



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

REDES NEURONALES PARA LA DETECCIÓN
DE GENES FORANEOS

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
LICENCIADA EN CIENCIAS DE LA
C O M P U T A C I Ó N
P R E S E N T A :
CITLALI CALDERÓN DE ANDA



DIRECTOR DE TESIS: DR. PEDRO EDUARDO MIRAMONTES VIDAL



2004 FACULTAD DE CIENCIAS
SECCION ESCOLAR



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE
DE LA BIBLIOTECA



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

"Redes Neuronales para la detección de genes foráneos"

realizado por

Citlali Calderón de Anda

con número de cuenta 09432330-2 , quien cubrió los créditos de la carrera de:

Ciencias de la Computación

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis

Propietario

Dr. Pedro Eduardo Miramontes Vidal

Propietario

M. en C. José Antonio Neme Castillo

Propietario

Dr. Germinal Cocho Gil

Suplente

M. en C. José de Jesús Galaviz Casas

Suplente

Biól. Luis José Delaye Arredondo

Consejo Departamental de
Matemáticas



Dr. Francisco Hernández Quiroz

FACULTAD DE CIENCIAS

CONSEJO DEPARTAMENTAL

MATEMÁTICAS

A ti Edgar, con mucho amor

Agradecimientos

A mis padres por su constante apoyo y cariño y por todo lo que he aprendido de ustedes.

A Edgar por que aun estando lejos has estado entrañablemente cerca, por ayudarme a crecer como persona, por enseñarme a ver la vida de otra manera, por todo el apoyo, comprensión y amor, por aguantar mis tensiones del fin, por todo.

A mis hermanos Froy y Miriam, y mis cuñados Elsa y Peter que están lejos pero que se que me apoyan. A mis tías Quetita y Guadalupe por toda su ayuda.

A Pedro Miramontes por su infinita paciencia y por todos los conocimientos que he adquirido gracias a él.

A Liliana por tu amistad y cariño, por tantos momentos inolvidables compartidos durante la carrera. A Gaby por tu amistad, por que sin ustedes dos no hubiese sido lo mismo. A Canek por tu amistad, por ayudarme innumerables veces en línea.

A Raul, Cindy, Emiliano y Edith por tantos años de amistad, y en especial por esta última etapa en la cual estuvieron muy cerca.

A toda la banda por su amistad y camaradería.

A la UNAM, a la facultad de ciencias y sus maestros que me hicieron ver las cosas de distinta manera.

A Frida por acompañarme muchas tardes al frente de la computadora.

Índice general

1. Introducción	8
2. Redes neuronales artificiales	12
2.1. Analogía de las redes neuronales artificiales con las biológicas	13
2.1.1. Neurona biológica	13
2.1.2. Neurona artificial	15
2.2. Características y clasificación	15
2.2.1. Tipos de funcionamiento y aprendizaje	16
2.3. Perceptrón simple	17
2.3.1. Separabilidad lineal	21
2.4. Perceptrón multicapa	23
2.4.1. Algoritmo de retropropagación	25
2.4.2. Explicación intuitiva de la retropropagación	28
2.4.3. Deficiencias de la retropropagación	28
2.5. Algunos ejemplos	31
3. Los genes	32
3.1. DNA	33
3.1.1. Eucariontes y procariontes	36
3.1.2. Transcripción y traducción	36
3.1.3. Código genético	38
3.2. Adquisición de nuevos genes	39
3.2.1. Transferencia horizontal	39
3.2.2. Virus	41
3.2.3. Transgénicos	41
4. Métodos	42
4.1. Introducción	42
4.2. La red neuronal	42

4.2.1. Arquitectura	43
4.2.2. Sensores	44
4.3. Implementación de la red	47
4.3.1. Entrenamiento y validación	47
4.3.2. Cálculo de los sensores	48
5. Resultados	50
6. Conclusiones	57
A. Los programas	59
Bibliografía	62

Índice de figuras

2.1. Neurona biológica.	14
2.2. Perceptrón simple.	18
2.3. Algunas funciones de activación: a) función escalón, b) función de umbral lógico, c) función sigmoideal.	20
2.4. Separabilidad lineal, las tres rectas son fronteras de decisión que separan al conjunto de elementos de la clase A de los de la clase B.	21
2.5. Un ejemplo no lineal, se requieren al menos dos rectas para separar el conjunto de puntos A del conjunto de puntos B.	21
2.6. Las tres funciones lógicas fundamentales, implementadas cada una por un perceptrón que emplea la función escalón como función de activación.	22
2.7. Perceptrón multicapa, son tres los tipos de capas: la capa de entrada o retina, las capas ocultas y la capa de salida	23
2.8. Tipos de regiones de decisión que pueden formarse: a) con un perceptrón simple, b) un perceptrón multicapa con una capa oculta, c) un perceptrón multicapa con dos capas ocultas.	24
3.1. Esquema simplificado de una doble hélice de DNA.	33
3.2. Las bases que componen la parte variable del DNA (el uracilo del RNA) arregladas según su naturaleza purínica o pirimídica.	34
3.3. La adenina sólo puede aparearse con la timina mediante dos puentes de hidrógeno, mientras que la citosina sólo lo hace con la guanina mediante tres puentes de hidrógeno.	35
3.4. Transcripción - traducción	37
3.5. Dos células de <i>E. coli</i> , intercambiando material genético mediante conjugación. Las bacterias pueden llevar a cabo la transferencia horizontal de genes, incluso entre individuos de diferentes especies	40
4.1. La arquitectura de la red neuronal.	44

4.2. Ángulos estructurales del DNA: a) Broche, b) Abertura, c) Hélice	45
4.3. Ventana móvil	48
5.1 <i>Escherichia coli</i> → 1 vs. <i>Mus musculus</i> → 0.	50
5.2 <i>Pseudomonas aeruginosa</i> → 1 vs. <i>Archaeoglobus fulgidus</i> → 0.	51
5.3 <i>Neisseria meningitidis</i> → 0 vs. <i>Haemophilus ducreyi</i> → 1.	52
5.4 <i>Neisseria meningitidis</i> → 0 vs. <i>Haemophilus ducreyi</i> → 1. Ejemplo anterior con datos alisados.	53
5.6 <i>Neisseria meningitidis</i> → 0 vs. <i>Haemophilus ducreyi</i> → 1. Ejemplo real con umbral.	55
5.5 <i>Neisseria meningitidis</i> → 0 vs. <i>Haemophilus ducreyi</i> → 1. Ejemplo real.	55
5.7 <i>Salmonella enterica</i> → 0 vs. <i>Escherichia coli</i> → 1.	55

Índice de cuadros

- 3.1. El código genético estándar. "Alto" es el codón de terminación de mensaje, los mensajes generalmente comienzan con el aminoácido Met 38
- 4.1. Resultados reportados por Quintana y Dickerson [Quintana *et al.*, 1992]. Los ángulos estructurales entre 2 bases son: *L* - low twist, *H* - high twist *I* - intermediate twist *V* - variable twist 46

Capítulo 1

Introducción

Las redes neuronales una herramienta para reconocer patrones

Desde que se desarrollaron las primeras computadoras, a mediados del siglo XX, no es exagerado afirmar que todas las áreas del conocimiento se han visto trastocadas por este avance científico y tecnológico.

Ha surgido una ciencia como tal, la ciencia de la computación, y a su vez una herramienta útil para la solución de problemas de otras disciplinas.

Con la computadora se han podido resolver problemas, que por la cantidad de datos, o el número de operaciones que se requieren para llegar a la solución, hubieran sido imposibles de tratar con otros medios. Así, por un lado, se ha ido desarrollando a pasos agigantados la tecnología en la que se basan las computadoras, lo que se traduce en máquinas cada vez más veloces y con una capacidad de almacenamiento muy grande, así como diversos tipos de arquitecturas (multiprocesamiento, almacenamiento distribuido, etcétera); y por otro lado los programas que controlan su funcionamiento también han progresado notablemente.

Sin embargo, existen problemas que por su naturaleza se podría pensar que sólo una máquina "inteligente" podría resolverlos. Por esta razón es que los científicos han intentado imitar algunas funciones del cerebro humano y la manera como funciona ciertos problemas.

Así nace la inteligencia artificial que es un área de las ciencias computacionales que, a grandes rasgos, intenta la creación de programas para máquinas que imiten el comportamiento y la comprensión humana, que sean capaz de aprender, reconocer, etcétera.

Como una parte de la inteligencia artificial se crearon las redes neuronales artificiales. Éstas son modelos matemáticos que intentan imitar algunas de las funciones

del cerebro humano (en muchos de los casos con bastante éxito), por ejemplo la capacidad de aprender. Existen varios tipos de redes neuronales, pero las que trataremos en este trabajo (perceptrones multicapa) pueden, mediante un entrenamiento previo aprender a reconocer o memorizar patrones.

El reconocimiento de patrones también puede verse como una clasificación de dichos patrones. La manera en que el perceptrón multicapa funciona es la siguiente: Se determina un número fijo de categorías en las cuales las muestras de entrada deben clasificarse. Para ello primero se requiere una fase de entrenamiento en la que se presentan a la red los patrones que debe aprender y la categoría en la cual debe clasificarlos. Entonces se le presenta a la red un patrón nuevo y desconocido pero que pertenece a alguna de las categorías aprendidas y ésta debe decidir a qué categoría se parece más. Por ejemplo, si se entrenase una red neuronal para que aprendiese a distinguir caracteres del alfabeto, y se le presentase a la red un semicírculo inferior, la red tendría que ser capaz de reconocer dicha línea circular como una u, claro está que durante la fase de entrenamiento se le presentaron otras letras u pero de diferente puño, es decir, la red aprende a generalizar y en este caso el patrón que se le mostró entraría dentro de la categoría de la letra u.

Vivimos rodeados de patrones, algunos perceptibles por el humano como por ejemplo los tonos musicales, los colores, las formas, etcétera; pero mucha de la información que se maneja en la naturaleza se presenta en la forma de patrones complejos. En estos casos es que se requiere de la ayuda de las computadoras, para poder detectar patrones que a simple vista no son discernibles por el humano.

En todas las células de los seres vivos se encuentra una molécula (el DNA), que es capaz de replicarse¹ y que además, contiene la información necesaria para dirigir las actividades del organismo. Esta información, que es hereditaria, es la información genética. A lo largo de toda la secuencia de nucleótidos que componen el DNA se encuentran los genes. Los genes normalmente se transfieren de padres a hijos (herencia), pero existen casos en los que la transferencia de genes se lleva a cabo entre dos organismos de distintas especies (transferencia horizontal), esto se da normalmente entre bacterias pero es observable en cualquier especie. Cuando un organismo incorpora a su genoma (toda la secuencia de DNA de ese organismo) un gen de otra especie, se dice que el nuevo gen es un *gen foráneo*.

La bioinformática es un área de investigación multidisciplinaria, en donde convergen dos ciencias: biología y computación y está impulsada por la incógnita del genoma humano y la promesa de una nueva era en la cual la investigación genómica puede ayudar dramáticamente a mejorar la condición y calidad de vida humana.

El objetivo de este trabajo es hacer uso de las redes neuronales para detectar

¹No lo hace por sí sola, sino por medio de una compleja maquinaria de replicación

genes foráneos. Se entrena a una red neuronal (perceptrón multicapa) con características del genoma de dos especies distintas para que aprenda a distinguirlas, más adelante se analiza una de las dos especies y se determina si esa secuencia de DNA contiene DNA de la otra especie.

Panorama general

El presente trabajo está conformado por 6 capítulos y un apéndice. Los capítulos 2 y 3 conforman el marco teórico de este trabajo, en uno se trata lo concerniente a las ciencias de la computación y en el otro, por tratarse de un trabajo de biocómputo, de biología.

En el capítulo 2, se presenta una introducción a las redes neuronales artificiales, en donde se hace notar la analogía que existe entre la neurona artificial y la neurona biológica; más adelante se trata con más detalle un tipo particular de red neuronal, el perceptrón multicapa, que es adecuado para el reconocimiento de patrones, se exponen sus características, su funcionalidad, el algoritmo que normalmente se emplea para su funcionamiento (retropropagación), sus limitaciones y por último algunos ejemplos en los cuales se han utilizado las redes neuronales.

En el capítulo 3, se da una introducción a la biología de los ácidos nucleicos, en particular del DNA. La intención de este capítulo es la de familiarizar al lector no versado en biología con la fenomenología de la genética molecular para lograr establecer un lenguaje común y poder así introducir los vectores de características que se emplearon para el análisis del DNA, así mismo se exponen los diferentes mecanismos por medio de los cuales, se adquieren nuevos genes en un genoma (genes foráneos).

En el capítulo 4, se exponen los métodos utilizados en este trabajo. En una breve introducción se explica el objetivo de este trabajo y cómo fue que se llevó a cabo. Más adelante se detallan las características de la red neuronal que se utilizó, tales como su arquitectura (número de capas y de neuronas en cada capa) y su función de optimización; se describen detalladamente los sensores (vectores de características) que alimentan a la red neuronal, así como también se explica el significado de cada uno de ellos y la razón por la cual ese conjunto de sensores caracteriza al DNA. También se describe como fue implementada la red neuronal, su entrenamiento y posterior validación así como la manera cómo se calcularon los sensores.

En el capítulo 5, se presentan los resultados obtenidos, se muestran graficados los datos de salida de la red neuronal de diferentes corridas realizadas; así mismo se analizan e interpretan dichas gráficas.

El trabajo concluye con un breve capítulo 6 de conclusiones donde se analizan los resultados y su relevancia, así como el posible trabajo a futuro continuando con

esta investigación.

Por último aparece un apéndice con parte del código fuente empleado en el desarrollo de este trabajo.

Capítulo 2

Redes neuronales artificiales

El cerebro humano es un procesador de información con características muy sobresalientes: es capaz de procesar muy velozmente grandes cantidades de información provenientes de nuestro entorno, combinarla y compararla con información almacenada y dar respuestas adecuadas aún en situaciones nuevas. Logra distinguir un susurro en una sala ruidosa, reconocer una cara en una calle mal iluminada, etcétera. Pero lo más impresionante es su capacidad de aprender a representar la información necesaria para desarrollar tales habilidades sin instrucciones explícitas para ello.

Aunque todavía se ignora mucho sobre la forma en que el cerebro aprende a procesar la información, se han desarrollado modelos que tratan de imitar algunas de las habilidades del cerebro, estos modelos son las llamadas *redes neuronales artificiales* o *modelos de computación conexionista*. Las redes neuronales artificiales son una rama de la inteligencia artificial.

Fue en el último siglo que los científicos intentaron imitar el complejo funcionamiento del cerebro. Durante la década de 1920 a 1930 se intentó utilizar la teoría de la conmutación telefónica como punto de partida de un sistema de conocimiento similar al del cerebro. Entre 1940 y 1950 los científicos comenzaron a pensar seriamente en las redes neuronales utilizando como concepto la noción de que las neuronas del cerebro funcionan como interruptores digitales.

En 1943, Warren McCulloch y Walter Pitts [McCulloch & Pitts, 1943] introdujeron el modelo de neuronas formales, estas eran un modelo conceptual de las neuronas biológicas y podían realizar tareas computacionales.

Seis años después de que McCulloch y Pitts mostraran sus redes neuronales, el psicólogo Donald O. Hebb [Hebb, 1949] expuso que éstas podían aprender. Su propuesta tenía que ver con la conductividad de la sinapsis, es decir, con las conexiones entre neuronas. Hebb expuso que la activación repetida de una neurona por otra

a través de una sinapsis determinada, aumenta su conductividad, y la hacía más propensa a ser activada sucesivamente, induciendo a la formación de un circuito de neuronas estrechamente conectadas entre sí. La hipótesis de Hebb, basada en investigaciones psicofisiológicas, presenta de manera intuitiva la manera en que las neuronas memorizan información y se plasman sintéticamente en la *regla de aprendizaje de Hebb* (también conocida como regla del producto). Esta regla indica que las conexiones entre dos neuronas se refuerzan si ambas son activadas.

En 1962, Frank Rosenblatt [Rosenblatt, 1962] publica los resultados de un ambicioso proyecto de investigación; el desarrollo del perceptrón, un identificador de patrones ópticos binarios y salida binaria. Las capacidades del perceptrón se extendieron al desarrollar la *regla de aprendizaje delta*, que permitía emplear señales continuas de entrada y salida.

A finales de la década de los sesenta Marvin Minsky y Seymour Papert, autoridades en la inteligencia artificial clásica, escribieron el libro "Perceptrons" [Minsky & Papert, 1969] en el cual mostraban las deficiencias del modelo de los perceptrones de una capa.

A pesar del duro golpe que supuso para las redes neuronales artificiales el libro de Minsky y Papert [Minsky & Papert, 1969], en la década de los setentas un puñado de investigadores continuaron con sus trabajos en esta área, los más notables Teuvo Kohonen, Stephen Grossberg y James Anderson. Se crearon nuevos tipos de redes neuronales como las redes de Kohonen o las redes de Hopfield. De estos modelos no se profundizara en este trabajo puesto que nuestro interés versó por los perceptrones multicapa.

Fue hasta la década de los ochentas que el interés por las redes neuronales resurgió gracias a descubrimientos teóricos como por ejemplo el error de retropropagación [Rumelhart, le Cun, Parker, 1986,1985,1985] del cual se hablará más adelante en este capítulo.

2.1. Analogía de las redes neuronales artificiales con las biológicas

2.1.1. Neurona biológica

El cerebro humano esta compuesto por centenas de billones de neuronas (aproximadamente 10^{14}) interconectadas entre sí (aproximadamente 10^4 conexiones por neurona) formando circuitos o redes que desarrollan funciones específicas. Tienen tamaño y forma variables, pero con los mismos componentes (Figura 2.1).

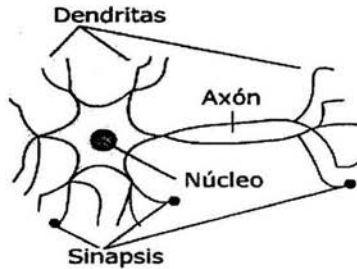


Figura 2.1: Neurona biológica.

Son tres los componentes principales de la neurona: el *soma* o cuerpo de la célula, las *dendritas* y el *axón*.

Una neurona recoge señales procedentes de otras neuronas a través de las dendritas, que son prolongaciones del cuerpo de la célula. La neurona emite impulsos de actividad eléctrica a lo largo del axón que es una fibra larga y delgada, que se divide en millares de ramificaciones. Las extremidades de estas ramificaciones llegan hasta las dendritas de otras neuronas y establecen unas conexiones llamadas *sinapsis*, en las cuales se produce una transformación del impulso eléctrico en un mensaje neuroquímico, mediante la liberación de unas sustancias llamadas neurotransmisores.

El efecto de los neurotransmisores sobre la neurona receptora puede ser excitatorio o inhibitorio, y es variable, de manera que podemos hablar de la fuerza o efectividad de una sinapsis. Las señales excitatorias e inhibitorias recibidas por una neurona se combinan, y en función de la estimulación total recibida, la neurona toma un cierto nivel de activación, que se traduce en la generación de breves impulsos nerviosos con una determinada frecuencia o tasa de disparo, y su propagación a lo largo del axón hacia las neuronas con las cuales está conectada.

De esta manera la información se transmite de unas neuronas a otras y va siendo procesada a través de las conexiones sinápticas y las propias neuronas. El aprendizaje de las redes neuronales se produce mediante la variación de la efectividad de las sinapsis, de esta manera cambia la influencia que unas neuronas ejercen sobre otras, de aquí se deduce que la arquitectura, el tipo y la efectividad de las conexiones en un momento dado, representan en cierto modo la memoria o estado de conocimiento de la red.

La fuerza inherente de una sinapsis no está dada por una sola sinapsis, sino por todas, tal y como lo postuló Hebb:

“Cuando un axón de la célula A está suficientemente próximo para excitar a una célula B o toma parte en su disparo de forma persistente, tiene lugar algún proceso de crecimiento o algún cambio metabólico en una de las células, o en las dos, de tal modo que la eficiencia de A, como una de las células que desencadena el disparo de B, se ve incrementada [Hebb, 1949].”

2.1.2. Neurona artificial

En el afán de crear máquinas con propiedades similares a las del cerebro humano se ha intentado imitar las características, estructura y funcionamiento real del cerebro. Es importante aclarar que estos modelos son idealizaciones burdas de las auténticas redes neuronales, en muchos casos de dudosa plausibilidad neurofisiológica, pero que, sin embargo, resultan interesantes cuando menos por sus capacidades de aprendizaje. Es importante hacer notar que aún se desconoce mucho acerca del funcionamiento del cerebro, pero aún así las neuronas artificiales logran imitar algunas de sus funciones.

La idea consiste en plantear un modelo artificial de neurona (como un sistema: con entradas, salidas y una determinada estructura interna) que tenga cierta semejanza en cuanto a estructura y funcionamiento con el modelo natural real.

Las neuronas artificiales se modelan mediante unidades de proceso; cada unidad o nodo se compone de: conexiones de entrada (dendritas), una función de propagación encargada de calcular la entrada total combinada de todas las conexiones (efecto combinado de las sinapsis), un núcleo central de proceso encargado de aplicar la función de activación (activación \rightarrow tasa de disparo), y la salida por donde se transmite el valor de activación a otras unidades (axón). Como sucede en el cerebro, las neuronas están interconectadas formando una red, por lo que las conexiones entre estas semejan las sinapsis y los pesos asociados a dichas conexiones representan la efectividad de la sinapsis.

Existen varias formas de conectar las neuronas; de las diferentes topologías así como de otras maneras de clasificar a las redes y de sus características más sobresalientes se hablara en la siguiente sección.

2.2. Características y clasificación

Las redes neuronales artificiales, como ya se vio en la sección anterior, son unidades de proceso, conectadas entre sí para formar una red. Los sistemas de cómputo tradicional procesan la información en forma secuencial; una computadora serial

consiste por lo general de un solo procesador que puede manipular instrucciones y datos que se localizan en la memoria, el procesador lee, y ejecuta una a una las instrucciones en la memoria; este sistema serial es secuencial, todo sucede en una sola secuencia determinada de operaciones. Las redes neuronales, en contraparte, son sistemas distribuidos y paralelos, es decir, la memoria se encuentra distribuida y son varios los procesadores trabajando al mismo tiempo¹.

Algunas características de las redes neuronales que las diferencian del modelo de cómputo secuencial son las siguientes:

Las redes neuronales presentan un *aprendizaje inductivo*, es decir, no se le indican las reglas para dar una solución, sino que extrae sus propias reglas a partir de ejemplos de aprendizaje, modificando su comportamiento en función de la experiencia. Estas reglas quedan almacenadas en las conexiones y no representadas explícitamente como en los sistemas basados en conocimiento.

Otra característica es la *generalización*, una vez entrenada la red, se le pueden presentar datos distintos a los usados durante el aprendizaje. La respuesta obtenida dependerá del parecido de los datos con los ejemplos de entrenamiento.

También pueden abstraer o tener *tolerancia al ruido* pues son capaces de extraer o abstraer las características esenciales de las entradas aprendidas, de esta manera pueden procesar correctamente datos incompletos o distorsionados.

Como ya se mencionó antes las redes neuronales son sistemas con *procesamiento paralelo*, las neuronas reales trabajan en paralelo; en el caso de las redes artificiales es obvio que si usamos un solo procesador no podrá haber proceso paralelo real; sin embargo hay un paralelismo inherente, lo esencial es que la estructura y modo de operación de las redes neuronales las hace especialmente adecuadas para el procesamiento paralelo real mediante multiprocesadores (se están desarrollando máquinas específicas para la computación neuronal).

El conocimiento acumulado por la red se halla distribuido en numerosas conexiones (*memoria distribuida*), esto tiene como consecuencia la tolerancia a fallos: una red neuronal es capaz de seguir funcionando adecuadamente a pesar de sufrir lesiones² con destrucción de neuronas o sus conexiones, ya que la información se halla distribuida por toda la red. Esto hace que las redes sean *robustas*.

2.2.1. Tipos de funcionamiento y aprendizaje

Una forma de clasificar a las redes neuronales es tomando en cuenta el problema para el cual serán aplicadas. Hay redes llamadas de *memoria asociativa* que recupe-

¹En teoría es así, pero en la práctica, la mayoría de las redes neuronales implementadas simulan el paralelismo, ya que las computadoras ordinarias sólo cuentan con un procesador.

²Hasta un cierto punto

ran un patrón completo dado un subpatrón, o dado el mismo patrón con cambios; también están las redes de *memoria heteroasociativa* que guardan asociaciones bidireccionales entre los patrones de activación y sus unidades de entrada y salida; existen las redes *clasificadoras de patrones* las cuales después de un entrenamiento aprenden a clasificar patrones de entrada; están las que sirven como *sistemas de reducción de la dimensión* que aprenden representaciones comprimidas de sus entradas; y las que son *sistemas de optimización y satisfacción-restricción* las cuales transforman sus estados de inicio a unos donde las metas de optimización y los problemas de restricción son satisfechos al máximo.

Otra forma de clasificar a las redes neuronales es por medio del tipo de aprendizaje que se lleva a cabo. El aprendizaje consiste en la presentación de patrones (vectores de características) a la red, y la subsiguiente modificación de los pesos de las conexiones siguiendo alguna regla de aprendizaje que trata de optimizar su respuesta, generalmente mediante la minimización del error o la optimización de alguna "función de energía". El modo de aprendizaje más sencillo consiste en la presentación de patrones de entrada junto a los patrones de salida deseados para cada patrón de entrada, por eso se llama *aprendizaje supervisado*. Si no se le presentan a la red los patrones de salida deseados, diremos que se trata de *aprendizaje no supervisado*, ya que no se le indica a la red que resultados debe dar, sino que se le deja seguir alguna regla de auto-organización. Un tercer tipo de aprendizaje, que se encuentra entre los dos anteriores, es el llamado *aprendizaje reforzado*, en este caso el supervisor se limita a indicar si la salida ofrecida por la red es correcta o incorrecta, pero no indica que respuesta debe dar.

En el presente trabajo nos ocuparemos de redes neuronales cuyo aprendizaje es supervisado y que sirven para clasificar y por lo tanto reconocer patrones, es decir, los perceptrones.

2.3. Perceptrón simple

Un posible modelo de neurona artificial es el *perceptrón simple*.

El perceptrón simple sirve para clasificar vectores, en dos clases. Tiene varias entradas y la conexión que existe entre estas y la salida se puede modificar por medio de unos pesos que se aplican a cada una de las entradas. El resultado de tal aplicación es sólo un valor a partir de varios (varias entradas sopesadas, es decir, multiplicadas cada una por un peso) lo cual se consigue mediante una suma y una función de activación.

La manera en que el perceptrón simple clasifica es la siguiente: primero se le entrena para que aprenda a distinguir a que clase pertenecen los objetos a clasificar,

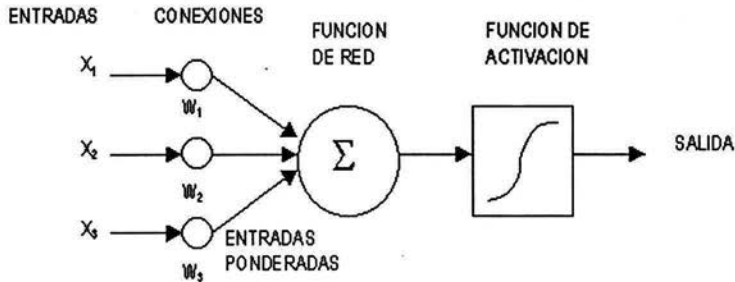


Figura 2.2: Perceptrón simple.

y una vez que ya se entrenó y los pesos de las conexiones se fijaron, entonces ya está listo para clasificar.

Como el perceptrón simple fija los pesos de las conexiones por medio de un aprendizaje supervisado, entonces a la hora del entrenamiento los objetos a clasificar van acompañados por la clase a la que pertenecen, para que aprenda a reconocer, esto en la fase del entrenamiento, pues una vez entrenado el perceptrón, en la fase de prueba, ya no es necesario saber a que clase pertenece el objeto pues de clasificarlo o reconocerlo se encargará el perceptrón.

Los componentes del perceptrón simple son los siguientes (Figura 2.2):

x_i Vectores de entrada

w_i Pesos sinápticos

Σ Función de red

$f()$ Función de activación

θ Umbral de disparo

y Salida

Vectores de entrada

Los vectores de entrada son los vectores de características, es decir, cada elemento del vector es una característica del objeto que se quiere clasificar (o reconocer). Por

ejemplo si tomamos como conjunto los individuos que contestaron una encuesta, cada respuesta de cada individuo es un componente del vector de características de dicho individuo y todas las respuestas del individuo formaran el vector que alimentará a el perceptrón para que este realice la clasificación. Los componentes del vector pueden ser números binarios, retomando el ejemplo de los individuos que respondieron una encuesta, podría ser: estado civil- soltero (0), casado (1), ocupación- estudiante (0), trabajador (1), etcétera; o bien los componentes también pueden ser números reales.

Si dos vectores son iguales, aunque provengan de distintos objetos a clasificar, el perceptrón los tomara como vectores del mismo objeto que por tanto clasificará dentro de la misma clase.

Pesos sinápticos

Estos hacen el papel de las conexiones sinápticas, el peso de la conexión equivale a la fuerza o efectividad de la sinapsis. La existencia de conexiones determina si es posible que una unidad influya sobre otra, el valor de los pesos y el signo de los mismos definen el tipo (excitatorio/inhibitorio) y la intensidad de la influencia.

Conforme se le van presentando los distintos vectores de entrada al perceptrón (en la fase de entrenamiento), los pesos sinápticos se irán modificando mientras que el resultado deseado no sea obtenido, hasta que el perceptrón quede entrenado y entonces los pesos quedaran fijos listos para la fase de prueba.

Función de propagación o de red

Calcula el valor de base o entrada total a la unidad, generalmente como simple suma ponderada de todas las entradas recibidas, es decir, de las entradas multiplicadas por el peso o valor de las conexiones. Equivale a la combinación de las señales excitatorias e inhibitorias de las neuronas biológicas.

Existen varias funciones de propagación, la más utilizada es la función lineal de base (2.1), que es una función de tipo hiperplano, esto es, una función de primer orden.

$$u_i(w, x) = \sum_{j=1}^n w_{ij}x_j \quad (2.1)$$

Otra función es la de base radial (2.2).

$$u_i(w, x) = \sqrt{\sum_{j=1}^n (x_j - w_{ij})^2} \quad (2.2)$$

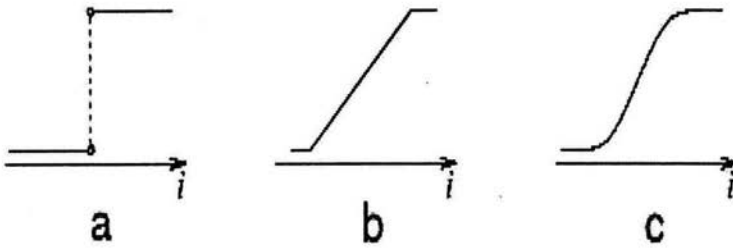


Figura 2.3: Algunas funciones de activación: a) función escalón, b) función de umbral lógico, c) función sigmooidal.

Función de activación

Esta función es quizás la característica principal o definitoria de las neuronas, la que mejor define el comportamiento de la misma. Se usan diferentes tipos de funciones, desde simples funciones de umbral a funciones no lineales. Se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total.

En un principio se pensó que las neuronas usaban una función de umbral, es decir, que permanecían inactivas y se activaban sólo si la estimulación total superaba cierto valor límite; esto se puede modelar con una función escalón, la más típica es el escalón unitario: la función devuelve 0 por debajo del valor crítico (umbral) y 1 por encima. Después se comprobó que las neuronas emitían impulsos de actividad eléctrica con una frecuencia variable, dependiendo de la intensidad de la estimulación recibida, y que tenían cierta actividad hasta en reposo, con estimulación nula. Estos descubrimientos llevaron al uso de funciones no lineales con esas características, como la función sigmooidal (2.3), con un perfil parecido al escalón de una función de umbral, pero continua. En la Figura (2.3) se presentan tres funciones de activación, la función de umbral lógico, de escalón y la sigmooidal respectivamente.

$$y_i = f(u_i) = \frac{1}{1 + e^{-\frac{u_i}{\sigma}}} \quad (2.3)$$

Otra función de activación es la gausiana (2.4).

$$f(u_i) = ce^{\frac{-u_i^2}{\sigma^2}} \quad (2.4)$$

Salida

La salida del perceptrón es el resultado de la función de activación, así que si por ejemplo la función empleada es la función escalón la salida será discreta, generalmente binaria; ahora bien si la función de activación es la función sigmoidea, la salida será un número real $y \in [0, 1]$.

La salida define a que categoría o clase pertenece el objeto cuyas características se pusieron en la entrada del perceptrón. Como ya vimos el perceptrón puede categorizar en dos clases un conjunto de elementos dado, así que se puede fijar la salida cuyo resultado sea 0 para una clase y la salida cuyo resultado sea 1 para la otra clase.

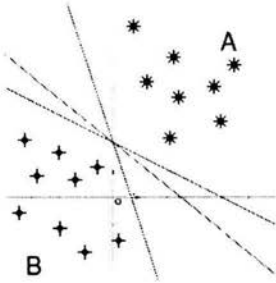


Figura 2.4: Separabilidad lineal, las tres rectas son fronteras de decisión que separan al conjunto de elementos de la clase A de los de la clase B.

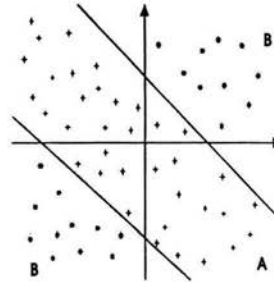


Figura 2.5: Un ejemplo no lineal, se requieren al menos dos rectas para separar el conjunto de puntos A del conjunto de puntos B.

2.3.1. Separabilidad lineal

El perceptrón simple sirve para hacer separaciones lineales (Figura 2.4), es decir, dadas las variables de entrada pertenecientes a un espacio multidimensional, el perceptrón formará dos regiones (clases) divididas por un hiperplano de dicho espacio multidimensional. Esto no es nada despreciable, pues resulta ser un método bastante sencillo para realizar separaciones lineales, pero como se verá más adelante no todos los conjuntos son separables linealmente. En la Figura (2.5) se muestra un ejemplo de ello.

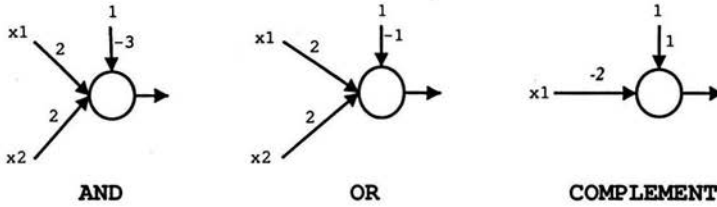


Figura 2.6: Las tres funciones lógicas fundamentales, implementadas cada una por un perceptrón que emplea la función escalón como función de activación.

Existen dos maneras de analizar la funcionalidad del perceptrón; una que ya mencionamos anteriormente, que es su capacidad de hacer separaciones lineales de conjuntos en dos clases; y la segunda viendo al perceptrón como una unidad lógica binaria. El perceptrón es capaz de implementar numerosas funciones lógicas incluyendo las tres operaciones fundamentales del álgebra Booleana (Figura 2.6): la conjunción (AND), la disyunción (OR) y el complemento (COMPLEMENT). Sin embargo el perceptrón no puede implementar todas las funciones lógicas posibles, por ejemplo no puede implementar la función XOR (OR exclusivo). Con n variables existen un total de 2^{2^n} funciones lógicas (2^n renglones en la tabla de verdad 2^{2^n} combinaciones de funciones). La fracción de esas funciones que el perceptrón puede implementar son las llamadas *funciones lógicas umbral* (threshold logic functions). Deduciendo una expresión exacta para el número de funciones lógicas umbral ($NTL(n)$) se prueba que la tarea del perceptrón simple es limitada.

Los siguientes límites lo demuestran [Muroga, 1971].

$$2^{n(n-1)/2} < NTL(n) \leq 2 \sum_{i=0}^n \binom{2^n - 1}{i} < 2^{n^2} \ll 2^{2^n} \quad (2.5)$$

Así que el número de funciones que puede implementar el perceptrón, es un pequeño porcentaje del número total de posibles funciones con n variables.

El problema de la separabilidad lineal se soluciona conectando perceptrones en cascada en dos o más capas, y formando así una red neuronal.

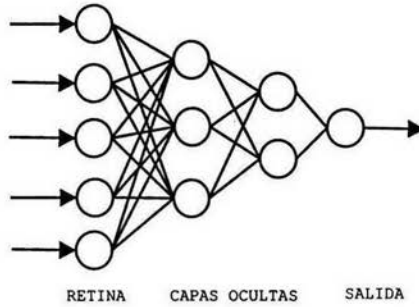


Figura 2.7: Perceptrón multicapa, son tres los tipos de capas: la capa de entrada o retina, las capas ocultas y la capa de salida

2.4. Perceptrón multicapa

Las capacidades de un solo perceptrón están limitadas por fronteras de decisión lineales y funciones lógicas simples. Sin embargo, colocando a los perceptrones en cascada, por capas, se pueden implementar regiones de decisión complejas y expresiones booleanas arbitrarias.

Los perceptrones individuales en la red son llamados neuronas, y normalmente se utiliza la función sigmoide para cada neurona (como función de activación). Son tres los tipos de capas: la capa de entrada o retina, las capas ocultas y la capa de salida (Figura 2.7). El vector de entrada alimenta a la capa de entrada, cada componente del vector va a una neurona y la salida de cada neurona va a cada neurona de la siguiente capa, esto es, cada neurona de la capa i está conectada con cada neurona de la capa $i + 1$, neuronas de una misma capa no están conectadas entre sí. Entonces, el número de neuronas de la capa de entrada está determinada por el número de componentes del vector de entrada, el número de neuronas en las capas ocultas es variable, puede tener más de una capa oculta³ y el número de neuronas de la capa de salida también es variable, este número determina el número de clases en que se quiere clasificar el conjunto. Por ejemplo con una neurona en la

³Aunque está demostrado [Hornik *et al.*, Funahashi, Cybenko, Hartman *et al.*, 1989,1989,1989,1990] que con una sola capa oculta es suficiente para aproximar cualquier función continua o con un número finito de discontinuidades, siempre y cuando la función de activación de las neuronas ocultas sea una función no lineal, cabe señalar que este teorema es de existencia, pues prueba que la red existe pero no indica como construirla y tampoco garantiza que la red aprenderá la función.

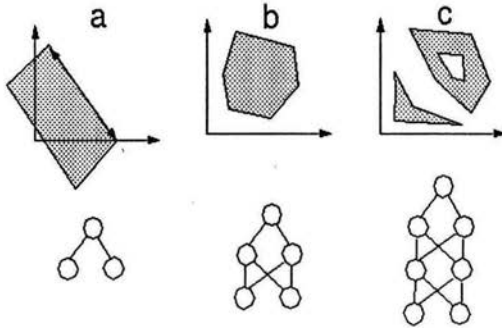


Figura 2.8: Tipos de regiones de decisión que pueden formarse: a) con un perceptrón simple, b) un perceptrón multicapa con una capa oculta, c) un perceptrón multicapa con dos capas ocultas.

capa de salida, si la salida es binaria, el conjunto se puede clasificar en dos, si son dos neuronas en cuatro y así sucesivamente.

En la Figura (2.8) se puede apreciar que el perceptrón de una capa forma regiones de decisión de medio plano delimitados por el hiperplano, que para dos capas se tienen regiones convexas ya sea cerradas o abiertas, y para las tres capas regiones arbitrarias limitadas por el número de neuronas en las capas.

Además del número de capas de una red, en función de como se interconectan unas capas con otras, podemos hablar de redes *recurrentes* (feed-back) y redes *no recurrentes* o redes en cascada (feed-forward). En las redes en cascada la información fluye unidireccionalmente de una capa a otra (desde la capa de entrada a las capas ocultas y de éstas a la capa de salida), y además, no se admiten conexiones intracapa. En las redes recurrentes la información puede volver a lugares por los que ya había pasado, formando bucles, y se admiten las conexiones intracapa (laterales), incluso de una unidad consigo misma.

Las conexiones entre una capa y otra pueden ser totales, es decir, que cada unidad se conecta con todas las unidades de la capa siguiente, o parciales, en las cuales una unidad se conecta con sólo algunas de las capas de la unidad siguiente, generalmente siguiendo algún patrón aleatorio o pseudo-aleatorio (por ejemplo, mediante algoritmos genéticos).

Desde una aproximación temporal se puede distinguir además entre conexiones

sin retardo y conexiones *con retardo*. Esto permite modelar aspectos dinámicos, por ejemplo para modelos psicofisiológicos de memoria.

Un último aspecto, referido a la manera en que las unidades calculan su activación en relación al tiempo es que, si en cada ciclo de aprendizaje se calcula la activación de todas las unidades de una capa, hablamos de redes *síncronas*, en caso contrario se habla de redes *asíncronas* o *probabilísticas*, en las cuales cada unidad de proceso tiene una cierta posibilidad de calcular su activación en cada ciclo de aprendizaje. Las redes asíncronas ofrecen la ventaja de ser menos propensas a caer en mínimos locales (Ver sección 2.4.3).

2.4.1. Algoritmo de retropropagación

Durante mucho tiempo no se dispuso de algoritmos para entrenar redes multicapa, y como las redes de una capa estaban muy limitadas en cuanto a lo que eran capaces de representar, el campo de las redes neuronales artificiales estaba estancado. La invención y perfeccionamiento del algoritmo de retropropagación dio un gran impulso al desarrollo de este campo. Tiene un buen fundamento matemático y a pesar de sus limitaciones ha expandido enormemente el rango de problemas donde se aplican las redes neuronales artificiales. Este algoritmo es una generalización de la regla de aprendizaje delta. La retropropagación consiste en propagar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas ocultas intermedias y ajustando los pesos de las conexiones con el fin de reducir dicho error.

Hay distintas versiones del algoritmo de retropropagación y distintas arquitecturas conexionistas a las que pueden ser aplicados. Aquí se explica una versión del algoritmo para redes que cumplan las siguientes características:

- No recurrentes
 - Función de activación sigmoideal
 - Capas totalmente interconectadas
 - Operación totalmente síncrona
1. Aleatorizamos los pesos de las conexiones.
 2. Presentamos un patrón de entrada y calculamos la salida.

3. Dada una unidad j -ésima de la capa de salida y unidades i -ésimas de la capa oculta inmediatamente anterior, calculamos la entrada total ponderada y la salida de activación de la misma.

$$x_j = \sum_{i=1}^n (y_i \cdot w_{ij}) \quad y_j = \frac{1}{1 + e^{-x_j}} \quad (2.6)$$

4. Una vez calculadas las actividades de todas las unidades de salida se estima el error, generalmente una función cuadrática de los errores individuales cometidos por cada unidad, siendo cada error individual la diferencia entre la salida deseada y la obtenida.

$$E = \frac{1}{2} \sum_j (d_j - y_j)^2 \quad (2.7)$$

donde d_j es la salida deseada para la unidad j -ésima.

Hay que recordar que el objetivo es calcular como varía el error al variar el peso de cada conexión (tasa de variación del error respecto al peso de una conexión, EP).

5. Cálculo de la rapidez de variación del error al cambiar la actividad de cada unidad de salida (EA , error respecto a la actividad)

$$EA_j = \frac{\partial E}{\partial y_j} = d_j - y_j \quad (2.8)$$

Es justamente la diferencia entre la salida deseada y la salida real obtenida, es decir, la diferencia entre la actividad deseada y la actividad real.

6. Cálculo de la rapidez de variación del error al cambiar la entrada total que recibe cada unidad de salida.

$$\delta_j = \frac{\partial E}{\partial k_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial k_j} = EA_j y_j (1 - y_j) = y_j (1 - y_j) (d_j - y_j) \quad (2.9)$$

Es igual a la tasa de variación del error al variar su activación multiplicando por la tasa de variación de la activación al cambiar su entrada (que es justamente la derivada de la función sigmoideal).

7. Cálculo de la rapidez de variación del error al ser modificado un peso de la conexión aferente a una unidad de salida.

$$EP_{ij} = \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial k_j} \frac{\partial k_j}{\partial w_{ij}} = \delta_j y_j \quad (2.10)$$

Es igual a la tasa de variación del error al variar su entrada, por la tasa de variación de la entrada al variar ese peso.

Hasta ahora hemos calculado el error respecto a la actividad sólo para las unidades de salida, ¿qué pasa con las unidades ocultas? En este caso no tenemos una estimación directa del error aportado por cada unidad oculta; aquí es donde interviene la retropropagación del error:

La unidad i -ésima de la capa oculta afecta a todas las unidades de salida, por lo tanto, para estimar como varía el error al variar la actividad de esta unidad oculta, habrá que sumar los efectos individuales de su actividad sobre todas las neuronas de salida. Cada efecto individual sobre la variación del error, será igual a la tasa de variación del error de la unidad de salida al cambiar su entrada total, multiplicado por la tasa de variación de su entrada al variar la actividad de la unidad oculta.

$$EA_i = \frac{\partial E}{\partial y_i} = \sum_{j=1}^m \left(\frac{\partial E}{\partial k_j} \frac{\partial k_j}{\partial y_i} \right) = \sum_{j=1}^m (\delta_j \cdot w_{ij}) \quad (2.11)$$

8. Conociendo EA para las unidades de cualquier capa podemos calcular δ y EP con las expresiones ya conocidas.

$$\delta_i = EA_i y_i (1 - y_i) = y_i (1 - y_i) \sum_{j=1}^m (\delta_j \cdot w_{ij}) \quad (2.12)$$

$$EP_{hi} = \delta_i \cdot y_h \quad (2.13)$$

9. Disponiendo de la tasa de variación del error respecto al peso de una conexión (EP), podemos usar distintas reglas para modificar ese peso con el objetivo de reducir dicho error. Una de las primeras reglas que aprovechó este algoritmo es la regla delta generalizada, que calcula el incremento a aplicar a un peso como una proporción directa de la tasa de variación del error.

siendo η el coeficiente de aprendizaje, típicamente con valores comprendidos entre 0.01 y 1.0

$$\Delta w_{ij} = \eta \delta_j y_j \quad (2.14)$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \quad (2.15)$$

2.4.2. Explicación intuitiva de la retropropagación

Las ecuaciones que se exponen antes pueden ser matemáticamente correctas, pero ¿qué es lo que significan? El proceso de la retropropagación es intuitivamente claro, cuando un patrón de entrenamiento es probado, los valores de activación son propagados hacia las neuronas de salida, y la salida de la red es comparada con los valores de salida deseados, normalmente se termina con un error en cada una de las neuronas de salida. Así que, el objetivo es convertir dicho error en cero. Y la manera de conseguirlo es hacer lo posible por cambiar los pesos de las conexiones en la red neuronal de tal manera que, la siguiente vez que se pruebe, el error sea cero con ese patrón de entrenamiento. Sabemos por la regla delta que para reducir el error tenemos que adaptar los pesos de acuerdo a:

$$\Delta w_{ho} = (d_o - y_o)y_h \quad (2.16)$$

Este es el primer paso, pero no es suficiente; cuando aplicamos solo esta regla, los pesos de las neuronas de entradas y de las neuronas de las capas ocultas nunca son modificados. Así que querríamos aplicar de nuevo la regla delta para las capas ocultas, pero no tenemos el valor de δ para las neuronas ocultas. Esto se soluciona con la regla de la cadena que hace lo siguiente: distribuye el error de una neurona de salida hacia todas las neuronas ocultas con las que esta conectada. Visto de otra manera, el error es una función multivariable y lo que se busca es minimizar dicho error, esto se consigue igualando la derivada de dicha función a cero.

2.4.3. Deficiencias de la retropropagación

Pese al aparente éxito del algoritmo de retropropagación, este presenta ciertos problemas, algunos referentes a su dudosa plausibilidad neurofisiológica, y otros a ciertos aspectos computacionales, que son los que comentaremos aquí.

La red se puede paralizar, cuando los pesos quedan ajustados a valores muy grandes, esto hace operar a las unidades del proceso con una activación muy próxima a 1, por lo tanto, el gradiente del error tiende a 0, en consecuencia no se producen

modificaciones en los pesos y el aprendizaje queda detenido. Por esta razón es conveniente aleatorizar los pesos de las conexiones con valores pequeños y usar la tasa de aprendizaje también pequeña aunque esto alenta el aprendizaje.

La red puede ser inestable temporalmente, esto sucede si usamos un coeficiente de aprendizaje elevado, pues se van a producir incrementos grandes en los pesos, de manera que es fácil pasarse de incremento y tener que tratar de compensarlo en el siguiente ciclo, por lo que se producirían oscilaciones continuas. Esto se soluciona usando un coeficiente pequeño, o para no tener un aprendizaje muy lento, modificar dicho coeficiente adaptativamente, es decir, aumentarlo si el error global disminuye, y disminuirlo en caso contrario.

Por otro lado, se pueden llegar a mínimos locales de la función de error puesto que el algoritmo de retropropagación usa la técnica del gradiente descendente, esto significa que sigue la “superficie del error” siempre hacia abajo, hasta alcanzar un mínimo local, pero no garantiza que se alcance una solución globalmente óptima.

Los resultados dependen de los valores iniciales, aleatorios, de las conexiones. Esto hace que sea conveniente entrenar varias veces la red neuronal con distintos valores iniciales y elegir la red con los pesos que mejor funcione.

A veces se requiere mucho tiempo para obtener soluciones sencillas. Este problema se reduce gracias al aumento de potencia de los procesadores y al uso de nuevas tecnologías; sin embargo, el tiempo de cómputo se incrementa mucho al aumentar el tamaño de la red. Si bien el volumen de cálculo es proporcional al número total de conexiones, en la práctica, al aumentar el tamaño de la red, hacen falta más ejemplos de aprendizaje, y eso provoca un aumento mucho mayor del tiempo de aprendizaje. Para incrementar la velocidad de convergencia se han desarrollado diferentes modificaciones del algoritmo.

Cabe señalar que existen algunas variantes del algoritmo de retropropagación que aminoran algunos de los problemas antes mencionados y que por tanto mejoran notablemente el desempeño, tal es el caso de la variante del algoritmo que recurre al *recocido simulado* (simulated annealing), que es un método que se inspira en el modo como se temple el acero en la industria metalúrgica: primero se calienta hasta temperaturas muy altas y luego se deja enfriar, de manera que pase de estados de alta energía a estados de baja energía, hasta que alcanza un mínimo energético global, así el acero queda mucho más resistente. Esta variante es utilizada para evitar caer en mínimos locales.

Por otro lado el algoritmo de retropropagación y sus diversas variantes no son la única alternativa para minimizar la función de error o visto de otra manera, para obtener los pesos sinápticos, esto también se puede llevar a cabo haciendo uso de los algoritmos genéticos.

Es importante hacer notar que son tres los factores que influyen en el buen

desempeño de un perceptrón multicapa, estos son:

1. El algoritmo de aprendizaje y el número de iteraciones que se efectúan, esto determina que tan buena es la minimización del error con el conjunto de entrenamiento.
2. El número de ejemplos para el entrenamiento, esto determina que tan bien están representadas las clases.
3. El número de neuronas ocultas, esto determina la capacidad de expresión de la red.

Al principio de esta sección se describieron algunos puntos, que son importantes para minimizar el error con el algoritmo de aprendizaje que se escoja (puede ser retropropagación, simulated annealing, algoritmos genéticos, etcétera); ahora veremos los efectos que tienen, el número de ejemplos para el entrenamiento y el número de neuronas ocultas, en el desempeño de la red.

La topología de una red neuronal, es decir, el número de nodos y la ubicación y el número de conexiones entre ellos, tiene un impacto significativo en el desempeño de la red y su habilidad para generalizar. La densidad de conexiones en una red neuronal determina su habilidad para almacenar información. Si una red no tiene suficientes conexiones entre nodos, el algoritmo de entrenamiento puede no converger nunca; la red neuronal no es capaz de aproximar la función. Por el otro lado, en una red densamente conectada, puede ocurrir el sobreajuste (*overfitting*). El sobreajuste es un problema de los modelos estadísticos donde se presentan demasiados parámetros. Esto es una mala situación porque en lugar de aprender a aproximar la función presente en los datos, la red simplemente puede memorizar cada ejemplo de entrenamiento. El ruido en los datos de entrenamiento se aprende entonces como parte de la función, a menudo destruyendo la habilidad de la red para generalizar.

Por otro lado el número de ejemplos para el entrenamiento también afecta el desempeño de la red. El error del conjunto de entrenamiento se incrementa con un conjunto de entrenamiento grande, el error del conjunto de prueba se decrementa con un conjunto de entrenamiento grande. Un error pequeño con un conjunto de entrenamiento pequeño no es garantía de un buen desempeño de la red. Incrementando el número de ejemplos de entrenamiento los dos errores convergen al mismo valor. Este valor depende de la capacidad de representación de la red, es decir, dados los pesos optimizados, que tan buena es la aproximación. Este error depende del número de capas ocultas y de la función de activación. Si el error del conjunto de entrenamiento no converge al error del conjunto de prueba el aprendizaje no encontró un mínimo global.

2.5. Algunos ejemplos

Las redes neuronales son ampliamente usadas, y con muy buenos resultados, en muy diversas áreas. Existen aplicaciones en el área de finanzas, en la medicina, en las ingenierías, en robótica, etcétera; hay aplicaciones en prácticamente todas las áreas del conocimiento así como en áreas tecnológicas.

Aquí se presentan tres ejemplos de aplicaciones de perceptrones multicapa.

Un ejemplo famoso es NETtalk [Sejnowski & Rosenberg, 1987] que toma como entradas textos escritos y como salidas deseadas los códigos elegidos para representar los fonemas correspondientes. Mediante la ayuda de un sintetizador (DECtalk) se transforman los códigos en fonemas. Durante el proceso de aprendizaje se observó como iba mejorando su habilidad desde un nivel de bebé hasta el nivel de un niño de 6 años, aprendiendo a hacer distinciones difíciles como pronunciar una *c* suave o fuerte según el contexto. Si bien esto ya se había conseguido antes, la novedad más importante reside en que mediante la red neuronal no es necesario definir y programar un montón de complejas reglas, pues la red extrae automáticamente el conocimiento necesario.

James McClelland y David Rumelhart [Rumelhart] estudiaron la forma en que construimos las reglas sobre el lenguaje, y trataron de enseñar a una red neuronal el pasado de los verbos ingleses. El sistema fue mejorando y al final era capaz de generalizar y conjugar verbos desconocidos.

Otro ejemplo ahora del área médica, es el presentado por Agnessa Babloyantz [Babloyantz & Ivanov, 1998] que utiliza redes neuronales para la clasificación de eventos cardiacos. Este método puede ser una herramienta poderosa para diagnósticos médicos.

Capítulo 3

Los genes

Desde que el hombre comenzó a sembrar cosechas o a criar animales, le debió de haber resultado obvio que cada semilla o cada óvulo fecundado debe de contener escondido un plan, o diseño, para el desarrollo del organismo. En la época moderna, la ciencia de la genética se desarrolló alrededor de la premisa de unos elementos invisibles portadores de información, denominados *genes*, que son distribuidos a las dos células hijas producidas en cada división celular. Antes de dividirse, una célula debe hacer una copia de sus genes con el fin de ceder a sus células hijas una colección completa de ellos.

Los genes de los espermatozoides y de los óvulos, transmiten la información genética de una generación a la siguiente.

Aunque la herencia de los caracteres biológicos parece misteriosa, lógicamente debe implicar la existencia de esquemas de átomos que sigan las leyes de la física y de la química: en otras palabras, los genes deben estar formados por moléculas. Al principio, la naturaleza de estas moléculas resultaba difícil de imaginar. ¿Qué tipo de molécula podría ser almacenada en una célula, dirigir las actividades de un organismo en desarrollo, y ser capaz además de una replicación exacta y casi ilimitada?

Hacia finales del siglo XIX los biólogos habían reconocido ya que los *cromosomas* que resultan visibles en el núcleo cuando una célula empieza a dividirse, eran los transportadores de la información heredada. Pero la evidencia de que es el ácido desoxirribonucleico (DNA) de estos cromosomas, la sustancia de la que están formados los genes, se obtuvo mucho más tarde a partir de unos estudios sobre bacterias. En 1944 Oswald T. Avery demostró que la adición del DNA purificado de una cepa de bacterias, confería propiedades hereditarias, características de dicha cepa, a una segunda cepa bacteriana ligeramente diferente. Este descubrimiento constituyó una sorpresa; de hecho, no fue aceptado de forma general hasta principios de los años 50, ya que hasta entonces se había creído que sólo las proteínas poseían una complejidad

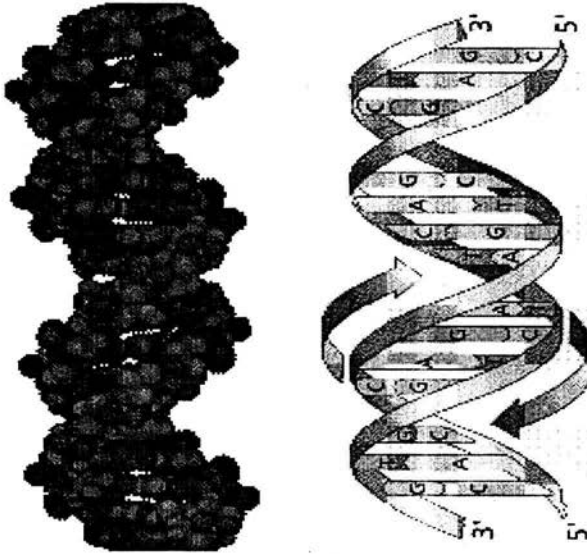


Figura 3.1: Esquema simplificado de una doble hélice de DNA.

conformacional suficiente como para ser portadoras de la información almacenada en los genes. Actualmente, la idea de que el DNA contiene la información genética (almacenada en su larga cadena de nucleótidos) es tan fundamental para el pensamiento biológico que a veces resulta difícil el darse cuenta del enorme abismo intelectual que superó.

3.1. DNA

La molécula de DNA es un polímero lineal (Figura 3.1) construido como una sucesión de cuatro monómeros distintos llamados nucleótidos. Cada monómero tiene una parte constante (moléculas de fosfato y desoxirribosa) y una parte variable que puede ser cualquiera de las siguientes cuatro bases nitrogenadas: Adenina (A), Guanina (G), Timina (T) y Citosina (C). El DNA es una doble hélice formada por dos polímeros lineales antiparalelos en forma de escalera, los “pasamanos” de la escalera son las moléculas de fosfato y de desoxirribosa mientras que los “peldaños” son una de las cuatro bases anteriormente mencionadas, una de cada lado unidas

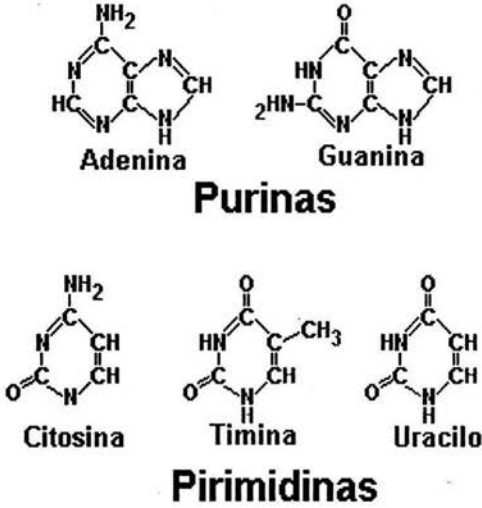


Figura 3.2: Las bases que componen la parte variable del DNA (el uracilo del RNA) arregladas según su naturaleza purínica o pirimídica.

entre sí mediante puentes de hidrógeno. Adicionalmente, se debe mencionar que las bases nitrogenadas pueden ser de dos tipos: la A y la G pertenecen a la familia de las purinas y son moléculas de dos anillos (*“grandes”*) (Figura 3.2); por otro parte, la T y la C pertenecen a la familia de las pirimidinas y son moléculas de un anillo (*“pequeñas”*).

Como cada peldaño está formado por una base de cada lado, para mantener homogénea la anchura de la escalera es necesario que si en un lado de la doble hélice se encuentra una purina, del lado opuesto le corresponda una pirimidina. Esto, aunado al hecho (Figura 3.3) de que no todas las uniones purina-pirimidina son posibles debido a la incompatibilidad de formar adecuadamente los enlaces de hidrógeno, determina el siguiente principio de complementariedad entre los lados de la escalera: cada vez que de un lado aparece una molécula de adenina del otro lado existe una timina con la cual está aparejada y viceversa, mientras que cada vez que aparezca una de citosina se hallará aparejada con una de guanina y viceversa. Por ello, la información contenida en la molécula se puede deducir de una sola rama de la doble hélice.

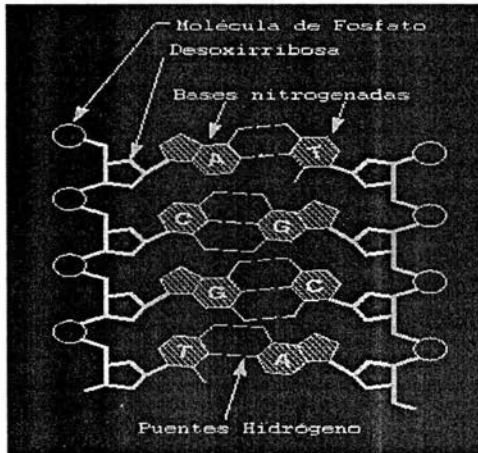


Figura 3.3: La adenina sólo puede aparearse con la timina mediante dos puentes de hidrógeno, mientras que la citosina sólo lo hace con la guanina mediante tres puentes de hidrógeno.

La unión de los peldaños de la escalera no es homogénea; las parejas A–T están unidas mediante dos puentes de hidrógeno, mientras que las parejas G–C lo están mediante tres puentes de hidrógeno. Esto sugiere que las secciones de la cadena más abundantes en G o en C sean, en promedio, más rígidas que aquellas en las cuales la abundancia relativa de A–T sea mayor, en el sentido de la energía necesaria para desnaturalizar (separar las dos ramas) la molécula. Esto, como se verá más adelante, no es necesariamente cierto. Por otra parte, la diferencia de tamaños entre las purinas y las pirimidinas hace que la escalera no sea geoméricamente uniforme; si de un sólo lado se alternan purinas y pirimidinas la escalera se encontrará dislocada, de manera que podemos afirmar que en este caso la molécula es, en promedio, rugosa. Si, localmente, se encuentran solamente purinas la cadena será lisa (en realidad la situación es mucho más complicada pero para nuestros propósitos esta descripción es suficiente). De aquí en adelante, llamaremos a las bases G y C *fuertes* y a A y T, *débiles*. Por otro lado las bases también se pueden emparejar tomando en cuenta su naturaleza amina o cetona, A y C son *aminas* y G y T son *centonas*.

La cadena de DNA contiene toda la información necesaria para la producción de cadenas de aminoácidos (polipéptidos) que, a su vez, dan lugar a las proteínas que son la materia prima tanto estructural como funcional de todo ser vivo. En el DNA

se encuentra codificada la herencia morfológica y funcional que una célula transmite a sus descendientes.

La doble hélice se abre como un cierre de cremallera, genéricamente hablando, para dos procesos diferentes de la vida celular. Uno de ellos es el proceso de replicación en el cual el DNA se duplica generando dos copias idénticas (salvo por mutaciones o errores de copiado) quedando una de estas copias en cada célula producto de la división celular. El otro proceso, llamado de transcripción-traducción, es aquel mediante el cual la información contenida en el DNA determina la manera en que una célula produce los polipéptidos que a la larga darán lugar a las proteínas. Un segmento de DNA que contiene la información necesaria para la síntesis de una proteína se llama *gen estructural*.

3.1.1. Eucariontes y procariontes

Basándose en criterios citológicos los seres vivos se pueden clasificar en: *procariontes* y *eucariontes*; las primeras no tienen un núcleo cuya pared separe al material genético del resto del protoplasma, un ejemplo de ellas son las bacterias. Las células eucariontes, por otra parte, poseen un núcleo bien definido en el cual reside el DNA, todos los organismos pluricelulares están conformados por células eucariontes, en estas, no toda la molécula de DNA está constituida por genes; de hecho, estos conforman una mínima parte de la secuencia. Los genes están separados entre sí por largos segmentos intergénicos cuya función, si es que la tienen, no ha sido aún dilucidada. Aún más, un gen está constituido por uno o varios segmentos llamados exones que contienen la información que codifica la producción de polipéptidos y por islas entre ellos llamadas intrones cuya utilidad o falta de ella tampoco se conoce cabalmente. Al conjunto de material genético de una célula se le llama *genoma*. El genoma humano consta aproximadamente de 2000 megabases (Mb)¹ de las cuales, casi el cinco por ciento codifica para la producción de polipéptidos o regula el funcionamiento de esta producción; la función del resto, en caso de haberla, es desconocida.

3.1.2. Transcripción y traducción

La manera en que se lleva cabo el proceso de transcripción-traducción es la siguiente: en algún momento de la vida celular el segmento de DNA correspondiente

¹Un virus tiene del orden de 10 Kb (Kb, mil bases), los animales y plantas más comunes tienen aproximadamente 2000 Mb, algunos sapos y salamandras pueden llegar a tener 20,000 Mb. Curiosamente, el organismo conocido que tiene el mayor número de bases en su genoma es un protozoario muy simple: la amiba *Chaos chaos* con alrededor de 1 Gb (un millón de Mb).

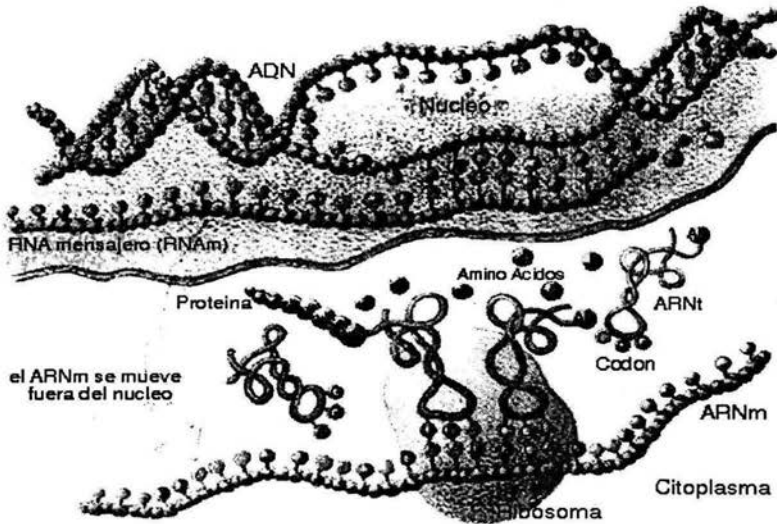


Figura 3.4: Transcripción – traducción

a un gen estructural se abre como se mencionó anteriormente y en lugar de duplicarse en una molécula similar, la sucesión de bases se transcribe en una molécula de ácido ribonucleico (RNA)², conocido, en esta etapa, como RNA premensajero.

Después, en las células eucariontes, con ayuda de un sistema formado por proteínas y RNA; conocido como *espliceosoma*, se eliminan los segmentos intrónicos y se “pegan”, en sucesión, las partes codificadoras; este proceso se conoce como *edición*. La molécula resultante, llamada RNA mensajero, viaja a una estructura de la célula conocida como ribosoma. En ésta, se lee secuencialmente el RNA y por cada tres bases se agrega un aminoácido en una cadena lineal conocida como *polipéptido* y que es la precursora de una proteína. A cada terna de bases se le llama *codón* (Figura 3.4).

² Siguiendo el principio de complementariedad mencionado arriba. De esta manera se obtiene una copia de una de las ramas de DNA, la única diferencia es que en el RNA se reemplaza la timina (T) por uracilo (U), que también es una pirimidina con propiedades geométricas y químicas muy semejantes a las de la timina.

1era posición	2da posición				3era posición
	T	C	A	C	
T	Fen	Ser	Tir	Cis	T
	Fen	Ser	Tir	Cis	C
	Leu	Ser	Alto	Alto	A
	Leu	Ser	Alto	Trp	G
C	Leu	Pro	His	Arg	T
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Tre	Asn	Ser	T
	Ile	Tre	Asn	Ser	C
	Ile	Tre	Lis	Arg	A
	Met	Tre	Lis	Arg	G
G	Val	Ala	Asn	Gli	T
	Val	Ala	Asn	Gli	C
	Val	Ala	Glu	Gli	A
	Val	Ala	Glu	Gli	G

Cuadro 3.1: El código genético estándar. "Alto" es el codón de terminación de mensaje, los mensajes generalmente comienzan con el aminoácido Met

3.1.3. Código genético

Dado que existen cuatro bases, pueden formarse sesenta y cuatro codones distintos de los cuales todos, excepto tres, dan lugar a aminoácidos (Cuadro 3.1). Sin embargo, en la naturaleza existen tan sólo veinte aminoácidos diferentes que se emplean en la construcción de proteínas, de lo que se infiere que hay codones que, aunque distintos, son sinónimos para un aminoácido.

3.2. Adquisición de nuevos genes

Hace 3.5 billones de años aparecieron las primeras células³, similares en apariencia a una bacteria moderna. Dado que es bastante posible que la vida haya tenido un origen común, todos los organismos vivos sean procariontes o eucariontes tienen como ancestro común a dichas células primitivas. Se piensa que la vida tiene un origen único pues todos los organismos contienen DNA en sus células y el código genético es “universal”⁴ para todas ellas, es decir, todas las células codifican los aminoácidos con las mismas reglas de correspondencia (codón-aminoácido). La evolución morfológica ha sido acompañada por la evolución de los genomas. Una indicación de la complejidad en el genoma es el número de genes, que varía de al menos 500 (*Mycoplasma genitalium*) en algunas bacterias hasta ~30,000 o tal vez más⁵ en los vertebrados como los humanos.

Existen dos maneras principales en que los genes pudieron ser adquiridos por un genoma:

- Por duplicación de algunos o todos los genes existentes en un genoma.
- Adquiriendo genes de otras especies.

Ambos eventos son importantes para la evolución del genoma, pero se ahondará en el segundo caso ya que es el tema que concierne a este trabajo.

3.2.1. Transferencia horizontal

La transferencia horizontal de genes es la transferencia de material genético entre células y genomas que pertenecen a especies no relacionadas, por procesos distintos a la reproducción. En el proceso inicial de reproducción, los genes son transferidos verticalmente de los padres a su descendencia; este proceso ocurre sólo en la misma especie o en especies muy relacionadas entre sí.

En las bacterias, existen tres mecanismos por medio de los cuales se pueden transferir genes entre especies⁶, estos tres mecanismos son la *conjugación*, la *transducción* y la *transformación*.

³A esta bacteria la han llamado LUCA (last universal common ancestor)

⁴El código genético mitocondrial es la única excepción a la universalidad del código, de manera que en algunos organismos los aminoácidos determinados por el mismo triplete o codón son diferentes en el núcleo y en la mitocondria.

⁵De hecho existe una planta con ~40,000 genes (*Arabidopsis thaliana*)

⁶¡Ojo!, esto no asegura que posteriormente formen parte del genoma



Figura 3.5: Dos células de *E. coli*, intercambiando material genético mediante conjugación. Las bacterias pueden llevar a cabo la transferencia horizontal de genes, incluso entre individuos de diferentes especies

La conjugación es un proceso en el cuál dos bacterias establecen contacto físico y una bacteria (la donadora) transfiere DNA a la segunda bacteria (la receptora) (Figura 3.5). El DNA transferido puede ser una copia de una parte o posiblemente de todo el cromosoma de la célula donadora, o un segmento de DNA cromosómico (más de 1Mb de longitud) de un plasmido (DNA circular).

Por otro lado la transducción es un mecanismo de transferencia génica que acontece cuando un bacteriófago (virus que infecta a bacterias) se adhiere a una bacteria y le inyecta su DNA. En el interior de la bacteria, el DNA inyectado se replica y el cromosoma de la bacteria se degrada. Lo normal es que el DNA vírico se empaquete en nuevas partículas víricas, que salen al exterior tras la lisis de la célula. Durante la transducción, sin embargo algunas partículas capturan DNA bacteriano (con genes bacterianos) y lo transportan hasta una segunda bacteria, que puede terminar incorporándolo en su propio cromosoma.

Mientras que la transformación es un proceso en el que las células absorben DNA libre liberado por una bacteria muerta. El DNA es atrapado e introducido por un complejo proteínico capaz de unirse al DNA, presente en la superficie de la bacteria. Durante el proceso, ciertas enzimas degradan una de las dos cadenas de la molécula. La otra puede acabar integrándose en el cromosoma de la bacteria. Raramente se

absorben mas de 50Kb.

3.2.2. Virus

Los virus son agentes causantes de enfermedades, que se multiplican únicamente en las células y que en virtud de su diminuto tamaño atraviesan los filtros ultra finos que retienen incluso a las bacterias más pequeñas. Son genes desnudos que han adquirido la capacidad de desplazarse de una célula a otra.

Es casi seguro que los virus han desempeñado un papel importante en la evolución de otros organismos por el hecho de presentar la capacidad extraordinaria de transferir DNA a través de las barreras entre especies. La mayoría de virus se recombinan frecuentemente con los comosomas de sus huéspedes, tomando al azar piezas de estos cromosomas y transportándolos a células u organismos diferentes.

3.2.3. Transgénicos

Otra manera de adquirir genes es por medio de modernas técnicas que se dan en la ingeniería genética, que altera la información genética propia de los organismos con la adición de muy pocos genes (generalmente uno) procedentes de otro organismo, normalmente de otra especie.

Capítulo 4

Métodos

4.1. Introducción

Como ya se expuso en el capítulo 2, los perceptrones multicapa son redes neuronales particularmente adecuadas para los problemas de reconocimiento de patrones. En nuestro caso lo que se buscó fue reconocer patrones en secuencias de DNA de dos especies distintas, para así poder determinar si hubo transferencia horizontal de genes entre éstas. La red neuronal fue entrenada con vectores de características de cada especie, para que aprendiera a distinguir entre secuencias de una y otra especie. Así, una vez entrenada la red, se le podían presentar vectores de características, de un segmento de DNA de una de las especies con las cuales se entrenó la red, y determinar si ese segmento (o gen) pertenece a dicha especie, o si pertenece a la otra.

Los componentes de los vectores de características son los sensores; más adelante se explicará como fueron obtenidos, por qué es que caracterizan al DNA y su relevancia.

En este trabajo se tomó como referencia un trabajo de investigación de Uberbacher [Uberbacher & Mural, 1991], donde se emplearon redes neuronales para analizar secuencias de DNA, y localizar regiones donde se codifican proteínas en el genoma humano.

4.2. La red neuronal

Se utilizó una red neuronal (perceptrón multicapa) entrenada con el algoritmo de retropropagación para la determinación de los pesos sinápticos adecuados, de tal manera que la función del error fuese mínima.

Como el perceptrón multicapa es una red cuyo entrenamiento es supervisado, los vectores de características van acompañados por su respectivo resultado deseado, 0 para un organismo y 1 para el otro.

Dado que la función de activación es la función sigmoide (función continua con codominio en $[0,1]$), a la hora de utilizar la red los resultados no son discretos y algunos vectores de características podrán dar como resultado cualquier valor entre el intervalo $[0,1]$. Por lo cual se aplicó un umbral, a los vectores que como resultado daban $x \geq 0,5$ se les asignó 1 y a los eran $x < 0,5$ se les asignó 0, aunque en muchos casos la aplicación del umbral no fue necesario pues como se verá en el próximo capítulo los resultados eran muy claros.

4.2.1. Arquitectura

En un principio la red neuronal propuesta tenía 4 capas; con 8 neuronas en la capa de entrada (tantas como sensores¹), una neurona en la capa de salida, y dos capas ocultas de 14 y 5 neuronas respectivamente. Se comenzó con esta arquitectura tomando como referencia el trabajo de Uberbacher [Uberbacher & Mural, 1991]. Pero después de muchas pruebas se observó que, en algunos casos la red no se entrenaba adecuadamente, es decir, la red no aprendía a reconocer los dos organismos, por lo que se realizaron cambios en la arquitectura. En los artículos de Hush [Hush & Horne, 1993] y de Lippmann [Lippmann, 1987] se exponen algunas heurísticas o reglas para determinar el número de neuronas por capa.

Hush y Horne proponen que el número de neuronas en la capa oculta debería ser menor que el número de ejemplos. Por otro lado el número de neuronas debe ser $3n$ donde n es igual al número de entradas, para regiones de decisión de tipo esférico.

Mientras que Lippmann propone que, si la región de decisión se forma como unión de hipercubos se necesitan $M(2n + 1) + 1$ ($2n$ en la capa oculta por cada hipercubo y una más para hacer el AND y una para el OR).

Tomando en cuenta lo antes descrito y haciendo múltiples pruebas fue que se escogió la arquitectura final que es la que se muestra en la Figura (4.1), con tres capas, ocho neuronas en la capa de entrada, cinco neuronas en la capa oculta y una neurona en la capa de salida.

¹Los sensores son los componentes del vector de características

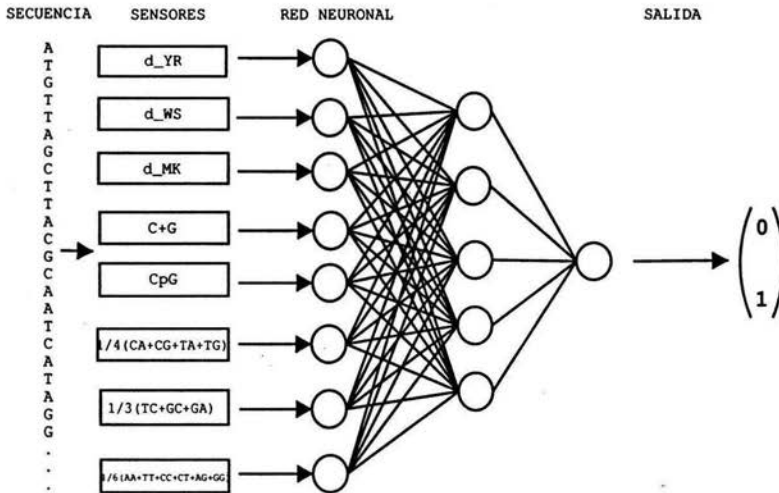


Figura 4.1: La arquitectura de la red neuronal.

4.2.2. Sensores

Los aquí llamados sensores son las variables que van a caracterizar a los objetos, estos son una recopilación de parámetros que caracterizan al DNA y se usan como datos de entrada en la red neuronal para que aprenda a reconocer con base en ellos.

El DNA no es únicamente una sucesión de letras, si así lo fuera bastaría con analizar dicha sucesión como un lenguaje en términos gramaticales. Pero cada una de las letras representa una base que es una molécula con características físico-químico particulares.

Las cuatro bases (A, C, G y T) –como se vio en capítulo 2– se pueden agrupar por pares de acuerdo a sus características físicas, químicas y geométricas. En efecto:

Recordemos que A y T se encuentran unidas por dos puentes de hidrógeno mientras que C y G se encuentran unidas por tres puentes de hidrógeno (Figura 3.3).

Análogamente, la pareja A y G corresponde al grupo de las purinas que son moléculas “grandes” y la pareja C y T son pirimidinas que son moléculas “pequeñas” (Figura 3.2).

De la misma manera se pueden emparejar a las bases tomando en cuenta su naturaleza amínica o cetónica, A y C son aminas y G y T son cetonas.

Estas tres formas de agrupar a las bases son las tres dicotomías, que en sus siglas

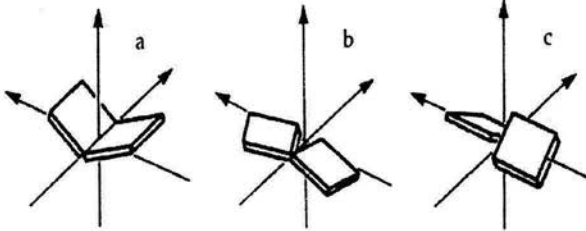


Figura 4.2: Ángulos estructurales del DNA: a) Broche, b) Abertura, c) Hélice

en ingles son:

- YR pirimidinas - purinas
- WS débiles - fuertes
- MK aminas - cetonas

Geoméricamente hablando, la molécula de DNA es una molécula irregular y rugosa; su flexibilidad y estabilidad no son uniformes a lo largo de toda la secuencia. Las propiedades termodinámicas y estructurales dependen de la secuencia de nucleótidos. La unidad más pequeña que sirve para medir la distribución de nucleótidos es el dinucleótido pues éste y sus bases complementarias definen las principales características locales del DNA como los ángulos intranucleótidos y las energías de apilamiento.

Con base en las tres dicotomías fue que Miramontes y colaboradores [Miramontes *et al.*, 1995] crearon el IDH (Index of DNA Homogeneity), el cual sirve para distinguir organismos con diferentes orígenes evolutivos. Las tres dicotomías pueden ser simultáneamente analizadas, la secuencia original de DNA se traduce en una secuencia binaria; por ejemplo para WS se le asigna un 0 a las bases "fuertes" y un 1 a las "débiles" y análogamente para las otras dos dicotomías. Para medir las variaciones locales de una secuencia de DNA dada, es necesario contar el número de ceros y de unos que se encuentran en cada una de las representaciones binarias; este es el propósito del IDH (Ver ecuación 4.1).

$$d = \frac{(N_{00}N_{11}) - (N_{01}N_{10})}{N_0N_1} \quad (4.1)$$

	C	T	A	G
G	H	I	H	L
A	I	I	L	L
T	H	L	V	V
C	L	L	V	V

Cuadro 4.1: Resultados reportados por Quintana y Dickerson [Quintana *et al.*, 1992]. Los ángulos estructurales entre 2 bases son: *L* - low twist, *H* - high twist *I* - intermediate twist *V* - variable twist

Donde N_{ij} es el número de dinucleótidos ij , N_0 y N_1 son el número total de ceros y de unos en la secuencia y su producto normaliza el índice al intervalo $[-1, 1]$.

El IDH mide el grado de alternancia de las bases, como se puede observar en la ecuación (4.1), el primer término cuenta los ceros o unos consecutivos, mientras que el segundo término cuenta los unos y ceros alternados; por esta razón si el IDH es cercano a 1 la secuencia es agregada; en contraparte si el IDH es cercano a -1 la secuencia es alternada, pues el segundo término predominará. Un IDH cercano a 0 corresponde a una secuencia aleatoria. Ahora bien, tomando en cuenta que el IDH se puede medir para las tres dicotomías, hay que hacer notar que para el caso de d_{YR} el índice nos proporciona información de que tan lisa o rugosa es la secuencia de DNA que se analiza y por otro lado el índice d_{WS} nos dice que tan flexible o rígida es la secuencia de DNA [Miramontes *et al.*, 1995].

Por otra parte se ha documentado que la abundancia $C+G$ y C_pG son variables que sirven para diferenciar grupos taxonómicos [Naya *et al.*, 2002].

Según han reportado Dickerson y colaboradores [Quintana *et al.*, 1992] los ángulos estructurales del DNA (Figura 4.2) se pueden agrupar en cuatro categorías que dependen de la correlación de primeros vecinos entre bases.

Las cuatro categorías son: *H* torcedura alta, *I* torcedura intermedia, *L* torcedura baja y *V* torcedura variable. En el cuadro (4.1) se pueden observar los tipos de torcedura que tiene cada posible pareja de bases vecinas.

Para ésto, se mide la matriz de correlaciones:

$$C_{ij} = f_{ij} - f_i * f_j \quad (4.2)$$

Donde f es la frecuencia, por lo que $f_i = \frac{\#i}{n}$ es, el número de apariciones de i entre el total de bases (n) y $f_{ij} = \frac{\#(i \text{ seguida de } j)}{n}$.

La fórmula de correlación (ecuación 4.2) indica cuando un dinucleótido ij está subrepresentado o sobrerrepresentado comparado con una distribución aleatoria. Ésta fórmula se ha ocupado para diversos análisis del DNA [Miramontes & Cocho, 2003].

De acuerdo con lo anterior, los sensores que se utilizaron son:

1. d_{YR}
2. d_{WS}
3. d_{MK}
4. $\%C+G (f_C + f_G)$
5. $\%C_pG (f_{CG})$
6. $\frac{1}{4}(C_{CA} + C_{CG} + C_{TA} + C_{TG})$
7. $\frac{1}{3}(C_{TC} + C_{GC} + C_{GA})$
8. $\frac{1}{6}(C_{AA} + C_{TT} + C_{CC} + C_{CT} + C_{AG} + C_{GG})$

Donde los primeros tres son los índices antes descritos para cada una de las dicotomías, los siguientes dos corresponden a los conteos de C sumado a los de G y a los de C seguida de G, y los últimos tres corresponden a combinaciones lineales de las parejas de bases cuyos tipos de torcedura son variable, alta y baja como se puede apreciar en la matriz de correlación (Cuadro 4.1).

4.3. Implementación de la red

4.3.1. Entrenamiento y validación

El entrenamiento consistió en pasarle a la red neuronal los sensores previamente calculados de cada organismo, cada vector de sensores iba acompañado de un 0 o



Figura 4.3: Ventana móvil

de un 1, dependiendo del organismo, para que la red fuese modificando sus pesos conociendo el valor esperado (entrenamiento supervisado, página 17). Los vectores de sensores de las dos especies se intercalaron para que el aprendizaje no fuese sesgado. La red se entrenó con 20,000 vectores de sensores.

Una vez entrenada la red, se verificó que hubiese sido bien entrenada, primero con un subconjunto de 100 vectores, escogidos aleatoriamente, del conjunto de entrenamiento; posteriormente con 100 vectores “nuevos” los cuales no se habían ocupado para el entrenamiento. Esta validación o verificación se hizo comprobando que el resultado de cada vector fuese igual, o cercano² al esperado. Cuando más del 95 % de vectores daban el resultado deseado se consideró que había sido un entrenamiento exitoso.

4.3.2. Cálculo de los sensores

Los sensores se calcularon en tres momentos distintos del proceso:

- i. Para el entrenamiento; para entrenar a la red los sensores se obtuvieron de subsecuencias de 300 bases de longitud escogidas aleatoriamente a lo largo de toda la secuencia. A cada vector se le adjunto otro dato, un 0 o un 1 para diferenciar entre las dos especies y que la red neuronal tuviera los valores esperados para cada vector.
- ii. Para la verificación; para verificar que la red aprendió a reconocer los vectores de sensores. Primero se escogieron aleatoriamente 100 vectores del conjunto de entrenamiento, posteriormente se calcularon 100 nuevos vectores de cada organismo con las características de los vectores de entrenamiento.

²Se aplicó un umbral, Si $|\text{esperado} - \text{obtenido}| < .5$ contaba como “bueno”

- iii. Para hacer las pruebas; una vez entrenada y validada la red, se obtuvieron los sensores de la especie que se quería analizar. Esto se hizo por medio de una ventana móvil de 200 bases de longitud que se desplaza de 30 en 30 bases.

La ventana móvil recorre toda la secuencia del genoma de un extremo al otro. Tiene 2 posibles variables; el tamaño de la ventana, esto es, el número de bases que formarán la subsecuencia y la longitud de desplazamiento o número de bases que se moverá para formar la siguiente subsecuencia. En cada instante en que se detiene y se obtiene una subsecuencia del genoma se calculan los sensores, que se irán guardando en un archivo, para que posteriormente la red neuronal previamente entrenada pruebe estos sensores. Si la longitud de desplazamiento es menor que el tamaño de la ventana, entonces la ventana se traslapa como se puede observar en la Figura (4.3).

Los genomas se obtuvieron de archivos tipo FASTA que se encuentran en la base de datos Genbank³. Se excluyeron las primeras y las últimas mil bases en cada genoma para evitar los efectos conocidos de sobrerepresentación de algún nucleótido.

³<http://www.ncbi.nlm.nih.gov/Entrez/>

Capítulo 5

Resultados

Las primeras pruebas se realizaron con organismos que están muy alejados filogenéticamente hablando, como el ratón (*Mus musculus*) y una bacteria (*Escherichia coli*). La red neuronal se entrenó con parte del genoma del ratón (el cromosoma 1) y con el genoma completo de *E. coli*; a los sensores (vectores de características) de la bacteria se les asignó el número 1 y a los sensores del ratón el número 0. Cabe señalar que ambas secuencias de DNA tienen aproximadamente el mismo número de bases (4639220 bases *E. coli* y 4171688 bases el cromosoma 1 del ratón).

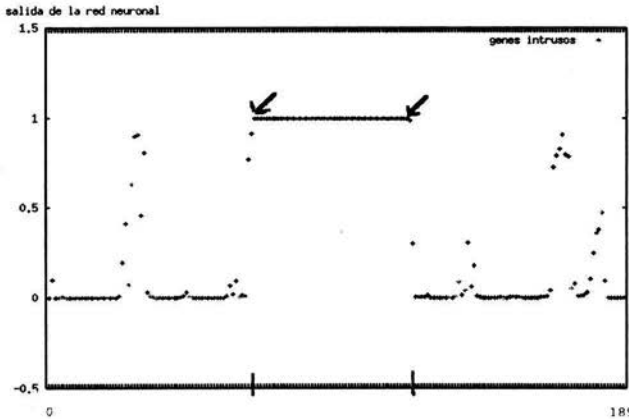


Figura 5.1: *Escherichia coli* → 1 vs. *Mus musculus* → 0.

Posteriormente se alteraron secuencias del ratón donde se incrustaban secuencias de DNA de *E. coli*, y se observaron excelentes resultados. Como se puede apreciar en la Figura (5.1), la gráfica es de una secuencia de 5670 bases aproximadamente¹ y una tercera parte de las bases es del DNA de *E. coli*, se observa claramente que el DNA artificialmente introducido se encuentra en la parte media de la secuencia.

Como se puede observar en la Figura (5.1) la detección del gen foráneo (la región que aparece señalada entre las flechas) es casi perfecta, llama la atención que sin proponérselo, se detectan brincos que pudieran tener alguna relevancia biológica.

A continuación, y siempre con la idea de tomar cada vez organismos más cercanos filogenéticamente para llegar a los límites de resolución de nuestro programa, tomamos a la bacteria *Pseudomonas aeruginosa* y a la arqueobacteria *Archaeoglobus fulgidus*.

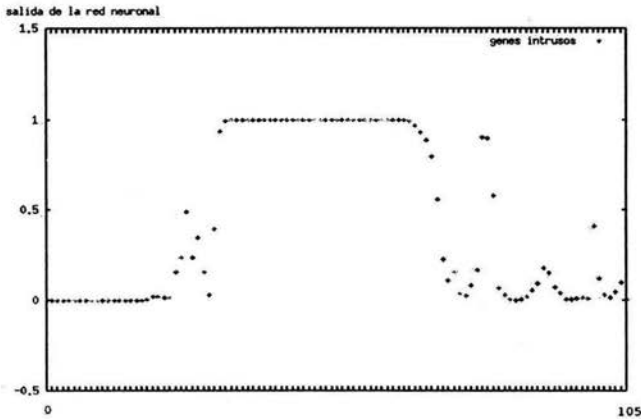


Figura 5.2: *Pseudomonas aeruginosa* → 1 vs. *Archaeoglobus fulgidus* → 0.

La red neuronal se entrenó con los genomas completos de los dos organismos (*Pseudomonas aeruginosa* 5996747 bases, *Archaeoglobus fulgidus* 2178400 bases), a las sensores de la bacteria se les asoció el número 1 y a los de la arqueobacteria el 0.

¹Hay que recordar que la ventana se desliza de 30 en 30 bases y que se obtuvieron 189 vectores de sensores.

De nueva cuenta se preparó una secuencia de DNA para el experimento. A una secuencia de *A. fulgidus* se le incrustó en la parte media alguna secuencia de DNA de *Pseudomonas aeruginosa*, la longitud de la secuencia que se preparó es de aproximadamente 3150 bases. Después de entrenada la red, se calcularon los sensores de esta secuencia por medio de la ventana móvil, para posteriormente pasárselos a la red, con lo que se obtuvieron 105 vectores de sensores.

En la Figura (5.2) se puede apreciar que una vez más se obtuvieron excelentes resultados, la región que aparece señalada entre las flechas corresponde al DNA que se incrusto perteneciente al genoma de *Pseudomonas aeruginosa*. No hay que olvidar que dado que la red neuronal ocupa la función sigmoideal como función de activación, los posibles resultados para cada vector de sensores pueden estar en el intervalo $[0, 1]$. Es por eso que puede haber puntos que no son ni 0 ni 1.

Para el siguiente ejemplo se tomaron en cuenta los eventos de transferencia horizontal reportados en la literatura, en algunos casos no se pudo llevar a cabo el experimento por no encontrarse en la actualidad totalmente secuenciados todos los genomas, tal es el caso de la bacteria *Pseudomonas mevalonii* [Boucher *et al.*, 2001]. Pero un evento de transferencia horizontal reportado, con el cual si se pudo trabajar fue el que se dio entre las bacterias *Neisseria meningitidis* y *Haemophilus ducreyi* [Kroll *et al.*, 1998]. Es importante notar que ambas son bacterias, organismos que están aún mas cercanos filogenéticamente hablando.

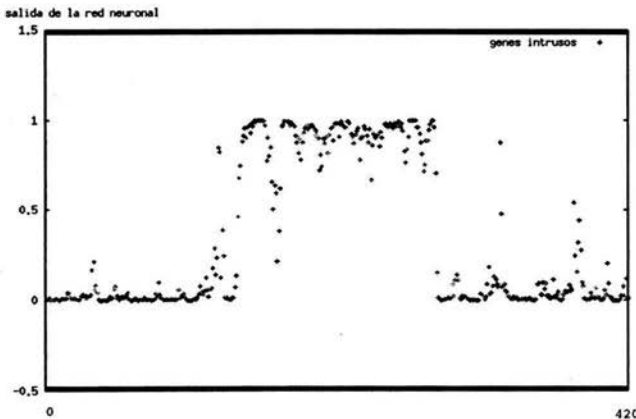


Figura 5.3: *Neisseria meningitidis* $\rightarrow 0$ vs. *Haemophilus ducreyi* $\rightarrow 1$.

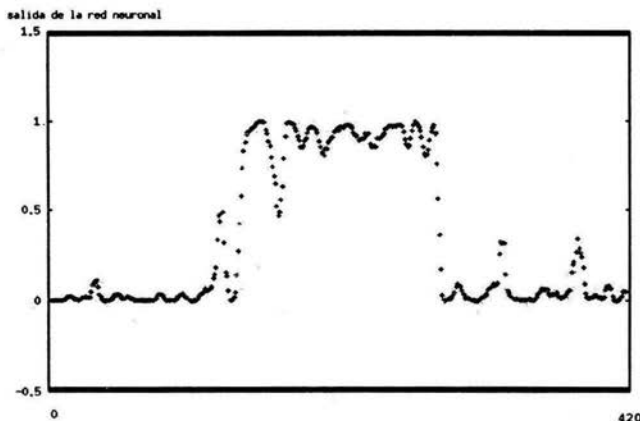


Figura 5.4: *Neisseria meningitidis* \rightarrow 0 vs. *Haemophilus ducreyi* \rightarrow 1. Ejemplo anterior con datos alisados.

En este caso la red neuronal también se entrenó con los genomas completos de los dos organismos, la bacteria *Neisseria meningitidis* cuenta con 2272351 bases, mientras que la bacteria *Haemophilus ducreyi* tiene 1699025 bases. A los vectores de *Neisseria meningitidis* se les asignó el número 0 como valor esperado y a los vectores de *Haemophilus ducreyi* el 1.

Antes de corroborar lo que el artículo reporta [Kroll *et al.*, 1998] (que el genoma de *Haemophilus* le transfirió genes al genoma de *Neisseria*), primero se verificó el desempeño de la red, es decir, que tan bien aprendió a reconocer los vectores de uno y otro organismo, con vectores que se utilizaron para el entrenamiento y posteriormente con nuevos vectores de los dos organismos. En todos los casos se obtuvieron buenos resultados. El éxito con vectores de entrenamiento fue en promedio de 91 %, con vectores nuevos de *Haemophilus* fue en promedio de 96 % y con vectores nuevos de *Neisseria* fue en promedio de 86 %. Cabe señalar que el desempeño con los sensores de *Neisseria* fue menor, éste puede ser un claro indicio de la transferencia horizontal.

Una vez verificado el desempeño, se creó un ejemplo, como en los casos anteriores, se tomó de la base 114381 a la base 122780 del genoma de *Neisseria* y justo en la parte media se le incrustó de la base 320251 a la base 324450 del genoma de *Haemophilus*. Con ésta secuencia se calcularon los 8 sensores por medio de la

ventana móvil, con lo que se obtuvieron 420 vectores de sensores. En la Figura (5.3) se puede apreciar la salida de la red neuronal para cada vector de sensores, aunque no se observa una meseta, claramente se puede ver cual es la región de DNA que se incrustó.

Con el fin de observar los resultados más claramente se volvió a graficar pero ahora recalculando los datos de salida de la red de la siguiente manera:

$$z_i = (x_{i-2} + x_{i-1} + x_i + x_{i+1} + x_{i+2}) \quad (5.1)$$

Como resultado se tuvo la gráfica que aparece en la Figura (5.4) que como se puede apreciar, los datos están alisados.

Ya que se observó que tanto a la hora de verificar con vectores, como con la secuencia preparada, se obtuvieron buenos resultados, se procedió a correr el programa de la ventana móvil a través de todo el genoma de la bacteria *Neisseria*. Se localizaron varias regiones en las cuales los valores de salida de la red fueran 1 o valores cercanos a 1.

En la Figura (5.5) se muestra una de las regiones antes mencionadas, que corresponde a la región comprendida entre los sensores calculados 6750–6950, es decir aproximadamente de la base 202500 a la base 208500.

Para poder apreciar claramente la región detectada se aplicó un umbral, es decir, a las salidas que fuesen $x \geq ,5$ se les asignó 1 y a las $x < ,5$ se les asignó 0. Esto se puede apreciar en la Figura (5.6).

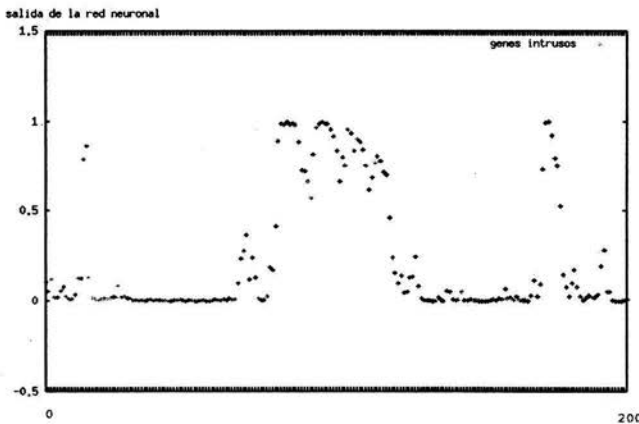


Figura 5.5: *Neisseria meningitidis* \rightarrow 0 vs. *Haemophilus ducreyi* \rightarrow 1. Ejemplo real.

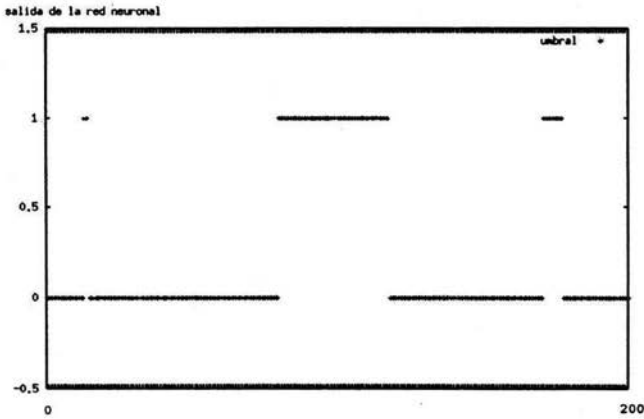


Figura 5.6: *Neisseria meningitidis* \rightarrow 0 vs. *Haemophilus ducreyi* \rightarrow 1. Ejemplo real con umbral.

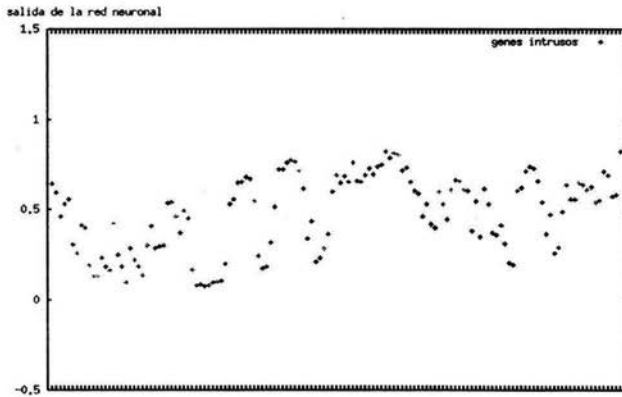


Figura 5.7: *Salmonella enterica* \rightarrow 0 vs. *Escherichia coli* \rightarrow 1.

Por último, lo que no es de sorprender por su semejanza genómica, se estudió a la pareja de genomas de *Escherichia coli* y *Salmonella enterica*² y no se obtuvieron buenos resultados. La red neuronal no logro aprender a reconocer a dichas bacterias, su bajo desempeño se vio reflejado a la hora de verificar con vectores del conjunto de entrenamiento, en promedio se obtenía 65% de efectividad, lo que no es desdeñable tomando en cuenta su semejanza estructural [Miramontes *et al.*, 1995].

En la Figura (5.7) se puede apreciar un ejemplo fabricado como los que aparecen al principio del capítulo, donde no se puede distinguir la región que se introdujo (en la parte media).

²subsp. *enterica* serovar *Typhistr.*

Capítulo 6

Conclusiones

La secuenciación de los genomas ha revelado una insospechada plasticidad genética entre las especies microbianas. En la actualidad se acepta que el intercambio horizontal de DNA puede ser la principal fuerza de innovación bioquímica y de adaptación y tolerancia a ambientes cambiantes. Los taxones básicos de la sistemática (especies, géneros, familias) no sirven para los procariotas, donde la transferencia horizontal de genes y los mecanismos de especiación son variados y complejos. Un ejemplo claro es la comparación de la información genética de dos conocidas cepas de *Escherichia coli*: la cepa K-12 habitual y la patógena O157:H7. El genoma de la primera tiene 4.6 Mb y el de la segunda 5.5 Mb. Comparten una secuencia de aproximadamente 4.1 Mb, que es equivalente a 3,574 proteínas codificadas. La cepa patógena codifica 1,387 nuevos genes, que han sido adquiridos por transferencia horizontal. Es decir, que consideramos dos cepas de una misma especie a una que tiene un 134% más de DNA que la otra.

Debido a la importancia evidente de la transferencia génica en la evolución, es importante poder detectar eventos de transferencia horizontal para poder reconstruir la historia evolutiva de los genes.

De la misma manera se puede recurrir a la herramienta descrita en este trabajo de tesis para detectar genes foráneos que hayan sido introducidos a algún genoma artificialmente (transgénicos, ver sección 3.2.3).

Este trabajo tiene una relevancia dual: por un lado se hace uso del conocimiento biológico para crear una herramienta útil para la detección de genes foráneos, y por otro lado se crea la red neuronal haciendo uso del conocimiento de las ciencias computacionales.

Por lo tanto, es importante señalar la importancia del trabajo interdisciplinario, pues tendiendo puentes entre las disciplinas científicas se podrán resolver muchos

problemas que aún permanecen sin resolver.

Con este trabajo se pudo comprobar la utilidad que tienen las redes neuronales en el reconocimiento de patrones, como ya se expuso en el capítulo 2, en particular el perceptrón multicapa cuyo uso se ha extendido y popularizado.

Los objetivos de este trabajo de investigación fueron cumplidos cabalmente (Ver página 9), ya que se pudo constatar el poder de las redes neuronales artificiales, en particular del perceptrón multicapa, que es un modelo de red neuronal adecuado para reconocer patrones. En la gran mayoría de los casos los entrenamientos de las redes fueron exitosos¹, por esta razón se puede decir que el conjunto de los ocho sensores (Ver página 47) contiene la información necesaria para categorizar al DNA de diversos organismos.

Es importante recordar que la detección de genes foráneos se llevó a cabo tomando en cuenta las características estructurales del DNA, esto es, no se toma en cuenta en ningún momento la funcionalidad de las regiones (se trabajó con todo el genoma sin importar si las regiones codificaban algo o no).

El posible trabajo futuro continuando con esta investigación incluye:

- Continuar utilizando la red neuronal para corroborar eventos de transferencia horizontal que hayan sido reportados en la literatura.
- Publicar los resultados exitosos de esta investigación.
- Buscar eventos de transferencia horizontal que no hayan sido reportados (pero que se tenga sospecha) y reportarlos.
- Añadir más sensores a los vectores de entrada para poder reconocer organismos muy parecidos estructuralmente².
- Comprobar que el conjunto total de sensores comprende el mínimo conjunto que caracteriza al DNA, es probable que el conjunto pueda ser restringido.

¹Excluyendo el caso de *Escherichia coli* y *Salmonella enterica*

²Ver capítulo anterior donde se nombra el caso de *Escherichia coli* y *Salmonella enterica*

Apéndice A

Los programas

Para llevar a cabo el objetivo de este trabajo, se requirieron de varios programas, un programa para calcular los sensores que alimentan a la red neuronal, un programa para implementar la red neuronal, otro para implementar la ventana móvil que calcula los sensores, y algunos pequeños programas para: “mochar” los archivos de tipo FASTA¹, intercalar los vectores de entrenamiento, medir el éxito del entrenamiento, etcétera. Los scripts se escribieron en el lenguaje de programación Perl, la red neuronal en C++ y lo referente a los sensores se programo en C.

Para calcular los sensores se escribió un programa en el lenguaje C, aunque dicho lenguaje de programación no es del agrado de quien esto escribe resulta imposible negar su eficacia en el manejo de bits, su velocidad de ejecución por ser un lenguaje compilado², entre muchas otras de sus bondades. Se emplearon campos de bits, esto para ahorrar espacio en la memoria, dado que sólo son cuatro las posibles bases (A, C, G y T) basta con 2 bits para almacenar cada base.

```
typedef struct
{
    unsigned int base:2;
    unsigned int WS:1;
    unsigned int YR:1;
    unsigned int MK:1;
}Base;
```

¹Estos archivos que contienen la secuenciación de los diferentes genomas se pueden obtener en: <http://www.ncbi.nlm.nih.gov/Entrez/>

²Sobre todo tratándose de cantidades de datos muy grandes como en este caso, los genomas que se analizaron tienen alrededor de seis millones de bases

De esta manera la secuencia de DNA se almacena en una lista de bases para poder recorrerla y así hacer los conteos necesarios. Cada struct Base contiene también tres variables (WS, YR, MK) para tener al mismo tiempo la traducción de la secuencia en las tres dicotomías; de igual manera se ocuparon bits para representarlas, ya que solo pueden tomar 2 valores (0, 1).

Aquí se presenta un ejemplo del uso de la lista de bases, la función subcadena que devuelve una subsecuencia de DNA.

```
Base *subcadena(Base *cadena, int inicio, int tam)
{ /* devuelve una subcadena de cadena que comienza en inicio de tamaño tam */

    int i;
    Base *temp,*sub = (Base *)malloc(sizeof(Base)*(tam+1));
    temp = sub;
    for(i=0;i<tam;i++,sub++){
        sub->base = cadena[inicio+i].base;
        sub->WS = cadena[inicio+i].WS;
        sub->YR = cadena[inicio+i].YR;
        sub->MK = cadena[inicio+i].MK;
    }
    return temp;
}
```

A continuación aparece un ejemplo de uno de los scripts escrito en Perl, cuya funcionalidad es eliminar las primeras 1000 bases y las últimas 1000 del archivo FASTA.

```
#!/usr/bin/perl -w

#mocha los primeros 1000 caracteres aprox de la secuencia de DNA
#y los ultimos 1000 aprox

my $archivo = $ARGV[0];

open(ARCHIVO, "$archivo");
open(SALIDA, "+>$archivo"."new");

while(<ARCHIVO>){
    if($. >= 15 and $_ !~ />gi/){
        print SALIDA $_;
    }
}
close(ARCHIVO);
```



```
seek SALIDA,0,0; #regresa al inicio
for(1..14){
  while ( <SALIDA> ) {
    $addr = tell(SALIDA) unless eof(SALIDA);
  }
  truncate(SALIDA, $addr)    or die "nose se puede trunca el archivo: $!";
  seek SALIDA,0,0; #regresa al inicio
}

close(SALIDA);
```

Bibliografía

- [Babloyantz & Ivanov] A. Babloyantz and V.V. Ivanov, "Neural Networks in Cardiac Arrhythmias," *Industrial Applications of Neural Networks* F.F. Soulié and P. Gallinari eds., World Scientific (1998), pp. 403–417.
- [Boucher *et al.*] Y. Boucher, H. Huber, S. I.'Haridon, K.O. Stetter, and W.F. Doolittle, "Bacterial Origin for the Isoprenoid Biosynthesis Enzyme HMG-CoA Reductase of the Archaeal Orders Thermoplasmatales and Archaeoglobales," *Mol. Biol. Evol.*, Vol. 18(7), (2001), pp. 1378–1388.
- [Cybenko] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, Vol. 2(4), (1989), pp. 303–314.
- [Funahashi] K.I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, Vol. 2(3), (1989), pp. 193–192.
- [Hartman *et al.*] E.J. Hartman, J.D. Keeler, and J.M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Computation*, Vol. 2(2), (1990), pp. 210–215.
- [Hebb] D.O. Hebb, *The Organization of Behavior*, Wiley, New York, N.Y., (1949).
- [Hornik *et al.*] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol. 2(5), (1989), pp. 359–366.
- [Hush & Horne] D.R. Hush and B.G. Horne, "Progress in supervised neural networks," *IEEE Signal Processing Magazine*, Vol. 10, (1993), pp. 8–39.
- [Kroll *et al.*] J.S. Kroll, K.E. Wilks, J.L. Farrant, and P.R. Langford, "Natural genetic exchange between *Haemophilus* and *Neisseria*: Intergeneric transfer of chromosomal genes between major human pathogens," *Proc. Natl. Acad. Sci.*, Vol. 95, (1998), pp. 12381–12385.

- [le Cun] Y. le Cun, "Une Procédure d'Apprentissage pour Réseau à Seuil Assymétrique," *Proc. Cognitiva 85: A la Frontière de l'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences*, CESTA, Paris, France, (1985), pp. 599-604.
- [Lippmann] R.P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE Acoustics, Speech, and Signal Processing Magazine*, Vol. 4, No.2, (1987), pp. 4-22.
- [McCulloch & Pitts] W.S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bull. Math. Biophysics*, Vol. 5, (1943), pp. 115-133.
- [Minsky & Papert] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, Mass., (1969).
- [Miramontes *et al.*] P. Miramontes, L. Medrano, C. Cerpa, R. Cedergren, G. Ferbeyre, and G. Cocho, "Structural and Thermodynamic Properties of DNA Uncover Different Evolutionary Histories," *J. Mol. Evol.*, Vol. 40, (1995), pp. 698-704.
- [Miramontes & Cocho] P. Miramontes and G. Cocho, "DNA dimer correlations reflect *in vivo* conditions and discriminate among nearest-neighbor base pair free energy parameter measures," *Physica A*, Vol. 321, (2003), pp. 577-586.
- [Muroga] S. Muroga, *Threshold Logic and Its Applications*, Wiley, New York, N.Y., (1971).
- [Naya *et al.*] H. Naya, H. Romero, A. Zavala, B. Alvarez, and H. Musto, "Aerobiosis Increases the Genomic Guanine Plus Cytosine Content (GC%) in Prokaryotes," *J. Mol. Evol.*, Vol. 55, (2002), pp. 260-264.
- [Parker] D.B. Parker, "Learning Logic," Tech. Report TR-47, Center for Computational Research in Economics and Management Science, MIT Cambridge, Mass., (1985).
- [Quintana *et al.*] J.R. Quintana, K. Grzeskowiak, K. Yanagi, and R.E. Dickerson, "Structure of a B-DNA Decamer with a Central T-A Step: C-G-A-T-T-A-A-T-C-G," *J. Mol. Biol.*, Vol. 225, (1992), pp. 379-395.

- [Rumelhart] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, eds., MIT Press, Cambridge, Mass. (1986), pp. 318-362.
- [Rosenblatt] F. Rosenblatt, *Principles of Neurodynamics*, Spartan Books, New York, N.Y., (1962).
- [Sejnowski & Rosemberg] T.J. Sejnowski and C.R. Rosemberg, "NETtalk: A Parallel Network that Learns to Read Aloud," (*Tech. Rep. Nos. JHU/EECS-86/01*). The John Hopkins University Electrical Engineering and Computer Science Department (1987).
- [Uberbacher & Mural] E.C. Uberbcher and R.J. Mural, "Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach," *Proc. Natl. Acad. Sci. USA*, Vol. 88, (1991), pp. 11261-11265.