encontrada entre los centroides es mayor que el radio del centroide, pero la clase coincide entre el centroide y el patrón.

Se fusionan el centroide y el patrón presentado. Esto representa incrementar el coeficiente de vitalidad del centroide, $C_V = C_V * \alpha^2$, incrementar el radio del

centroide, $R_P = R_P + d(X,P)$ y modificar el vector de pesos en $P=P+1/2(X-P)$, donde X es el vector de entrada y P es el vector de pesos del centroide.

Sino: $WC \neq PC$

La mínima distancia reportada corresponde a un centroide que no clasifica al patrón presentado porque las dos clases, la del patrón y la del centroide, no coinciden, ello implica que no existe un centroide que pueda clasificar a ese patrón.

Por lo tanto, crear un centroide para clasificar el patrón.

$wp = X$ $R_P = R_O$, $C_V = \alpha$ y $WC=PC$

4. Después de identificar el centroide ganador se modifican los centroides que no pudieron clasificar al patrón, a través de una penalización que disminuye su coeficiente de vitalidad.

   Decrementar el coeficiente de vitalidad en los centroides no ganadores, $C_V = C_V / \alpha$

5. Verificar cuáles son los centroides a eliminar de acuerdo con los valores de su coeficiente de vitalidad y su radio. Se comparan esos valores con el radio mínimo y el factor de decisión para el coeficiente de vitalidad, proporcionados inicialmente por el usuario.

   Si $C_V \leq \beta_v$ or $R_P \leq g_v$ **entonces** eliminar el centroide

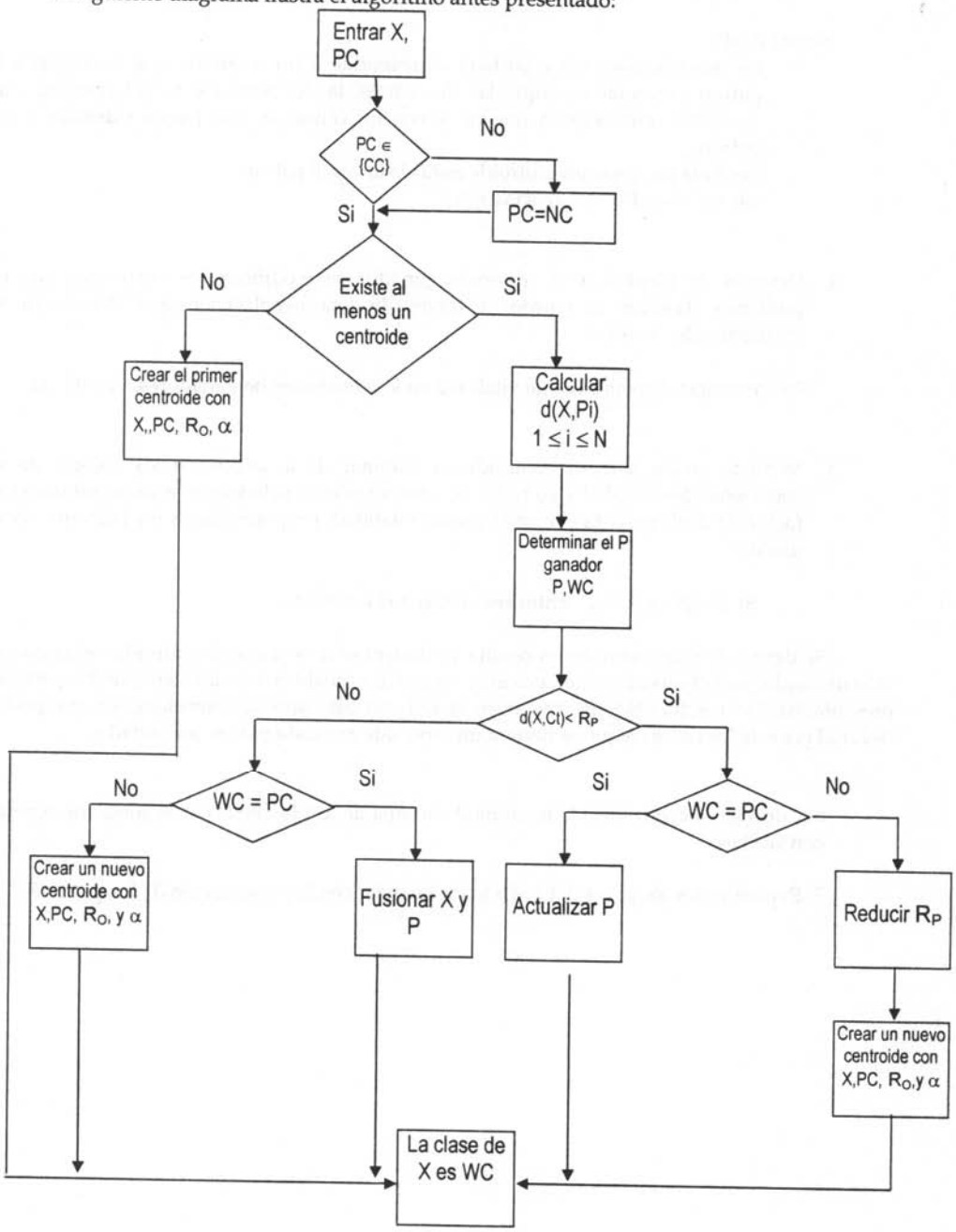
Si alguna de estas expresiones resulta verdadera se debe a que el centroide verificado ha sido utilizado para clasificar un número muy escaso (o probablemente uno solo) de los patrones presentados. No resulta eficiente contar en la red con este tipo de centroides, ya que podría llevar al peor de los casos en que se tuviera un centroide por cada patrón presentado.

6. Conectar el centroide a la neurona de la capa de salida que le corresponde de acuerdo con su clase.

7. Repetir todos los pasos del 1 al 6 hasta finalizar con la presentación de los patrones.

2.2.2 *Diagrama de flujo*

El siguiente diagrama ilustra el algoritmo antes presentado:



2.3 Principales comportamientos de la red de acuerdo al algoritmo

Para mostrar cómo trabaja el algoritmo se considerara un pequeño conjunto de patrones de expresión de genes reales.

2.3.1 *Creación de la red supervisada*

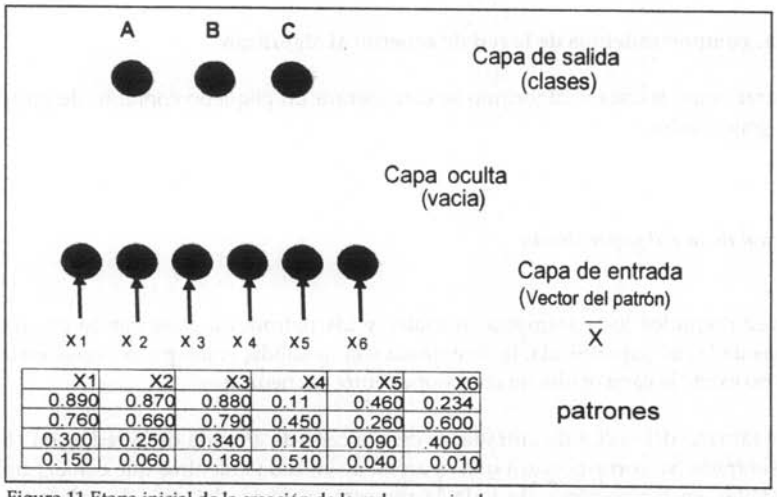
Una vez definidos los parámetros iniciales y los patrones a clasificar se crean las dos primeras capas de la red supervisada: la de entrada y la de salida, como puede verse en la figura 11, en donde no existe la capa oculta, ni conexiones entre las neuronas.

Con el tamaño del vector de entrada, se crea la capa de entrada de la red, cada elemento del vector de entrada (x_i) corresponde a una neurona de entrada, mientras que con el número de clases presentadas en los patrones de toda la muestra se forma la capa de salida de la red neuronal. Entonces, las capas de entrada y salida se crean en la etapa inicial de definición de la topología de la red y la información sobre su número de neuronas proviene de la muestra de patrones.

	X1	X2	X3	X4	X5	X6	Clase
Gen1	0.890	0.870	0.880	-0.250	0.460	-0.420	A
Gen2	0.760	0.660	0.790	-0.450	0.260	-0.600	A
Gen3	-0.300	-0.250	-0.340	-0.120	-0.090	-0.400	B
Gen4	-0.150	-0.060	0.180	-0.510	-0.040	0.010	C

Número de clases: m = 3

Número de neuronas de entrada = 6

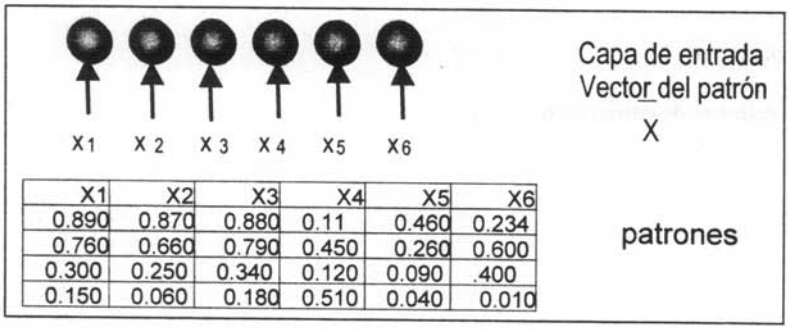


X1	X2	X3	X4	X5	X6
0.890	0.870	0.880	0.11	0.460	0.234
0.760	0.660	0.790	0.450	0.260	0.600
0.300	0.250	0.340	0.120	0.090	.400
0.150	0.060	0.180	0.510	0.040	0.010

Figura 11 Etapa inicial de la creación de la red supervisada.

2.3.2 *Presentación de patrones a la red*

Ya creada la parte inicial de la red y capturados los patrones de entrada, comienza la presentación de los patrones a la red para su clasificación, propagando uno a la vez, cada patrón se presenta a la red en forma de vector. En la capa de entrada cada neurona toma el valor de expresión del gene en un experimento particular. En la figura 12 se muestra como la dimensión del vector del patrón es igual al número de neuronas de la capa de entrada.



X1	X2	X3	X4	X5	X6
0.890	0.870	0.880	0.11	0.460	0.234
0.760	0.660	0.790	0.450	0.260	0.600
0.300	0.250	0.340	0.120	0.090	.400
0.150	0.060	0.180	0.510	0.040	0.010

Figura 12 Presentación de un patrón a la capa de entrada de la red.

2.3.3 *Creación de centroides en la red*

Una vez que la capa de entrada recibe el vector de expresión de un gene, esta información es propagada hacia la capa oculta, a través de las conexiones que unen las capas de entrada con la oculta. Si no existe ninguna neurona en la capa oculta se crea el primer centroide con los valores iniciales dados como parámetros ($R_P = R_O$, $C_V = \alpha$, $wp = X$) y los del patrón presentado. En la figura 9 puede verse cuando se crea el primer centroide de la red. En general, los centroides se crean cuando el patrón presentado no encuentra algún centroide que lo clasifique, ello puede deberse a:

a) No existe ningún centroide en la RNA

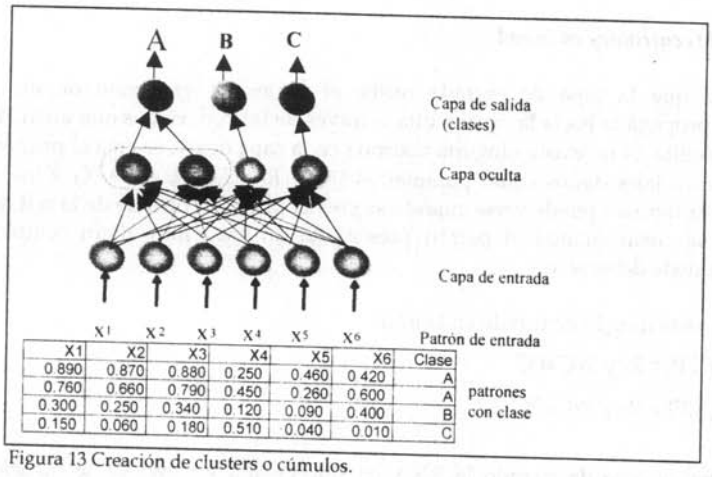
b) Si $d(X,P) < R_P$ y $WC \neq PC$

c) Si $d(X,P) \geq R_P$ y $WC \neq PC$

El primer caso se da cuando la RNA ha sido creada y como ya se mencionó en ese momento no hay centroides en la capa oculta, al presentar el primer patrón de entrada se crea el primer centroide que lo clasifique. En el segundo caso, la clase del centroide ganador (*WC*) no coincide con la clase del patrón (*PC*) por lo que se crea su centroide. Por último, en el tercer caso no existe ningún centroide que pueda clasificar al patrón, como lo indica la diferencia de clases y la distancia calculada mayor al radio R_P. Como se trata de un modelo de aprendizaje supervisado, una restricción impuesta en el proceso de clasificación es que la clase del patrón (*PC*) coincida con la del centroide ganador (*WC*), de lo contrario un nuevo centroide viene creado para garantizar la clasificación del patrón.

2.3.4 *Formación de clusters o cúmulos*

La presentación de múltiples patrones de expresión a la red permite la formación de los clusters o cúmulos. Los clusters están integrados por grupos de centroides con un alto grado de similitud ya que pertenecen a una misma clase de salida, pero a la vez con un mínimo grado de semejanza con los centroides de otros cúmulos. En la figura 13 se ejemplifican tres clusters integrantes de la capa oculta; un cluster puede estar representado por una sola neurona cuando los patrones que clasifica tienen un alto grado de similitud. La dinámica de la red en el modelo supervisado se refiere a que se crean y eliminan centroides dentro de cada cúmulo hasta que se logra una estabilización que represente correctamente a todos los patrones de expresión presentados. Así pues, la topología final de la red depende de los cúmulos formados en la capa oculta.



	X1	X2	X3	X4	X5	X6	Patrón de entrada
X1	X2	X3	X4	X5	X6	Clase	
0.890	0.870	0.880	0.250	0.460	0.420	A	
0.760	0.660	0.790	0.450	0.260	0.600	A	patrones
0.300	0.250	0.340	0.120	0.090	0.400	B	con clase
0.150	0.060	0.180	0.510	0.040	0.010	C	

Figura 13 Creación de clusters o cúmulos.

2.3.5 *Obtención de las medidas de similitud*

Las medidas de similitud permiten distinguir la semejanza que existe entre el patrón de genes y los centroides existentes en la red. Esta medida de similitud es obtenida por las distancias euclidiana, de Manhatan, de Mahalanobis, de Hamming y el coeficiente de correlación de Pearson. Una vez que existen centroides en la capa oculta, cada vez que se presenta un patrón a la red se obtiene la distancia o diferencia entre el patrón presentado y el vector de pesos de cada neurona centroide. El usuario determina, de acuerdo al objetivo de su estudio biológico y al tipo de datos (binarios, reales o ternarios) de los vectores, la medida que le pueda proporcionar mayor y mejor información.

2.3.6 *Establecimiento del centroide ganador*

Entre todas las diferencias obtenidas por la medida de similitud aplicada a los patrones de entrada y vector de pesos de cada centroide se elige la menor y el centroide al que le corresponde es considerado como el ganador. Esa diferencia (distancia) es comparada con el radio (R_P) de dicho centroide y considerando también las clases a las que pertenecen tanto el centroide como el patrón de entrada se toma la decisión acerca de si este centroide representa realmente a ese patrón de entrada. De acuerdo a la decisión tomada se puede reforzar al centroide ganador o crearse uno nuevo para que represente correctamente al patrón de entrada. En el algoritmo consideramos 4 casos a evaluar cuando se ha obtenido el centroide ganador:

a) Si $d(X,P) < R_P$ y $WC=PC$
b) Si $d(X,P) < R_P$ y $WC \neq PC$
c) Si $d(X,P) \geq R_P$ y $WC=PC$
d) Si $d(X,P) \geq R_P$ y $WC \neq PC$

Para cada caso es necesario tomar la decisión de si el centroide ganador clasifica o no al patrón presentado a la red y de acuerdo a esto las evaluaciones hechas para esos cuatro casos son: en el caso a) el centroide es el que clasifica correctamente al patrón de entrada, ya que su distancia está dentro del rango de su radio (R_p) y las clases del patrón y centroide son iguales. En el caso b), a pesar de que el patrón cae dentro del rango del radio del centroide las clases no son iguales, el R_p es reducido y se crea un centroide ad-hoc para clasificar este patrón. En el caso c), las clases son iguales pero la distancia computada entre ambos vectores cae fuera del R_p y como consecuencia el patrón es fusionado al centroide. Finalmente, en el último caso d), el centroide que computa la menor distancia al patrón de entrada tiene un R_p menor que la distancia calculada y su clase tampoco coincide con la clase del patrón, por lo que es necesario crear un nuevo centroide para clasificar al patrón. Las acciones anteriores son ejecutadas en esta versión supervisada de la red.

2.3.7 *Incremento de radio del centroide (R_p) y del coeficiente de vitalidad (C_v) en el centroide ganador*

Cuando se considera que el centroide ganador clasifica correctamente al patrón de entrada se refuerza el centroide en su parámetro C_v. Al reforzar al centroide ganador en C_v se indica que el centroide es eficaz para representar patrones de expresión de genes, por lo que se justifica su permanencia en la red. El incremento del coeficiente de vitalidad se realiza de acuerdo a la siguiente expresión:

$$C_v = C_v * \alpha^2$$

Se incrementa el C_v con relación al factor de incremento α, proporcionado como parámetro de la red.

Cuando se realiza la fusión de un patrón de entrada y un centroide el R_P es incrementado de acuerdo a la siguiente expresión:

$$R_P = \tfrac{1}{2} d(X,P) + R_P$$

2.3.8 *Modificación del vector de pesos en el centroide ganador*

El centroide que ha resultado ganador modifica su vector de pesos (*wp*) para reforzarlo y contribuir al aprendizaje, como sucede en las redes neuronales. La modificación se hace incrementando en una proporción la diferencia del vector de pesos actual con respecto al vector de entrada.

$wp = wp + factor * d(\overline{X}, \overline{P_i})$, donde $\overline{P_i}$ es el centroide ganador y $\overline{X}$ el vector de entrada.

El *factor* varía, es un parámetro cuyo valor es menor en los casos de fusión y mayor en otro caso.

2.3.9 *Modificación de centroides perdedores*

A los centroides que no representan al patrón de entrada actual y que son considerados como perdedores reciben una penalización reflejada en una disminución del coeficiente de vitalidad (C_v). Con ello se va debilitando el centroide que no representa ese patrón. La expresión para efectuar el decremento del coeficiente de vitalidad es la siguiente:

$C_v = C_v / \alpha$ con α mayor que 1

2.3.10 *Verificación de C_v y R_p para eliminación de centroides*

Después de la presentación de cada patrón de entrada a la red, se compara el coeficiente de vitalidad y el radio de todos los centroides existentes en la capa oculta con las constantes de decisión para eliminación de centroides. Aquí es donde repercute el incremento o decremento que sufrieron esos atributos del centroide durante la presentación de patrones. Las comparaciones realizadas son: el C_v contra la constante de decisión para eliminación del coeficiente de vitalidad (β_v) y el R_P contra la constante de radio mínimo (g_v). Para decidir eliminar un centroide es suficiente que resulte cierta alguna de las dos siguientes comparaciones:

$C_v \leq \beta_v$ o $R_P \leq g_v$

Si resulta cierta la evaluación se eliminará el centroide ya que significa que sólo representa a uno o a un número muy pequeño de patrones que podrían incluirse en otro centroide de esa clase o porque el centroide podría representar patrones atípicos.

2.3.11 *Fusión de centroide y patrón de entrada*

El caso en que la distancia entre el patrón y el centroide ganador es superior al R_p de ese centroide y ambos coinciden en cuanto a la clase, se hace una fusión del centroide ganador con el patrón, ya que se considera que ese centroide representa correctamente al patrón de entrada. Por lo cual, el centroide es reforzado al incrementar su radio y el coeficiente de vitalidad. La condición necesaria para la fusión es:

$d(X,P) \geq R_P$ y $WC=PC$

2.3.12 *Evolución de la RNA*

La RNA evoluciona hasta alcanzar la estructura más adecuada correspondiente a la muestra de patrones presentada. Cada neurona de salida está unida a sus centroides que le permiten clasificar correctamente a los patrones de su clase. Los centroides que componen el cúmulo de cada clase son los necesarios para la clasificación, puede considerarse que la red ya está entrenada cuando después de presentar todos los patrones de la muestra (una época k) la modificación sufrida en la capa oculta es nula o mínima.

2.4 Conclusiones

El modelo de RNA presentado, basado en el modelo de Sánchez et al. [29], utiliza un paradigma de aprendizaje supervisado, ya que considera como entradas los patrones de expresión de genes que cuentan con una clase de identificación y al inicio del funcionamiento de la RNA es necesario proporcionar las clases en las cuales deben venir clasificados los patrones. El algoritmo fue enriquecido para lograr nuestro objetivo de manera eficiente. Entre las modificaciones realizadas se encuentran:

1) El cambio de las funciones de activación de las capas de entrada, para considerar valores discretos (binarios y ternarios).
2) La verificación de la permanencia de un centroide, en el modelo de Sánchez et al. era hecha después de la presentación de cada patrón, ya que se trataba de un modelo para patrones que cambiaban su distribución en el tiempo. Debido a que los patrones de expresión de genes siguen una distribución establecida, en nuestra variante del modelo la eliminación de centroides se efectúa al final de la presentación de cada época.
3) La función de activación de las neuronas ocultas puede ser elegida entre las distancias: Euclidiana, Manhatan y Coeficiente de Correlación de Pearson.
4) Pensando en el modelo no supervisado que se mostrará en el siguiente capítulo, se agregó una neurona en la capa de salida para garantizar la clasificación de los patrones de entrada cuya clase no estuviese especificada entre las neuronas de salida consideradas en la topología inicial de la red. De esta forma, no se pierde la información correspondiente a estos patrones para posteriores estudios, pudiéndose determinar que tan diferentes son estos patrones del resto de los patrones clasificados por la red, analizándose si podrían ser incluidos en alguna de las clases proporcionadas a priori o al contrario, si se trata de patrones atípicos.

Capítulo 3.
Modelo no supervisado para el reconocimiento De patrones de expresión de genes

En este capítulo presentamos la variante no supervisada del modelo neuronal que proponemos para la formación de clusters que permitan una correcta clasificación de los patrones de expresión de genes. En el capítulo anterior hablamos acerca de un modelo neuronal para la clasificación de patrones de expresión de genes supervisado. A pesar que el problema del análisis de patrones de expresión de genes requiere de técnicas no supervisadas, principalmente, la adaptación del modelo de Sánchez et al. al problema del análisis de patrones de expresión de genes en la variante supervisada nos proporcionó dos importantes resultados a considerar en la propuesta de la variante no supervisada de este modelo:

1) Las características de los clusters formados por la red, en particular como para determinados conglomerados son requeridos más de un centroide o neurona para la adecuada representación de todos los patrones de entrada que pertenecen al cluster.
2) Parte de la información biológica necesaria para guiar este análisis, la cual viene expresada en los valores alcanzados por los principales parámetros que caracterizan el modelo neuronal (como el radio del centroide, las constantes de eliminación de centroides considerando el radio y el coeficiente de vitalidad, así como el valor de α usado por los mecanismos de recompensa o penalización de centroides).

Las modificaciones y enriquecimientos que proponemos al modelo neuronal de Sánchez et al.[29] en este capítulo nos permitirán abordar el problema del análisis de patrones de expresión de genes a través de una variante no supervisada, la cual es necesaria debido a que el problema principal para la clasificaciòn de genes es el desconocimiento de su clase, a la vez que podremos explotar el conocimiento de la estructuración topológica de los clusters así como el conocimiento biológico acumulado en los parámetros de la red proporcionados por la variante supervisada. La dinámica de la red cambia, deberán formarse dos capas durante el proceso de presentación de patrones y no solo una como en el modelo supervisado. En este modelo no supervisado no se cuenta en la fase inicial ni con la capa de salida (las clases) ni con la capa oculta (los clusters), sino que se formarán de acuerdo a las características de los patrones presentados. El tipo de patrones son como los que se muestran en la figura 1, donde puede apreciarse que no están acompañados de la clase a la que pertenecen, este conocimiento no se posee. La función de la RNA será la propuesta de las clases "biológicamente significativas" para agrupar los patrones de expresión de genes presentados a la red. Al igual que en la variante supervisada, una clase o categoría representada por una neurona de la capa de salida puede estar asociada a uno o más centroides creados en la capa oculta, pero todos representando el mismo cluster.

	X1	X2	X3	X4	X5	X6	X7
Gen1	0.63360006	0.36395448	0.2775541	0.18370205	0.13851574	0.01478122	0.02231959
Gen2	0.6856	0.38101485	0.13915451	0.14589752	0.1281656	0.03239061	0.17677666
Gen3	0.5152	0.41951007	0.28107697	0.26799217	0.20017615	0.0416222	0.05573052
Gen4	0.65440005	0.42169732	0.24207346	0.21646598	0.13080817	0.03159018	0.05826373
Gen5	0.2416	0.22615924	0.1977856	0.18314198	0.13146883	0.14194238	0.88326716
Gen6	0.2824	0.279965	0.17664821	0.20666482	0.13014753	0.06259339	0.11926606

Figura 1 Patrones de expresión de genes para los cuales no se conoce la clase.

3.1 Topología del modelo no supervisado de la RNA

3.1.1 Arquitectura de la RNA

Como ya se mencionó en el capítulo anterior, el modelo neuronal propuesto por Sánchez et al.[29] puede adaptarse para el clustering no supervisado. La arquitectura del modelo neuronal no supervisado también considera tres capas de neuronas: la de entrada, la de neuronas ocultas o centroides y la de salida, como se puede apreciar en la figura 2. En el caso del modelo no supervisado el conocimiento de las clases a las que pertenecen los patrones no se posee a priori, emitir una propuesta de las clases es la meta de la red. Considerando lo antes dicho, al inicio de la dinámica de la red solo viene creada la capa de entrada, mientras que las capas oculta y de salida se generan dinámicamente conforme arriban los patrones de expresión de genes. Del mismo modo que en el modelo supervisado, en la capa oculta las neuronas se organizan dinámicamente formando cúmulos y vecindades de centroides como resultado de la aplicación de una medida de similitud, mientras que en la capa de salida se crea una neurona por cada cluster formado en la capa oculta, cada neurona en la capa de salida representa una clase o categoría de patrones. Y la capa de salida se forma con las clases representadas por los cúmulos de centroides. Si denotamos por m el número de neuronas en la capa de salida y por k el número e neuronas en la capa oculta, entonces $m <= k$, alcanzándose la igualdad únicamente en el caso de que cada clase viene representada por un centroide, lo cual sería un caso límite.

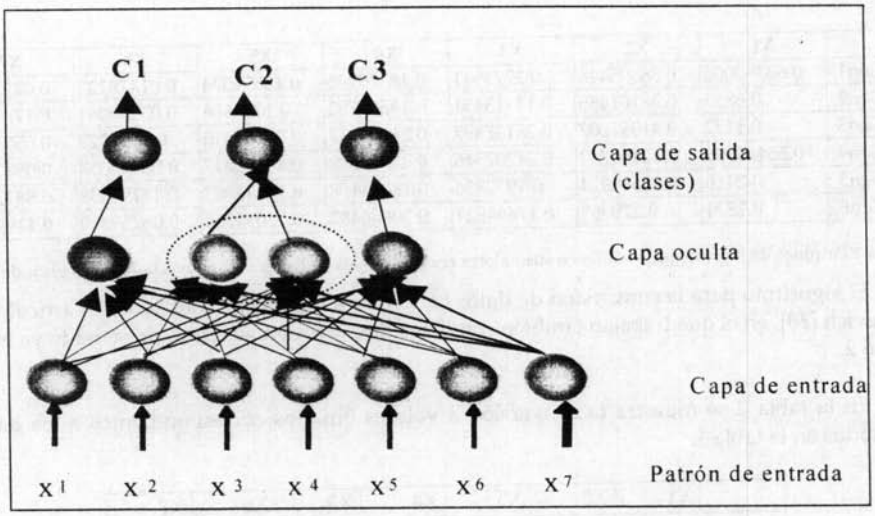


Figura 2 Arquitectura de la red neuronal.

3.1.1.1 La capa de entrada

Al igual que en el modelo supervisado la capa de entrada se genera al inicio de la estructuración de la red, a partir del conocimiento de la dimensión de los patrones de expresión de genes. El objetivo de esta capa es proyectar los valores de cada patrón de expresión de genes al interior de la red. Al igual que en el modelo supervisado, la capa de entrada recibe un vector de valores reales y propaga al interior de la red el mismo vector de valores reales o un vector de valores binarios o un vector de valores ternarios, según sea el tipo de codificación requerida. Las funciones utilizadas por las neuronas de esta capa son las mismas que fueron presentadas en el modelo supervisado. En la tabla 1 se muestra un pequeño conjunto de datos reales o continuos, los cuales no son convertidos a través de su paso por la capa de entrada.

En la capa de entrada la función de red es la función identidad ($x = f(x)$), lo que da como resultado que el dato que está ingresando sea el mismo que pasa a la función de activación de la neurona, mientras que la función de activación varia. La función de salida es también la función identidad ($x = f(x)$) con la finalidad de proyectar hacia el interior de la red el tipo de datos codificados por la función de activación.

	X1	X2	X3	X4	X5	X6	X7
Gen1	0.63360006	0.36395448	0.2775541	0.18370205	0.13851574	0.01478122	0.02231959
Gen2	0.6856	0.38101485	0.13915451	0.14589752	0.1281656	0.03239061	0.17677666
Gen3	0.5152	0.41951007	0.28107697	0.26799217	0.20017615	0.0416222	0.05573052
Gen4	0.65440005	0.42169732	0.24207346	0.21646598	0.13080817	0.03159018	0.05826373
Gen5	0.2416	0.22615924	0.1977856	0.18314198	0.13146883	0.14194238	0.88326716
Gen6	0.2824	0.279965	0.17664821	0.20666482	0.13014753	0.06259339	0.11926606

Tabla 1 Patrones de ingreso que mantienen sus valores reales después de haber sido proyectados al interior de la red.

El algoritmo para la conversión de datos continuos a binarios lo tomamos del artículo de Shmulevich [30], en el que trabajan también con datos de expresión genética, presentado ya en el capítulo 2.

En la tabla 2 se muestra la conversión a valores binarios correspondientes a los datos presentados en la tabla 1.

	X1	X2	X3	X4	X5	X6	X7
Gen1	1	1	1	0	1	0	0
Gen2	1	1	1	0	1	0	0
Gen3	0	0	0	0	0	0	0
Gen4	0	0	1	0	0	1	1

Tabla 2 Patrones de ingreso binarios correspondientes a los datos originales mostrados en la tabla.

La función escalón extendida para obtener datos ternarios [1]:

$f(x) = 0$ si $0.5 > x > -0.5$
$f(x) = 1$ si $x >= 0.5$
$f(x) = -1$ si $x =< -0.5$

En la tabla 3 se muestra la conversión a valores binarios correspondientes a los datos presentados en la tabla 1.

	X1	X2	X3	X4	X5	X6	X7
Gen1	1	0	0	0	0	0	0
Gen2	1	0	0	0	0	0	0
Gen3	1	0	0	0	0	0	0
Gen4	1	0	0	0	0	0	0
Gen5	0	0	0	0	0	0	1
Gen6	0	0	0	0	0	0	0

Tabla 3 Patrones de ingreso binarios correspondientes a los datos originales mostrados en la tabla.

[1] Esta función escalón fue tomada del artículo de Kim *et al.* 2000.

3.1.1.2 La capa oculta

La capa oculta, formada por los centroides o neuronas ocultas de la red, se utiliza para crear los cúmulos o clusters que representan a las clases de salida encontradas según las características de los patrones de expresión de genes. En la figura 2 podemos ver que esta capa se encuentra físicamente entre las capas de entrada y la de salida. Las neuronas de entrada se encuentran conectadas directamente a todas las neuronas creadas en la capa oculta y propagan a éstas los valores de los patrones presentados, en tanto que los centroides creados sólo se conectan a una sola neurona de la capa de salida, aquella que representa la clase asociada al cluster. Cuando la red es creada no existe la capa oculta (la red no posee memoria inicialmente), sino que se formará a partir de la presentación a la red de los patrones de expresión de genes. Lo anterior significa que las características topológicas del espacio neuronal oculto vendrán determinadas por las características particulares presentes en los patrones de entrada; de ellos depende el número de centroides creados así como la caracterización de los cúmulos o cluster por uno o más centroides. Esta capa de la RNA sufre diversas modificaciones durante la dinámica de la red, ya que los centroides se crearán, modificarán y eliminarán de acuerdo a las características dadas por los patrones. El grado de similitud que existe entre los centroides del mismo cluster es mayor que el que pueden tener con los centroides de otros clusters que representan a otras clases. Por lo anterior, para conjuntos de patrones muy similares entre sí se esperarían menos centroides para representar su clase.

Las funciones de activación de las neuronas ocultas se describen a continuación. La función de red es la distancia obtenida entre el patrón de entrada y el vector de peso del centroide. Las métricas de distancia a utilizar son: la de Manhatan, la Euclidiana y el coeficiente de correlación de Pearson, la de Mahalanobis y la de Hamming, las cuales son las funciones más frecuentemente reportadas en la literatura especializada. Estas funciones son las mismas que fueron utilizadas por la versión supervisada RNA (ver capítulo 2).

La función de salida que proyecta la información hacia la capa de salida está dada por la siguiente expresión:

salida=1/(distancia+1)

Según la anterior expresión, cuanto mayor es la distancia calculada entre el vector de pesos del centroide y el patrón de entrada, menor será el valor de salida proporcionado por la neurona oculta, como puede verse en la sección 2.1.1.3 del capítulo 2. El centroide que consiga computar una menor distancia al patrón de entrada resultará un candidato a vencedor y por lo tanto a recibir una recompensa la cual puede englobar su radio, vector de peso y coeficiente de vitalidad (como se verá más adelante en el algoritmo).

Para soportar la dinámica de red que se desarrolla a nivel de la capa oculta, las neuronas cuentan con tres parámetros principales: el vector de peso, el radio y el coeficiente de vitalidad. Cuando se crea un centroide el vector de peso toma el valor del vector del patrón de entrada que provocó la creación del centroide, mientras que los restantes dos parámetros, radio y coeficiente de vitalidad, vienen inicializados con los valores proporcionados como parámetros de la red. Los restantes centroides (aquellos que no lograron clasificar el patrón de entrada) son penalizados a través de una modificación que se efectúa en su coeficiente de vitalidad. El radio es un parámetro indispensable para llevar a cabo el proceso de clustering. Cada vez que se

computa la distancia entre el vector de peso del centroide y el patrón de entrada, el valor resultante es comparado con el radio y como resultado de esta comparación se decide cuál comportamiento de la capa oculta ejecutar (por ejemplo, crear un nuevo centroide, reducir el radio del centroide actual, etc.). Por otra parte, el coeficiente de vitalidad refleja la eficacia del centroide en la representación de patrones de entrada. Centroides con un coeficiente de vitalidad elevado resultan ser prototipos idóneos de clusters.

Existen dos factores de decisión, uno para el radio y otro para el coeficiente de vitalidad y contra estos son comparados tanto el radio como el coeficiente de vitalidad de los centroides. Esta comparación es realizada para determinar qué centroides serán eliminados por su poca utilidad en la clasificación de patrones y que pueden ser absorbidos por otros. Esto hace que la capa oculta cuente con los centroides necesarios de acuerdo a las características de los patrones de entrada.

3.1.1.3 La *capa de salida*

La capa de salida es la que muestra las diferentes clases a las que pertenecen los patrones de expresión de genes que se han presentado a la red, en este modelo no supervisado esta capa se va creando de acuerdo a las necesidades de clasificación dictaminadas por los patrones presentados a la red. A diferencia del modelo supervisado (ver capítulo 2) esta capa no existe cuando la topología inicial de la red es generada, sino que la misma comienza a generarse dinámicamente en el transcurso de la presentación de los patrones de expresión de genes a la red.

Cada neurona de la capa de salida se encuentra asociada a los centroides de uno y solo un cluster. Entre las neuronas de la capa de salida no se ha concebido algún proceso competitivo. Estas reciben el patrón producido por los centroides del cluster que representan a través de conexiones pesadas fijas y producen una activación y salida proporcional o igual a este valor.

La función inicial de cada neurona en la capa de salida es la suma pesada (ver figura 3), que representa el producto escalar del vector de entrada por el vector de pesos asociados a cada neurona de salida.

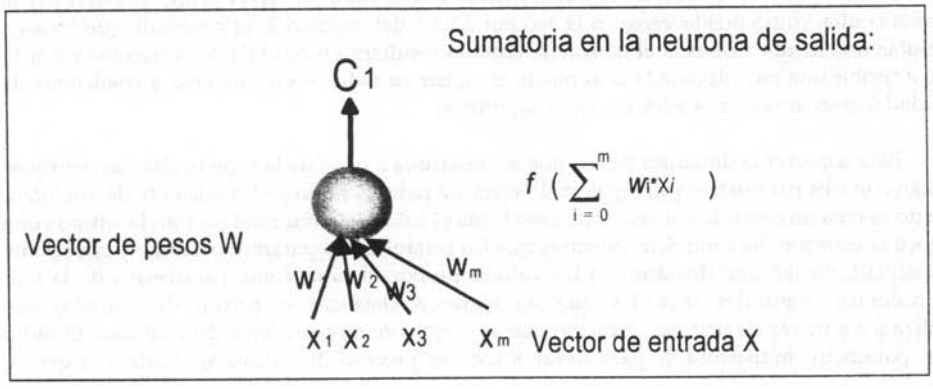


Figura 3 Función de activación en la capa de salida.

La expresión de la suma pesada es la siguiente:

$w_1x_1+w_2x_2+\ldots+w_mx_m$ que es igual a $\sum_{i=1}^{m} w_i x_i$

donde cada peso está representado por w y el valor de salida de cada neurona de la capa anterior es x con $1 \leq i \leq m$. *[21]*

La función de salida que obtiene el valor final de esta capa y por tanto de la red es la función identidad.

$f(x) = x$

La neurona de salida que proporcione el valor mayor indicará que a esa clase pertenece el patrón de expresión de genes que ha sido presentado a la RNA, se disparán todas las neuronas de salida pero una tiene el valor mayor, aquella que recibe el mayor impulso proveniente de la capa oculta donde por medio de la competencia entre los centroides se determina la clase del patrón. Esta es una de las diferencias con la red de mapa auto-organizado (ver capítulo 1), pues en ese algoritmo de competitividad se lleva a cabo en la capa de salida la competencia por la clasificación.

3.1.2 Conexiones entre las neuronas

En la RNA se establecen conexiones entre las diferentes capas de neuronas. Las neuronas de la capa de entrada se conectan siguiendo un patrón de completamente conectado con las neuronas de la capa oculta, es decir, cada neurona de la capa de entrada se conecta con todas las neuronas de la capa oculta. El patrón de conexión puede ser apreciado en la figura 4.

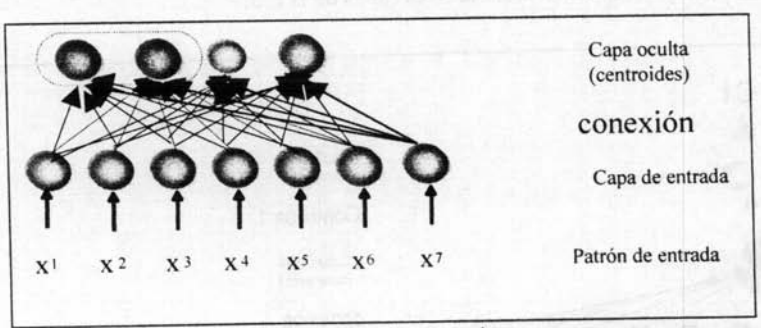


Figura 4 Conexión entre la capa de entrada y la capa oculta.

Entre las neuronas de las capas oculta y de salida la conexión no sigue el patrón de conexión completa, en este caso las neuronas pertenecientes a un determinado cúmulo de la capa oculta se conectarán con la neurona de la capa de salida que representa la clase de este

cúmulo o cluster, como puede verse en la figura 5. De esta forma, cada centroide de la capa oculta se conectará a una sola neurona de salida a la cual representa la clase a la cual pertenece el centroide.

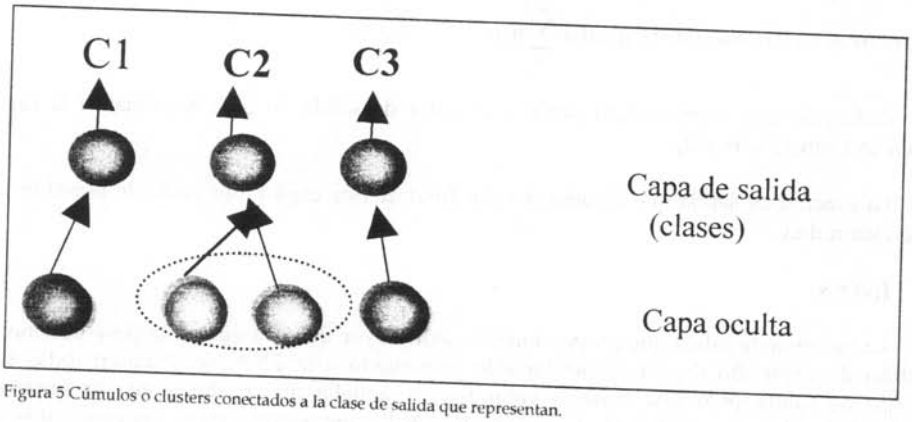


Figura 5 Cúmulos o clusters conectados a la clase de salida que representan.

Debido a que al inicio del proceso de clustering no se cuenta con centroides creados que formen cúmulos ni con neuronas de salida o clases, no es posible observar ningún patrón de conexión en la red. Cuando un centroide es creado en la capa oculta, el vector de peso del centroide es inicializado con los valores provenientes del vector de entrada del patrón de expresión de gene que ha provocado la creación del centroide. A partir del momento en que la capa de entrada reciba la información de un patrón de expresión de genes y realice la propagación del vector hacia el interior de la red se crearán las primeras conexiones en la red. Con el primer patrón presentado dará inicio la creación del primer centroide entonces la red mostrará la conexión de la capa de entrada y la capa oculta y a continuación se creará la primera neurona de salida que clasifica el patrón, obteniéndose así la conexión entre el primer centroide y la primera neurona de salida de la RNA. Lo anterior puede apreciarse en la figura 6, donde se ha establecido la primera conexión entre las tres capas de la red.

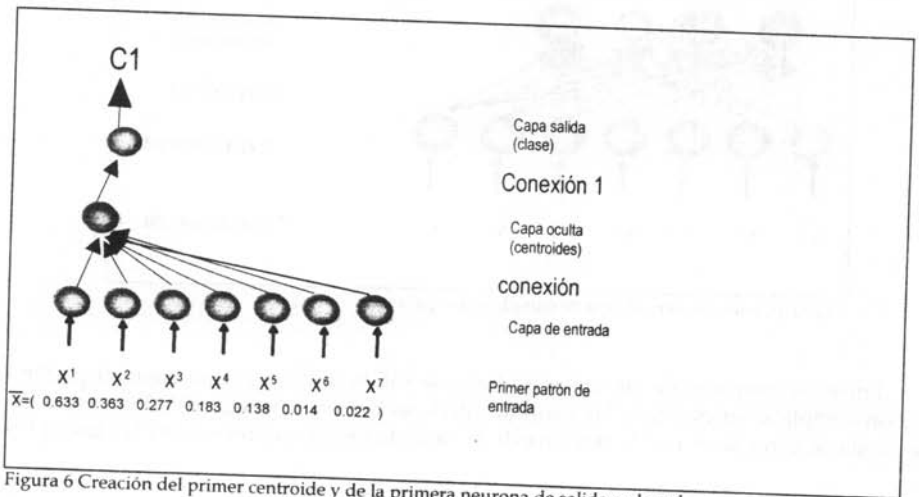


Figura 6 Creación del primer centroide y de la primera neurona de salida en la red.

A pesar de que en ambos modelos, supervisado y no supervisado, no existen conexiones al inicio del entrenamiento, la diferencia estructural entre ambos consiste en que en el modelo no supervisado la topología inicial de la red viene conformada solo por la capa de entrada.

3.1.3 Flujo de información entre las capas de la RNA

El flujo de información en el modelo neuronal es hacia adelante (del ingles, *feed-forward*). De las neuronas de una capa inferior se obtiene un valor de salida que se pasa a la siguiente capa para formar en su conjunto el vector de entrada, el flujo de esta información sólo es hacia adelante y no hay retropropagación de la información(ver capítulo 1). Lo anterior significa que los valores de salida de la capa de entrada pasan a la capa oculta y el resultado de esta primera representación interna hecha por la red es pasado a la capa de salida.

3.2 Funcionamiento de la red

3.2.1 Algoritmo no supervisado

Este algoritmo funciona con el conocimiento previo de los parámetros iniciales: dimensión de los patrones de entrada, valores de radio inicial y coeficiente de vitalidad, así como radio y coeficiente mínimos que determinen la permanencia de los centroides. En comparación con el modelo supervisado varía en que no se considera como parámetro inicial el conocimiento de las clases de los patrones. Los parámetros iniciales requeridos para el modelo no supervisado son los siguientes:

C_V	Coeficiente de vitalidad
R_P	Radio del centroide
R_o	Radio inicial
g_v	Constante de decisión para eliminación por R_P
β_v	Constante de decisión para eliminación por C_V
X	Patrón de entrada
N	Dimensión del patrón X
α	Factor para modificación del C_V

C_V Coeficiente de vitalidad, es una medida que proporciona información acerca de que tan frecuentemente ha resultado ganador el centroide en cuestión, lo que se traduce en una medida de su eficacia como representativo de patrones de expresión de genes.

R_P Radio del centroide, define la vecindad del centroide. Los patrones que se encuentran a una distancia del centroide menor a su radio vienen clasificados por este centroide.

R_o Radio inicial, es proporcionado por el usuario y utilizado cuando un nuevo centroide se crea.

g_v es el valor mínimo que puede tomar el radio, prototipos con un radio menor a este valor serán eliminados de la red.

β_v es el valor mínimo que puede tomar el coeficiente de vitalidad, prototipos con un C_V menor a este valor serán eliminados de la red.

X es el patrón de entrada y representa el vector con los valores de expresión de un mismo gene en N experimentos realizados.

d(X,P) es la distancia entre el patrón de ingreso y el vector de peso del centroide, constituye el criterio usado como base para definir los cúmulos de patrones.

P_i es el centroide que se está evaluando actualmente, con $1<i<n$.

WC cada centroide representan una clase y se indica con este valor, más de un centroide puede pertenecer a una misma clase, estos centroides forman los cúmulos o clusters.

CS al crear la neurona de salida se le asigna este valor el cual inicialmente puede ser un número entero y representa la clase asociada al cluster de la capa oculta.

α es el factor para la modificación del C_V de cada centroide.

El algoritmo de clasificación de patrones utilizado por la red es el siguiente:

1. Presentar a la red neuronal el patrón de expresión de gene, este patrón se presenta como un vector (X).

2. Verificar si existe ya un centroide en la capa oculta.

   Al inicio cuando se crea la red no existe ningún centroide o neurona en la capa oculta, por lo cual esta pregunta tiene una respuesta negativa sólo para el primer patrón presentado a la red. En el momento de propagar la información del primer patrón se crea la primera neurona o centroide en la capa oculta y también la primer neurona de salida.

   2.1 Si **no** existen centroides:

   Crear el primer centroide con los datos iniciales para el radio (R_P), coeficiente de vitalidad (C_v) dados por el radio inicial (R_o) y el factor de vitalidad (α) respectivamente, y el vector de pesos con los valores del vector de entrada.

   $R_P = R_o$, $C_v = \alpha$, $wp=X$ *y* $CS=PC=WC$

   Crear la primera neurona de la capa de salida con su conexión hacia la capa oculta.

   2.2 Si existen centroides o neuronas en la capa oculta:

   Calcular la distancia entre el patrón presentado a la red y cada centroide de la capa oculta,
   $d(X,P_i)$, $1 \leq i \leq m$

   Dicha distancia corresponde a la medida de similitud entre los dos vectores evaluados, el de pesos del centroide y el de entrada del patrón.

El centroide ganador es el que tenga menor distancia.

2.2.1 Si $d(X,P) < R_P$

El patrón es clasificado por el centroide ganador.
Esto quiere decir que la diferencia entre el patrón presentado y el vector de pesos del centroide es menor al radio del prototipo.

Entonces, el patrón toma la clase del centroide que lo clasifica PC=WC. Se actualiza el centroide ganador, reforzándolo para clasificar ese patrón en su vector de pesos y se incrementa el coeficiente de vitalidad (C_v), $C_v = C_v * \alpha^2$

2.2.2 Sino: $d(X,P) > R_P$

La menor distancia encontrada entre los centroides es mayor que el radio del centroide

Si $(d(X,P)-R_p) < R_o/2$
Fusiona el patrón al centroide que tuvo la menor distancia, por lo que tomará la clase este centroide PC=WC, y reforzará al centroide en su vector de pesos y con $C_v = C_v * \alpha^2$.

Sino :Si $(d(X,P)-R_p) < R_o$
Crear un centroide para clasificar al patrón, con la clase del centroide que tuvo la menor distancia, por lo que tomará la clase del cúmulo de este centroide.
$R_P = R_o$, $C_v = \alpha$, $wp=X$ y $PC=WC$

Sino: $(d(X,P) - R_p) >= R_o$
La mínima distancia reportada corresponde a un centroide que no clasifica al patrón presentado. Crear un centroide y una nueva clase de salida para ese centroide.

$R_P = R_o$, $C_v = \alpha$, $wp = X$ y $PC = WC = CS$

3. Modificar los centroides que no pudieron clasificar al patrón, penalizándolos a través de una reducción de su coeficiente de vitalidad.

$C_v = C_v / \alpha$

4. Verificar cuáles son los centroides a eliminar de acuerdo con los valores de su coeficiente de vitalidad y su radio. Se comparan esos valores con el radio mínimo y el factor de decisión para el coeficiente de vitalidad, proporcionados inicialmente por el usuario.

Si $C_v \leq \beta_v$ or $R_P \leq g_v$ entonces eliminar el centroide

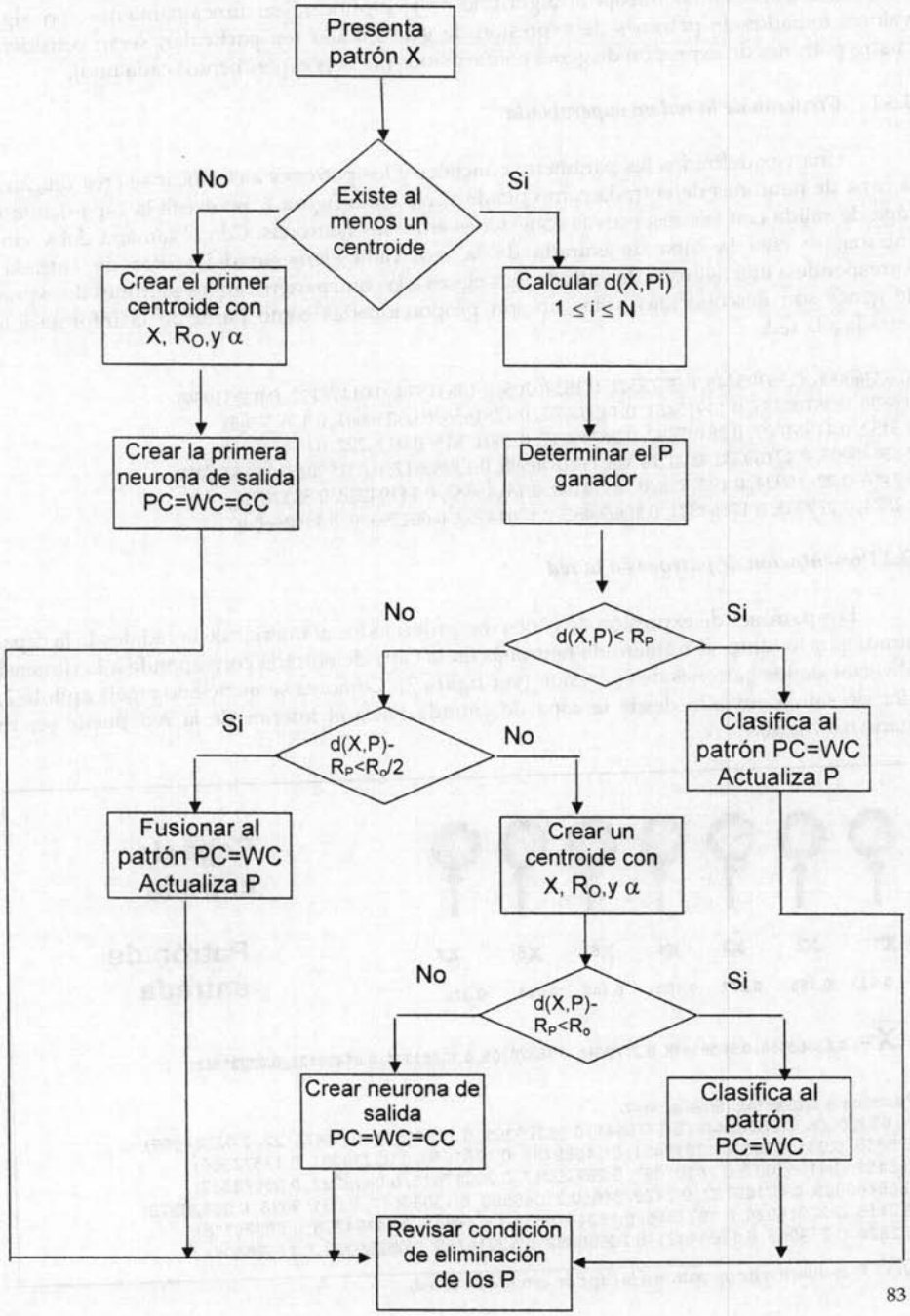
Si alguna de estas expresiones resulta verdadera se debe a que el centroide verificado ha sido utilizado para clasificar un número muy escaso (y muy posiblemente uno solo) de los

patrones presentados. Resultando poco eficiente contar con este tipo de centroides, ya que podría llevar al peor de los casos en que se tuviera un centroide por cada patrón presentado.

5. Repetir todos los pasos, 1-4 hasta que se presenten todos los patrones durante el número de épocas necesarias.

3.2.2 *Diagrama de flujo*

El siguiente diagrama ilustra el algoritmo antes presentado:



3.3 *Comportamientos principales de la red de acuerdo al algoritmo*

Para mostrar cómo trabaja el algoritmo se ejemplificará su funcionamiento con algunos valores tomados de patrones de expresión de genes reales (en particular, serán considerados cuatro patrones de expresión de genes conformados por seis experimentos cada uno).

3.3.1 *Creación de la red no supervisada*

Una vez definidos los parámetros iniciales y los patrones a clasificar se crea únicamente la capa de neuronas de entrada como puede verse en la figura 7, no existe la capa oculta ni la capa de salida con sus respectivas conexiones entre las neuronas. Con el tamaño del vector de entrada, se crea la capa de entrada de la red, cada elemento del vector de entrada (x_i) corresponde a una neurona de entrada. Las clases a las que pertenecen los patrones de expresión de genes son desconocidas, estas no son proporcionadas como parte de la información de entrada a la red.

(0.63360006, 0.36395448, 0.2775541, 0.18370205, 0.13851574, 0.01478122, 0.02231959)
(0.6856, 0.38101485, 0.13915451, 0.14589752, 0.1281656, 0.03239061, 0.17677666)
(0.5152, 0.41951007, 0.28107697, 0.26799217, 0.20017615, 0.0416222, 0.05573052)
(0.65440005, 0.42169732, 0.24207346, 0.21646598, 0.13080817, 0.03159018, 0.05826373)
(0.2416, 0.22615924, 0.1977856, 0.18314198, 0.13146883, 0.14194238, 0.88326716)
(0.2824, 0.279965, 0.17664821, 0.20666482, 0.13014753, 0.06259339, 0.11926606)

3.3.2 *Presentación de patrones a la red*

Los patrones de expresión de genes son proyectados al interior de la red desde la capa de entrada por lo tanto, el número de neuronas en la capa de entrada corresponde a la dimensión del vector de los patrones de expresión (ver figura 7). Como ya se mencionó en el capítulo 2, el valor de salida enviado desde la capa de entrada hacia el interior de la red puede ser real, binario o ternario.

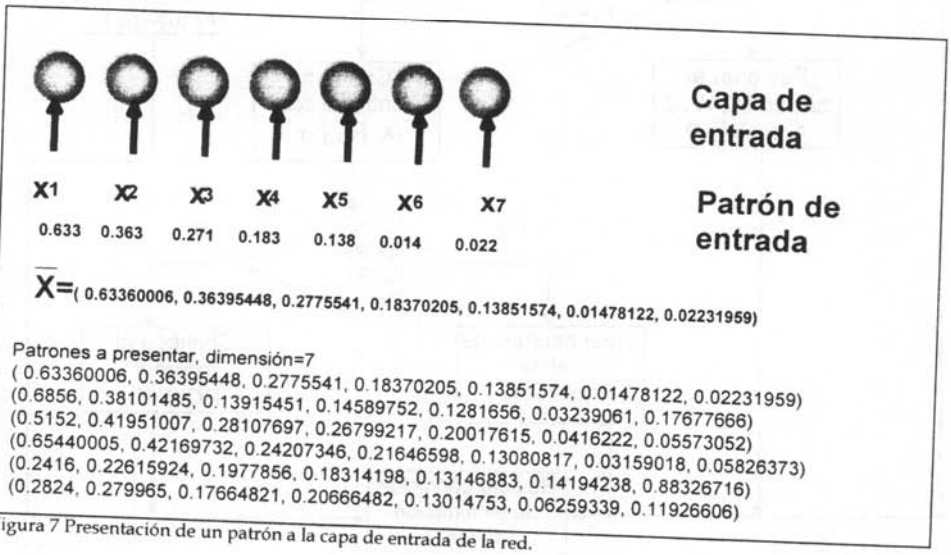


Figura 7 Presentación de un patrón a la capa de entrada de la red.

3.3.3 Creación de centroides en la red

Una vez que la capa de entrada recibe el patrón de expresión del gene, la información es propagada hacia todas las neuronas de la capa oculta. Si no existe ninguna neurona en la capa oculta se crea el primer centroide con los valores iniciales dados como parámetros y con un vector de pesos igual a los valores de entrada, en la figura 8 puede verse cuando se crea el primer centroide de la red.

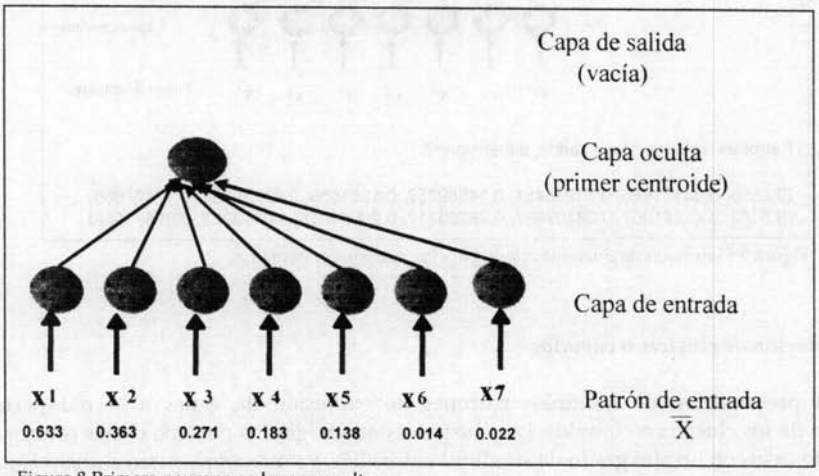


Figura 8 Primera neurona en la capa oculta.

En seguida, es creada la neurona de salida con una clase (CS) que es asignada como la clase para el centroide (WC) y para el patrón presentado (PC).

$$R_P = R_v,\ C_v = \alpha,\ wp = X\ y\ PC = WC = CS$$

Durante la dinámica de la red no supervisada los centroides son creados cuando el patrón presentado no encuentra algún centroide que lo pueda clasificar, y puede deberse a:

a) No existe ningún centroide en la RNA

b) Si $d(X,P) \geq R_P$

El primer caso se da cuando la RNA acaba de crearse y no hay centroides en la capa oculta, como se dijo anteriormente. En el segundo caso, existen centroides en la capa oculta pero no es posible encontrar alguno que clasifique correctamente al patrón de entrada así que se crea otro centroide, en la figura 9 ese centroide pertenece a otra clase de salida que tampoco existía por lo que también se crea una nueva clase de salida.

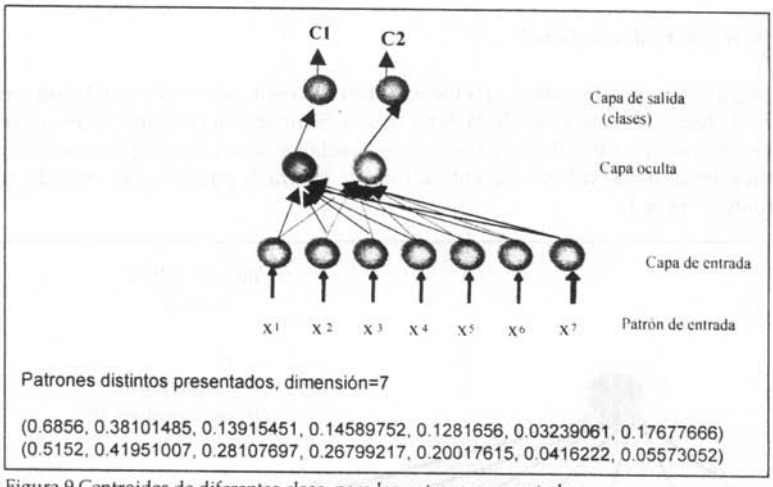


Figura 9 Centroides de diferentas clase, para los patrones presentados.

3.3.4 *Formación de clusters o cúmulos*

La presentación de múltiples patrones de expresión de genes a la red permite la formación de los clusters o cúmulos. Los clusters están integrados por uno o más centroides, en este último caso con un alto grado de similitud entre ellos, ya que representan a una misma clase de salida, mientras que el grado de similitud con centroides que representan otras categorías diferentes es menor. En la figura 10 se ejemplifican 2 clusters integrantes de la capa oculta; un cluster puede estar representado por una sola neurona cuando los patrones que clasifica tienen un alto grado de similitud. Dentro de la dinámica en la red en el modelo no supervisado consideramos que los centroides de un cúmulo no tienen de diferencia entre ellos un valor superior al radio inicial.

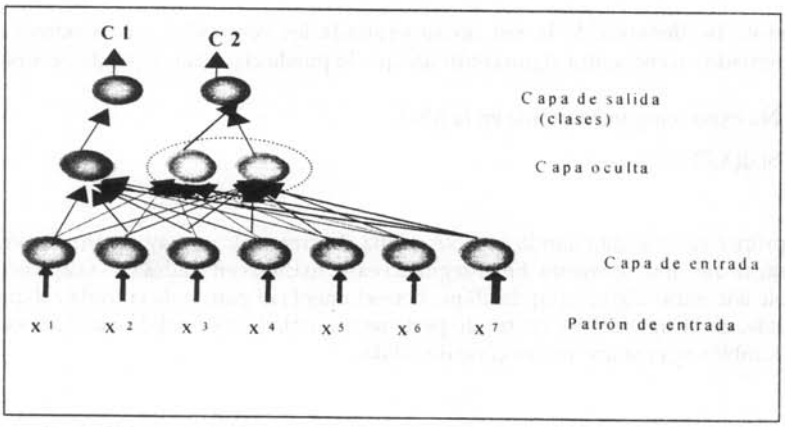


Figura 10 Creación de clusters o cúmulos.

3.3.5 Obtención de las medidas de similitud

Al igual que en el modelo supervisado, las medidas de similitud permiten distinguir la semejanza que existe entre el patrón presentado y los centroides existentes en la red para lograr la clasificación del patrón. Esta medida de similitud es obtenida por las distancias presentadas antes en la sección 1.1.2, el usuario determina, de acuerdo al objetivo de su estudio biológico y al tipo de datos (binarios, reales o ternarios) de los vectores, la medida que le pueda proporcionar mayor y mejor información. Cuando existen centroides en la capa oculta, cada vez que se presente un patrón de entrada se obtiene la distancia o diferencia entre los vectores del patrón presentado y el vector de pesos de cada centroide, para determinar posteriormente qué centroide es el ganador de la competencia por la clasificación del patrón.

3.3.6 Establecimiento del centroide ganador

Entre todas las diferencias obtenidas por la medida de similitud aplicada a los patrones de entrada y a cada centroide se elige la menor considerando a ese centroide como el ganador. Esa diferencia, distancia, es comparada con el radio (R_P) de dicho centroide para determinar finalmente si el centroide ganador clasifica o no al patrón. De acuerdo a la decisión tomada será posible reforzar al centroide ganador o crear uno nuevo que represente correctamente al patrón de entrada. En el algoritmo no supervisado consideramos diferentes casos a evaluar cuando se ha obtenido el centroide ganador:

a) Si $d(X,P) < R_P$
b) Si $d(X,P) \geq R_P$

Para cada caso es necesario tomar la decisión de si el centroide ganador clasifica o no al patrón presentado a la red.

3.3.7 Fusión de centroide y patrón

Si $d(X,P) \geq R_P$ entonces obtener
  Si $(d(X,P)-R_P) < R_o/2$
  entonces fusionar el patrón con el centroide

Al fusionar el patrón recibe la misma clase a la que clasifica el centroide, el centroide es actualizado con el incremento de su C_v y su vector de pesos (wp) es reforzado.

3.3.8 Incremento del coeficiente de vitalidad (C_v) en el centroide ganador

Cuando se considera que el centroide ganador clasifica correctamente al patrón de entrada, entonces se refuerza el centroide en coeficiente de vitalidad: C_v, con ello se indica que el centroide es utilizado en la representación de patrones, por lo que justifica su existencia.

El incremento del coeficiente de vitalidad es:

$$C_v = C_v * \alpha^2$$

Se incrementa el C_v con relación al factor de incremento α

Además se actualiza su vector de pesos (*wp*) con

$wp = wp + factor * d(X, P_i)$, donde P_i es el centroide ganador y X el vector de entrada.

El *factor* varía, es un parámetro cuyo valor es menor en los casos de fusión y mayor en otro caso.

3.3.9 *Modificación del vector de pesos en el centroide ganador*

El centroide que ha resultado ganador modifica su vector de pesos para reforzarlo y contribuir al aprendizaje, como sucede en las redes neuronales. La modificación se hace incrementando en una proporción la diferencia del vector de pesos actual con respecto al vector de entrada.

$wp = wp + factor * d(\overline{X}, P_i)$, donde P_i es el centroide ganador y $\overline{X}$ el vector de entrada.

El *factor* varía, es un parámetro cuyo valor es menor en los casos de fusión y mayor en otro caso.

3.3.10 *Modificación de centroides perdedores*

A los centroides que no representan al patrón y que son considerados como perdedores se les decrementa el coeficiente de vitalidad (C_v). Con ello se va debilitando el centroide que no representa ese patrón.

Decremento del coeficiente de vitalidad en los centroides:

$$C_v = C_v / \alpha$$

3.3.11 *Verificación de Cv y Rp para eliminación de centroides*

Cada vez que se presenta un patrón se compara al final el coeficiente de vitalidad y el radio de todos los centroides existentes en la capa oculta con los parámetros de decisión para eliminación de centroides. Aquí es donde repercute el incremento o decremento que sufrieron esos atributos del centroide durante la presentación de patrones. Las comparaciones realizadas son: el C_v contra la constante de decisión para eliminación del coeficiente de vitalidad (β_v) y, el R_P contra la constante de radio mínimo (g_v). Para decidir eliminar un centroide es suficiente que resulte cierta alguna de las dos comparaciones siguientes:

$C_V \leq \beta_v$ o $R_P \leq g_v$

Si resulta cierta la evaluación se eliminará el centroide porque significa que sólo representa a uno o a un número muy pequeño de patrones que podrían incluirse en otro centroide de esa clase.

3.3.12 *Creación de neuronas de salida en la red*

En este modelo no supervisado, como ya se mencionó, las neuronas de salida se crearán con la dinámica de la red. La primera neurona de salida se crea con el primer centroide, como puede verse en la figura 11, puesto que ese centroide permite clasificar para una clase, cuando se crean cúmulos en la capa oculta se crean, como siguiente paso, también neuronas de salida. Finalmente existen tantas neuronas de salida como cúmulos formados en la capa oculta.

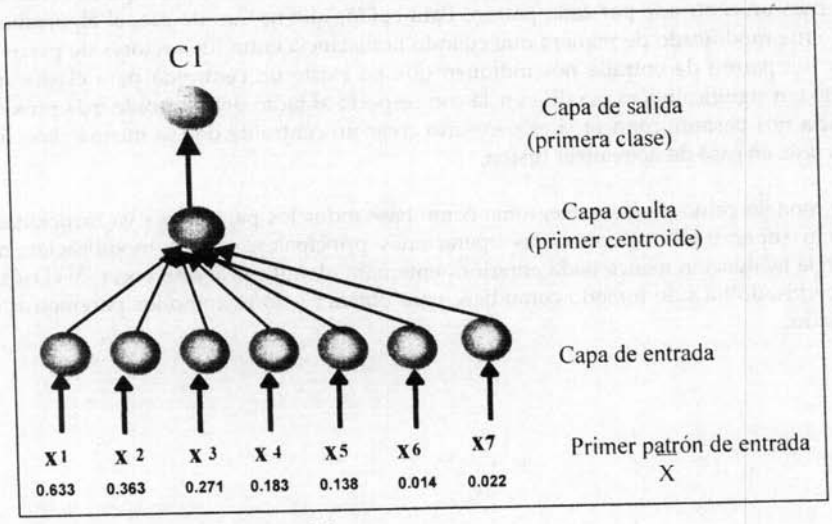


Figura 11 Creación de capa de salida.

3.3.13 *Evolución de la RNA*

La RNA evoluciona hasta establecer su estructura más adecuada a la muestra de patrones presentada, se crean las neuronas de salida necesarias para clasificar a los patrones y los centroides que permitan representar por medio de cúmulos a esas clases. Cada neurona de salida está unida a sus centroides que le permiten clasificar correctamente a los patrones de su clase. Se considera que la red ya está entrenada cuando después de presentar los patrones de la muestra la modificación sufrida en la capa oculta es nula o mínima.

3.4 Conclusiones

El modelo presentado ahora ha sido modificado para aceptar patrones que no cuentan con la clase a la que pertenecen, debiendo la red encontrar este dato por medio de la aceptación de los patrones por lo centroides de la capa oculta.

La clasificación en este modelo no supervisado es más compleja que en el supervisado, pues la red debe evaluar a partir de la diferencia encontrada entre el vector de pesos de los centroides y el vector de entrada, si la clase del patrón ya existe o debe crearse. Este punto es muy importante, porque una mala clasificación podría darnos como resultados extremos que cada patrón pertenece a una clase distinta o todos a la misma, y dentro de la capa de centroides podría crearnos un centroide por cada patrón. Para cuidar que no suceda eso, el algoritmo no supervisado fue modificado de manera que cuando la distancia entre los vectores de pesos del centroide y del patrón de entrada nos indiquen que no existe un centroide para clasificar se evaluará qué tan significativa es esa diferencia con respecto al radio del centroide más próximo. Esa diferencia nos permite conocer si es necesario crear un centroide de esa misma clase o de una nueva clase, en caso de no realizar fusión.

Este modelo como puede verse, toma como base todos los parámetros ya mencionados en el modelo supervisado, así como las operaciones principales, con su modificación más relevante en la evaluación mencionada anteriormente para identificar la diferencia de clases. El modelo supervisado ha sido tomado como base para obtener valores como los parámetros del no supervisado.

Capítulo 4.
Diseño orientado a objetos del modelo neuronal

El diseño del modelo neuronal propuesto, en sus dos diferentes versiones, supervisado y no supervisado, se realizó utilizando una metodología orientada a objetos. El modelo orientado a objetos nos permite identificar claramente los elementos necesarios para integrar nuestra red neuronal, así como sus comportamientos y las relaciones que pueden darse entre ellos. Para el modelado orientado a objetos se utilizó la Técnica de Modelado de Objetos (OMT), creada por James Rumbaugh [37]. La fase del desarrollo se materializa en la creación de tres modelos diferentes del sistema:

- Modelo de objetos
- Modelo dinámico
- Modelo funcional

El modelo de objetos pretende obtener una representación de la estructura estática de los objetos que conforman la red neuronal, lo mismo que de sus relaciones, por medio de los diagramas de clases y objetos. El modelo dinámico abstrae los aspectos del sistema que están sujeto a cambios continuos, utilizando diagramas de estados. El modelo funcional define las transformaciones que se producen en los valores de los datos dentro de la red, a través de diagramas de flujo de datos.

En el presente capítulo se mostrarán algunos ejemplos del diseño de la red neuronal realizado para la implementación del sistema del modelo de red neuronal propuesto.

4.1 El modelo de objetos

En este modelo se definen claramente las entidades que intervienen en el sistema para la red neuronal. Los diagramas utilizados en este modelo son grafos cuyos nodos corresponden a objetos de las clases y los arcos definen las relaciones entre ellas. Con los diagramas de este modelo pueden representarse gráficamente los objetos y clases, atributos, operaciones y relaciones y asociaciones. El modelo estático de objetos presenta también las clases en que están agrupados, las cuales describen los atributos similares, identificando sus relaciones comunes con otros objetos en una semántica común. Los objetos de una clase tienen los mismos atributos y los mismos patrones de comportamientos.

4.1.1 Clases de objetos

Dentro de nuestro modelo neuronal se tienen como clases principales las neuronas de las capas de entrada, oculta y de salida, la de supervisor, la de red y la generadora. La clase NEURONA es abstracta, es decir que en ella se definen propiedades o atributos y algunas

operaciones de forma general, dejando el refinamiento para la implementación concreta de la clase, en nuestro caso, para sus subclases: Nentrada, Ncentroide y Nsalida, que representan a las neuronas de las capas de entrada, oculta y de salida, respectivamente, como se muestra en el diagrama de la figura 1.

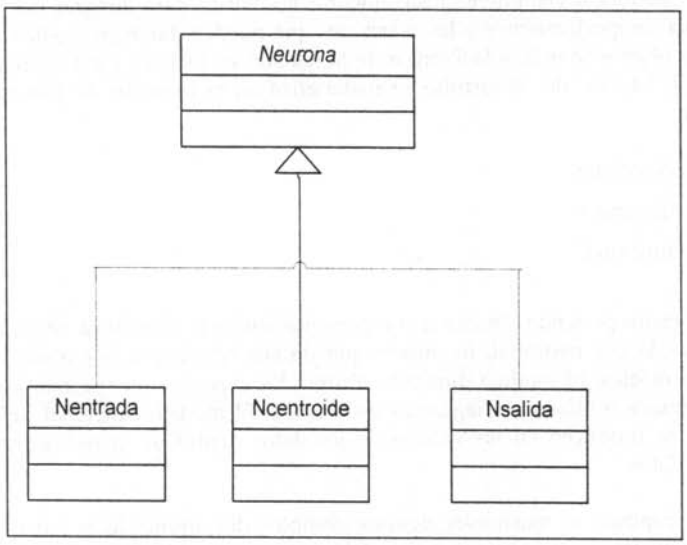


Figura 1 Diagrama jerárquico de clases.

4.1.1.1 Los atributos

Los atributos corresponden a las características o propiedades que definen a los objetos. Los atributos comunes a los objetos de las neuronas de las tres capas (entrada, oculta y salida) de la RNA se encuentran definidos en la clase abstracta neurona. Mas, como se muestra en la figura 2, para cada objeto de las neuronas de entrada, de los centroides y de las neuronas de salida se requieren además otros atributos que diferencian cada uno de los objetos neuronales en la capa respectiva.

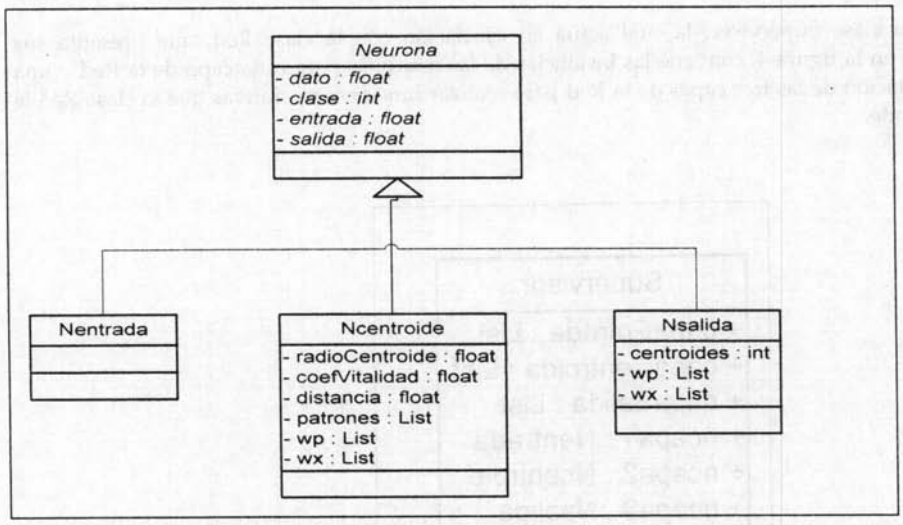


Figura 2 Atributos de los objetos de las clases neuronas.

La clase Red tiene como atributos los parámetros iniciales proporcionados para la formación de las capas de entrada, oculta y de salida como el número de neuronas de entrada requeridas para los patrones que se presentarán, las clases que puede reconocer la Red, parámetros para la creación, existencia y eliminación de centroides, entre otras, lo que se encuentra indicado en la figura 3.

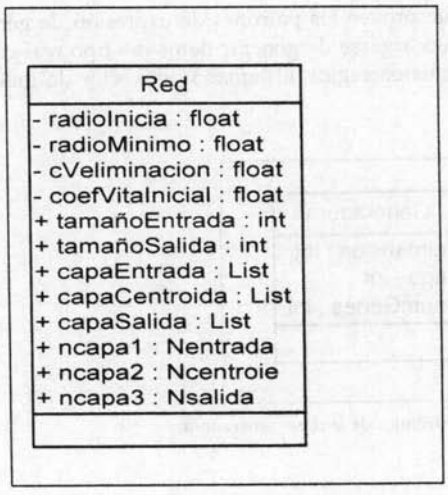


Figura 3 Atributos de los objetos de la clase Red.

La clase Supervisor, la cual actúa en asociación con la clase Red, que presenta sus atributos en la figura 4, contiene las instancias de las neuronas para cada capa de la Red y una representación de las tres capas de la Red para realizar funciones operativas que la clase Red le encomiende.

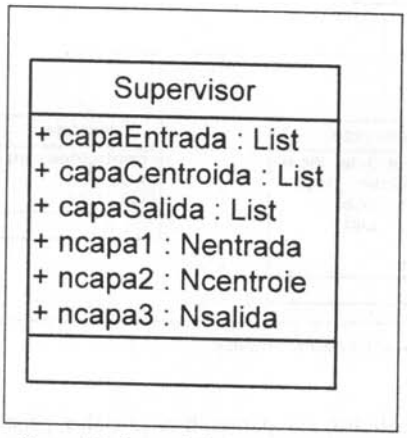


Figura 4 Atributos de la clase Supervisor.

La clase Generadora, que provee los patrones de expresión de genes, con los atributos mostrados en la figura 5 para encargarse de generar datos del tipo real correspondientes a los experimentos realizados a los microarreglos, el tamaño del vector de muestra y el número de muestras.

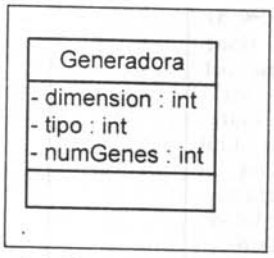


Figura 5 Atributos de la clase Generadora.

La clase Tabla contiene los atributos de los datos que representan la muestra de patrones de expresión de genes, como el número de experimentos medidos para integrar el perfil de expresión de un número específico de genes correspondiente al atributo de filas de la tabla, el tipo de patrones supervisados o no supervisados y su característica para normalizar o no los datos. La figura 6 muestra los atributos de esta clase.

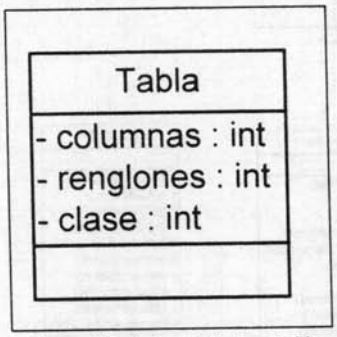


Figura 6 Atributos de la clase Tabla.

4.1.1.2 Los métodos

Los métodos son la implementación de las operaciones o transformaciones aplicadas en y por los objetos de una clase. Los métodos o comportamientos que se incluyen en la clase abstracta Neurona son, como puede verse en la figura 7, aquellos correspondientes a la obtención de datos de entrada y cálculos de funciones internas de activación y transformación de datos, los cuales son implementados de manera específica en las subclases Nentrada, Ncentroide y Nsalida, que son métodos abstractos en la clase superior. Mientras que los métodos para mostrar datos sí se implementan en la superclase Neurona ya que su comportamiento es general para cualquier tipo de neurona.

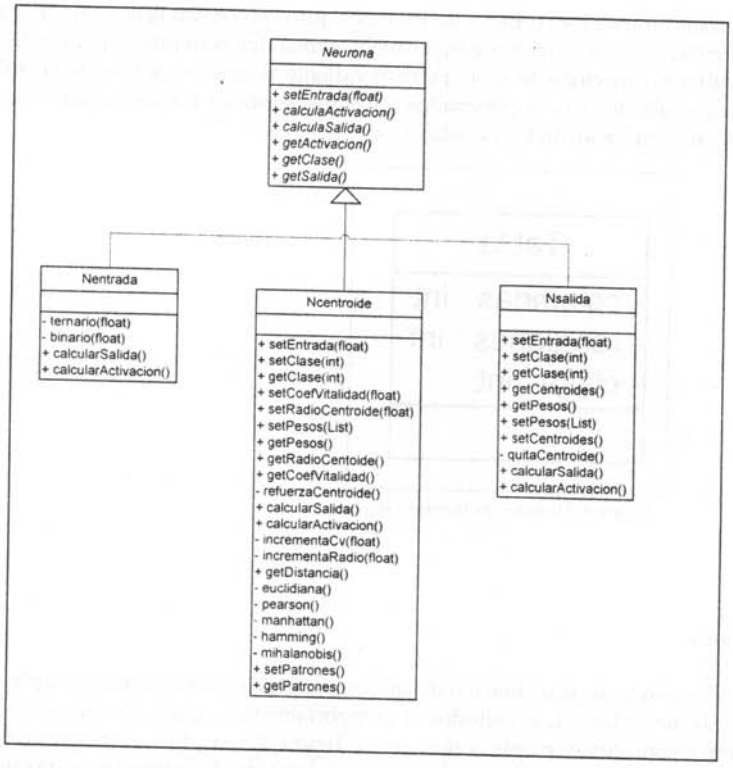


Figura 7 Métodos de los objetos neuronas de entrada, centroides y de salida.

Los métodos de la clase Red son aquellos que corresponden a la manipulación directa de las neuronas de cada capa que la integra, como lo son la creación de las neuronas de las capas de entrada, de salida y la oculta, además de la eliminación y fusión de centroides, así como la destrucción de clases no necesarias. Estos métodos se muestran en la figura 8.

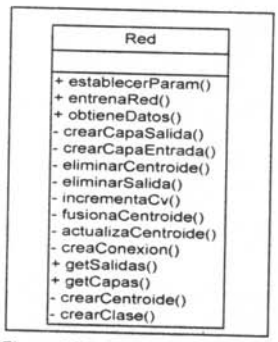


Figura 8 Métodos de la clase Red.

La clase Supervisor contiene los métodos necesarios para manipular la información obtenida por el patrón de expresión de genes en las tres capas para llevar a cabo la clasificación. Como puede verse en la figura 9, la clase Red se encarga de dirigir sus operaciones.

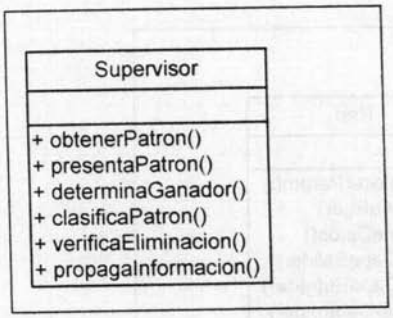


Figura 9 Métodos de los objetos de la clase Supervisor.

La clase Generadora encargada de preparar los patrones de expresión de genes para ser presentados a la Red, tiene los métodos, como puede verse en la figura 9, para leer desde un archivo los datos, capturarlos en tablas, ejecutar las conversiones de los datos a tipo binario o ternario

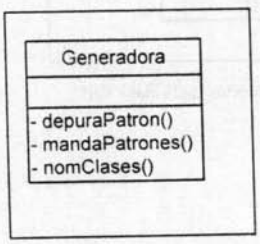


Figura 9 Métodos de la clase Generadora.

Los métodos de la clase Tabla, presentados en la figura 10, corresponden a las operaciones requeridas para lectura y escritura de datos desde archivos, la normalización de datos para dejarlos dentro del rango 0 a 1.

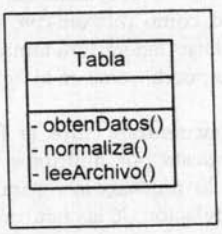


Figura 10 Métodos de la clase Tabla.

La clase Red es la que lleva el control de todo el funcionamiento de la red neuronal por lo que presenta los métodos necesarios para la manipulación de las demás clases, principalmente de las correspondientes a las neuronas de entrada, ocultas y de salida. En la figura 11 puede verse la lista de métodos que la componen.

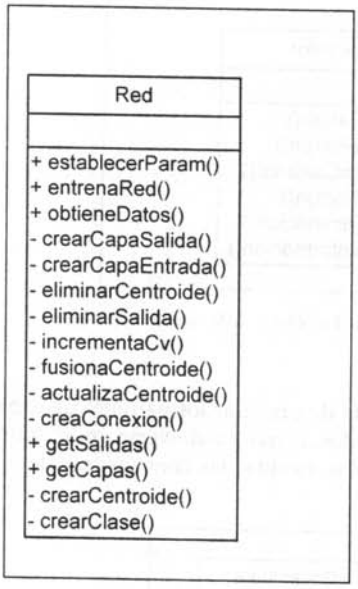


Figura 11 Métodos de la clase Red.

4.1.2 Asociaciones o relaciones

Las asociaciones entre las clases determinan el comportamiento del sistema, es una parte básica puesto que mediante las relaciones definimos la forma en que los objetos se comunican, lo que también se denomina comportamiento. Las relaciones son conexiones físicas o conceptuales entre casos concretos de objetos.

Entra la clase Red y la clase Supervisor existe una asociación uno-a-uno, del lado de la Red se ve como una instancia de la Red *requiere* una instancia de Supervisor, mientras que de parte de la clase Supervisor hacia la Red como *colabora-con*, es decir, que una instancia del Supervisor *colabora-con* una de la Red. La clase Generadora también presenta una relación uno-a-uno con la clase Supervisora. Gráficamente puede verse en la figura 12b

Las asociaciones establecidas se encuentran entre la Red y las neuronas (Nentrada, Ncentroide y Nsalida), la Red está compuesta por múltiples neuronas de esas clases y esas neuronas forman parte de la Red, como lo muestra la figura 12a es una asociación de tipo agregación. Entre las capas neuronales, la relación de las neuronas de entrada con las de la capa oculta es de muchos a muchos, todas las neuronas de entrada tienen relación directa con las

neuronas de la clase oculta, es decir, con las Ncentroide, por lo que es de muchos a muchos. Entre la capa de entrada y la de salida no existe relación en este tipo de red. Las neuronas de salida mantienen relación con neuronas de la capa oculta, de uno a muchos, ya que una neurona de salida puede relacionarse con muchas Ncentroides, pero una Ncentroide sólo puede relacionarse con una Nsalida, indicando así a la clase que representa. Lo anterior se muestra en la figura 12b.

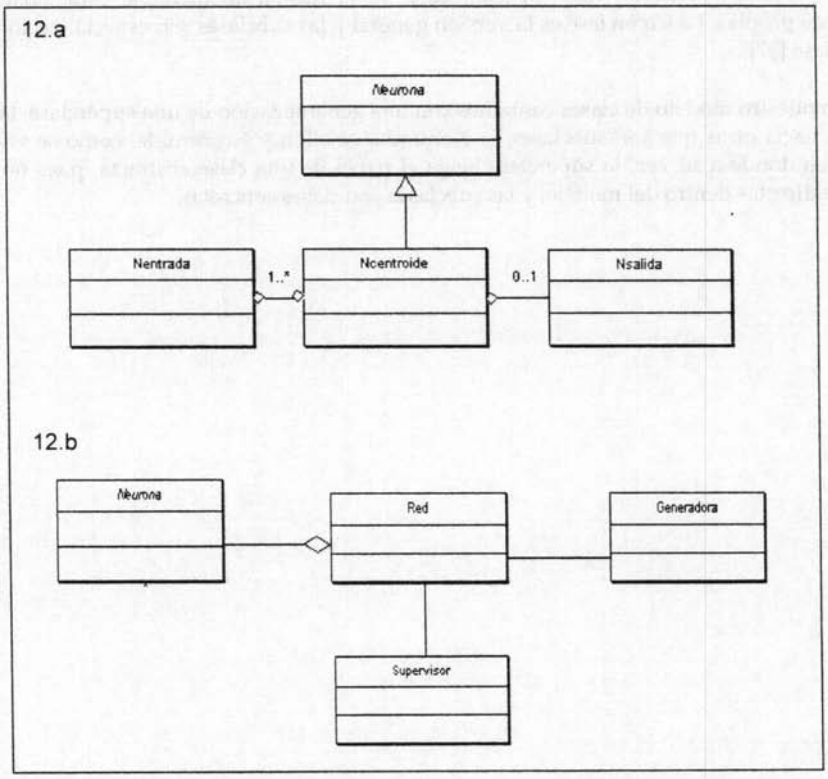


Figura 12 Asociaciones entre las clases del modelo neuronal, supervisado y no supervisado.

4.1.3 Herencia - generalización

En el paradigma orientado a objetos es muy importante la herencia, cualidad por medio de la cual los objetos heredan los atributos y los métodos dentro de una estructura jerárquica. La generalización es la relación que existe entre una clase y las subclases que se derivan de la misma, se generan versiones más especializadas de la misma añadiéndole características y operaciones propias. La superclase es la versión general y las subclases son especializaciones de la superclase [37].

En nuestro modelo de clases contamos con una generalización de una superclase, la clase Neurona, hacia otras que son subclases, la Nentrada, Nsalida y Ncentroide, como se ve en la figura 1, en donde a su vez, la superclase juega el papel de una clase abstracta, pues no tiene instancias directas dentro del modelo, y las subclases son clases concretas.

El modelo estático completo de las clases de objetos de todo el modelo neuronal queda de la siguiente forma:

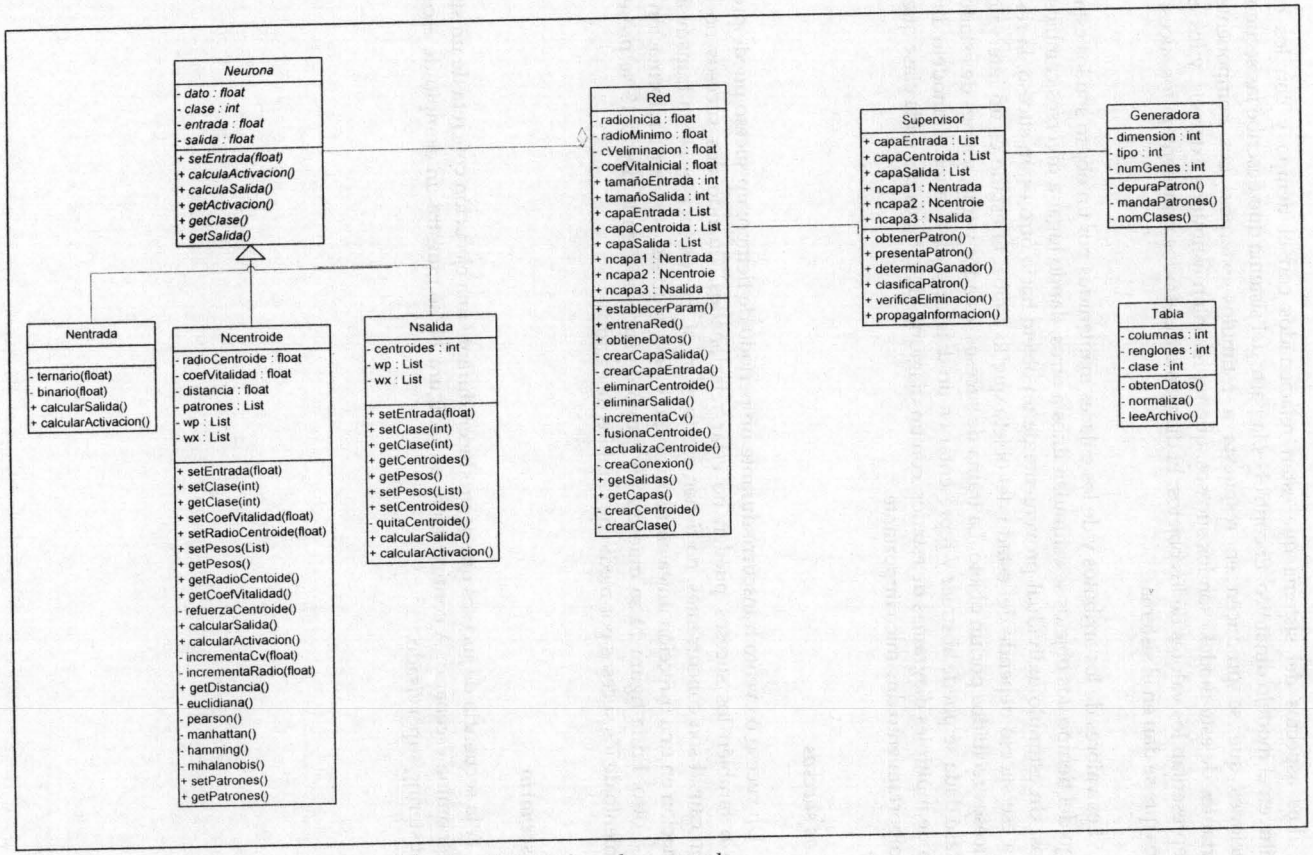


Figura 13 Diagrama de clases del modelo completo de la red neuronal.

4.2 El modelo dinámico

Los aspectos del sistema que están relacionados con el tiempo y con los cambios constituyen el modelo dinámico. El control es la parte del sistema que describe las secuencias de operaciones que se producen en respuesta a estímulos externos. Los componentes más importantes de este modelo son los sucesos, que representan estímulos externos, y los estados, que representan los valores de los objetos. El diagrama de estados representará los sucesos y los estados que se dan en el sistema.

Los valores de los atributos y de los enlaces mantenidos por un objeto son los estados. A lo largo del tiempo, los objetos se estimulan unos a otros, dando lugar a diversos cambios en los estados. Un estímulo individual proveniente de un objeto hacia otro es un suceso, la respuesta dada a ese suceso depende del estado del objeto que lo recibe. El estado es un intervalo entre dos sucesos recibidos por un objeto. La trama de sucesos, estados y transiciones de estados para una clase dada se puede abstraer y representar en un diagrama de estados. El modelo dinámico consta de múltiples diagramas de estados, con un diagrama de estados por cada clase que posea un comportamiento dinámico importante.

4.2.1 *Los sucesos*

Un suceso o evento transcurre durante un período de tiempo; un suceso puede preceder a otro o también los sucesos pueden no estar relacionados. Cuando dos sucesos no tienen relación causal son concurrentes, no tienen efecto entre sí. En los sucesos se da transmisión de información en una dirección única entre un objeto y otro, todo suceso aporta información de un objeto a otro. En la figura 14 se muestra un grupo de sucesos y en la figura 15 se presenta el seguimiento de los sucesos por medio de cambios de estados.

4.2.2 *Escenario*

A la secuencia de sucesos que se producen durante una ejecución concreta de un sistema se le denomina escenario. A continuación, en la figura 14, se muestra un ejemplo de escenario para clustering supervisado.

Tomar un patrón de expresión de genes
Cada neurona de la capa de entrada toma un valor del patrón presentado
Realizan cálculos en las neuronas de entrada
Salen los valores obtenidos de cada neurona de entrada
Se propagan hacia las siguientes capas esos valores
Se calcula la distancia con cada centroide
Elige el centroide ganador
Refuerza el centroide ganador
Inhibición de los otros centroides, perdedores
Salen los valores hacia la capa de salida
Refuerza el valor de la neurona de la clase del centroide ganador
Se muestra la clase a la que pertenece el patrón de expresión de genes

Figura 14 Escenario para el clustering supervisado de un patrón de expresión de genes.

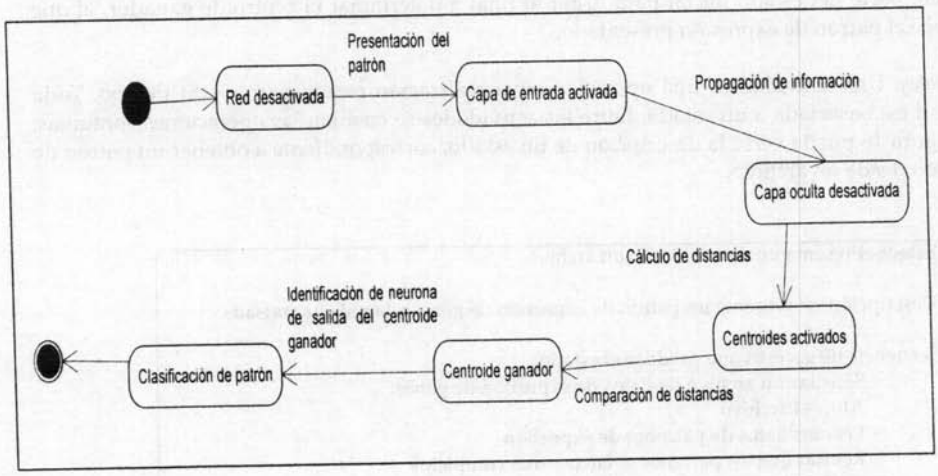


Figura 15 Diagrama de estados para encontrar el centroide ganador.

4.2.3 Estados

Un estado es una abstracción de los valores de los atributos y de los enlaces de un objeto. Los conjuntos de valores se agrupan dentro del estado de acuerdo con las propiedades que afectan el comportamiento del objeto. Un estado especifica la respuesta del objeto a los sucesos entrantes. La respuesta de un objeto a un suceso puede incluir una acción o un cambio de estado por parte del objeto.

4.2.3.1 Diagrama de estados

Mediante el diagrama de estados se relacionan sucesos y estados. Cuando se recibe un suceso, el estado siguiente depende del actual, así como del suceso; un cambio de estado causado por un suceso es lo que se denomina transición. El diagrama de estados es un grafo en donde los nodos son los estados y los arcos son las transiciones dadas por sucesos. Dado que todo objeto posee sus propios valores de atributos, cada objeto posee su propio estado, que es el resultado de la especial secuencia de sucesos que haya recibido. Todo objeto es independiente de los demás objetos.

El modelo dinámico es una colección de diagramas de estados que interactúan entre sí a través de sucesos compartidos. En este modelo se describe un conjunto de objetos concurrentes, cada cual con su propio estado y con su propio diagrama de estados. Los objetos son inherentemente concurrentes y pueden cambiar de estado de manera independiente. En la figura 15 se muestra un ejemplo de diagrama de estados del funcionamiento del modelo neuronal, parte del estado inicial para llegar al final a determinar el centroide ganador, al que pertenece el patrón de expresión presentado.

Actividades. Una actividad es una operación cuya realización requiere un cierto tiempo. Toda actividad está asociada a un estado. Entre las actividades se cuentan las operaciones continuas. En la figura 16 puede verse la descripción de un estado, correspondiente a obtener un patrón de expresión desde un archivo.

Estado: Presenta un patrón desde un archivo

Descripción: se presenta un patrón de expresión de genes a la red supervisada

Secuencia de sucesos que produce el estado:
- Solicitar un archivo de datos de expresión de genes
- Abrir el archivo
- Leer los datos de patrones de expresión
- Revisar que los patrones se encuentren completos
- Colocar los patrones en la tabla
- Condición del estado:
  - Red = no creada
  - Clases de patrones = presentes
- Se cumple condición:
  - Crear capas de entrada y salida

Figura 16 Descripción de un estado.

4.3. El modelo funcional

El modelo funcional muestra la forma en que se derivan los valores producidos en un cálculo a partir de los datos introducidos, sin tener en cuenta el orden en el cual se calculan esos valores. Consta de múltiples diagramas de flujo de datos que muestran el paso de valores desde las entradas externas, a través de operaciones y almacenes internos de datos, hasta las salidas externas. También incluyen restricciones entre valores dentro del modelo de objetos.

4.3.1 Diagrama de flujo de datos (DFD)

Un diagrama de flujo de datos es un grafo que muestra el flujo de valores de datos desde sus fuentes en los objetos mediante procesos que los transforman hasta sus destinos en otros objetos. En estos diagramas se encuentran los procesos que transforman datos, flujos de datos que los trasladan, objetos actores que producen y consumen datos, y los almacenes de datos que guardan los datos pasivamente.

Proceso. El proceso se encarga de transformar los valores de los datos, un proceso puede tener efectos laterales si contiene componentes no funcionales.

Flujo de datos. El flujo de datos conecta la salida de un objeto o un proceso con la entrada de otro objeto o proceso. Representa un valor de datos intermedio dentro de un cálculo que no es modificado por el flujo de datos.

Almacén de datos. Un almacén de datos es un objeto pasivo dentro de un diagrama de flujo de datos donde se guardan éstos para su posterior utilización, se limita a responder a solicitudes de almacenamiento y acceso de datos.

En los siguientes diagramas se muestran ejemplos del flujo de datos del modelo neuronal y en la figura 17 se muestra la descripción de la función clasificación de patrones del modelo de clustering supervisado y no supervisado, el DFD nivel 0 para la realización de clustering por parte de la red neuronal. En la figura 18 se puede apreciar el DFD nivel 1 para el clustering. En la figura 19 el DFD nivel 2 para el clustering supervisado y en la figura 20 el DFD nivel 2 para el clustering supervisado referente a la clasificación del primer patrón.

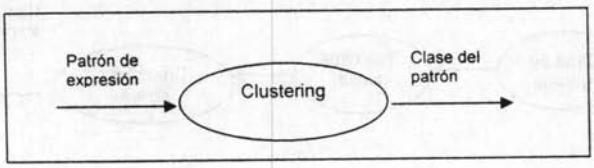


Figura 17 Diagrama de flujo de datos del modelo neuronal.

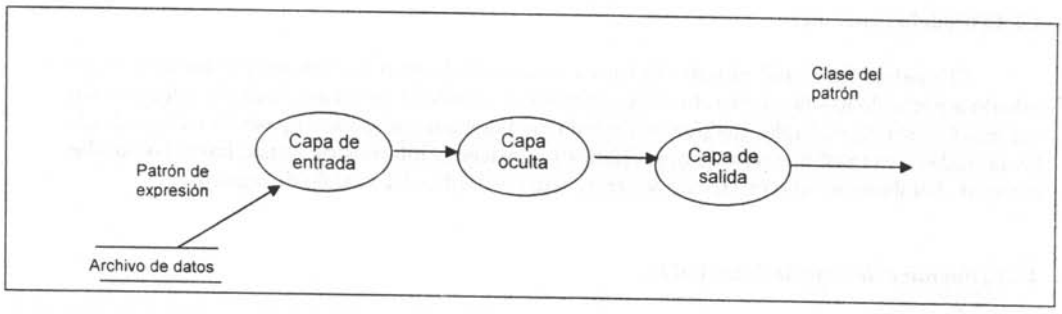


Figura 18 Diagrama de flujo de datos nivel 1 para clustering.

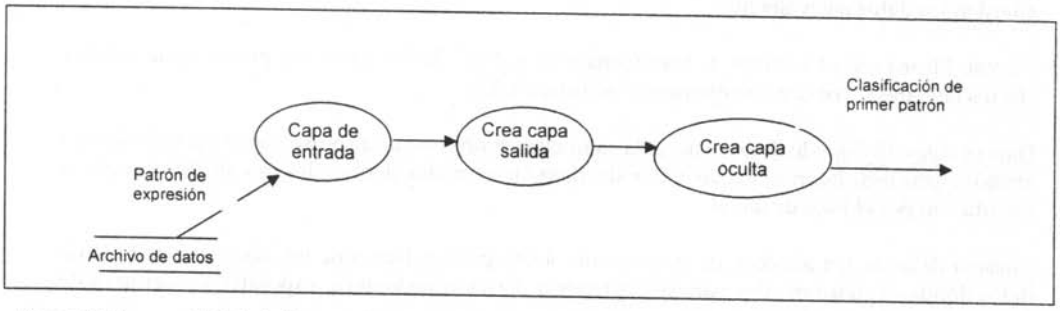


Figura 19 Diagrama de flujo de datos nivel 2 para clustering supervisado.

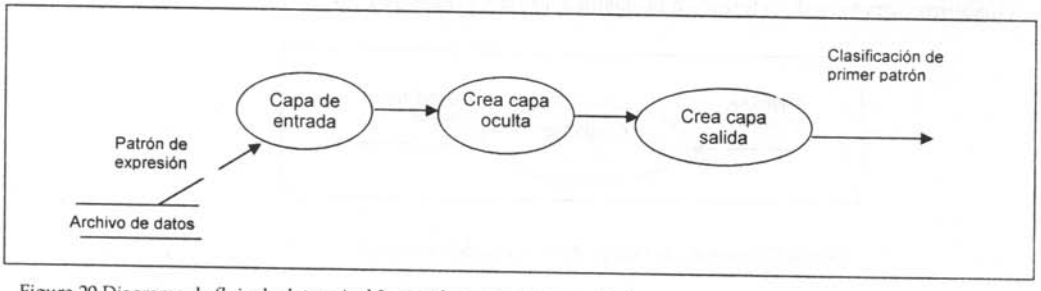


Figura 20 Diagrama de flujo de datos nivel 2 para clustering no supervisado.

El diagrama de flujo de datos para presentación de un patrón a la capa de entrada de la Red puede verse en la figura 21.

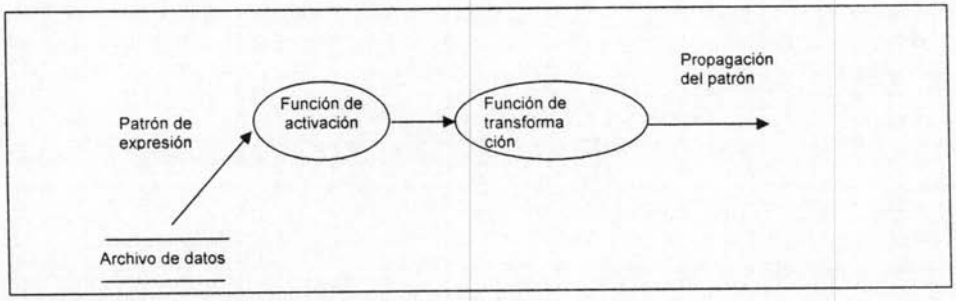


Figura 21 DFD para una neurona en la capa de entrada.

La descripción de la realización de clustering por el modelo neuronal se realiza como se expresa en la figura 22, en la cual se hace una descripción funcional.

```
Abrir archivo de datos
Leer los patrones de expresión de genes del archivo
Mientras existan patrones de expresión:
        Presentar el patrón de expresión a la Red
        Colocar los valores del patrón en las neuronas de entrada
        Aplicar funciones de activación y transformación de las neuronas
        Sacar valor de transformación de cada neurona de entrada
        Propagar datos hacia las demás capas de la red
        Determinar centroide ganador
        Reforzar centroide ganador
        Inhibir centroides perdedores
        Presentar salida de centroides a las neuronas de salida
        Mostrar clase del patrón
 Fin-mientras
Fin
```

Figura 22 Descripción de la función clasificar patrón.

El desarrollo del modelo por medio de la metodología OMT hace que la implementación del sistema sea sencillo ya que se utiliza también un lenguaje de programación que pertenece al paradigma de programación Orientado a Objetos, Java. Una vez que el análisis y diseño se concretan adecuadamente, con los niveles de abstracción necesarios, el paso al lenguaje Java es utilizar esa misma abstracción para la creación de las clases de objetos necesarios para el modelo de red neuronal propuesto.

Capítulo 5.
Pruebas y experimentos: clasificación de patrones de expresión de genes

En los capítulos anteriores se presentó un nuevo modelo de red neuronal que responde a la tarea de clasificación y clustering no supervisado de patrones de expresión de genes. Los resultados obtenidos a través de los experimentos realizados con ambas formas de aprendizaje se presentan en este último capítulo, donde puede apreciarse el funcionamiento del sistema o software que implementa el modelo neuronal. Los algoritmos presentados en los capítulos 3 y 4 así como el modelo orientado a objetos de la red están plasmados en el software que se presenta en este capítulo. En el capítulo se realizan diversos experimentos con datos obtenidos de bases de datos de patrones de expresión de genes que pueden encontrarse en la Internet y que pertenecen a datos reales publicados por instituciones especializadas. Por medio de la demostración del funcionamiento del modelo neuronal con patrones reales se verificara la funcionalidad del modelo neuronal. Además se presentan algunas comparaciones con otros modelos neuronales que también trabajan con esos patrones de expresión de genes. Los resultados del aprendizaje dependen de diversos factores como los parámetros de la red, el tipo de datos (binario, ternario o real), la distancia elegida (euclidiana, mahalanobis, manhathan) y la normalización de los datos. Sin embargo, se trató de hacer los experimentos en condiciones semejantes a la de otros modelos con el fin de poder realizar comparaciones útiles.

En primer lugar será presentada la red trabajando con patrones de expresión que cuentan con la clase a la que pertenecen, lo que sugiere utilizar el modelo supervisado. Posteriormente se mostrara el funcionamiento de la red con patrones de expresión que no están acompañados por la clase a la que pertenecen, en donde se mostrara el desempeño de la red no supervisada. Además de ver como funciona el modelo propuesto con estos dos métodos de aprendizaje es importante revisar como actúa con las diferentes medidas de distancia.

5.1 Los patrones de expresión de genes

Los patrones de expresión de genes que se utilizan en estos experimentos fueron obtenidos en la Internet en diversas fuentes, procurando que ya hubiesen sido utilizados por métodos referidos en la introducción con la finalidad de permitirnos hacer ciertas comparaciones.

Elegimos 3 muestras de patrones de expresión de genes, las cuales se presentan en la fase de experimentos y en tablas en el anexo A. De estos datos obtuvimos los datos originales extraídos del microarreglo, sin haberse normalizado. Para realizar la normalización de los datos recurrimos a nuestro algoritmo. Los patrones de expresión de genes utilizados para los experimentos corresponden a bases datos publicadas en la Internet, de levaduras y de linfoma obtenidos de los sitios: http://www.broad.mit.edu/cancer/, http://gepas.bioinfo.cnio.es

Estos sitios se eligieron ya que proporcionan los resultados o el software para analizar los perfiles de expresión de genes.

5.2 La interfaz del software de aplicación

La aplicación está hecha en Java 1.3, lo cual permite correrla tanto en plataforma Windows como Linux. En primer lugar, al ejecutar la aplicación se selecciona el tipo de clustering que se llevará a cabo, supervisado o no supervisado, como muestra la figura 1. De acuerdo a la elección del tipo de clustering se espera que en la tabla de datos los patrones se encuentren acompañados de la clase a la que pertenecen, en el caso supervisado, y en el otro caso sólo se mostrarán los perfiles de expresión de los patrones. La importancia de ejecutar el modelo de clustering supervisado consiste en que nos permite conocer a priori los valores de determinados parámetros de la red neuronal, aquellos que contienen conocimiento biológico del dominio del problema.

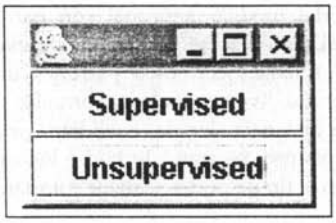


Figura 1 Elección entre clustering supervisado y no supervisado.

La siguiente acción a realizar para el uso del software es cargar una tabla que cuente con los patrones de expresión de genes, como se ve en en la figura 2. En la figura 3, se presenta una tabla con datos obtenidos de un archivo con patrones de expresión de genes, esto puede ser a través de la lectura de un archivo de texto que sea cargado en la tabla o el llenado de la tabla escribiendo los datos.

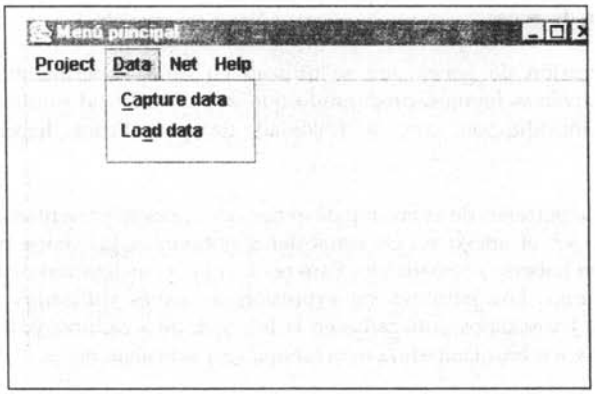


Figura 2 Obtención de tabla de datos con perfiles de expresión de genes.

En la tabla obtenida se representan en las filas los diferentes genes y los experimentos realizados por las columnas. Los datos completos de esta muestra pueden verse en el anexo A tabla 1.

Esta tabla representa la matriz de patrones de expresión de genes, compuesta por n renglones de genes y m columnas de mediciones, que integran el perfil de expresión del gene, como se observa a continuación:

$$\begin{pmatrix} G_{11} & G_{12} & G_{13}\ldots\ldots G_{1m} \\ G_{21} & G_{22} & G_{23}\ldots\ldots G_{2m} \\ G_{31} & G_{32} & G_{33}\ldots\ldots G_{3m} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ G_{n1} & G_{n2} & G_{n3}\ldots\ldots G_{nm} \end{pmatrix}$$

En el caso de que algún gene no cuente con el total de mediciones para los experimentos, ese gene es desechado y no se incluye en la matriz de datos, así sólo se tendrán datos de los genes que cuenten con un perfil completo para todos los experimentos hechos.

Después de aplicar la normalización, los datos quedan como se muestra en la figura 3, la normalización hecha con la expresión (1.0) permite que los valores de la matriz queden en el rango de [0...1], lo que es más eficiente para el cálculo de distancias en el clustering. En este caso se realiza la normalización columna por columna.

$$c = \frac{c_i - c_{min}}{c_{max} - c_{min}} \tag{1.0}$$

donde C_i es el valor original de la columna, C_{min} es el valor mínimo de la columna y C_{max} es el valor más grande de la columna. Esta forma de normalización evita el dominio de algún elemento.

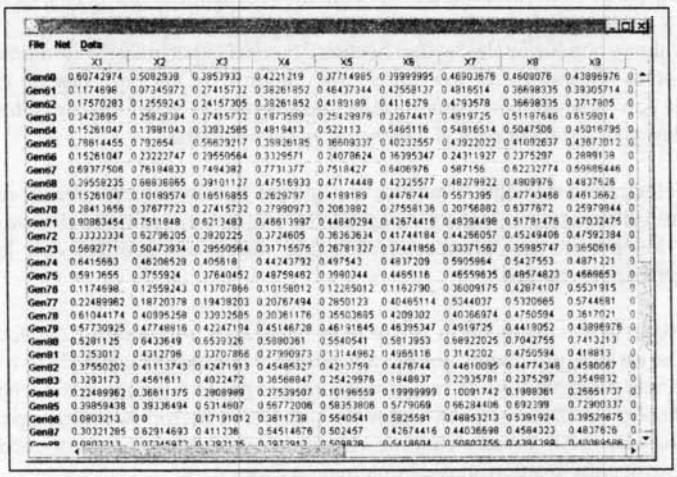


Figura 3 Columnas y filas correspondientes a los perfiles de expresión de genes obtenidos en los experimentos, después de normalizar.

Los datos presentados en la tabla muestran el número de genes cuya expresión fue medida, un renglón por gene, y el número de experimentos en los que se observo la expresión de todos los genes, dados por las columnas, y la clase a la que se cree pertenece el perfil de expresión de los genes, la ultima columna corresponde a la clase. Hablamos de una creencia porque al presentar los patrones a la red puede darse el caso de que la red lo clasifique en otra clase distinta lo que indica que puede existir un error inicial.

Ya que se cuenta con los datos es necesario proporcionar los parámetros iniciales para ejecutar la aplicación y que inicie la clasificación. Los parámetros son proporcionados por medio de la siguiente ventana presentada en la figura 4.

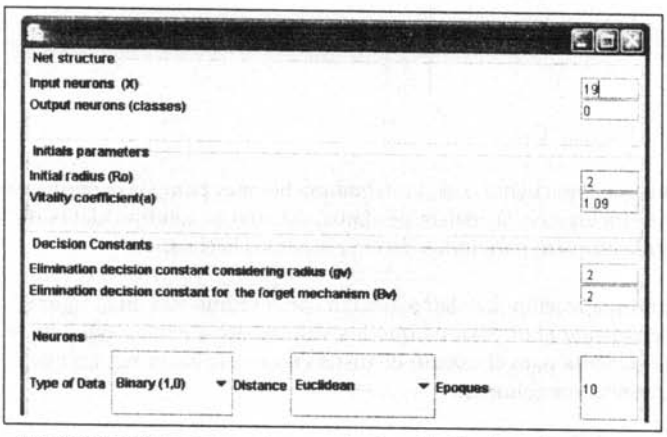


Figura 4 Pantalla para proporcionar parámetros iniciales de configuración

Los valores dados para la estructura de la red corresponden al número de neuronas de entrada, que coincide con la cantidad de experimentos realizados a los genes, es decir, el número de columnas; y la cantidad de clases o neuronas de salida. El radio inicial y coeficiente de vitalidad son valores con los cuales se han hechos diversos experimentos (tomando valores entre 0.2 y 0.5) y por la naturaleza de los datos provenientes de los patrones de expresión se ha encontrado que con esos valores responde adecuadamente la red. El tipo de datos a manejar son continuos, reales, y la distancia para el clustering es la euclidiana.

Ahora elegimos la opción crear red con la que físicamente se crea la red neuronal, es decir, se forman las capas de entrada y de salida de la red con el número de neuronas necesario de acuerdo a los parámetros proporcionados que deben ser acordes con los patrones de expresión de genes presentados.

Al finalizar la ejecución de la red puede mostrarse un conjunto de datos estadísticos, como se muestra en la figura 5, que ayudan a verificar el comportamiento de la red. Por ejemplo, el total de patrones presentados a la red, el número de épocas o iteraciones hechas con todos los patrones, el número de clase formadas, cuantos centroides fueron necesarios para clasificar los patrones de una clase. Al conocer cuantos centroides se requirieron para cada clase se ve la similitud que existe entre los patrones de una misma clase.

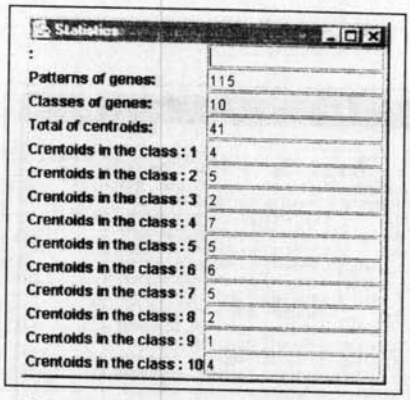


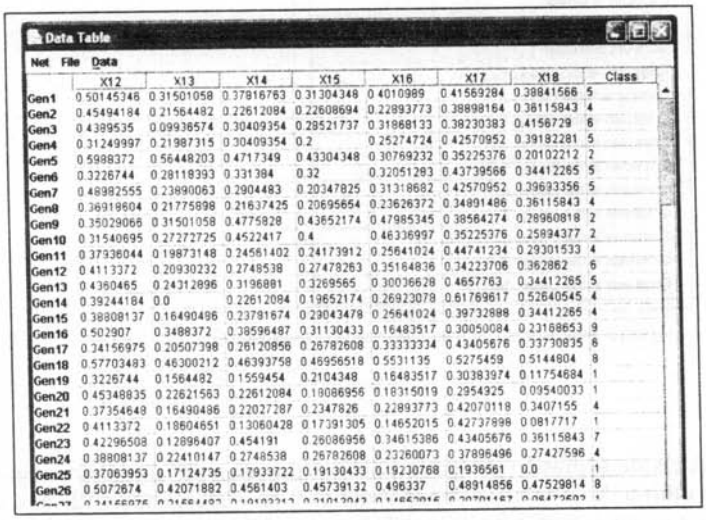
Figura 5 Estadísticas con resultados de los experimentos.

5.3 Experimentos

Utilizando bases de datos de expresión de genes públicas provenientes de microarreglos realizamos experimentos utilizando el modelo supervisado y el no supervisado. Además se realiza una comparación con otros software desarrollados para el análisis de patrones de expresión de genes.

5.3.1 *Clustering supervisado*

La muestra utilizada consta de 109 perfiles de expresión de genes medidos en 18 experimentos. Los perfiles van acompañados por su clase, como presenta la figura 6, por esta razón podemos realizar el clustering supervisado, cuya característica es que el algoritmo de clustering es guiado por la clase a la que pertenecen los patrones.



Data Table

Net File Data

	X12	X13	X14	X15	X16	X17	X18	Class
Gen1	0.50145346	0.31501058	0.37816763	0.31304348	0.4010989	0.41569284	0.38841566	5
Gen2	0.45494184	0.21564482	0.22612084	0.22608694	0.22893773	0.38898184	0.36115843	4
Gen3	0.4389535	0.09936574	0.30409354	0.28521737	0.31868133	0.38230383	0.4156729	6
Gen4	0.31249997	0.21987315	0.30409354	0.2	0.25274724	0.42570952	0.39182281	5
Gen5	0.5988372	0.56448203	0.4717349	0.43304348	0.30769232	0.35225376	0.20102212	2
Gen6	0.3226744	0.28118393	0.331384	0.32	0.32051283	0.43739566	0.34412265	5
Gen7	0.48982555	0.23890063	0.2904483	0.20347825	0.31318682	0.42570952	0.39633356	5
Gen8	0.36918604	0.21775898	0.21637425	0.20695654	0.23626372	0.34891486	0.36115843	4
Gen9	0.35029066	0.31501058	0.4775828	0.43652174	0.47985345	0.38564274	0.28960818	2
Gen10	0.31540695	0.27272725	0.4522417	0.4	0.46336997	0.35225376	0.25894377	2
Gen11	0.37936044	0.19873148	0.24561402	0.24173912	0.25641024	0.44741234	0.29301533	4
Gen12	0.4113372	0.20930232	0.2748538	0.27478263	0.35164836	0.34223706	0.362862	6
Gen13	0.4360465	0.24312896	0.3196881	0.3269565	0.30036628	0.4657763	0.34412265	5
Gen14	0.39244184	0.0	0.22612084	0.19652174	0.26923078	0.61769617	0.52640545	4
Gen15	0.38808137	0.16490486	0.23791674	0.23043478	0.25641024	0.39732898	0.34412265	4
Gen16	0.502907	0.3488372	0.38596487	0.31130433	0.16483517	0.30050084	0.23188653	9
Gen17	0.24156975	0.20507398	0.26120856	0.26782608	0.33333334	0.43405676	0.33730835	8
Gen18	0.57703483	0.46300212	0.46393758	0.46956518	0.5531135	0.5275459	0.5144804	8
Gen19	0.3226744	0.1564482	0.1559454	0.2104348	0.16483517	0.30383974	0.11754684	1
Gen20	0.45348835	0.22621563	0.22612084	0.19086956	0.18315019	0.2954325	0.09540033	1
Gen21	0.37354648	0.16490486	0.22027287	0.2347826	0.22893773	0.42070118	0.3407155	4
Gen22	0.4113372	0.18604651	0.13060428	0.17391305	0.14652015	0.42737898	0.0817717	1
Gen23	0.42296508	0.12896407	0.454191	0.26086956	0.34615386	0.43405676	0.36115843	7
Gen24	0.38808137	0.22410147	0.2748538	0.26782608	0.23260073	0.37896496	0.27427596	4
Gen25	0.37063953	0.17124735	0.17933722	0.19130433	0.19230768	0.1936561	0.0	1
Gen26	0.5072674	0.42071882	0.4561403	0.45739132	0.496337	0.48914856	0.47529814	8

Figura 6 Datos de perfiles de expresión de genes acompañados con su clase.

Las siguientes tablas presentan los resultados de la aplicación de clustering supervisado a los datos mencionados, se crean 9 clases, las que ya eran presentadas por la muestra de genes. Sin embargo, en los experimentos varía el número de centroides, estos centroides representan una mayor similitud entre ciertos patrones de cada clase, pues al modificar los parámetros el resultado de la agrupación se modifica. Esa modificación de parámetros para la formación de agrupaciones o de clusters permite a los usuarios llegar al resultado que ellos consideren idóneo de acuerdo al conocimiento previo que tengan del problema a resolver (en este caso, el conocimiento biológico). En las tablas siguientes, 1a y1b se presentan los valores de los parámetros para clasificar con datos reales, variando los parámetros para comparar los resultados en el número de centroides necesarios para clasificar en 9 cúmulos o clases a los 109 patrones.

Radio inicial	0.2
Coeficiente de vitalidad	1.08
Constante de eliminación para el radio	0.2
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Euclidiana
Datos de tipo	Reales
Número de patrones de expresión	109
Clases creadas	9
Número de centroides	36

Tabla 1a Creación de 9 clusters y 36 centroides con valores de punto flotante

El número de centroides es mayor que el de las clases, como se ve en las tablas 1a y 1b, esto se explica porque nuestro modelo de clustering permite extraer clases de diferenciación entre los patrones, pero dentro de cada clase se hace un segundo nivel de comparación, ya que existen patrones más semejantes unos que otros en la misma clase lo cual es representado por los centroides, así en una clase puede haber más de un centroide. Esta característica del modelo permite obtener una clasificación más refinada.

Radio inicial	0.5
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.8
Constante de eliminación para el coeficiente de vitalidad	0.55
Distancia	Euclidiana
Datos de tipo	Reales
Número de patrones de expresión	109
Clases creadas	9
Número de centroides	27

Tabla 1b Creación de 9 clusters y 27 centroides con valores de punto flotante.

Las clases formadas por los centroides quedan como se muestra en la figura 7, cada pequeña gráfica corresponde a un centroide se puede observar el número de la clase a la que pertenece la gráfica, en la tabla inferior de la figura tenemos el número de centroides por clase y el número de patrones que clasifico cada clase. El grupo de centroides forma un cúmulo o cluster y se identifica con una clase. En la tabla 2 se indican los resultados de todas las clases con sus centroides y patrones de los clusters ilustrados en la figura 7.

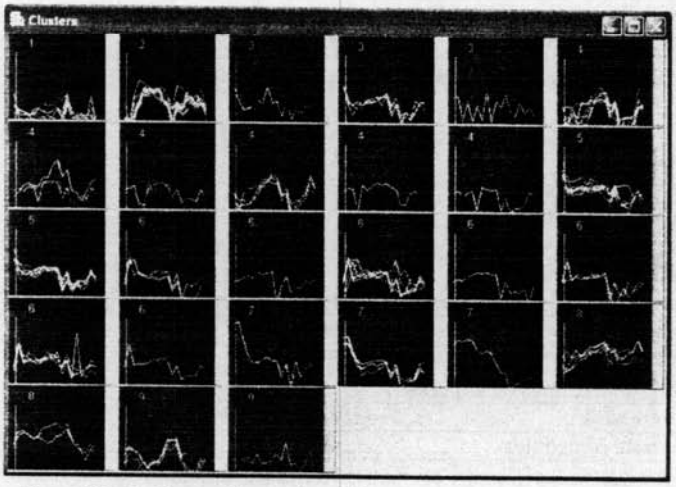


Figura 7 Centroides creados por el modelo supervisado para 9 clases de genes.

Clase	Número de centroides	Patrones clasificados
1	1	8
2	1	14
3	3	7
4	6	18
5	2	20
6	7	20
7	3	7
8	2	8
9	2	7

Tabla 2 Resultados de clustering supervisado para 9 clusters y 27 centroides.

Los parámetros y resultados de las tablas 3a y 3b fueron tomados de la aplicación de clustering a los mismos 109 perfiles de expresión, pero transformando las entradas a datos binarios (1,0), para evaluación de la similitud se utilizó la distancia de Hamming.

Radio inicial	0.2
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.2
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Hamming
Datos de tipo	Binario
Número de patrones de expresión	109
Clases creadas	9
Número de centroides	92

Tabla 3a Creación de 9 clusters y 96 centroides con valores binarios.

Radio inicial	1
Coeficiente de vitalidad	2
Constante de eliminación para el radio	1.2
Constante de eliminación para el coeficiente de vitalidad	1
Distancia	Hamming
Datos de tipo	Binario
Número de patrones de expresión	109
Clases creadas	9
Número de centroides	48

Tabla 3b Creación de 9 clusters y 48 centroides con valores binarios.

Cuando se utiliza el tipo de datos binario es preferible partir de un radio inicial con valor de 1, ya que se manipulan los valores 0 y 1 en los patrones de entrada, puede verse esto en la diferencia del número de centroides creados, hay menor dispersión de datos.

Esta muestra también fue analizada por el clustering supervisado con datos ternarios (0,1,-1), los valores de los parámetros y de los resultados se presentan en las siguientes tablas 4a y 4b.

Radio inicial	0.2
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.2
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Euclidiana
Datos de tipo	Ternario
Número de patrones de expresión	109
Clases creadas	9
Número de centroides	51

Tabla 4a Creación de 9 clusters y 51 centroides con valores ternarios.

Radio inicial	0.5
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.8
Constante de eliminación para el coeficiente de vitalidad	0.55
Distancia	Euclidiana
Datos de tipo	Ternario
Número de patrones de expresión	109
Clases creadas	9
Número de centroides	42

Tabla 4b Creación de 9 clusters y 42 centroides con valores ternarios.

5.3.2 *Clustering no supervisado*

Se realizaron experimentos con perfiles de expresión genética que no estaban acompañados por su clase, aplicando así el clustering no supervisado. El clustering no supervisado se caracteriza porque los perfiles de expresión no van acompañados del dato referente a la clase, sino que las clases se forman durante el proceso.

Primero se trabajo con el modelo supervisado ya obtenido previamente para tener conocimiento a priori del comportamiento de la red para este tipo de patrones (tabla 5a).

Radio inicial	0.19879
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.21009
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Euclidiana
Datos de tipo	Reales
Número de patrones de expresión	121
Clases creadas	9
Número de centroides	36

Tabla 5a Creación de 9 clusters y 36 centroides con valores de punto.

Y se obtuvo con la muestra de 121datos que el número de clases aumento lo mismo que el de centroides como se ve en la tabla 5b, haciendo mayor la diferencia entre clases y dentro de las clases más centroides.

Clases creadas	14
Número de centroides	49

Tabla 5b Creación de 14 clusters y 49 centroides con valores de punto flotante.

Por otro lado, trabajamos por medio de clustering no supervisado para obtener los grupos de la muestra de 121 genes utilizados previamente, pero dejando que el modelo forme sus cúmulos, se obtuvieron los resultados mostrados en las tablas 6a, 6b, 6c y 6d. En la figura 8 se presentan las gráficas de los cúmulos compuestos por los centroides, es preciso recordar que cada gráfica representa un centroide agrupados en clusters para identificar una clase.

Radio inicial	0.2
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.25
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Euclidiana
Datos de tipo	Reales
Número de patrones de expresión	121
Clases creadas	10
Número de centroides	30

Tabla 6a Creación de 10 clusters y 30 centroides con valores de punto flotante.

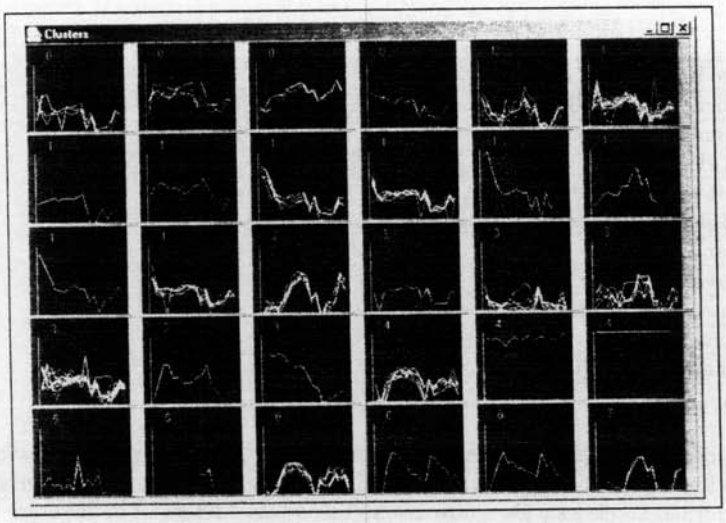


Figura 8 Gráficas de cada centroide integrante de los cúmulos.

Radio inicial	0.15
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.18
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Manhattan
Datos de tipo	Reales
Número de patrones de expresión	121
Clases creadas	13
Número de centroides	21

Tabla 6b Creación de 13 clusters con valores de punto flotante y distancia de Manhattan.

Radio inicial	1.2
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	1.5
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Mahalanobis
Datos de tipo	Reales
Número de patrones de expresión	121
Clases creadas	10
Número de centroides	18

Tabla 6c Creación de 10 clusters con valores de punto flotante y distancia de Mahalanobis

Radio inicial	1.3
Coeficiente de vitalidad	2
Constante de eliminación para el radio	1.5
Constante de eliminación para el coeficiente de vitalidad	.6
Distancia	Euclidiana
Datos de tipo	Ternarios
Número de patrones de expresión	121
Clases creadas	15
Número de centroides	35

Tabla 6d Creación de 15 clusters con valores de punto flotante

La diferencia observada en los experimentos, entre el número de clases y centroides obtenidas entre el modelo supervisado y no supervisado se debe a que en el no supervisado actúa libremente para formar las clases y los centroides, siguiendo las diferencias dadas entre los valores calculados por las distancias y el valor de los centroides creados.

En la figura 8, se presentan los clusters creados por otro software denominado j-express, el cual también se utiliza para agrupar genes de acuerdo a su expresión. Esta forma de clusters varia en cuanto a que cada gráfica es un cluster, no permite identificar dentro de cada cluster aquellos genes que cuenten con más semejanzas. Además siempre se le indica que número de clusters se desean y por lo tanto cuántas clases se pretenden encontrar.

Utilizando una muestra de 826 genes representativos de levadura provenientes de http://gepas.bioinfo.cnio.es, dichos patrones no se encuentran acompañados de la clase a la que pertenecen por lo cual han sido utilizado únicamente para pruebas con el clustering no supervisado, estos 826 genes fueron utilizados en 7 diferentes experimentos para generar su perfil de expresión en una tabla de 826 renglones x 7 columnas. Los experimentos hechos con los valores indicados en la tabla 7a, crean las clases ilustradas en la figura 9, que se resumen en la tabla 8.

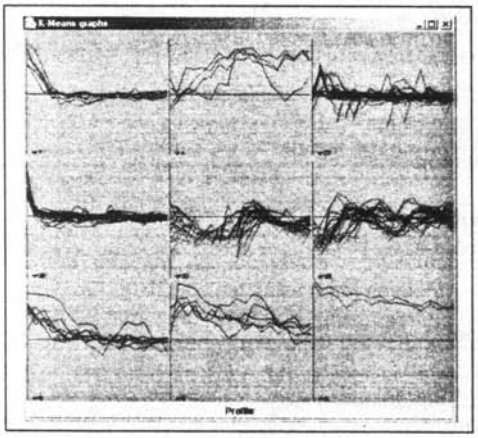


Figura 8 Cúmulos formados por el software *j-express*.

Radio inicial	0.2
Coeficiente de vitalidad	1.09
Constante de eliminación para el radio	0.23
Constante de eliminación para el coeficiente de vitalidad	0.2
Distancia	Euclidiana
Datos de tipo	Reales
Número de patrones de expresión	826
Clases creadas	11
Número de centroides	22

Tabla 7a Creación de 11 clusters con valores de punto flotante.

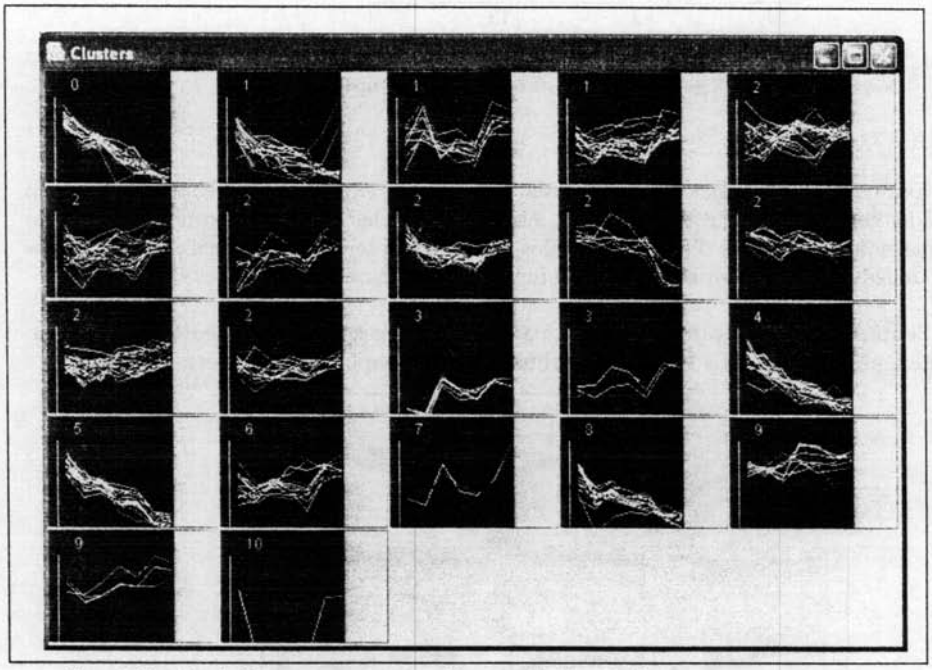


Figura 9 Creación de 10 grupos con sus centroides para clasificación de genes.

Después de 150 épocas el resultado de la creación de cúmulos indica que existen 11 clases para clasificar los 826 patrones de expresión de genes, con un total de 22 centroides. Los cúmulos de las clases están formados como se indica en la tabla 8.

Cúmulos de las Clases	Número de centroides	Patrones clasificados
1	1	55
2	3	164
3	8	251
4	1	8
5	1	162
6	1	133
7	1	14
8	1	2
9	1	23
10	2	13
11	1	1

Tabla 8 Resultados de 11 clusters con valores de punto flotante.

Existen algunos genes muy similares como se ve en la clase 5, en la cual fueron clasificados con un solo centroide 162 genes, mientras que en la clase 3 los patrones de expresión de los genes tienen algunas diferencias que los separan en 8 centroides, lo cual indica que esos genes pueden diferir muy probablemente su funcionalidad en otras condiciones.

Si utilizamos el software de J-express para revisar estos genes con K-medias se obtiene la siguiente figura 10, dando las 11 clases que obtuvimos con la aplicación de nuestro modelo.

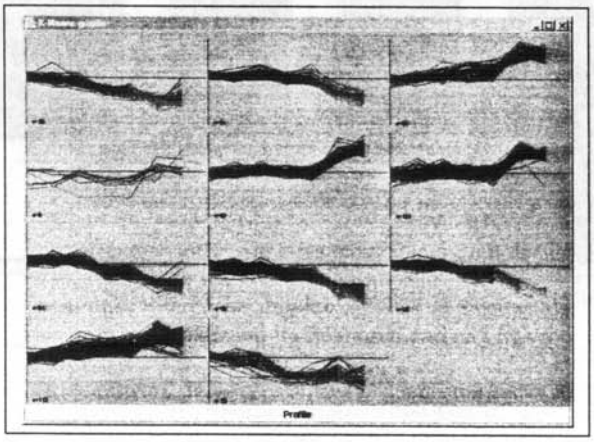


Figura 10 Cúmulos obtenidos con *k-medias* en j-express.

Un ejemplo de la comparación entre los resultados obtenidos por nuestro modelo con los del método *k-medias* de J-express se presenta en la tabla 9 para 4 clases tomadas del experimento con datos de la tabla 7a. En la tabla 10 se muestra una matriz de comparación para los datos de la tabla 6a con 5 clases. Las categorías de los renglones corresponden a las encontradas por nuestra aplicación mientras que en las columnasa las del software J-express.

	Categorías J-express			
	A	**B**	**C**	**D**
A	7	1	0	0
B	0	1	0	0
C	0	1	20	3
D	1	0	4	24

Tabla 9 Comparación de 4 clusters con J-express.

	Categorías J-express				
	A	**B**	**C**	**D**	**E**
A	10	1	0	0	0
B	0	18	2	1	0
C	0	0	10	0	0
D	0	0	0	2	0
E	0	0	0	0	9

Tabla 10 Comparación de 5 clusters con J-express.

La comparación se hizo de acuerdo al nombre de los genes que acompaña a los datos de expresión. entre las listas formadas por esas comparaciones tenemos:

clase 6
centroide 1
YBL002W
YBL003C
YDR224C
YDR225W
YOR247W
YOR248W
YMR215W
centroide 2
YPL163C
centroide 3
YER001W

clase 7
centroide 1
YPL232W
YGR084C

Aquí tenemos la clase 6 y 7 con sus centroides, los genes de cada clase pertenecen a una misma clase dada por J-express. La clase 6 corresponde a la E y la 7 a la D de la tabla 9.

clase 4
centroide 1
YAL067C
YAL018C
YCL075W
YDL024c
YOR358W
centroide 2
YMR322C
YOR100C
YOR386W

clase 1
centroide 1
YAL003W
YAL012W
YAL023C
YAL025C
YAL036C
YAL038W
YAL043C
YAL046C
YAL059W
YAR071W
YAR073W
YAR074C
YBL024W
YBL027W
YBL039C
YBL054W
YBL068W
YBL076C
YBL087C
YBR032W
YBR034C
YBR048W
YBR069C
YBR079C

Aquí tenemos la clase 4 y 1 con sus centroides, los genes de cada clase pertenecen a una misma clase dada por J-express. La clase 4 corresponde a la A y la 1 a la C de la tabla 10. Los nombres subrayados son genes cuya clasificación es diferente en J-express.

5.4 Conclusión

Las pruebas realizadas con el clustering supervisado nos permitieron identificar que el modelo neuronal es aplicable a la tarea de reconocimiento de patrones de expresión de genes. Encontramos que en la mayoría de las clases de patrones de expresión de genes se integran por subgrupos que se presentan en forma de centroides dentro de los clusters que corresponden a las diferentes clases. La posibilidad de encontrar estos subgrupos es propia de nuestro modelo ya que para otras herramientas cada cluster representa un único grupo de genes comunes, dando un solo nivel de similitud. Mientras que nosotros presentamos dos niveles de similitud, uno correspondiente a las clases (clusters) y otro entre los centroides que integran estas clases (subgrupos), lo que puede llevar al descubrimiento de más características para los patrones de expresión de genes.

Lo más importante es encontrar los parámetros iniciales que permitan crear los cúmulos o clases y centroides necesarios para la clasificación, pues como se presenta en las diferentes tablas de resultados puede variar mucho el resultado del clustering. Entre más pequeños son los valores, mayor es el número de cúmulos y de centroides, es decir, la clasificación de perfiles es más rigurosa o fina, lo cual puede llevar hasta una diferenciación poco significativa tendiendo a mostrar a cada uno de los genes como parte de un cúmulo y llevando al peor de los casos, en el cual existen tantos centroides como genes. Esto también repercute en el procesamiento, pues la red es mayor y por ende el número de cálculos requeridos.

Además puede apreciarse la diferencia encontrada entre la clasificación obtenida con números reales, binarios y ternarios. En el nivel de procesamiento, la diferencia entre utilizar valores binarios y ternarios es importante pues el número de cálculos es superior. El usuario puede guiarse para el establecimiento de los parámetros iniciales de la red por la presentación gráfica final de los valores obtenidos en cada cúmulo y centroide, así como en las estadísticas obtenidas. Aunque no se conozca el resultado final de la formación de cúmulos, el usuario tiene el conocimiento básico para identificar el grado de diferenciación que se pretende alcanzar.

Los sistemas o aplicaciones creadas para el análisis de microarreglos, como los que se presentan en la introducción, responden a diversas necesidades de los datos o a los valores como vienen de los microarreglos, lo cual se pudo constatar al buscar las bases de datos reportadas en la documentación referida. Esto fue un impedimento para comparar con algunas otras aplicaciones, al no poder expresar los datos de igual forma. El software j-express fue el único que permitió utilizar datos semejantes.

CONCLUSIONES

El trabajo presentado es una herramienta que contribuye a una de las tareas principales de las ciencias de la computación como es la automatización del procesamiento de análisis de información del genoma, especialmente de la expresión genética. De esta forma se logró la construcción de un modelo neuronal basado en la formación de clusters en tiempo real para el análisis de patrones de expresión de genes. El trabajo de la tesis incluyó el análisis, diseño e implementación de dos modelos neuronales, uno con supervisión y otro sin supervisión, así como de la interfaz de usuario para la interacción con el sistema.

La primera fase de trabajo en este proyecto consistió en el refinamiento de una red neuronal propuesta por Sánchez *et al.* [29] para contar con un primer modelo neuronal cuya tarea fue la identificación de patrones de expresión de genes. Durante esta etapa de trabajo, la red neuronal puede recibir tanto patrones de expresión de genes para los cuales se conoce la clase o categoría a la que pertenecen, como patrones de expresión de los cuales se desconoce su clase. La red neuronal es capaz de decidir si estos últimos patrones pueden ser clasificados como elementos de las clases ya conocidas. Para la solución de esta tarea la topología de la red neuronal considera la existencia de una capa de neuronas de entradas correspondiente a la dimensión de los vectores de entrada (patrones de expresión de genes) y una capa de neuronas de salida cuya dimensión inicial corresponde a la cantidad de clases de expresión conocidas. La capa de neuronas ocultas es generada dinámicamente a partir de la presentación de los vectores de entrada a la red. Es a nivel de esta capa donde se forman los clusters de neuronas que representarán las diferentes clases o categorías de patrones de expresión de genes. La importancia de esta primera fase de trabajo consistió sobretodo en conocer las características topológicas de una red neuronal dinámica para el análisis de patrones de expresión de genes, después de la presentación de diversos conjuntos de datos de expresión de genes acompañados por la clase o categoría a la que pertenecían.

La segunda fase de trabajo envolvió el refinamiento del primer modelo neuronal obtenido, de forma tal que las clases o categorías de expresión sean creadas por la propia red. Este refinamiento conllevó a restringir toda la información de entrada a la red a solamente la contenida patrones de expresión de genes. sin información alguna de las clases o categorías a las que pertenecían los patrones de expresión. Durante la presentación de los patrones de expresión, la red debe crear dinámicamente la capa de neuronas ocultas y la capa de neuronas de salida. La topología inicial de la red consistía solamente de una capa de entrada, cuya dimensión es la misma que la de los vectores de expresión de genes de entrada. Por cada cluster de neuronas ocultas se creó una neurona en la capa de salida, la cual representó la clase definida por dicho cluster.

Una característica importante del modelo neuronal propuesto es que los clusters creados en la capa oculta pueden venir representados por más de una neurona o centroide, este hecho responde a que dentro de un cluster existen subconjuntos de patrones de expresión que son entre sí más cercanos por su nivel de expresión y pueden llevar a un nivel mayor de

refinamiento representando características más afines. En tanto que, los subconjuntos de genes entre un cluster y otro presentan mayores diferencias por lo que se encuentran en diferentes clases.

Las alternativas de datos binarios y ternarios, además de los de punto flotante, unidas con las diferentes distancias como medidas de similitud puede utilizarse con todas las muestras, sin embargo, es más difícil establecer los parámetros iniciales de la red para este tipo de datos. A su vez, podemos concluir que cuando los datos provienen de muestras en números flotantes es mejor continuar con ese tipo de números, no convertirlos en binarios o ternarios ya que el uso de valores binarios y ternarios implica un mayor trabajo de procesamiento que no se ve reflejado en los resultados finales. Por otro lado, la distancia euclidiana ha sido la más eficiente para la tarea de clustering de expresión de genes pues el nivel de procesamiento es menor que en otras.

Por otro lado, el procesamiento realizado en el modelo no supervisado es más complejo que en el supervisado, pues la red debe evaluar a partir de la diferencia encontrada entre el vector de pesos de los centroides y el vector de entrada si el centroide representativo del patrón ya existe o debe crearse y en tal caso decidir si pertenece a un nuevo cluster o a uno ya conocido.

La idea de crear centroides dentro de los cúmulos o subclases es novedosa, no aparece esta alternativa en la literatura revisada. Estas subclases representan información extra que arroja la clasificación para pensar en un refinamiento de dicha clasificación al mostrar mayor semejanza entre ciertos subconjuntos de genes. Por lo tanto podemos sugerir este modelo neuronal como una buena alternativa para la clasificación de genes de acuerdo a su perfil de expresión.

Trabajo futuro

Sería interesante probar el desempeño de la red neuronal con patrones de genes integrados por un número mayor a 20 experimentos, paro esto se requiere implementar la aplicación en una computadora de alto rendimiento, por la cantidad de datos procesados y complejidad de la misma red neuronal.

Una vez resueltos los problemas de identificación y clasificación de patrones de expresión, el modelo neuronal propuesto deberá evolucionar para solucionar una tercera tarea, la asociación de patrones de expresión de genes, lo cual será parte de un trabajo futuro. Durante esta etapa, la red recibirá como entrada un patrón de expresión correspondiente a los niveles de expresión de dos o varios genes en un único experimento más la información que se desea que produzca como salida, la cual corresponde al nivel de expresión de otro gen, dentro del mismo experimento realizado. La tarea a resolver por la red consistirá en sintetizar una función que efectúe la asociación entre los patrones que son presentados a la red. La síntesis de esta función requiere de una etapa de entrenamiento, durante la cual neuronas y clusters de neuronas son creados como parte de la capa oculta.

REFERENCIAS

[1] Adams, M., Serial analysis of gene expression:ESTs get smaller, PubMed, April 18(4), pp. 261-262, 1996.

[2] Bicciato S., Pandin, Analysis of an associative memory neural network for pattern identification in gene expression data, Workshop on data mining in bioinformatics, 2001.

[3] Brown M., Noble W., Lin D., Cristianinis N., Walsh C., Furey T., Ares M. y Haussler D., Knowledge-based analysis of microarray gene expression data by using support vector machines, Vol. 97 No. 1 2000 pp. 262-267, PNAS.

[4] D'Haeseleer P., Liang S. y Somogyi R., Genetic network inference: from co-expression clustering to reverse engeneering, Incyte Pharmaceuticals, Inc., 1999.

[5] Eisen M., Spellman O., Brown P. y Botstein D., Correlación de pearson y euclidiana: cluster analysis and display of genome -wide expression patterns, Proc. Natural academic science USA, Vol. 95, 1998, pp. 14863-14868.

[6] Gibbons, F., Which clustering algorithms best use expression data to group genes by function?, ISMB 2001.

[7] González-Pérez P., Cárdenas M., Gershenson C. y Lagúnez-Otero J., Integration of computational techniques for modelling of signal transduction, Instituto de Química, Universidad Nacional Autónoma de México, 2001.

[8] Haykin, S., Neural networks : A comprehensive foundation, Macmillan, USA, 1999.

[9] Hen-Hu Yu, Handbook of neural network signal processing, CRC Press, USA, 2001.

[10] Herrero J., Valencia A. y Dopazo J., A hierarchical unsupervised growing neural network for clustering gene expression patterns, Bioinformatics, Vol. 17 No. 2, 2001, pp. 126-136.

[11] Herwig, Large-scale clustering of cDNA-fingerprinting data, letter, Genome Research, Vol. 9, 1999, pp. 1093-1105.

[12] Hilsenbeck S., Friedrichs W., Schiff R., O'Connell R., Hansen R., Osborne K., Fuqua S., Statistical analysis of array expression data as applied to the problem of tamoxifen resistence, Journal of national cancer institute, vol. 91, no.5 1999 pp. 453-460.

[13] Human genome project information: http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml

[14] Kevin, Murphy, Mian, Modelling gene expression data using dynamic bayesian networks,

Computer science division California U. 1998.

[15] Kim S., Dougherty E., Chen Y., Sivakumar K., Meltzer P., Trent J. and Bittner M., Multivariate measurement of gene expression relationships, Genomics, Vol. 67, 2000, pp. 201-209.

[16] Kohonen T., The self-organizing map, Proc of the IEEE 78, 1990, pp. 1464-1480.

[17] Liang P., SAGE genie: a suite with panoramic view of gene expression, PNAS, Vol. 99 No. 18, 2002, pp. 11547-1548.

[18] Martín del Brío B., Redes neuronales y sistemas borrosos, Textos universitarios, Madrid 1997.

[19] Martín-Sánchez, F., Impactos de la Bionformática y la genómica en la medicina del siglo XXI, Unidad de bioinformática- BOITIC, Madrid España.

[20] McCord, M, A practical guide to neural netwoks, Addison Wesley, USA, 1991.

[21] Mehrotra, K., Elements of artificial neural networks, MIT Press, USA, 1997.

[22] Murphy David, Gene expression studies using microarrays: principles, problems, and prospects, APS refresher course report, 2002.

[23] Parbhane R., Tambe S., Kulkarni B., ANN modeling of DNA sequences: new strategies using DNA shape code, Computers & Chemistry Vol. 24, No. 6, 2000, pp. 699-711.

[24] Quackenbush, Jonh, Computational analysis of microarray data, Nature reviews, Vol. 2, 2001, pp. 418-427.

[25] Raychaudhuri S., Sutphin P., Chang J. y Altman R., Basic microarray analysis : grouping and feature reduction, Trends in Biotechnology, Vol. 19, No. 5, 2001, pp. 189-193.

[26] Revista Cómo ves?, Edición especial, Las ciencias del genoma, año 4 núm. 37, UNAM, diciembre 2001.

[28] Rumbaugh *et al.*, Modelo y diseño orienteado a objetos, Prentice Hall, España 1995.

[29] Sánchez, *et al.*, A neuronal network to learn patterns from non stationary environments, The Firts Joint Mexico-US International Workshop on Neural Networks and Neurocontrol, México 1998, pp. 310-325.

[30] Schmulevich, I., Zhang, Binary analyisis and optimization-based normalization of gene expression data, Bioinformatics, Vol. 18 No. 4, 2002, pp. 555-565.

[31] Smith D., et al. Assessing cúmulos and Motifs from Gene Expression Data, ISMB 2001.

[32] Soberón, La ingeniería genética y la nueva biotecnología, Fondo de cultura económica, México, 2001.

[33] Somogyi R. Making sense of expression data, A trends guide, Incyte Pharmaceuticals, Inc., 1999.

[34] Tamayo P., Slonim D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E., Golub T.,

Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, Proc Natl Acad Sci, USA, Vol. 96, No. 6, 1999, pp. 2907-2912.

[35] Thijs G., Moreau Y., Rombauts S., De Moor B., Rouze P., Recognition of gene regulatory sequences by bagging of neural networks, Proceedings International Conference on Artificial Neural Networks, Escocia, 1999, pp. 988-993.

[36] Törönen P., Kolehmainen M., Wong G. y Castrén E., Analysis and visualization of gene expression data using self-organizing maps, Federation of European Biochemical Societies, Finlandia, 1999, pp. 142-146.

[37] UML para java de sun (manual)

[38] Xiang-Chen, cDNA microarray technologi and its application, Biotechonology advances 18, 2000, pp. 35-46.

[39] Zaki, Recent advances in data mining for bioinformatics, SIGKDD, Vol. 4, issue 2, pp. 112-114.