

03063
6
24.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y SISTEMAS

Teoría de la Red Neuronal Canónica
tesis que en opción al grado de
Maestro en Ciencias de la Computación
presenta
Juan Raymundo Iglesias León

tutor
M. en I. B. B. Pedro Pablo González Pérez

México, D.F.

1997

**TESIS CON
FALLA DE ORIGEN**



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.

PAGINACION VARIA

COMPLETA LA INFORMACION

A mi padre ...

INDICE

Índice	V
Introducción	1
1. Forma Canónica de la Lógica	5
1.1 Definiciones básicas.....	5
1.2 Definición axiomática del álgebra booleana.....	9
1.3 Álgebra de predicados.....	10
1.4 Forma normal.....	15
1.5 Forma canónica.....	15
2. Redes neuronales biológicas	25
2.1 El sistema nervioso.....	25
2.2 La neurona biológica.....	26
2.3 Sinapsis.....	27
2.4 Potenciales.....	30
2.5 Sensitividad.....	33
3. Redes neuronales artificiales	37
3.1 Modelo de un neuroide.....	37
3.2 Modelo de una red neuronal artificial.....	40
3.3 Aprendizaje.....	44
3.4 Gesta de las redes neuronales artificiales.....	46
3.5 Redes neuronales artificiales en comparación a otras técnicas.....	48

3.6 Modelos clásicos.....	50
4. El neuroide lógico.....	53
4.1 Modelo.....	53
4.2 Propiedades con respecto a \vee	55
4.3 Propiedades con respecto a \wedge	59
4.4 Corolarios.....	63
5. La red neuronal canónica.....	67
5.1 Modelo.....	67
5.2 Propiedades.....	69
5.3 Aprendizaje.....	73
6. Red neuronal canónica vs. modelos clásicos.....	79
6.1 Neuroides.....	79
6.2 Modelo.....	81
6.3 Aprendizaje.....	83
6.4 Conversión a la red neuronal canónica.....	85
7. Conclusiones.....	89
Bibliografía.....	91
Ilustraciones.....	95
Tablas.....	97
Teoremas.....	99
Lemas.....	101
Corolarios.....	103

**" I suspect that a deeper mathematical study of nervous system
will affect our understanding
in fact it may alter the way in wich we look at mathematics and logic proper."**

- John von Neumann -

INTRODUCCIÓN

Debido a que los seres inteligentes que conocemos hasta el momento realizan su trabajo intelectual a través de un sistema nervioso, y a que un sistema nervioso está conformado por neuronas interconectadas en redes, existen grandes esfuerzos de investigación alrededor a las redes neuronales biológicas. Las redes neuronales artificiales, también conocidas como modelos conexionistas o sistemas de procesamiento distribuido paralelo, son sistemas de procesamiento de información que simulan aspectos relevantes de la arquitectura y funcionamiento de las redes neuronales biológicas. Pueden distinguirse dos principales áreas de interés en torno a las redes neuronales artificiales, estas son la aplicativa teórica y la aplicativa práctica.

El área aplicativa teórica busca la simulación de las actividades de un sistema nervioso a partir de evidencias neurofisiológicas o neuroanatómicas para comprobar hipótesis acerca de los mecanismos subyacentes en las redes neuronales biológicas. Un ejemplo de esta área lo constituye la neuroetología computacional, disciplina que abarca el estudio y simulación de los mecanismos neurobiológicos de los animales inferiores, tales como insectos, batracios o moluscos.

El área aplicativa práctica busca la simulación de las actividades del sistema nervioso en general que permitan solucionar un problema real. Algunos ejemplos de esta área están en el tratamiento computacional para la visión y el habla, en la industria para el control automático de procesos y el diagnóstico de fallas, en las comunicaciones para la compactación de la información, y en la economía para la predicción de fluctuaciones en los precios de mercado.

Son varios los desarrollos de redes neuronales artificiales que han resultado sonados éxitos de laboratorio e incluso comerciales en algunos casos [McCord, Illingworth 1990]. Tales redes han sido basadas en modelos de redes neuronales artificiales generalmente aceptados a los que se ha denominado modelos clásicos. Sin embargo, al construir una red aplicativa práctica cuando se tiene un problema a resolver, o bien al construir

una red aplicativa teórica cuando no se dispone de evidencias neurofisiológicas o neuroanatómicas, el diseñador debe solucionar varias interrogantes de diseño encaminadas a encontrar la topología y funcionamiento adecuados de la red:

- ¿Cuántas capas neuronales deben de existir en la red?
- ¿Cuántos neuroides debe tener cada capa neuronal?
- ¿Qué patrón de conectividad debe establecerse entre los neuroides?
- ¿Cómo deben ser inicializados los parámetros de los neuroides para estabilizar la red?
- ¿Qué modelo de neuroide debe usarse en qué capa neural?
- ¿Qué valores debe asumir cada vector de pesos?
- ¿Qué valores debe asumir cada vector de entradas?
- ¿Que valor debe asumir cada valor umbral?
- ¿Qué función debe usarse para computar el potencial del neuroide?
- ¿Qué función de transferencia debe emplearse?
- ¿Qué algoritmo de entrenamiento se debe seguir?
- ¿Qué modificaciones deben realizarse en cada ciclo del algoritmo de entrenamiento?
- ¿Qué función de aprendizaje se debe aplicar?

Ninguno de los modelos clásicos existentes ha dado explicación formal a estas preguntas de diseño. Por lo tanto existen algunos problemas asociados a los modelos clásicos de redes neuronales artificiales que aún no han sido resueltos satisfactoriamente:

- No se ha especificado un algoritmo genérico para la extracción del conocimiento de la red.
- No se ha mostrado un método para justificar las respuestas de la red, tal y como sucede en los sistemas expertos¹.
- No se sabe cuántas y cuáles son las modificaciones exactas que deben realizarse en cada ciclo del algoritmo de entrenamiento para asegurar la convergencia de la red a la solución del problema.
- No se conoce cuantos ciclos requiere el algoritmo de entrenamiento en el mejor y peor de los casos para asegurar la convergencia de la red a la solución del problema.
- No se sabe qué puede aprender y qué no puede aprender la red.
- No se ha garantizado la convergencia de la red a la solución de un problema
- No se ha podido demostrar que la red siempre aprenderá a responder correctamente a cierta entrada.

La falta de una explicación formal a las preguntas de diseño así como los problemas asociados a los modelos clásicos son precisamente los causantes de algunas de las desventajas de las redes neuronales artificiales con respecto a otras técnicas de la inteligencia artificial:

- El número de intentos que deben hacerse antes de que una red aprenda puede ser significativamente grande por lo que el entrenamiento puede resultar difícil y tedioso.
- Los algoritmos de entrenamiento son creados intuitivamente al no existir un estándar con fundamentos teóricos sólidos y contundentes a seguir.

¹ Existen algunas propuestas de sistemas de justificación de respuestas para cierto tipo de redes neuronales artificiales [Gallant 1988].

- ➡ En ocasiones las respuestas que arrojan las redes neuronales artificiales no son precisas.
- ➡ No se pueden simular actividades cíclicas como lo puede ser el contar.

Si para las anteriores cuestiones de diseño no se han propuesto soluciones formales, ¿Cómo es que actualmente existen redes neuronales artificiales que resuelven problemas? La respuesta es sencilla, esto ha sido posible a través de mucho tiempo de experimentación basada en similitudes halladas en los circuitos neuronales de los sistemas nerviosos de los seres vivos, o bien en modelos dinámicos similares e incluso en ideas intuitivas o situaciones aleatorias y empíricas. Es por esto que el diseño de una red neuronal artificial ha sido descrito por autores reconocidos como un proceso "misterioso" y en base a un gran esfuerzo en ciclos intento-error [McCord, Illingworth 1990]. Si la red no responde como se desea entonces se intenta con una función de transferencia diferente, tal vez con otra regla de aprendizaje, quizá moviendo los umbrales de los neuroides.

En este trabajo se propone un modelo de red neuronal artificial llamado la red neuronal canónica, el cual, a través de demostraciones formales, brinda soluciones a las cuestiones de diseño anteriormente planteadas, eliminando de esta forma los problemas asociados a los modelos clásicos revisados en los párrafos previos.

En el capítulo 1 se introducen las definiciones básicas y notaciones usadas en los capítulos restantes. Se detallan además los conceptos teóricos lógico-matemáticos bajo los cuales se sustenta el modelo de red neuronal canónica.

Las bases biológicas sobre las que están fundamentados los modelos de redes neuronales artificiales en general y en particular el modelo de la red neuronal canónica son explicadas en el capítulo 2. Aquí también se describen los principales componentes del sistema nervioso, los tipos de interconexiones que se suscitan entre las neuronas, así como los aspectos más relevantes de la anatomía y fisiología neuronal

En el capítulo 3 se presentan los términos comúnmente empleados en el área de las redes neuronales artificiales. Se describen los componentes del modelo de un neuroide y de una red neuronal artificial. Se revisan los métodos de aprendizaje más usados y se finaliza con una breve reseña histórica de las redes neuronales artificiales y una descripción somera de los modelos clásicos más relevantes.

Con el capítulo 4 se inicia propiamente la descripción del modelo propuesto, pues se define el modelo de neuroide que es usado en la red neuronal canónica. Las propiedades de comportamiento del neuroide lógico son formalmente demostradas con lo que los fundamentos teóricos de la red neuronal canónica quedan de esta forma bien cimentados.

El modelo de la red neuronal canónica es explicado detenidamente en el capítulo 5. Aquí se detallan las características del modelo en general y se propone un caso específico llamado la red neuronal canónica universal, a través de la cual se demuestran formalmente las propiedades tanto del modelo como de su forma de aprendizaje.

Finalmente, el capítulo 6 es un compendio de las principales ventajas y desventajas del modelo propuesto con respecto a los modelos clásicos. Son analizadas las características de ambos enfoques en cuanto a los neuroideos usados, las propiedades de los modelos de redes neuronales artificiales y la forma en que se realiza el aprendizaje. El capítulo termina con un conjunto de recomendaciones acerca de cómo los modelos clásicos pudieran enriquecerse con los resultados de la red neuronal canónica.

1. FORMA CANÓNICA DE LA LÓGICA

1.1 DEFINICIONES BÁSICAS

Un **conjunto** es una colección de elementos con alguna propiedad en común. Un conjunto de una cantidad numerable de elementos se denota de la forma $A = \{a_1, a_2, a_3\}$, lo que significa que a_1 , a_2 y a_3 están en el conjunto A . La cardinalidad de un conjunto A , denotada como $|A|$, es el número de elementos que están en A . El **universo de discurso**, denotado como U , es un conjunto en el que están todos los elementos usados para referirse a la situación de un contexto específico. El **conjunto de valores de verdad**, denotado como \mathcal{D} , es un conjunto formado por los elementos 0 y 1 que se interpretarán en adelante como "falso" y "verdadero" respectivamente, o bien "no" y "sí", o bien "cero" y "uno". Una **variable** x definida en un conjunto A es un símbolo que puede representar a cualquier elemento de A . En lo siguiente los símbolos i, j, m, n serán usados también frecuentemente para denotar variables. En adelante una notación del tipo $x_m, \dots, x_i, \dots, x_n$ se referirá a un conjunto de $n - m + 1$ variables diferentes, para $n \geq i \geq m$ a menos que se especifique lo contrario. Un **símbolo de constante** a definido en un conjunto A es un símbolo que representa a un y solo a un elemento de A . En lo siguiente los símbolos b y c serán usados también frecuentemente para denotar símbolos de constante. En adelante una notación del tipo $a_m, \dots, a_i, \dots, a_n$ se referirá a un conjunto de $n - m + 1$ símbolos de constante diferentes, para $n \geq i \geq m$ a menos que se especifique lo contrario.

Una **n -ada de elementos** denotada como (x_1, \dots, x_n) es un conjunto en el que existe un orden entre los elementos, de manera que pueden asignarse de manera única y sucesiva del primer número ordinal hasta el n -ésimo número ordinal a los elementos x_1 a x_n , respectivamente. En otros contextos a una n -ada se le conoce también con el nombre de **vector**. Si $V = (x_1, \dots, x_i, \dots, x_n)$ es un vector o n -ada entonces $V_i = x_i$ y a cada V_i se le llama **escalar**. Una **relación** es un conjunto cuyos elementos son n -adas. El **orden de una relación** R , denotado como R^n , es n donde n es el total de elementos de una n -ada de R ; se dice entonces que R es una

relación n -aria. El i -ésimo dominio de una relación R con n -adas $(x_1, \dots, x_i, \dots, x_j, \dots, x_n)$ para $n \geq j > i \geq 1$, es el conjunto formado por todos los elementos x_i de la misma forma, el j -ésimo contradominio de R es el conjunto formado por todos los elementos x_j . Una función f de A_1 en A_2 , denotada como $f: A_1 \rightarrow A_2$, es una relación en la que a cada elemento del dominio A_1 le puede corresponder no más de un elemento del contradominio A_2 . Una función lógica es una función cuyo contradominio es \mathcal{O} .

Un operador n -ario es un símbolo que representa a una función cuyo dominio es una n -ada de elementos (x_1, \dots, x_n) . Un operador lógico es un operador cuyo contradominio es \mathcal{O} . Los siguientes son operadores lógicos binarios usados frecuentemente para cualesquiera dos conjuntos A_1 y A_2 :

- $x \in A_1$ es 1 si el elemento x está en A_1 , de otra forma es 0.
- $x \notin A_1$ es 1 si el elemento x no está en A_1 , de otra forma es 0. Formalmente, $x_1 \notin A_1$ si y solo si para todo x_2 , si $x_2 \in A_1$ entonces $x_1 \neq x_2$.
- $A_1 \subseteq A_2$ es 1 si A_1 es subconjunto de A_2 , de otra forma es 0. Formalmente, $A_1 \subseteq A_2$ si y solo si para todo x , si $x \in A_1$ entonces $x \in A_2$.
- $A_1 \not\subseteq A_2$ es 1 si A_1 no es subconjunto de A_2 , de otra forma es 0.
- $A_1 = A_2$ es 1 si A_1 es igual a A_2 , de otra forma es 0. Formalmente, $A_1 = A_2$ si y solo si $A_1 \subseteq A_2$ y $A_2 \subseteq A_1$.
- $A_1 \neq A_2$ es 1 si A_1 no es igual a A_2 , de otra forma es 0.
- $A_1 \subset A_2$ es 1 si A_1 es subconjunto propio de A_2 , de otra forma es 0. Formalmente, $A_1 \subset A_2$ si y solo si $A_1 \subseteq A_2$ y $A_2 \not\subseteq A_1$.
- $A_1 \not\subset A_2$ es 1 si A_1 no es subconjunto propio de A_2 , de otra forma es 0.

El operador de conjuntos \cup , denominado unión, se define como: para todo x , $x \in A_1 \cup A_2$ si y solo si $x \in A_1$ o $x \in A_2$. El operador de conjuntos \cap , denominado intersección, se define como: para todo x , $x \in A_1 \cap A_2$ si y solo si $x \in A_1$ y $x \in A_2$. El operador de conjuntos $-$, denominado diferencia, se define como: para todo x , $x \in A_1 - A_2$ si y solo si $x \in A_1$ y $x \notin A_2$. El conjunto vacío, denotado como \emptyset , es un conjunto en el que no existe elemento alguno; formalmente, para todo x , $x \notin \emptyset$. Si $f: A_1 \rightarrow A_2$ es definida como $f = \{(x_1, y_1), \dots, (x_n, y_n)\}$ entonces $f(x_i) = y_i$; es evidente entonces que $x_i \in A_1$ y $f(x_i) \in A_2$.

El producto cartesiano de los conjuntos A_1, \dots, A_n , denotado como $A_1 \times \dots \times A_n$ o $\prod_{i=1}^n A_i$ se define como: $(x_1, \dots, x_n) \in \prod_{i=1}^n A_i$ si y solo si $x_i \in A_i$. A cada n -ada $(x_1, \dots, x_i, \dots, x_n) \in \prod_{i=1}^n A_i$ se le denomina permutación. Observe que todas las permutaciones de un $\prod_{i=1}^n A_i$ representan todos los arreglos posibles en un orden definido de los elementos de los conjuntos A_i . La cardinalidad de $\prod_{i=1}^n A_i$ puede conocerse multiplicando sucesivamente las cardinalidades de los conjuntos A_i que conforman las n -adas $(x_1, \dots, x_i, \dots, x_n) \in \prod_{i=1}^n A_i$, esto es expuesto formalmente en el Teorema 1.1.

Teorema 1.1 Principio permutatorio. $|\prod_{i=1}^n A_i| = \prod_{i=1}^n |A_i|$

Demostración. Por inducción sobre n .

1. si $n=1$ entonces $|\prod_{i=1}^1 A_i| = \prod_{i=1}^1 |A_i|$:

1.1. $n=1$

1.2. $(x_1, \dots, x_1, \dots, x_n) \in \prod_{i=1}^1 A_i$ si y solo si $x_1 \in A_1$.

1.3. $(x_1) \in \prod_{i=1}^1 A_i$ si y solo si $x_1 \in A_1$

1.4. $|\prod_{i=1}^1 A_i| = |A_1|$

1.5. $|\prod_{i=1}^1 A_i| = \prod_{i=1}^1 |A_i|$

1.6. $|\prod_{i=1}^1 A_i| = \prod_{i=1}^1 |A_i|$

2. si $n=j+1$ entonces $|\prod_{i=1}^{j+1} A_i| = \prod_{i=1}^{j+1} |A_i|$:

2.1. $n=j+1$

2.2. $|\prod_{i=1}^{j+1} A_i| = \prod_{i=1}^{j+1} |A_i|$

2.3. $(x_1, \dots, x_n, \dots, x_n) \in \prod_{i=1}^{j+1} A_i$ si y solo si $x_i \in A_i$.

2.4. $(x_1, \dots, x_n, \dots, x_j, x_{j+1}) \in \prod_{i=1}^{j+1} A_i$ si y solo si $x_i \in A_i$.

2.5. $|\prod_{i=1}^{j+1} A_i| = |\prod_{i=1}^j A_i| \times |A_{j+1}|$

2.6. $|\prod_{i=1}^{j+1} A_i| = \prod_{i=1}^j |A_i| \times |A_{j+1}|$

2.7. $|\prod_{i=1}^{j+1} A_i| = \prod_{i=1}^{j+1} |A_i|$

2.8. $|\prod_{i=1}^{j+1} A_i| = \prod_{i=1}^{j+1} |A_i|$

■

Una clase es un conjunto cuyos elementos son conjuntos. La clase potencia del conjunto A denotada como 2^A es una clase definida como: $x \in 2^A$ si y solo si $x \subseteq A$. A cada $x \in 2^A$ se le denomina combinación. Observe que todas las combinaciones de 2^A representan todos los arreglos posibles de los elementos de un conjunto A sin importar el orden de los elementos. De la definición de 2^A se infiere que 2^A puede crearse recursivamente mediante los siguientes axiomas de construcción:

1. si $A = \emptyset$ entonces $2^A = \{\emptyset\}$

2. si $A = B \cup \{x\}$ y $x \notin B$ entonces $2^A = \{S_1, \dots, S_n, S_1 \cup \{x\}, \dots, S_n \cup \{x\}\}$, para $2^B = \{S_1, \dots, S_n\}$

La cardinalidad de 2^A puede conocerse multiplicando 2 por sí mismo tantas veces como elementos existan en A , esto mismo se presenta formalmente en el Teorema 1.2.

Hipótesis

Definición de Π

De (1.1) substituyendo en (1.2)

De (1.3) pues existe una sola n -ada por cada elemento de A_1

De (1.4) por paso algebraico

De (1.1) substituyendo en (1.5)

Hipótesis

Hipótesis inductiva

Definición de Π

De (2.1) substituyendo (2.3)

De (2.4) pues existe una misma $x_1, \dots, x_i, \dots, x_j$ para cada x_{j+1}

De (2.2) substituyendo en (2.5)

De (2.6) por paso algebraico

De (2.1) substituyendo en (2.7)

Teorema 1.2 Principio Combinatorio. $|2^A| = 2^{|A|}$ **Demostración.** Por inducción sobre A :1. si $A = \emptyset$ entonces $|2^A| = 2^{|A|}$:1.1. $A = \emptyset$ 1.2. si $A = \emptyset$ entonces $2^A = \{\emptyset\}$ 1.3. $2^A = \{\emptyset\}$ 1.4. $|2^A| = 1$ 1.5. $|A| = 0$ 1.6. $2^{|A|} = 1$ 1.7. $|2^A| = 2^{|A|}$ 2. si $A = B \cup \{x\}$ y $x \in B$ entonces $|2^A| = 2^{|A|}$:2.1. $A = B \cup \{x\}$ y $x \in B$ 2.2. $|2^B| = 2^{|B|}$ 2.3. $2^A = \{S_1, \dots, S_n, S_1 \cup \{x\}, \dots, S_n \cup \{x\}\}$, para $2^B = \{S_1, \dots, S_n\}$ 2.4. $|2^A| = 2 \times |2^B|$ 2.5. $|2^A| = 2 \times 2^{|B|}$ 2.6. $|2^A| = 2^{|B|+1}$ 2.7. $|B| + 1 = |A|$ 2.8. $|2^A| = 2^{|A|}$

Hipótesis

De (1.1) por axioma 1 de construcción de 2^A

De (1.1) por (1.2)

De (1.3) por definición de $||$ De (1.1) por definición de $||$

De (1.5)

De (1.4) y (1.6)

Hipótesis

Hipótesis inductiva

De (2.1) por axioma 2 de construcción de 2^A De (2.3) pues para cada $S_i \in 2^B$ existen $S_i \in 2^A$ y $S_i \cup \{x\} \in 2^A$ para $n \geq 1$

De (2.2) substituyendo en (2.4)

De (2.5) por paso algebraico

De (2.1) por definición de $||$

De (2.7) substituyendo en (2.6)

Un **sistema algebraico** está formado por un conjunto de valores, un conjunto de operadores y un conjunto de postulados. Los **postulados** de un álgebra son axiomas que forman los supuestos básicos mediante los cuales es posible deducir los teoremas de tal álgebra. Los postulados más comunes usados para formular los sistemas algebraicos son:

- **Cierre.** Un conjunto A está cerrado con respecto a un operador binario $*$ si y solo si, para todo x, y , si $x, y \in A$ entonces $x * y = z$ y $z \in A$.
- **Asociatividad.** Un operador binario $*$ cerrado en el conjunto A es asociativo si y solo si, para todo x, y, z si $x, y, z \in A$ entonces $(x * y) * z = x * (y * z)$.
- **Conmutatividad.** Un operador binario $*$ cerrado en el conjunto A es conmutativo si y solo si, para todo x, y si $x, y \in A$ entonces $x * y = y * x$.
- **Valor identidad.** Un elemento $a \in A$ es un valor identidad con respecto a un operador binario $*$ cerrado en A si y solo si, para todo x , si $x \in A$ entonces $a * x = x * a = x$
- **Inversa.** Un conjunto A que tiene el valor identidad a con respecto al operador binario $*$ cerrado en A , se dice que tiene una inversa si y solo si, para todo x , si $x \in A$ entonces existe $z \in A$ tal que $x * z = a$.
- **Distributividad.** Si $*$ y $\#$ son dos operadores binarios cerrados en A , se dice que $*$ es distributivo sobre $\#$ si y solo si, para todo x, y, z , si $x, y, z \in A$ entonces $x * (y \# z) = (x * y) \# (x * z)$

1.2 DEFINICIÓN AXIOMÁTICA DEL ÁLGEBRA BOOLEANA

El álgebra booleana fue introducida por George Boole [Boole 1854] como un tratamiento sistemático de la lógica mediante un sistema algebraico. Posteriormente, E. V. Huntington estableció los primeros postulados para la definición formal de la lógica booleana [Huntington 1904], junto a los que se han usado otros conjuntos de postulados [Birkhoff, Bartee 1970].

Las aplicaciones del álgebra booleana han sido muchas y con muy importantes repercusiones en nuestros días. Casi un siglo después de Boole, C. E. Shannon demostró que las propiedades de los circuitos estables con interruptores pueden representar al álgebra booleana [Shannon 1938]. Shannon le llamó a su descubrimiento el álgebra de interruptores, que es la precursora de la electrónica digital y por lo tanto de la tecnología usada en el hardware de las computadoras en nuestros días. Además de la lógica y la electrónica digital, el álgebra booleana ha sido también empleada intensivamente en la teoría de conjuntos y la teoría de la probabilidad entre otras áreas.

El álgebra booleana es un sistema algebraico definido en un conjunto de valores β con los operadores binarios $\#$, $*$ y el operador unario $@$, donde se satisfacen los postulados de Huntington:

1. a. Cierre con respecto al operador $\#$
b. Cierre con respecto al operador $*$
2. a. Un valor identidad con respecto a $\#$, designado por a : $x\#a = a\#x = x$
b. Un valor identidad con respecto a $*$, designado por b : $x*b = b*x = x$
3. a. Conmutativo con respecto a $\#$: $y\#x = x\#y$
b. Conmutativo con respecto a $*$: $y*x = x*y$
4. a. Distributivo con respecto a $*$ sobre $\#$: $x*(y\#z) = (x*y)\#(x*z)$
b. Distributivo con respecto a $\#$ sobre $*$: $x\#(y*z) = (x\#y)*(x\#z)$
5. para todo x , si $x \in \beta$ entonces existe $@x \in \beta$ tal que:
 - a. $x\#@x = b$
 - b. $x*\@x = a$
6. Existen cuando menos dos valores $x, y \in \beta$ tales que $x \neq y$

Observe que los postulados de Huntington no incluyen el postulado de asociatividad, sin embargo, la asociatividad es válida para el álgebra booleana y se deriva tanto para el operador $\#$ como el $*$ precisamente de los postulados de Huntington. En la Tabla 1.1 se muestran los postulados de Huntington junto con los teoremas básicos derivados del álgebra booleana, mismos que han sido ampliamente estudiados y cuya demostración es bastante asequible [Birkhoff, Bartee 1970] [Birkhoff, Maclane 1965] [Hohn 1966].

	(a) $x \# a = x$	(b) $x^a b = x$
Postulado 2, valor identidad	(a) $x \# y = y \# x$	(b) $x^a y = y^a x$
Postulado 3, conmutativo	(a) $x^a (y \# z) = (x^a y) \# (x^a z)$	(b) $x \# (y^a z) = (x \# y)^a (x \# z)$
Postulado 4, distributivo	(a) $x \# (@x) = b$	(b) $x^a @x = a$
Postulado 5,	(a) $x \# x = x$	(b) $x^a x = x$
Teorema 1	(a) $x \# b = b$	(b) $x^a a = a$
Teorema 2	$@(@x) = x$	
Teorema 3, involución	(a) $x \# (y \# z) = (x \# y) \# z$	(b) $x^a (y^a z) = (x^a y)^a z$
Teorema 4, asociativo	(a) $@(x \# y) = (@x) \# (@y)$	(b) $@(x^a y) = (@x) \# (@y)$
Teorema 5, de De Morgan	(a) $x \# (x^a y) = x$	(b) $x^a (x \# y) = x$
Teorema 6, absorción		

Tabla 1.1 Resumen de postulados y teoremas básicos del álgebra booleana

Existe una importante propiedad del álgebra booleana llamada el principio de dualidad, el cual establece que una igualdad entre dos expresiones booleanas $e_1 = e_2$ permanece válida en $e_1' = e_2'$ si se obtiene siguiendo las siguientes reglas:

1. se intercambian de manera simultánea los operadores $\#$ por $*$ y los operadores $*$ por $\#$
2. se intercambian de manera simultánea los valores identidad a por b y los valores identidad b por a

El dual $e_1' = e_2'$ de $e_1 = e_2$ es una igualdad formada a partir de $e_1 = e_2$ a través de las dos reglas anteriores del principio de dualidad. Debido al principio de dualidad los incisos b de los postulados y teoremas de la Tabla 1.1 pueden ser obtenidos de los incisos a correspondientes y viceversa, en otras palabras, estos incisos son duales uno del otro.

1.3 ÁLGEBRA DE PREDICADOS

El álgebra booleana trabaja con un conjunto β de valores no especificado. Por ello, para definir un álgebra booleana se deben especificar los elementos del conjunto β así como la definición de la función de cada operador. Finalmente, una vez especificados los elementos y operadores debe demostrarse que se satisfacen los seis postulados de Huntington. Existen numerosos trabajos con propuestas de álgebras booleanas [Hohn 1966] [Whitesitt 1961] [Birkoff, Bartee 1970]. Entre las álgebras booleanas destaca el álgebra de predicados como la formalización del tratamiento sistemático a la lógica mediante un sistema algebraico introducido por Boole [Boole 1854].

El álgebra de predicados es un álgebra booleana definida en un conjunto de elementos $\beta = \Theta$ en la que existen dos operadores binarios \vee , \wedge y un operador unario \neg para el cumplimiento del postulado 5 de Huntington. Los operadores \vee , \wedge y \neg se definen como:

- $1 \vee 1 = 1$; $1 \vee 0 = 1$; $0 \vee 1 = 1$; $0 \vee 0 = 0$
- $1 \wedge 1 = 1$; $1 \wedge 0 = 0$; $0 \wedge 1 = 0$; $0 \wedge 0 = 0$
- $\neg 1 = 0$; $\neg 0 = 1$

Los operadores \vee y \wedge son llamados disyunción y conjunción respectivamente y se leen "o" y "y" en el mismo orden. El operador \neg es llamado negación o complemento y se lee "no" o "no es cierto que ...". En adelante

al hablar de un álgebra booleana se asumirá que se está haciendo referencia al álgebra de predicados y que por lo tanto $\beta = \oplus$.

Para el álgebra de predicados una variable booleana es una variable definida en \oplus . Un símbolo de constante booleano en el álgebra de predicados es un símbolo de constante definido en \oplus . Una expresión booleana dentro del álgebra de predicados se define como:

1. un valor de \oplus ya sea 1 o 0, o
2. una variable booleana, o
3. si e es una expresión booleana entonces $(\neg e)$ es una expresión booleana, o
4. si e_1 y e_2 son expresiones booleanas entonces $(e_1 \vee e_2)$ y $(e_1 \wedge e_2)$ son expresiones booleanas.

La interpretación de una expresión booleana en la que ocurren las variables booleanas x_1, \dots, x_n , denotada como ψ , es una función $\psi: \{x_1, \dots, x_n\} \rightarrow \oplus$. En otras palabras, una ψ asigna a cada variable booleana x un valor de \oplus ya sea 1 o 0. Es evidente por su definición que $\psi = \{(x_1, a_1), \dots, (x_n, a_n)\}$ donde cada x_i es una variable booleana y cada a_i es un símbolo de constante booleana. En adelante, al conjunto de todas las interpretaciones de una expresión booleana en las que ocurren las variables booleanas x_1, \dots, x_n se denotará como Ψ_{x_1, \dots, x_n} , formalmente, $(a_1, \dots, a_n) \in \prod_{i=1}^n \oplus$ si y solo si $\{(x_1, a_1), \dots, (x_n, a_n)\} \in \Psi_{x_1, \dots, x_n}$. De su definición se desprende que $|\Psi_{x_1, \dots, x_n}|$ es igual a 2^n donde n es el número de variables booleanas que intervienen en una expresión, en otras palabras, dada una expresión con n variables booleanas diferentes existen exactamente 2^n interpretaciones diferentes para tal expresión; esto mismo es enunciado formalmente en el Teorema 1.3

Teorema 1.3 $|\Psi_{x_1, \dots, x_n}| = 2^n$

Demostración.

1. $(a_1, \dots, a_n) \in \prod_{i=1}^n \oplus$ si y solo si $\{(x_1, a_1), \dots, (x_n, a_n)\} \in \Psi_{x_1, \dots, x_n}$

Definición de Ψ_{x_1, \dots, x_n}

2. $|\Psi_{x_1, \dots, x_n}| = |\prod_{i=1}^n \oplus|$

De (1) pues existe una interpretación por cada permutación de $\prod_{i=1}^n \oplus$

3. $|\prod_{i=1}^n \oplus| = \prod_{i=1}^n |\oplus|$

Teorema 1.1

4. $|\prod_{i=1}^n \oplus| = \prod_{i=1}^n 2$

De (3) por definición de \oplus y $||$

5. $|\prod_{i=1}^n \oplus| = 2^n$

De (4) por paso algebraico

6. $|\Psi_{x_1, \dots, x_n}| = 2^n$

De (2) substituyendo en (5)

■

En la Tabla 1.2 se han enumerado las 2^3 interpretaciones ψ de tres variables, en adelante cada $\psi \in \Psi_{x_1, \dots, x_n}$ se enumerará como $1+y$, donde y es el número decimal equivalente al número binario formado por la permutación de los valores de \oplus que ψ asigna a x_1, \dots, x_n .

Denominación	x_1	x_2	x_3
ψ_1	0	0	0
ψ_2	0	0	1
ψ_3	0	1	0
ψ_4	0	1	1
ψ_5	1	0	0
ψ_6	1	0	1
ψ_7	1	1	0
ψ_8	1	1	1

Tabla 1.2 Interpretaciones de tres variables

La **substitución**, denotada como ψ/e , de la interpretación $\psi = \{(x_1, a_1), \dots, (x_i, a_i), \dots, (x_n, a_n)\}$ en la expresión e donde ocurren x_1, \dots, x_n es una expresión resultante de remplazar en e todas las variables x_i por los valores a_i que les corresponden según ψ . Dado que en el Teorema 1.3 se había demostrado ya que existen exactamente 2^n interpretaciones ψ diferentes, sabemos por lo tanto que existen 2^n formas de hacer una sustitución en una expresión; esta misma idea es plasmada en el Corolario 1.1.

Corolario 1.1 (del Teorema 1.3) Existen 2^n sustituciones para una misma expresión booleana e .

La **evaluación**, denotada como $\xi(e, \psi)$, de una expresión booleana e dada la interpretación ψ , es una función definida como:

1. si e tiene la forma $\neg e$ entonces $\xi(\neg e, \psi) = \neg \xi(e, \psi)$
2. si e tiene la forma $e_1 \vee e_2$ entonces $\xi(e_1 \vee e_2, \psi) = \xi(e_1, \psi) \vee \xi(e_2, \psi)$
3. si e tiene la forma $e_1 \wedge e_2$ entonces $\xi(e_1 \wedge e_2, \psi) = \xi(e_1, \psi) \wedge \xi(e_2, \psi)$
4. si e no tiene ninguna de las formas anteriores entonces $\xi(e, \psi) = \psi/e$

Es evidente que por su definición el contradominio de ξ es Θ . Se dice que dos expresiones booleanas e_1 y e_2 son iguales, denotando $e_1 = e_2$, si y solo si para todo i , $\xi(e_1, \psi_i) = \xi(e_2, \psi_i)$. En lo siguiente se asumirá que la precedencia entre los operadores al evaluar una expresión es $(1^\circ) \neg$, $(3^\circ) \wedge$, $(4^\circ) \vee$. La precedencia entre los operadores puede ser útil para eliminar paréntesis redundantes en una expresión booleana.

Sabemos por el Teorema 1.3 que existen exactamente 2^n interpretaciones distintas, además, el Corolario 1.1 afirma que por lo tanto que hay 2^n sustituciones distintas, es evidente entonces que también existirán exactamente 2^n evaluaciones distintas para una expresión. Lo anterior en este párrafo es remarcado en el Corolario 1.2.

Corolario 1.2 (del Teorema 1.3 y Corolario 1.1) Existen 2^n evaluaciones para una misma expresión booleana e .

La **tabla de verdad** de una expresión booleana e es una tabla en la que se indican las 2^n evaluaciones de e . La Tabla 1.3 es la tabla de verdad para las expresiones booleanas más simples que pueden formarse a partir de dos variables booleanas x y y con los operadores \wedge , \vee y \neg . Observe en la Tabla 1.3 que los renglones de las

columnas para x y y denotan las 2^n interpretaciones posibles de x y y , mientras que los renglones de las columnas de $x \vee y$, $x \wedge y$, $\neg x$ denotan las 2^n evaluaciones de las expresiones correspondientes.

x	y	$x \vee y$	$x \wedge y$	$\neg x$	$\neg y$
0	0	0	0	1	1
0	1	1	0	1	0
1	0	1	0	0	1
1	1	1	1	0	0

Tabla 1.3 Evaluaciones para expresiones de dos variables booleanas con los operadores \vee , \wedge y \neg .

A continuación se demuestra que el álgebra de predicados satisface los postulados de Huntington:

1. a. Cierre con respecto al operador \vee
- b. Cierre con respecto al operador \wedge

Basta con mirar la Tabla 1.3 para que sea evidente el cumplimiento del cierre para 1.a y 1.b

2. a. Un elemento identidad con respecto a \vee designado por 0: $x \vee 0 = 0 \vee x = x$

A partir de la Tabla 1.3 se observa que $0 \vee 0 = 0$ y $0 \vee 1 = 1 \vee 0 = 1$

- b. Un elemento identidad con respecto a \wedge designado por 1: $x \wedge 1 = 1 \wedge x = x$

A partir de la Tabla 1.3 se observa que $1 \wedge 1 = 1$ y $0 \wedge 1 = 1 \wedge 0 = 0$

3. a. Conmutativo con respecto a \vee : $x \vee y = y \vee x$
- b. Conmutativo con respecto a \wedge : $x \wedge y = y \wedge x$

Basta con mirar la simetría de la Tabla 1.3 en los operadores \vee y \wedge para que sea evidente el cumplimiento de la conmutatividad para 3.a y 3.b

4. a. Distributivo con respecto a \vee sobre \wedge : $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

En la Tabla 1.4 se muestra la validez de la equivalencia anterior a través de una tabla de verdad

- b. Distributivo con respecto a \wedge sobre \vee : $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

En la Tabla 1.5 se muestra la validez de la equivalencia anterior a través de una tabla de verdad

5. para todo x , si $x \in \beta$ entonces existe $\neg x \in \beta$ tal que

a. $x \vee \neg x = 1$

A partir de la Tabla 1.3 se observa que $0 \vee \neg 0 = 0 \vee 1 = 1$ y $1 \vee \neg 1 = 1 \vee 0 = 1$

b. $x \wedge \neg x = 0$

A partir de la Tabla 1.3 se observa que $0 \wedge \neg 0 = 0 \wedge 1 = 0$ y $1 \wedge \neg 1 = 1 \wedge 0 = 0$

6. Existen cuando menos dos elementos $x, y \in \beta$ tales que $x \neq y$

Dichos elementos son precisamente 1 y 0 con $1 \neq 0$

x	y	z	$x \wedge z$	$x \vee (\neg y \wedge z)$	$x \wedge y$	$x \vee z$	$(x \vee y) \wedge (x \vee z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	1	0	0	0
0	1	1	0	1	0	1	0
1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	0
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Tabla 1.4 Cumplimiento del postulado 4.a de Huntington

x	y	z	$y \vee z$	$\neg(y \vee z)$	$x \wedge y$	$\neg x \vee z$	$(x \wedge y) \vee (\neg x \vee z)$
0	0	0	0	1	0	1	0
0	0	1	1	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	1	0	0	1	0
1	0	0	0	1	0	0	0
1	0	1	1	0	0	1	0
1	1	0	1	0	1	0	1
1	1	1	1	0	1	0	1

Tabla 1.5 Cumplimiento del postulado 4.b de Huntington

Habiendo demostrado ya que el álgebra de predicados es un álgebra booleana, se infiere entonces que el principio de dualidad y los teoremas básicos del álgebra booleana expuestos en la sección 1.2 son válidos también para el álgebra de predicados. En la Tabla 1.6 se exponen nuevamente los postulados y teoremas básicos del álgebra booleana ahora aplicados al álgebra de predicados.

Postulado 2, valor identidad	(a) $x \vee 0 = x$	(b) $x \wedge 1 = x$
Postulado 3, conmutativo	(a) $x \vee y = y \vee x$	(b) $x \wedge y = y \wedge x$
Postulado 4, distributivo	(a) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$	(b) $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
Postulado 5	(a) $x \vee \neg x = 1$	(b) $x \wedge \neg x = 0$
Teorema 1	(a) $x \vee x = x$	(b) $x \wedge x = x$
Teorema 2	(a) $x \vee 1 = 1$	(b) $x \wedge 0 = 0$
Teorema 3, involución	$\neg(\neg x) = x$	
Teorema 4, asociativo	(a) $x \vee (y \vee z) = (x \vee y) \vee z$	(b) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$
Teorema 5, de De Morgan	(a) $\neg(x \vee y) = (\neg x) \wedge (\neg y)$	(b) $\neg(x \wedge y) = (\neg x) \vee (\neg y)$
Teorema 6, absorción	(a) $x \vee (x \wedge y) = x$	(b) $x \wedge (x \vee y) = x$

Tabla 1.6 Resumen de postulados y teoremas básicos para el álgebra de predicados

1.4 FORMA NORMAL

Una función booleana es una función cuyo dominio es un conjunto de variables booleanas y cuyo contradominio es Θ . Observe que por su definición una función booleana es una función lógica. Una función booleana puede especificarse a través de una expresión booleana, por ejemplo, la notación $f(x,y,z) = (x \wedge y \vee x \wedge z) \wedge (\neg x \wedge \neg y \vee y \wedge z)$ especifica que la función booleana f es igual a 1 si $(x \wedge y \vee x \wedge z) \wedge (\neg x \wedge \neg y \vee y \wedge z)$ es igual a 1 y 0 de otra forma. Por consiguiente, dos funciones booleanas $f_1 = e_1$ y $f_2 = e_2$ son iguales si y solo si $e_1 = e_2$. El complemento o negación de una función booleana f es $\neg f$, la cual es 0 si f es igual a 1 y es igual a 1 si f es igual a 0.

Una literal, denotada como l_i , es una variable booleana x o su negación $\neg x$. En adelante l_i denotará una literal formada a partir del i -ésimo elemento de un conjunto de variables booleanas $x_1, \dots, x_i, \dots, x_n$. Así mismo, l_i' denotará a $\neg x$ si $y=0$ o bien a x si $y=1$. De igual manera, L_i denotará a un conjunto cuyos elementos son las posibles l_i de x_i . Es evidente que $L_i = \{x_i, \neg x_i\}$ y que por lo tanto $|L_i| = 2$.

En lo siguiente a la disyunción de todos los elementos de un conjunto A se le denotará como $\vee A$, mientras que la conjunción de todos los elementos de A se denotará como $\wedge A$. Además, la conjunción y disyunción de un conjunto cuyos elementos son x_1, \dots, x_n se denotará también como $\bigwedge_{i=1}^n x_i$ y $\bigvee_{i=1}^n x_i$ respectivamente.

Un término booleano disyuntivo o término disyuntivo es una expresión booleana formada por la disyunción de una o más literales. Un término booleano conjuntivo o término conjuntivo es una expresión booleana formada por la conjunción de una o más literales. Un término booleano o término es un término conjuntivo o disyuntivo.

Un polinomio en forma normal disyuntiva o polinomio disyuntivo se define como $\vee A$, donde A es un conjunto posiblemente vacío de términos conjuntivos. Un polinomio en forma normal conjuntiva o polinomio conjuntivo se define como $\wedge A$, donde A es un conjunto posiblemente vacío de términos disyuntivos. Un polinomio booleano en forma normal o polinomio es un polinomio conjuntivo o disyuntivo.

Observe que cualquier expresión que no se encuentre en forma normal puede convertirse a la forma normal y viceversa a través del postulado 4 distributivo de Huntington.

1.5 FORMA CANÓNICA

Un término booleano canónico conjuntivo o término canónico conjuntivo, denotado t , se define como $\bigwedge_{i=1}^n l_i$.

Un término booleano canónico disyuntivo o término canónico disyuntivo, denotado $\neg t$, se define como $\bigvee_{i=1}^n l_i$.

Un término booleano canónico o término canónico es un término canónico conjuntivo o disyuntivo. En adelante T denotará al conjunto de todos los términos canónicos conjuntivos t que pueden formarse del conjunto de variables booleanas x_1, \dots, x_n . De la misma forma $\neg T$ denotará al conjunto de todos los términos

canónicos disyuntivos $\neg t$ que pueden formarse del conjunto de variables booleanas x_1, \dots, x_n . Formalmente, $(l_1, \dots, l_n) \in \prod_{i=1}^n L_i$ si y solo si $l_1 \wedge \dots \wedge l_n \in T$, igualmente, $(l_1, \dots, l_n) \in \prod_{i=1}^n L_i$ si y solo si $l_1 \vee \dots \vee l_n \in \neg T$. Además, ζ denotará a un conjunto posiblemente vacío de términos canónicos conjuntivos, es decir, $\zeta \in T$. Finalmente, δ denotará a un conjunto posiblemente vacío de términos canónicos disyuntivos, es decir, $\delta \in \neg T$.

El número de términos canónicos conjuntivos distintos que pueden formarse con n variables es exactamente 2^n . Esto es debido a que cada literal l_i que compone un término canónico conjuntivo puede tener dos formas ya sea x_i o $\neg x_i$, de manera que si se calculan por el Teorema 1.1 las permutaciones posibles que pueden formarse de las literales se obtiene dos multiplicado por sí mismo n veces. De la misma forma se puede deducir que el número de términos canónicos disyuntivos distintos que pueden formarse con n variables es exactamente 2^n . Lo dicho anteriormente en este párrafo es formalmente expuesto en el Teorema 1.4 y el Corolario 1.3.

Teorema 1.4 $|T| = 2^n$

Demostración.

1. $(l_1, \dots, l_n) \in \prod_{i=1}^n L_i$ si y solo si $l_1 \wedge \dots \wedge l_n \in T$

Definición de T

2. $|T| = \left| \prod_{i=1}^n L_i \right|$

De (1) pues existe un término canónico conjuntivo por cada permutación de $\prod_{i=1}^n L_i$

3. $\left| \prod_{i=1}^n L_i \right| = \prod_{i=1}^n |L_i|$

Teorema 1.1

4. $\left| \prod_{i=1}^n L_i \right| = \prod_{i=1}^n 2$

De (3) por definición de $| \cdot |$ y L_i

5. $\left| \prod_{i=1}^n L_i \right| = 2^n$

De (4) por paso algebraico

6. $|T| = 2^n$

De (2) substituyendo en (5)

■

Corolario 1.3 (del Teorema 1.4) $|\neg T| = 2^n$

En realidad los 2^n términos canónicos conjuntivos y disyuntivos pueden determinarse a partir de las interpretaciones de una función booleana. Para obtener los 2^n términos canónicos conjuntivos de un conjunto de n variables booleanas se hace la conjunción de las n variables para cada interpretación, con cada variable negada si el valor correspondiente de la interpretación es 0 y sin negar si el valor de la permutación correspondiente es 1. Formalmente, si $\Psi_i = \{(x_1, a_1), \dots, (x_i, a_i), \dots, (x_n, a_n)\}$ entonces $t_i = l_1^{a_1} \wedge \dots \wedge l_i^{a_i} \wedge \dots \wedge l_n^{a_n}$, donde x_1, \dots, x_n es un conjunto de variables booleanas y a_1, \dots, a_n son constantes booleanas. Los 2^n términos canónicos conjuntivos para tres variables son mostrados como ejemplo en la Tabla 1.7. De la misma forma, para obtener los 2^n términos canónicos disyuntivos de un conjunto de n variables booleanas se hace la disyunción de las n variables para cada interpretación, con cada variable negada si el valor correspondiente de la interpretación es 1 y sin negar si el valor de la interpretación correspondiente es 0. Formalmente, si $\Psi_i = \{(x_1, a_1), \dots, (x_i, a_i), \dots, (x_n, a_n)\}$ entonces $\neg t_i = \neg l_1^{a_1} \vee \dots \vee \neg l_i^{a_i} \vee \dots \vee \neg l_n^{a_n}$, donde x_1, \dots, x_n es un conjunto de variables booleanas y a_1, \dots, a_n son constantes booleanas. Los 2^n términos canónicos disyuntivos para tres

variables son mostrados también en la Tabla 1.7. Es evidente a través del teorema booleano básico 5 de De Morgan, que debido a la forma en que se ha especificado su construcción cada término canónico disyuntivo $\neg t_i$, resulta ser la negación de su término canónico disyuntivo homólogo t_i y viceversa, lo que se puede corroborar fácilmente observando la Tabla 1.7.

Interpretaciones				T		$\neg T$	
	x	y	z	Denominación	Término	Denominación	Término
ψ_1	0	0	0	t_1	$\neg x \wedge \neg y \wedge \neg z$	$\neg t_1$	$x \vee y \vee z$
ψ_2	0	0	1	t_2	$\neg x \wedge \neg y \wedge z$	$\neg t_2$	$x \vee y \vee \neg z$
ψ_3	0	1	0	t_3	$\neg x \wedge y \wedge \neg z$	$\neg t_3$	$x \vee \neg y \vee z$
ψ_4	0	1	1	t_4	$\neg x \wedge y \wedge z$	$\neg t_4$	$x \vee \neg y \vee \neg z$
ψ_5	1	0	0	t_5	$x \wedge \neg y \wedge \neg z$	$\neg t_5$	$\neg x \vee y \vee z$
ψ_6	1	0	1	t_6	$x \wedge \neg y \wedge z$	$\neg t_6$	$\neg x \vee y \vee \neg z$
ψ_7	1	1	0	t_7	$x \wedge y \wedge \neg z$	$\neg t_7$	$\neg x \vee \neg y \vee z$
ψ_8	1	1	1	t_8	$x \wedge y \wedge z$	$\neg t_8$	$\neg x \vee \neg y \vee \neg z$

Tabla 1.7 Términos canónicos conjuntivos y disyuntivos obtenidos a partir de las interpretaciones de tres variables

Observe en la Tabla 1.7 que cada t_i resulta ser el término canónico conjuntivo que evaluado como expresión booleana hace verdadera a la interpretación ψ_i y falsa a las interpretaciones restantes. Esta propiedad se cumple en general para cualquier interpretación de cualquier número de variables debido (1^o) a que existen únicamente 2^n términos canónicos conjuntivos distintos, (2^o) a que por su definición existe solo una forma de escribir cada término canónico conjuntivo, y obviamente (3^o) a la forma en que se ha especificado la construcción de t_i . De la misma manera, en la Tabla 1.7 salta a la vista que cada $\neg t_i$ resulta ser el término canónico disyuntivo que evaluado como expresión booleana hace falsa a la interpretación ψ_i y verdadera a las interpretaciones restantes. Las propiedades expuestas en este párrafo son enunciadas formalmente a través del Teorema 1.5 y del Corolario 1.5 que se valen del Lema 1.1 y del Corolario 1.4 para su demostración. Finalmente, que t_i hace verdadera a la evaluación ψ_i y que $\neg t_i$ hace falsa a la evaluación ψ_i , es remarcado en el Corolario 1.6 y el Corolario 1.7.

Lema 1.1 si $\xi(t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 1$ entonces $t = I_{a_1}^{x_1} \wedge \dots \wedge I_{a_n}^{x_n}$

Demostración. Por inducción sobre ψ_i

1. si $\xi(t, \psi_i = \{(x_1, a_1)\}) = 1$ entonces $t = I_{a_1}^{x_1} \wedge \dots \wedge I_{a_n}^{x_n}$:

1.1. $\xi(t, \psi_i = \{(x_1, a_1)\}) = 1$

1.2. $a = 1 \vee a = 0$

1.3. si $a_1 = 1$ entonces $t = I_{1}^{x_1}$:

1.3.1. $a_1 = 1$

1.3.2. $\xi(t, \psi_i = \{(x_1, 1)\}) = 1$

1.3.3. $\xi(x_1, \psi_i = \{(x_1, 1)\}) = 1$

1.3.4. $t = x_1$

1.3.5. $I_{1}^{x_1} = x_1$

Hipótesis

De (1.1) por definición de ψ

Hipótesis

De (1.3.1) substituyendo en (1.1)

De (1.3.2) por definición de ξ

De (1.3.2) y (1.3.3)

De (1.3.1) por definición de $I_{1}^{x_1}$

1.3.6. $t = I_{x_1}^{a_1}$

1.4. si $a_1 = 0$ entonces $t = I_{x_1}^{a_1}$:

1.4.1. $a_1 = 0$

1.4.2. $\xi(t, \psi_i = \{(x_1, 0)\}) = 1$

1.4.3. $\xi(\neg x_1, \psi_i = \{(x_1, 0)\}) = 1$

1.4.4. $t = \neg x_1$

1.4.5. $I_{x_1}^{a_1} = \neg x_1$

1.4.6. $t = I_{x_1}^{a_1}$

1.5. $t = I_{x_1}^{a_1}$

1.6. $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_n}^{a_n}$

2. si $\xi(t, \psi_i = \{(x_1, a_1), \dots, (x_j, a_j), (x_{j+1}, a_{j+1})\}) = 1$ entonces $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_n}^{a_n}$:

2.1. $\xi(t, \psi_i = \{(x_1, a_1), \dots, (x_j, a_j), (x_{j+1}, a_{j+1})\}) = 1$

2.2. $\xi(t', \psi_i = \{(x_1, a_1), \dots, (x_j, a_j)\}) = 1$ y $t' = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j}$

2.3. $\xi(t', \psi_i = \{(x_1, a_1), \dots, (x_j, a_j)\}) = 1$

2.4. $t' = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j}$

2.5. $\psi_i = \psi_k \cup \{(x_{j+1}, a_{j+1})\}$

2.6. $\xi(t, \psi_k \cup \{(x_{j+1}, a_{j+1})\}) = 1$

2.7. $a_{j+1} = 1$ o $a_{j+1} = 0$

2.8. si $a_{j+1} = 1$ entonces $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge I_{x_{j+1}}^{a_{j+1}}$:

2.8.1. $a_{j+1} = 1$

2.8.2. $\xi(t, \psi_k \cup \{(x_{j+1}, 1)\}) = 1$

2.8.3. $\xi(t' \wedge x_{j+1}, \psi_k \cup \{(x_{j+1}, 1)\}) = 1$

2.8.4. $\xi(I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge x_{j+1}, \psi_k \cup \{(x_{j+1}, 1)\}) = 1$

2.8.5. $I_{x_{j+1}}^{a_{j+1}} = x_{j+1}$

2.8.6. $\xi(I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge I_{x_{j+1}}^{a_{j+1}}, \psi_k \cup \{(x_{j+1}, 1)\}) = 1$

2.8.7. $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge I_{x_{j+1}}^{a_{j+1}}$

2.9. si $a_{j+1} = 0$ entonces $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge I_{x_{j+1}}^{a_{j+1}}$:

2.9.1. $a_{j+1} = 0$

2.9.2. $\xi(t, \psi_k \cup \{(x_{j+1}, 0)\}) = 1$

2.9.3. $\xi(t' \wedge \neg x_{j+1}, \psi_k \cup \{(x_{j+1}, 0)\}) = 1$

2.9.4. $\xi(I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge \neg x_{j+1}, \psi_k \cup \{(x_{j+1}, 0)\}) = 1$

2.9.5. $I_{x_{j+1}}^{a_{j+1}} = \neg x_{j+1}$

2.9.6. $\xi(I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge I_{x_{j+1}}^{a_{j+1}}, \psi_k \cup \{(x_{j+1}, 0)\}) = 1$

2.9.7. $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge I_{x_{j+1}}^{a_{j+1}}$

2.10. $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_j}^{a_j} \wedge I_{x_{j+1}}^{a_{j+1}}$

2.11. $t = I_{x_1}^{a_1} \wedge \dots \wedge I_{x_n}^{a_n}$

De (1.3.4) y (1.3.5)

Hipótesis

De (1.4.1) substituyendo en (1.1)

De (1.4.2) por definición de ξ

De (1.4.2) y (1.4.3)

De (1.4.1) por definición de $I_{x_1}^{a_1}$

De (1.4.4) y (1.4.5)

De (1.2), (1.3) y (1.4) por casos

De (1.5) y (1.1) pues $n = 1$

Hipótesis

Hipótesis inductiva

De (2.2) de la conjunción a la parte

De (2.2) de la conjunción a la parte

De (2.1) y (2.3)

De (2.5) substituyendo en (2.1)

De (2.1) por definición de ψ

Hipótesis

De (2.8.1) substituyendo en (2.6)

De (2.3) y (2.8.2) por definición de ξ

De (2.4) substituyendo en (2.8.3)

De (2.8.1) por definición de $I_{x_{j+1}}^{a_{j+1}}$

De (2.8.5) substituyendo en (2.8.4)

De (2.6) y (2.8.6)

Hipótesis

De (2.9.1) substituyendo en (2.6)

De (2.3) y (2.9.2) por definición de ξ

De (2.4) substituyendo en (2.9.3)

De (2.9.1) por definición de $I_{x_{j+1}}^{a_{j+1}}$

De (2.9.5) substituyendo en (2.9.4)

De (2.6) y (2.9.6)

De (2.7), (2.8) y (2.9) por casos

De (2.10) y (2.1) pues $n = j + 1$ **Corolario 1.4** (del Lema 1.1). si $\xi(\neg t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 0$ entonces $\neg t = \neg I_{x_1}^{a_1} \wedge \dots \wedge \neg I_{x_n}^{a_n}$

Teorema 1.5 para todo i , si $\xi(t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 1$ entonces $t = t_i$

Demostración. Por reducción al absurdo, exhibiendo contradicción en pasos (7) y (8)

1. existe i tal que $\xi(t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 1$ y $t \neq t_i$

2. $\xi(t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 1$ y $t \neq t_i$

3. $\xi(t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 1$

4. $t = l_i^n \wedge \dots \wedge l_i^n$

5. $\psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}$

6. $t_i = l_i^n \wedge \dots \wedge l_i^n$

7. $t = t_i$

8. $t \neq t_i$

■

Corolario 1.5 (del Teorema 1.5 y Corolario 1.4) para todo i , si $\xi(\neg t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 0$ entonces $\neg t = \neg t_i$

Corolario 1.6 (del Teorema 1.5) $\xi(t_i, \psi_i) = 1$

Corolario 1.7 (del Corolario 1.5) $\xi(\neg t_i, \psi_i) = 0$

Un polinomio booleano en forma canónica disyuntiva o polinomio canónico disyuntivo, denotado como $\vee \zeta$, es una expresión formada por la disyunción de un conjunto ζ posiblemente vacío de términos canónicos conjuntivos. Un polinomio booleano en forma canónica conjuntiva o polinomio canónico conjuntivo, denotado como $\wedge \delta$, es una expresión formada por la conjunción de un conjunto δ posiblemente vacío de términos canónicos disyuntivos. Del Teorema 1.4 sabemos que existen exactamente 2^n términos canónicos conjuntivos, por lo tanto es evidente que el número máximo de términos canónicos conjuntivos que puede tener un polinomio canónico disyuntivo es 2^n . De la misma manera, por el Corolario 1.3 sabemos que existen exactamente 2^n términos canónicos disyuntivos, por lo que el número máximo de términos canónicos disyuntivos que puede tener un polinomio canónico conjuntivo es 2^n . Las dos anteriores propiedades son remarcadas en el Corolario 1.8 y el Corolario 1.9.

Corolario 1.8 (del Teorema 1.4). El número máximo de términos canónicos conjuntivos que puede tener un polinomio canónico disyuntivo es 2^n .

Corolario 1.9 (del Corolario 1.3). El número máximo de términos canónicos disyuntivos que puede tener un polinomio canónico conjuntivo es 2^n .

Un polinomio booleano en forma canónica o polinomio canónico es un polinomio canónico conjuntivo o disyuntivo. En adelante D denotará al conjunto de todos los polinomios canónicos disyuntivos que pueden formarse del conjunto de variables booleanas x_1, \dots, x_n , formalmente, $\zeta \in 2^T$ si y solo si $\vee \zeta \in D$. De la misma forma C denotará al conjunto de todos los polinomios canónicos conjuntivos que pueden formarse del conjunto de variables booleanas x_1, \dots, x_n , formalmente, $\delta \in 2^{-T}$ si y solo si $\wedge \delta \in C$. El número total de

polinomios canónicos disyuntivos que pueden formarse con n variables es 2^n lo que es fácilmente demostrable debido a tres factores, el primero es que el Teorema 1.4 dice que en T existen 2^n términos canónicos, el segundo es que por definición $\zeta \in 2^T$ si y solo si $V\zeta \in D$ y el tercero es que sabemos por el Teorema 1.2 que la cardinalidad de 2^T es $2^{|T|}$. De igual manera es demostrable que el número total de polinomios canónicos conjuntivos que pueden formarse con n variables es 2^{2^n} . Las dos anteriores propiedades son escritas formalmente en el Teorema 1.6 y el Corolario 1.10.

Teorema 1.6 $|D| = 2^n$

Demostración.

1. $\zeta \in 2^T$ si y solo si $V\zeta \in D$

2. $|2^T| = |D|$

3. $|2^T| = 2^{|T|}$

4. $|T| = 2^n$

5. $|2^T| = 2^{2^n}$

6. $|D| = 2^{2^n}$

■

Definición de D

De (1) pues existe un polinomio canónico en D por cada combinación en 2^T

Teorema 1.2

Teorema 1.4

De (4) substituyendo en (3)

De (2) substituyendo en (5)

Corolario 1.10 (del Teorema 1.6) $|C| = 2^{2^n}$

Es fácil reconocer a través del Teorema 1.5 que para evaluar un polinomio canónico disyuntivo a 1 bajo una interpretación ψ , es necesario incluir a t_i en tal polinomio. De esta manera sabemos que es posible representar a cualquier ψ , que se evalúe a 1 a través de un polinomio canónico disyuntivo en el que esté el t_i correspondiente. Finalmente, el que podamos representar a cualquier ψ , evaluada a 1 significa que podemos representar a cualquier función booleana mediante un polinomio canónico disyuntivo. Lo dicho anteriormente es establecido formalmente en el Teorema 1.7 y el Corolario 1.12. Además, a través del teorema 5 básico booleano de De Morgan se hace evidente que lo explicado para los polinomios canónicos disyuntivos en el Teorema 1.7 y el Corolario 1.12, se cumple también para los polinomios canónicos conjuntivos, siendo formalizada esta propiedad en el Corolario 1.11 y el Corolario 1.13.

Teorema 1.7 $t_i \in \zeta$ si y solo si $\xi(V\zeta, \psi) = 1$

Demostración.

1. si $t_i \in \zeta$ entonces $\xi(V\zeta, \psi) = 1$:

Hipótesis

1.1. $t_i \in \zeta$

De (1.1) por paso algebraico

1.2. $\zeta = (\zeta - \{t_i\}) \cup \{t_i\}$

De (1.2) por definición de $V\zeta$

1.3. $V\zeta = V((\zeta - \{t_i\}) \cup \{t_i\})$

De (1.3) por definición de V

1.4. $V((\zeta - \{t_i\}) \cup \{t_i\}) = V(\zeta - \{t_i\}) \vee \{t_i\}$

De (1.4) por definición de ξ

1.5. $\xi(V((\zeta - \{t_i\}) \cup \{t_i\}), \psi) = \xi(V(\zeta - \{t_i\}), \psi) \vee \xi(\{t_i\}, \psi)$

Corolario 1.6

1.6. $\xi(\{t_i\}, \psi) = 1$

De (1.6) substituyendo en (1.5)

1.7. $\xi(V((\zeta - \{t_i\}) \cup \{t_i\}), \psi) = \xi(V(\zeta - \{t_i\}), \psi) \vee 1$

De (1.7) por Teorema 2 básico

1.8. $\xi(V((\zeta - \{t_i\}) \cup \{t_i\}), \psi) = 1$

- 1.9. $\xi(V_G, \Psi) = 1$ De (1.3) substituyendo en (1.8)
2. si $\xi(V_G, \Psi) = 1$ entonces $t_i \in \mathcal{C}$: Por reducción al absurdo mostrando contradicción en (2.3) y (2.4)
- 2.1. $\xi(V_G, \Psi) = 1$ y $t_i \notin \mathcal{C}$ Negación de (2)
- 2.2. para todo t , si $t \in \mathcal{C}$ entonces $\xi(t, \Psi) = 1$:
- 2.2.1. $t \in \mathcal{C}$ Hipótesis
- 2.2.2. $t_i \in \mathcal{C}$ De (2.1) de la conjunción a la parte
- 2.2.3. para todo t , si $t \in \mathcal{C}$ entonces $t \neq t_i$ De (2.2.2) por definición de \mathcal{C} y por que $\mathcal{C} \subseteq T$
- 2.2.4. $t \neq t_i$ De (2.2.1) por (2.2.3)
- 2.2.5. para todo t , si $t \neq t_i$ entonces $\xi(t, \Psi) = 1$ Teorema 1.5 girado
- 2.2.6. $\xi(t, \Psi) = 1$ De (2.2.4) por (2.2.5)
- 2.3. $\xi(V_G, \Psi) = 1$ De (2.2) por definición de V
- 2.4. $\xi(V_G, \Psi) = 1$ De (2.1) de la conjunción a la parte
3. $t_i \in \mathcal{C}$ si y solo si $\xi(V_G, \Psi) = 1$ De (1) y (2)



Corolario 1.11 $\neg t_i \in \delta$ si y solo si $\xi(\wedge \delta, \Psi) = 0$

Corolario 1.12 (del Teorema 1.7) para todo f , si f es una función booleana entonces f es representable por un polinomio booleano canónico disyuntivo

Corolario 1.13 (del Corolario 1.11) para todo f , si f es una función booleana entonces f es representable por un polinomio booleano canónico conjuntivo.

Sabemos por el Teorema 1.7 que si la evaluación de un polinomio canónico disyuntivo es verdadera bajo una interpretación Ψ , entonces significa que el término canónico disyuntivo t_i forma parte de tal polinomio canónico disyuntivo. Por lo tanto, podemos deducir el Teorema 1.8 que resulta de gran relevancia pues indica que si bajo cualquier interpretación la evaluación de un polinomio canónico disyuntivo V_{G_1} es igual a la evaluación de otro polinomio canónico disyuntivo V_{G_2} entonces significa que V_{G_1} y V_{G_2} están formados por los mismos términos canónicos conjuntivos, es decir, que si V_{G_1} tiene diferentes términos canónicos conjuntivos que V_{G_2} entonces sus evaluaciones también serán distintas. Más claramente, el Teorema 1.8 nos indica que cada uno de los 2^{2^n} polinomios canónicos disyuntivos que es posible formar tienen evaluaciones distintas. Lo expresado anteriormente en este párrafo para los polinomios canónicos disyuntivos es igualmente válido para el caso de los polinomios canónicos conjuntivos debido al teorema 5 básico booleano de De Morgan, lo que es afirmado en el Corolario 1.14.

Teorema 1.8 $\xi(V_{G_1}, \Psi) = \xi(V_{G_2}, \Psi)$ si y solo si $\mathcal{C}_1 = \mathcal{C}_2$

Demostración.

1. si $\xi(V_{G_1}, \Psi) = \xi(V_{G_2}, \Psi)$ entonces $\mathcal{C}_1 = \mathcal{C}_2$:

1.1. $\xi(V_{G_1}, \Psi) = \xi(V_{G_2}, \Psi)$ Hipótesis

1.2. para todo t_i , si $t_i \in \mathcal{C}_1$ entonces $t_i \in \mathcal{C}_2$:

1.2.1. $t_i \in \mathcal{C}_1$ Hipótesis

1.2.2. $\xi(V_{c_1}, \Psi) = 1$

1.2.3. $\xi(V_{c_2}, \Psi) = 1$

1.2.4. $t_i \in c_2$

1.3. para todo t_i , si $t_i \in c_2$ entonces $t_i \in c_1$:

1.3.1. $t_i \in c_2$

1.3.2. $\xi(V_{c_2}, \Psi) = 1$

1.3.3. $\xi(V_{c_1}, \Psi) = 1$

1.3.4. $t_i \in c_2$

2. si $c_1 = c_2$ entonces $\xi(V_{c_1}, \Psi) = \xi(V_{c_2}, \Psi)$ 3. $\xi(V_{c_1}, \Psi) = \xi(V_{c_2}, \Psi)$ si y solo si $c_1 = c_2$

De (1.2.1) por el Teorema 1.7

De (1.2.2) por (1.1)

De (1.2.3) por el Teorema 1.7

Hipótesis

De (1.3.1) por el Teorema 1.7

De (1.3.2) por (1.1)

De (1.3.3) por Teorema 1.7

Por definición de = y ξ

De (1) y (2)

Corolario 1.14 (del Teorema 1.8) $\xi(\wedge \delta_1, \Psi) = \xi(\wedge \delta_2, \Psi)$ si y solo si $\delta_1 = \delta_2$

Sabemos por el Corolario 1.12 que cualquier función es representable en forma canónica disyuntiva. Por otra parte el Teorema 1.8 nos dice que cualquiera de los polinomios booleanos que es posible crear tiene una evaluación distinta a cualquier otro polinomio booleano que pueda crearse. Por lo tanto, del Corolario 1.12 y el Teorema 1.8 se deriva que cualquier función es representable en forma canónica disyuntiva de una y solo una manera, lo que es expresado formalmente en el Corolario 1.15. Lo anteriormente dicho para la forma disyuntiva es igualmente válido a través del Corolario 1.13 y del Corolario 1.14 para la forma canónica conjuntiva, esto mismo se presenta en el Corolario 1.16.

Corolario 1.15 (del Corolario 1.12 y el Teorema 1.8) para todo f , si f es una función booleana entonces f es representable por un y solo un polinomio canónico disyuntivo

Corolario 1.16 (del Corolario 1.13 y el Corolario 1.14) para todo f , si f es una función booleana entonces f es representable por un y solo un polinomio canónico conjuntivo

Por el Corolario 1.15 sabemos que cualquier función puede representarse en forma canónica disyuntiva de una y solo una manera, además sabemos por el Teorema 1.6 que existen exactamente 2^n polinomios canónicos disyuntivos distintos para n variables. Por lo tanto, es evidente que pueden crearse exactamente 2^n funciones booleanas diferentes a partir de n variables. Este mismo resultado pudo haberse obtenido mucho más atrás calculando las combinaciones que pueden crearse a partir de Ψ , sin embargo se decidió mostrarse hasta este momento para recalcar la afinidad entre las funciones booleanas y los polinomios canónicos. Las ideas presentadas anteriormente en este párrafo son enunciadas formalmente en el Corolario 1.17.

Corolario 1.17 (del Corolario 1.15 y el Teorema 1.6) Existen exactamente 2^n funciones booleanas diferentes que pueden formarse a partir de n variables booleanas.

Dado que existe una correspondencia directa entre polinomios canónicos y funciones booleanas, es posible especificar cualquier función booleana en forma canónica disyuntiva a partir de la tabla de verdad de la función, creando un término canónico conjuntivo para cada combinación de variables que produce un 1 en

la función y formando entonces la disyunción de todos los términos canónicos conjuntivos creados. De la misma manera, una función booleana puede especificarse en forma canónica conjuntiva a partir de una tabla de verdad dada, creando un término canónico disyuntivo para cada combinación de variables que produce un 0 en la función y tomando entonces la conjunción de todos los términos canónicos disyuntivos creados.

Una función booleana también puede especificarse en forma canónica en la notación canónica disyuntiva abreviada que tiene la forma $f(x_1, \dots, x_n) = \vee(1, 2, 3, 4)$ en la que el símbolo \vee representa la disyunción de los términos canónicos conjuntivos t_1, t_2, t_3 y t_4 que hacen la función $f(x_1, \dots, x_n)$ igual a 1. De la misma forma, una función booleana puede especificarse en la notación canónica conjuntiva abreviada que tiene la forma $f(x_1, \dots, x_n) = \wedge(1, 2, 3, 4)$ en la que el símbolo \wedge representa la conjunción de los términos canónicos disyuntivos $\neg t_1, \neg t_2, \neg t_3$ y $\neg t_4$ que hacen la función $f(x_1, \dots, x_n)$ igual a 1.

Usando la notación canónica abreviada se puede convertir fácilmente una forma canónica disyuntiva en conjuntiva y viceversa, lo único que se requiere es intercambiar los símbolos \wedge y \vee , para finalmente listar los números faltantes de los 2^n posibles.

La forma canónica muy rara vez contiene el menor número de literales posibles para especificar una función booleana, debido a que cada término canónico disyuntivo o conjuntivo debe contener por definición todas las variables de una función. Debido a esta desventaja las funciones booleanas son generalmente representadas en forma normal.

Una forma canónica puede convertirse a una forma normal y viceversa. Para convertir una forma normal disyuntiva a una forma canónica disyuntiva lo único que debe hacerse es insertar un $(x\vee\neg x)$ para cada variable x no presente en el polinomio en forma normal. Para convertir una forma estándar conjuntiva a una forma canónica conjuntiva lo único que debe hacerse es insertar un $(x\wedge\neg x)$ por cada variable x no presente en el polinomio.

Para convertir una forma canónica a una forma normal se han propuesto numerosos métodos de minimización para reducir una expresión canónica a sus mínimas literales [Veitch 1952], [Karnaugh 1953], [Quine 1952], [McCluskey 1956]. Los métodos de minimización están basados en el mismo principio de resolución [Robinson 1965]. Aún cuando el principio de resolución no es detallado en este escrito, puede ser de importante aplicación en futuras investigaciones de la teoría de la red neural canónica pues, como es propuesto en la sección 7, su aplicación podría dar lugar a una reducción sustancial del procesamiento necesario para el funcionamiento de una red neural canónica.

2. REDES NEURONALES BIOLÓGICAS

2.1 EL SISTEMA NERVIOSO

El sistema nervioso constituye el soporte material a través del cual un ser vivo capta señales externas e internas a su organismo y reacciona ante ellas mediante la generación de respuestas endocrinas, autonómicas y conductuales. En el caso de los organismos más desarrollados y en particular del hombre, el sistema nervioso alcanza un grado de desarrollo tal que permite la generación de respuestas totalmente adaptativas, así como una abstracción elevada de las señales que procesa. Esto libera al humano de una interacción directa con tales señales para reaccionar o proceder con respecto a ellas.

El sistema nervioso está constituido principalmente por células conocidas con el nombre de neuronas. La neurona o neurocito es la unidad funcional del sistema nervioso. Las neuronas son células especializadas en la conducción de impulsos y por medio de ellas se concretan todas las funciones del sistema nervioso. El encéfalo está conformado por lo menos por 10^{16} neuronas, cada una de las cuales se comunica generalmente con otras 10^3 a 10^7 neuronas. El sistema nervioso, tal y como se muestra en la Ilustración 2-1, se divide en dos partes principales sistema nervioso central y sistema nervioso periférico.

El sistema nervioso periférico se subdivide en aferente y eferente. El sistema aferente consiste principalmente en las células nerviosas llamadas neuronas aferentes o sensoriales o sensitivas que transmiten información de entrada en forma de impulsos hacia el sistema central desde los receptores periféricos ubicados en el interior o exterior del organismo. El sistema eferente consiste principalmente en las células nerviosas llamadas neuronas eferentes o motoras que transmiten información de salida en forma de impulsos desde el sistema nervioso central hasta los músculos y glándulas. El sistema nervioso periférico se encuentra irrigando la totalidad del cuerpo humano.

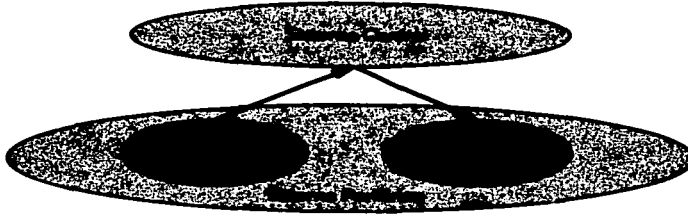


Ilustración 2-1 Organización del sistema nervioso

El sistema nervioso central está formado principalmente por las células nerviosas llamadas neuronas asociativas o de asociación cuya función es interpretar los impulsos de las neuronas aferentes o de otras neuronas asociativas para entonces emitir información en forma de impulsos a otras neuronas eferentes o asociativas. Al sistema nervioso central se le atribuye el ejercicio del razonamiento, creatividad, planeación, memorización y demás actividades de las que supone el humano se compone la inteligencia. El sistema nervioso central se encuentra distribuido en el encéfalo y en la médula espinal. Son conocidas como funciones integradoras, las actividades cerebrales como memoria, sueño, vigilia, y respuestas emocionales.

La memoria es la capacidad de evocar estímulos generados con anterioridad. A la memoria se le ha clasificado como memoria de corto plazo y memoria de largo plazo. La memoria de corto plazo es la capacidad de evocar estímulos durante un tiempo relativamente corto para después olvidarlos debido a la poca importancia que revisten. La memoria a largo plazo es la capacidad de evocar estímulos que nunca son olvidados. El olvido puede explicarse como un proceso que permite no retener datos intrascendentes en una memoria presumiblemente finita.

2.2 LA NEURONA BIOLÓGICA

Las principales partes de una neurona son el soma, las dendritas, la zona de integración, el axón y la membrana, tal y como se puede ver en la Ilustración 2-2. El soma o pericarion o cuerpo celular es la zona donde se encuentra la maquinaria bioquímica para la síntesis de enzimas y otras moléculas esenciales para la vida de la neurona. Las dendritas son prolongaciones aferentes de la neurona formadas por ramificaciones múltiples que nacen del soma de la neurona. A través de las dendritas, la neurona recibe los impulsos de entrada.

La zona de integración o cono axónico es el lugar de unión entre el axón y el cuerpo de la neurona. Es en la zona de integración donde se procesan los impulsos recibidos de las dendritas, si los impulsos recibidos de las dendritas rebasan en su conjunto un nivel llamado umbral entonces en la zona de integración se genera un impulso que es transmitido por el axón.

El axón o cilindroje es una prolongación eferente única y muy larga de la neurona. El axón es el encargado de transmitir los impulsos generados por la zona de integración hacia otras neuronas o bien hacia efectores como glándulas y músculos. La longitud total del axón varía generalmente desde un milímetro hasta más de dos metros en las conexiones hacia los pies.

La membrana de la neurona es una estructura laminar de composición lípido-proteica, que delimita espacios y tiene permeabilidad selectiva, es decir, regula el paso de solutos. Esto último no se rige sólo por leyes osmóticas, sino que existe un transporte activo de substancias hacia adentro o afuera de la membrana y que gasta energía debido a que se trata de un mecanismo enzimático.

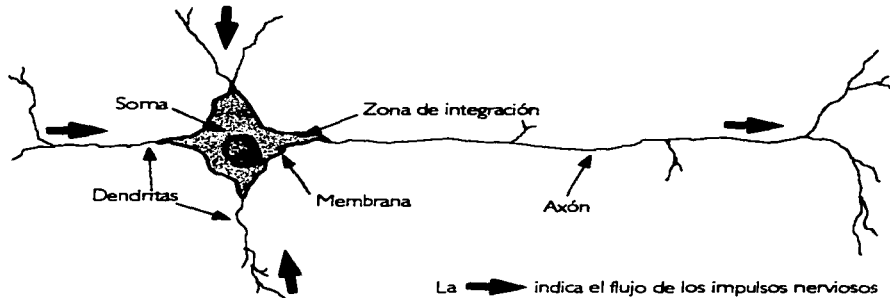


Ilustración 2-2 Principales partes de la neurona

Las neuronas se clasifican de acuerdo a su anatomía en multipolares, bipolares y unipolares. Las neuronas multipolares tienen varias dendritas y un solo axón; son encontradas en el sistema nervioso central principalmente. Las neuronas bipolares tienen una dendrita y un axón; son encontradas en la retina, oído y olfato. Las neuronas unipolares tienen una prolongación dividida en dos ramas, central y periférica que cumplen las funciones del axón y dendrita respectivamente; son encontradas en el sistema nervioso periférico. Reciben el nombre de nervios las prolongaciones dendríticas y axónicas de las neuronas aferentes y eferentes.

2.3 SINAPSIS

Dos son las características más notables de la fisiología de la neurona, su limitada facultad regenerativa y su capacidad de excitabilidad, capacitancia y conductividad de mensajes eléctricos a los que se denomina impulsos nerviosos. La excitabilidad es la capacidad de respuesta de la neurona a estímulos transformándolos en impulsos nerviosos. La capacitancia es la capacidad de la neurona para acumular energía eléctrica. La

conductividad es la capacidad de la neurona de transmitir un impulso generado por ella hacia otras neuronas o hacia un órgano efector, como los músculos y las glándulas.

A la zona de unión funcional entre dos neuronas se le llama **sinapsis**. En la Ilustración 2-3 se muestra una sinapsis. La sinapsis es la estructura única para la transferencia de impulsos de una neurona a otra, en otras palabras, la sinapsis es la vía de comunicación entre dos neuronas. Un conjunto de dos o más neuronas interconectadas entre sí a través de sinapsis recibe el nombre de **red neuronal**.

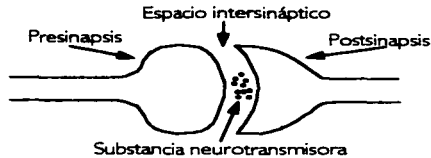


Ilustración 2-3 Conducción de un impulso desde un axón presináptico hacia una dendrita postsináptica

La sinapsis consta de tres partes principales que son la presinápsis, la postsinapsis y el espacio intersináptico. La **presinapsis** está formada por el órgano eferente de la neurona transmisora y está representada por el axón. La **postsinapsis** está formada por el órgano aferente de la neurona receptora y está representada generalmente por la dendrita, sin embargo, pueden funcionar también como postsinapsis el soma o incluso otro axón. El **espacio intersináptico** o hendidura intersináptica es el espacio físico de aproximadamente 200 Å existente entre la presinapsis y la postsinapsis. El paso de un del pulso nervioso de la neurona presináptica a la postsináptica no se efectúa transmitiendo impulsos eléctricos directamente, sino que a nivel de la sinapsis, el órgano presináptico libera ciertas sustancias químicas llamadas neurotransmisoras que recorren el espacio intersináptico y son captadas por el órgano postsináptico. El retardo sináptico es el tiempo en el que un impulso atraviesa la sinapsis, aproximadamente 0.5 milisegundos.

Las sinapsis se clasifican en axodendríticas, axosomáticas y axoaxónicas. La **sinapsis axodendrítica** es la sinapsis de un axón con una dendrita. La **sinapsis axosomática** es la sinapsis de un axón con el soma de una neurona. Finalmente, la **sinapsis axoaxónica** es la sinapsis de un axón con otro axón.

Una red neuronal está organizada a manera de una extensa y compleja red de comunicaciones entre las neuronas a través de las interconexiones de sus sinapsis. Un **circuito** es un patrón de interconexión sináptica establecido en una red neuronal para la conducción de impulsos. En los sistemas biológicos se han identificado cuatro tipos principales de circuitos, el divergente, el convergente, el paralelo y el de reverberación, todos ellos mostrados en la Ilustración 2-4. En los **circuitos divergentes** el impulso proveniente de una sola neurona presináptica estimula a dos o más neuronas postsinápticas. En los **circuitos convergentes** el impulso proveniente de dos o más neuronas presinápticas estimula a una sola neurona postsináptica.

Tal vez el circuito que ha acaparado mayor interés de investigación ha sido el reverberante. En los circuitos reverberantes el impulso de alguna parte del circuito es retransmitido a las neuronas presinápticas a manera de retroalimentación. Como características interesantes en los circuitos reverberantes, se ha descubierto que una vez activado puede durar desde uno cuantos segundos hasta varias horas; la duración dependerá del número de neuronas y de su disposición en el circuito. Entre las respuestas corporales que al parecer son resultado de señales reverberantes se encuentran el ritmo respiratorio, las actividades musculares coordinadas, el despertarse y el dormirse. Además, algunos investigadores y el mismo autor, sostenemos que los circuitos de reverberación guardan relación con la memoria.

En los circuitos paralelos una sola célula presináptica estimula a un grupo de neuronas que a su vez establecen sinapsis con alguna célula postsináptica común. La ventaja de los circuitos paralelos es que la neurona presináptica puede enviar impulsos consecutivos, conforme los reciba, hasta un máximo de 2000 por segundo; este tipo de circuito no tiene sistema de retroalimentación por lo que se interrumpe toda vez que las neuronas que lo componen hayan transmitido sus impulsos a la neurona postsináptica. Algunos investigadores piensan que se emplea este circuito en actividades que requieren precisión o rapidez.

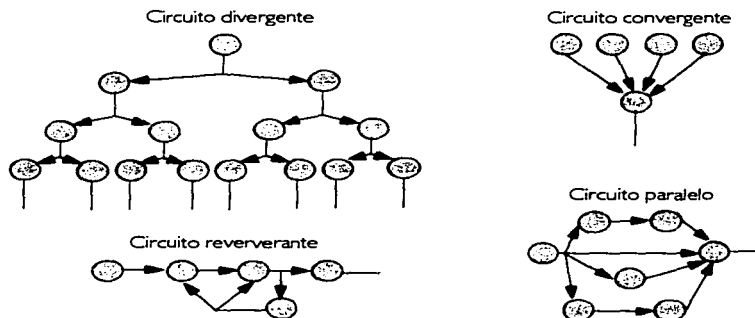


Ilustración 2-4 Circuitos de redes neuronales

Como se ha indicado anteriormente, el funcionamiento de una neurona está basado en la recepción, almacenamiento y transmisión de impulsos eléctricos. Se conoce como potencial de una neurona al voltaje almacenado por la neurona en un momento dado. El potencial de una neurona aumenta o disminuye dependiendo de los estímulos recibidos de sus sinapsis. Si una sinapsis hace reducir el potencial de una neurona se dice que se trata de una sinapsis inhibitoria . Si una sinapsis hace aumentar el potencial de una neurona se dice que se trata de una sinapsis excitatoria . No todas las neuronas presinápticas inhiben o excitan con igual intensidad a una neurona postsináptica, sino que unas lo hacen con mayor o menor magnitud que las otras. El peso sináptico es la intensidad con la que una neurona presináptica estimula inhibitoria o excitatoriamente a una neurona postsináptica.

De acuerdo a la fisiología de su sinapsis una neurona puede ser excitadora o inhibidora, es decir, que participan en la excitación o en la inhibición de una neurona postsináptica respectivamente. El sistema central contiene tanto neuronas excitadoras como inhibidoras, mientras que el sistema nervioso periférico solo tiene neuronas excitadoras.

2.4 POTENCIALES

Los potenciales o voltajes de una neurona son clasificados como potencial de reposo y potencial de acción que pueden observarse gráficamente en la Ilustración 2-5. El potencial de reposo de una neurona es el voltaje en milivolts de la neurona en condiciones de poca estimulación. Debido a su falta de estimulación, una neurona con potencial de reposo no genera impulso alguno. El potencial de reposo se encuentra generalmente en los -70 milivolts y se mantiene estable gracias a las propiedades de la membrana neuronal. Durante el potencial de reposo se dice que la neurona está polarizada.

El potencial de acción de una neurona es el voltaje en milivolts de la neurona en condiciones de alta estimulación. Debido a su gran estimulación, una neurona con potencial de acción genera un impulso en la zona de integración que es transmitido por el axón hacia otras neuronas o glándulas, o bien hacia tejido muscular. El potencial de acción es propiciado, al igual que el potencial de reposo, por las propiedades de la membrana neuronal. Durante el proceso de realización de un potencial de acción se observa un cambio abrupto en el funcionamiento de la neurona que puede dividirse en tres etapas, (1ª) la despolarización, (2ª) el potencial de acción y (3ª) la repolarización.

En la despolarización la neurona sube su voltaje hasta llegar a los 0 volts, límite en el cual se dice que la neurona está despolarizada. A partir de la despolarización se inicia el potencial de acción o pulso nervioso, cuya duración es de un milisegundo en promedio. En el potencial de acción el voltaje continua subiendo para alcanzar de manera repentina rangos de hasta 30 milivolts. Una vez que se ha llegado a los 30 milivolts se inicia la etapa de repolarización para restablecer el potencial de reposo. En la repolarización el voltaje de la neurona se decremenca cayendo incluso ligeramente por debajo de los -70 milivolts para finalmente recuperar voltaje y establecer una tendencia al potencial de reposo. Una vez terminada la repolarización, la neurona queda preparada para iniciar una nueva transmisión de un pulso nervioso. De hecho, en tanto no ocurra la repolarización, la neurona no podrá transmitir impulsos. El lapso durante el cual ocurre la recuperación de la neurona recibe el nombre de período refractario.

..

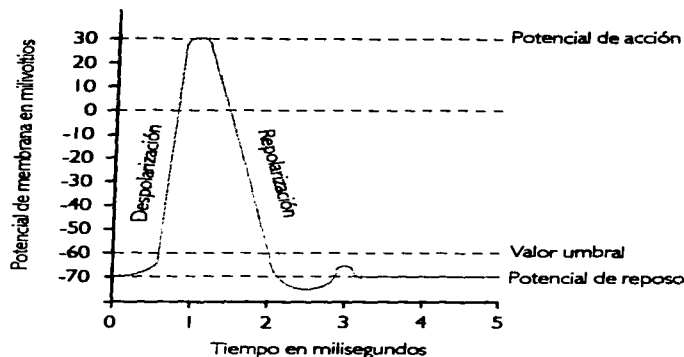


Ilustración 2-5 Voltaje en la neurona durante el desencadenamiento de un potencial de acción

El nivel de umbral es el valor crítico en milivolts que una vez alcanzado por el potencial de la neurona crea un cambio en el comportamiento eléctrico neuronal para producir el potencial acción. Recibe el nombre de estímulo subumbral o subliminal el que es resultado de la integración de las sinapsis excitatorias e inhibitorias y cuya intensidad resulta insuficiente para llegar al nivel de umbral, siendo por lo tanto incapaz de desencadenar un impulso nervioso por sí mismo. Se denomina estímulo umbral o liminal al que es resultado de la integración de las sinapsis excitatorias e inhibitorias y que tiene la intensidad suficiente para llegar al nivel de umbral, por lo que es capaz de originar un impulso nervioso. El estímulo umbral se logra ya sea por sumación temporal o por sumación espacial. La sumación espacial ocurre cuando la integración de las sinapsis excitatorias e inhibitorias en un mismo tiempo dan un resultado positivo tan alto que se llega al nivel de umbral, generando entonces el potencial de acción. La sumación temporal ocurre cuando repetidas integraciones de sinapsis excitatorias e inhibitorias dan resultados positivos que por si mismos no alcanzarían el valor umbral, pero que siendo constantes van acumulándose en la neurona e incrementando paulatinamente potencial hasta alcanzar el nivel de umbral.

En general, el impulso recibido por una sinapsis excitatoria no es suficiente para originar el potencial de acción en una neurona postsináptica, sin embargo, tal impulso sí acerca el potencial de reposo de la neurona al nivel umbral, de manera que la neurona se vuelve más excitable a impulsos provenientes de las neuronas presinápticas y aumentan la posibilidad de generar un impulso. El acercamiento del potencial de reposo de la neurona al nivel umbral debido al impulso recibido por una sinapsis excitatoria que no es suficiente para originar el potencial de acción es llamado potencial excitatorio postsináptico o efecto de facilitación. El potencial excitatorio postsináptico dura solo unos milisegundos y es durante estos instantes en los que a través de la sumación temporal o espacial puede originarse un impulso nervioso. El comportamiento del potencial excitatorio postsináptico es mostrado en la Ilustración 2-6.

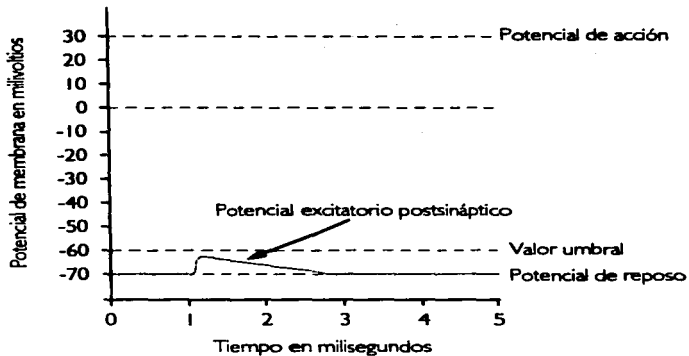


Ilustración 2-6 Voltaje en la neurona durante un potencial excitatorio postsináptico

Los impulsos de las sinapsis inhibitorias hiperpolarizan negativamente a la neurona postsináptica atenuando los impulsos enviados por las neuronas excitatorias, de manera que se retarda o impide la creación del potencial de acción a través de la sumación temporal o espacial. El cambio de una neurona postsináptica que da como resultado el aumento en la negatividad del potencial de la neurona recibe el nombre de potencial inhibitorio postsináptico. El comportamiento del potencial inhibitorio postsináptico es mostrado en la Ilustración 2-7.

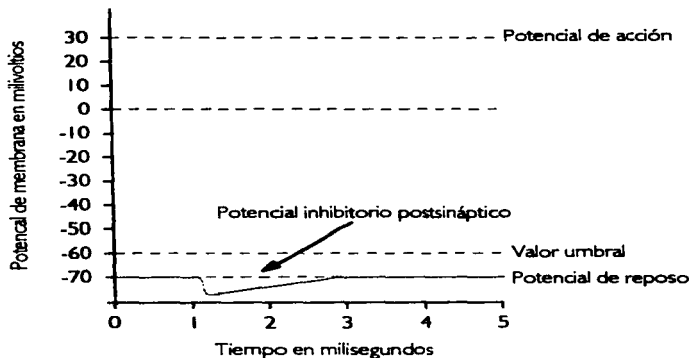


Ilustración 2-7 Voltaje en la neurona durante un potencial inhibitorio postsináptico

A manera de resumen, el potencial de una neurona es determinado por los impulsos recibidos en sus sinapsis excitatorias e inhibitorias. Los impulsos de las sinapsis excitatorias e inhibitorias recibidas son integrados por la neurona respondiendo a ellos en cualquiera de las siguientes maneras:

- Si el efecto inhibitor es más fuerte que el excitador entonces el resultado es que el potencial de la neurona se decrementa hiperpolarizando la neurona y creando un potencial inhibitorio postsináptico.
- Si el efecto excitador es más fuerte que el inhibitor pero sin alcanzar el nivel de umbral entonces el potencial de la neurona se incrementa despolarizando la neurona y creando un potencial excitatorio postsináptico.
- Si el efecto excitador es mayor que el inhibitor y además igual o mayor que el nivel umbral entonces el resultado es el desencadenamiento de un impulso nervioso.

Los impulsos nerviosos son transmitidos a lo largo de la neurona con una intensidad máxima y constante en las condiciones prevaletentes. La velocidad de transmisión de impulsos nerviosos es independiente de la intensidad del estímulo, sin embargo, la velocidad sí puede ser alterada por diversos factores externos, como la presencia de materiales tóxicos en las células, la fatiga y las enfermedades. Una neurona puede conducir desde unos 10 hasta unos 2 500 impulsos por segundo dependiendo del diámetro de su axón, entre más grueso es el axón mayor es el número de impulsos por segundo que puede transmitir.

Los impulsos nerviosos son conducidos unidireccionalmente en la sinapsis, desde el axón presináptico hasta el receptor postsináptico. En otras palabras, los impulsos siempre viajan en una dirección y no pueden regresar a otra neurona presináptica. Tal mecanismo evita que los impulsos sigan direcciones inadecuadas.

2.5 SENSITIVIDAD

Una sensación es la percepción de las impresiones derivadas de la presencia de un estímulo externo o interno. La percepción es el proceso mental por el que se reconoce la naturaleza del objeto que produce el estímulo. Las sensaciones conscientes o percepciones ocurren en las áreas corticales del encéfalo, en otras palabras, vemos, oímos, degustamos, olfateamos y sentimos tácilmente en el encéfalo, si bien en apariencia vemos con los ojos, escuchamos con los odos, degustamos con la boca, olfateamos con la nariz y sentimos en la región en la que se presenta un estímulo. La proyección es el proceso mental realizado en el encéfalo consistente en referir las sensaciones al sitio en que ocurrió la estimulación.

La modalidad de una sensación es la forma o propiedad distinguible en la que se presenta una sensación. Las sensaciones son clasificadas de acuerdo a su modalidad en dolor, presión, tacto, equilibrio, audición, visión, olfato, gusto y posición corporal.

Las sensaciones se captan por medio de los receptores que son las terminaciones dendríticas de las neuronas aferentes. Las sensaciones se clasifican de acuerdo al receptor del que proviene la sensación en exteroceptivas, interoceptivas y propioceptivas. Las sensaciones exteroceptivas son las originadas por estímulos externos al organismo que son captadas por receptores llamados exteroceptores. Los exteroceptores son dendritas localizadas en los tejidos superficiales que transmiten sensaciones auditivas, visuales, olfatorias, táctiles, de presión, de temperatura y dolor.

Los sensaciones interoceptivas o vicerceptivas son las originadas por estímulos de las actividades viscerales que son captadas por receptores llamados interoceptores. Los interoceptores son dendritas localizadas en los vasos sanguíneos y las vísceras que transmiten sensaciones de dolor, presión, gusto, fatiga, hambre, sed, así como de náusea.

Los sensaciones propioceptivas son las originadas por estímulos concernientes a la posición de las partes del cuerpo y al movimiento que son captadas por receptores llamados propioceptores. Los propioceptores son dendritas presentes en músculos, tendones articulares y oído interno que transmiten sensaciones acerca de la tensión muscular, posición, además del equilibrio.

Durante su funcionamiento la neurona muestra fenómenos de gran relevancia para el funcionamiento del organismo en su totalidad, tales fenómenos han sido llamados de plasticidad neuronal y son la sensibilización, la habituación y la potenciación posttetánica. La sensibilización es la predisposición de una neurona a llegar al nivel umbral ante estímulos pequeños.

La transmisión en el nivel de la sinapsis depende de la producción y la liberación de las sustancias neurotransmisoras, por lo que la estimulación prolongada o frecuente de la neurona presináptica da por resultado la disminución en la cantidad disponible de dichas sustancias. Como consecuencia, la neurona postsináptica no sufre estimulación. La habituación o adaptación o fatiga sináptica es el retraimiento de una neurona para alcanzar el nivel umbral una vez que lo ha hecho en repetidas ocasiones ante un estímulo determinado. La habituación se presenta cuando por ejemplo, una persona al vestirse puede sentir la ropa que se está poniendo, sin embargo, al paso de un tiempo la sensación de la ropa desaparece. A la habituación se le atribuye un papel importante en el comportamiento de un organismo biológico, pues al parecer evita una sobrecarga de procesamiento en la red neuronal debida a la presencia de estímulos que ya han sido percibidos con anterioridad.

La potenciación posttetánica o postimagen o imagen residual es la tendencia de una neurona a alcanzar el nivel umbral por sí misma, es decir, sin estimulación alguna, que se presenta generalmente después de suministrar a la neurona un estímulo de manera continua durante un período y quitar repentinamente tal estímulo. Este fenómeno se presenta por ejemplo cuando después de observar una luz brillante se cierran los ojos y se sigue observando la luz durante algunos segundos.

Una de las sensaciones más importantes es la del dolor que cumple funciones de protección del organismo al indicar las situaciones en las que algún miembro del organismo presenta un daño. El dolor es provocado por el exceso de estimulación de un receptor; cuando los estímulos que provocan sensaciones sobrepasan cierto

nivel de intensidad producen dolor. Los receptores del dolor están presentes en casi todos los tejidos del organismo y su habituación es escasa o nula para no pasar como inadvertida una sensación que pudiera causar una lesión irreparable.

La capacidad del encéfalo para localizar el origen del dolor se relaciona con experiencias pasadas. En la mayor parte de los casos del dolor cutáneo, la sensación es referida o proyectada a su localización precisa. Sin embargo, en la mayor parte de los casos del dolor visceral, la sensación no es referida al punto de estimulación sino que se sienten superficialmente situándose al mismo nivel que el órgano estimulado. Este fenómeno recibe el nombre de dolor referido.

3. REDES NEURONALES ARTIFICIALES

3.1 MODELO DE UN NEUROIDE

Un neuroide o elemento de procesamiento o nodo, es un conjunto formado por un vector de entradas, un vector de pesos sinápticos, un valor de umbral, una función de integración y una función de transferencia. El neuroide es una unidad funcional de procesamiento que, como su propio nombre sugiere, es resultado de la abstracción fisiológica de una neurona biológica. El modelo de un neuroide es la descripción de un neuroide en base a la especificación de su vector de entradas, su vector de pesos sinápticos, su valor de umbral, su función de integración y su función de transferencia.

El vector de entradas, es un vector de n variables numéricas que representan las n señales que ingresan al neuroide para ser procesadas. En adelante t denotará al vector de entradas de un neuroide, mientras que se usará t_k para hacer referencia al k -ésimo elemento de t . Un vector de entradas, como puede verse en la Ilustración 3-1, es la abstracción de los impulsos que son transmitidos desde las n presinápsis de entrada o desde el tejido sensible hacia una neurona biológica.

El vector de pesos sinápticos, es un vector de n variables numéricas que son usadas para modular la intensidad de cada uno de los n valores correspondientes del vector de entradas. En adelante ω denotará al vector de pesos sinápticos de un neuroide, mientras que se usará ω_k para hacer referencia al k -ésimo elemento de ω . A través del vector de pesos sinápticos es posible hacer que un valor del vector de entradas represente para el neuroide un valor inhibitorio, excitatorio o nulo. Incluso mediante el vector de pesos sinápticos puede disminuirse o acrecentarse la inhibición o excitación de un valor del vector de entradas t_k con respecto a los valores restantes de t . Para hacer que un t_k sea inhibitorio basta que ω_k sea menor que cero; para que t_k sea excitatorio ω_k tiene que ser mayor que cero; para hacer nulo a t_k entonces ω_k debe ser igual a cero. Una t_k será más excitatorio o inhibitorio entre mayor o menor sea el valor de ω_k .

respectivamente. Un vector de pesos sinápticos, como puede verse en la Ilustración 3-1, es la abstracción del grado de eficacia de las sinapsis de entrada de una neurona biológica.

El valor umbral, denotado como Θ , es un valor numérico que una vez alcanzado activa al neuroide para que envíe una señal especial de salida. Como se observa en la Ilustración 3-1, el valor umbral de un neuroide es la abstracción del nivel umbral encontrado en las neuronas biológicas.

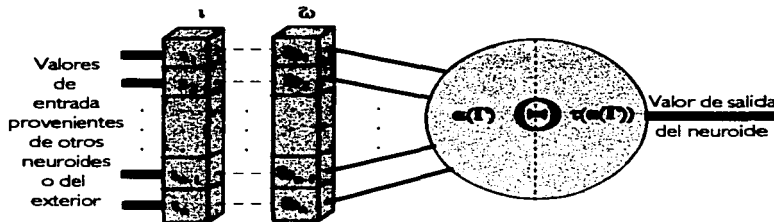


Ilustración 3-1 Representación gráfica del modelo de un neuroide

La función de integración, denotada $\alpha(\Gamma)$, es una función que integra el nivel de excitación o inhibición del neuroide en el tiempo $\Gamma-1$ con todas las señales que entran al neuroide en un tiempo Γ arrojando como resultado un valor que indica el nivel de excitación o inhibición del neuroide en el tiempo Γ . La función de integración, como puede notarse en la Ilustración 3-1, es la abstracción del proceso de sumación espacial y temporal realizada por la membrana de una neurona biológica para integrar el potencial existente de la neurona con los potenciales excitatorios e inhibitorios postsinápticos provenientes de sus sinapsis de entrada. Una infinidad de funciones pueden usarse como función de integración dependiendo del funcionamiento deseado del neuroide modelado. De ellas, la más comúnmente empleada es la función producto punto o producto interno o suma pesada entre los vectores v y w , esto es, $\alpha(\Gamma) = \sum_{i=1}^n v_i w_i$. Observe que el producto punto es una función que hace caso omiso de $\alpha(\Gamma-1)$, es decir, del nivel de excitación o inhibición del neuroide en el tiempo $\Gamma-1$. Por lo tanto, la función producto punto hace solo una sumación espacial y no temporal de las entradas al neuroide. Otro tipo de funciones pueden usarse en conjunción con el producto punto para no pasar inadvertido a $\alpha(\Gamma-1)$, de manera que exista una sumación temporal además de la espacial. Un ejemplo sencillo, podría ser $\alpha(\Gamma) = \sum_{i=1}^n v_i w_i + f(\Gamma)$, donde $f(\Gamma) = 0$ si $\alpha(\Gamma-1) \geq \Theta$, de otra forma $f(\Gamma) = \alpha(\Gamma-1)$. La sumación temporal es esencial si se desean modelar los interesantes fenómenos de plasticidad neuronal observados en los sistemas nerviosos biológicos.

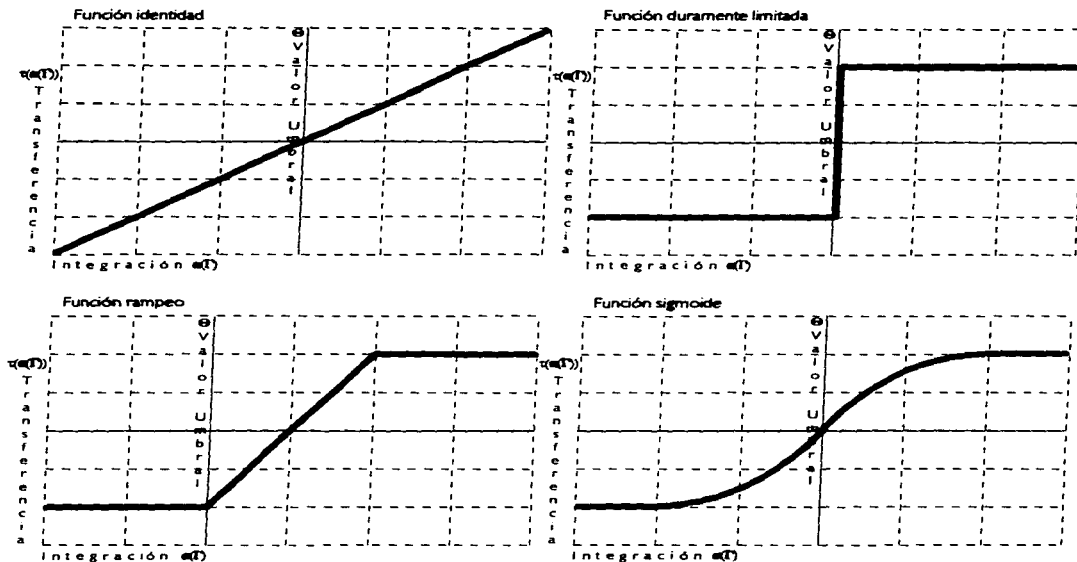


Ilustración 3-2 Gráficas del comportamiento de las funciones de transferencia comúnmente empleadas

La función de transferencia, denotada como $\tau(\alpha(\Gamma))$, es una función que calcula el valor de salida del neuroide en un tiempo Γ . La función de transferencia es generalmente una función no lineal, es decir, una función cuyo resultado no guarda proporción alguna con respecto a su parámetro. Una función de transferencia, como puede verse gráficamente en la Ilustración 3-1, es la abstracción del impulso nervioso que es creado en la zona de integración de una neurona si se rebasa el nivel umbral y que es transmitido por el axón hacia otra neurona o hacia un efector. Existe una infinidad de funciones pueden usarse para especificar la función de transferencia, las más comúnmente empleadas se describen a continuación, todas ellas son no lineales a excepción de la función identidad:

- La función identidad, definida como $\tau(\alpha(\Gamma)) = \alpha(\Gamma)$

- ➔ La función umbral (threshold), definida como $\tau(\alpha(\Gamma)) = \begin{cases} 0 & \text{para } \alpha(\Gamma) < \Theta \\ \alpha(\Gamma) & \text{de otra forma} \end{cases}$
- ➔ La función duramente limitada (hard limited), definida como $\tau(\alpha(\Gamma)) = \begin{cases} -1 & \text{para } \alpha(\Gamma) < \Theta \\ 0 & \text{para } \alpha(\Gamma) = \Theta \\ 1 & \text{para } \alpha(\Gamma) > \Theta \end{cases}$
- ➔ La función rampeo (ramping), definida como $\tau(\alpha(\Gamma)) = \begin{cases} 0 & \text{para } \alpha(\Gamma) < \Theta \\ \alpha(\Gamma) & \text{para } 1 \geq \alpha(\Gamma) \geq \Theta \\ 1 & \text{para } \alpha(\Gamma) > 1 \end{cases}$
- ➔ La función sigmoide, definida como $\tau(\alpha(\Gamma)) = \begin{cases} 1 - \frac{1}{1 + \alpha(\Gamma)} & \text{para } \alpha(\Gamma) \geq \Theta \\ -1 + \frac{1}{1 - \alpha(\Gamma)} & \text{para } \alpha(\Gamma) < \Theta \end{cases}$
- ➔ La función sigmoide logarítmica, definida como $\tau(\alpha(\Gamma)) = \frac{1}{1 + e^{-(\alpha(\Gamma) - \Theta)}}$

En la Ilustración 3-2 se muestra gráficamente el comportamiento de las funciones de transferencia descritas anteriormente. Observe que la función sigmoide, además de ser diferenciable, guarda gran similitud con el comportamiento eléctrico de las neuronas biológicas. Es por esta razón que las funciones sigmoideas son la opción comúnmente usada por la mayoría de los modelos de neuroides.

5.2 MODELO DE UNA RED NEURONAL ARTIFICIAL

Una red neuronal artificial es un conjunto formado por una función de inicialización, una función de interconexión, una función de aprendizaje, un algoritmo de entrenamiento y una n -ada no vacía de capas neuronales. La red neuronal artificial es un sistema de procesamiento de información que, como su propio nombre sugiere, es resultado de la abstracción fisiológica de las redes neuronales encontradas en los sistemas nerviosos biológicos. El modelo de una red neuronal artificial es la descripción de una red neuronal artificial en base a la especificación de su función de inicialización, su función de interconexión, su función de aprendizaje, su algoritmo de entrenamiento y su n -ada no vacía de capas neuronales.

Una n -ada de capas neuronales, denotada como U , es una n -ada en la que cada uno de sus elementos es una capa neuronal. Una capa neuronal es una n -ada en la que cada uno de sus elementos es un neuroide. Es necesario introducir la siguiente nomenclatura para lograr la exacta descripción del funcionamiento de una red:

- ➔ U_i denotará a la i -ésima capa de una red.
- ➔ U_{ij} denotará al j -ésimo neuroide de la i -ésima capa neuronal.

- $v_{i,j,k}$ denotará al k -ésimo elemento del vector de entradas del j -ésimo neuroide que está en la i -ésima capa neuronal.
- $\omega_{i,j,k}$ denotará al k -ésimo elemento del vector de pesos sinápticos del j -ésimo neuroide que está en la i -ésima capa neuronal.
- $\alpha_{i,j}(\mathbf{I})$ denotará a la función de integración del j -ésimo neuroide que está en la i -ésima capa neuronal.
- $\tau_{i,j}(\alpha_{i,j}(\mathbf{I}))$ denotará a la función de transferencia del j -ésimo neuroide que está en la i -ésima capa neuronal.

Para la especificación de una n -ada de capas neuronales de una red, deben establecerse $\{U_i\}$, $\{U_i\}$ y el modelo de neuroide usado para cada $U_{i,j}$. La $\{U_i\}$ indicará el número de capas neuronales por las que estará compuesta una red. La $\{U_i\}$ establecerá el número de neuroides por los que está compuesta cada capa neuronal. Por último el modelo de neuroide usado para cada $U_{i,j}$ describirá a cada neuroide usado en cada capa.

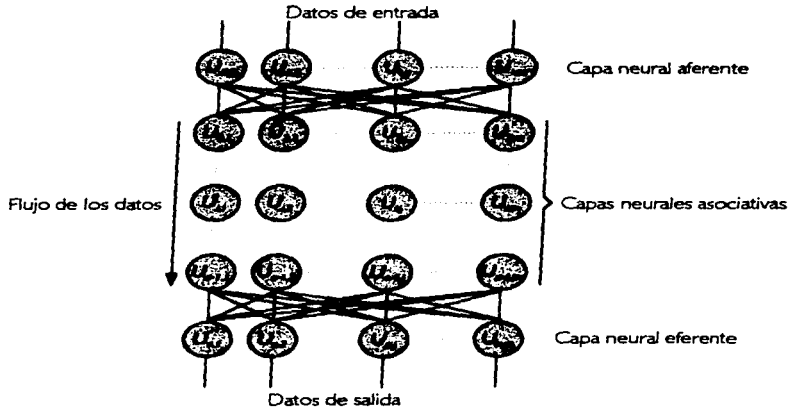


Ilustración 3-3 Esquema de una red de n capas, cada una de las cuales tiene m neuroides.

La mayoría de los modelos de redes neuronales artificiales proponen el uso de una capa neuronal aferente, una o más capas neuronales asociativas y una capa neuronal eferente. De hecho, lo más comúnmente encontrado en los modelos neuronales es que $\{U_i\} = 3$, donde U_1 es una capa neuronal aferente, U_2 es una capa neuronal asociativa y U_3 es una capa neuronal eferente. La capa neuronal aferente o capa neuronal de entrada, tiene normalmente la función de recibir datos sensoriales de entrada provenientes del exterior para toda la red y proyectarlos sobre una capa neuronal asociativa. Una capa neuronal asociativa o capa neuronal oculta o capa neuronal interna, generalmente procesa la información proveniente de una capa neuronal de

entrada o de otra capa neuronal asociativa y entrega sus resultados a otra capa neuronal asociativa o a una capa neuronal eferente. La capa neuronal eferente, o capa neuronal de salida, regularmente tiene la función de recibir los datos de una capa neuronal asociativa y entregarlos al exterior como resultado del procesamiento de la red. El cálculo de $|U|$ se ha planteado como un problema sin resolver por los modelos de redes neuronales artificiales clásicos, pues si bien es fácil deducir que se requiere una capa neuronal aferente y una capa neuronal eferente, ningún modelo de red clásico ha podido demostrar formalmente cuántas capas neuronales asociativas se requieren para garantizar la convergencia a la solución de un problema. La n -ada de capas neuronales de una red es la abstracción de la organización del sistema nervioso de los seres vivos, en los que pueden identificarse al menos tres capas neuronales interconectadas, la del sistema nervioso aferente equivalente a la capa neuronal aferente, la del sistema nervioso central equivalente a la capa neuronal asociativa y la del sistema nervioso eferente equivalente a la capa neuronal eferente. La representación gráfica de una red con la interconexión de sus capas es mostrada en la Ilustración 3-3.

El cálculo de $|U_i|$ es sencillo, pues al tratarse de una capa neuronal aferente, los modelos de redes neuronales artificiales generalmente proponen colocar un neuroide en U_i por cada dato de entrada a la red. El cálculo de $|U_n|$ no es más difícil, pues al referirse a la capa neuronal eferente, los modelos de redes neuronales artificiales comúnmente plantean situar un neuroide en U_n por cada dato de salida que debe arrojar la red como resultado de su procesamiento. Sin embargo, el cálculo de $|U_n|$ para $n > g > 1$, es un problema para el que no ha existido respuesta satisfactoria por parte de los modelos de redes neuronales artificiales clásicos, pues si bien se puede calcular $|U_1|$ y $|U_n|$ sabiendo respectivamente cuántas entradas habrá y cuántas salidas se desean, ningún modelo de red clásico ha demostrado formalmente cuántos neuroides se requieren en las capas neuronales asociativas para garantizar la convergencia a la solución de un problema.

En cuanto al modelo de neuroide usado para cada U_{ij} , los modelos de redes neuronales artificiales generalmente proponen el uso de un mismo modelo de neuroide para todos los neuroides de una misma capa neuronal U_i , sin embargo, es común que el modelo de neuroide usado en una capa neuronal aferente sea diferente al usado en las capas neuronales asociativas y el de las capas neuronales anteriores diferente al usado en una capa neuronal eferente. La diferencia más significativa entre las capas neuronales, en cuanto al modelo de neuroide usado, radica en la mayoría de los casos en la función de transferencia empleada. En las capas neuronales aferentes y eferentes se encuentra con gran frecuencia el uso de la función identidad como función de transferencia, mientras que en las capas neuronales asociativas es más usual que los modelos de redes neuronales artificiales propongan funciones sigmoides como funciones de transferencia. En ocasiones el modelo de neuroide usado para la capa neuronal aferente incluye también alguna función adicional de tratamiento o análisis de los datos de entrada, como lo puede ser una conversión de entradas analógicas a digitales o una normalización o cualquier otro tipo de transformación que facilite el procesamiento de los datos realizado por la capa neuronal asociativa. No existe modelo clásico de red alguno, que haya demostrado formalmente qué modelo de neuroide deba usarse en cada U_{ij} para garantizar la convergencia a la solución de un problema.

Una función de inicialización es una función que asigna a cada $\omega_{ij,k}$ y a cada Θ_{ij} un valor al momento de iniciar la actividad de la red, es decir, antes de comenzar cualquier ciclo de procesamiento o entrenamiento de

la red. Al establecer un valor inicial a los valores umbrales y a cada elemento del vector de pesos sinápticos de todos los neuroides de la red, la función de inicialización puede ser de gran ayuda al tratar de estabilizar el funcionamiento de la red en sus primeras iteraciones. Lo más común es encontrar que los modelos de redes neuronales artificiales proponen como función de inicialización a una función que retorne un valor aleatorio acotado entre dos límites. También es frecuente que la función de inicialización sea un valor constante, mientras que en algunos casos se hace uso de un algoritmo para calcular el valor de inicialización. No hay modelo clásico de red alguno que haya demostrado formalmente cuál debe ser la función de inicialización para garantizar la más rápida convergencia a la solución de un problema. Sin embargo, teóricamente, independientemente de la función de inicialización, una red debe converger a la solución de un problema si su función de aprendizaje y su algoritmo de entrenamiento así lo garantizan.

Una función de interconexión es una función que asigna a toda $\tau_{g,o,k}$, para $n \geq g > 1$, el valor de alguna $\tau_{i,j}(\alpha_{i,j}(\Gamma))$. La función de interconexión, al asignar la salida de un neuroide a la entrada de otro neuroide, establece un patrón de conectividad entre los neuroides distribuidos en las distintas capas neuronales de una red. Dos son las funciones de interconexión más comúnmente empleadas por los modelos de redes neuronales artificiales, la función de interconexión total y la función de interconexión recurrente. La función de interconexión total asigna a toda $\tau_{g,o,k}$, para $n \geq g > 1$, el valor de $\tau_{g-1,k}(\alpha_{g-1,k}(\Gamma))$. La función de interconexión total transmite la salida de todos los neuroides de una capa neuronal hacia una entrada de cada uno de los neuroides de la capa neuronal siguiente. La función de interconexión recurrente es una función que asigna a alguna $\tau_{g,o,k}$, el valor de alguna $\tau_{h,i}(\alpha_{h,i}(\Gamma))$, para $n \geq h \geq g > 1$. La función de interconexión recurrente transmite la salida de al menos un neuroide de una capa neuronal hacia la entrada de algún neuroide de la misma capa neuronal o de una capa neuronal previa. En la Ilustración 3-4 puede observarse una función de interconexión que puede catalogarse como localmente recurrente con interconexión total. En adelante $\tau_{(o,g)-(i,j)}$ denotará al elemento del vector de entradas del g -ésimo neuroide de la o -ésima capa neuronal cuyo valor asignado por la función de interconexión es el valor retornado por la función de transferencia el j -ésimo neuroide de la i -ésima capa neuronal. De la misma forma, $\omega_{(o,g)-(i,j)}$ denotará al elemento del vector de pesos sinápticos correspondiente a $\tau_{(o,g)-(i,j)}$. Ningún modelo de red clásico a demostrado formalmente que cierta función de interconexión garantizará la convergencia de la red a la solución de un problema. La función de interconexión es la abstracción de los axones neuronales que transmiten los impulsos nerviosos hacia sus sinapsis con otras neuronas.

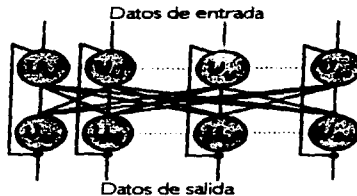


Ilustración 3-4 Red de dos capas, las cuales tienen una función de interconexión que conecta totalmente la salida de cada neuroide de la capa uno con todos los neuroides de la capa dos, pero además, la salida de cada neuroide de la capa dos se conecta a la entrada de un neuroide de la capa uno. Esta función de interconexión es un híbrido entre interconexión total y recurrente.

A través de la función de interconexión se crea una red de comunicaciones entre los neuroides de una red. La topología de una red neuronal artificial es un patrón de interconexión establecido en la red por la función de interconexión para la comunicación entre los neuroides. Las principales topologías de red propuestas por los modelos de redes neuronales artificiales son la divergente/convergente y la reverberante, mismas que pueden observarse gráficamente en la Ilustración 2-4. La topología divergente/convergente es producida por una función de interconexión total. La divergencia está presente en el hecho de que, entre cada dos capas neuronales, la salida de cada neuroide de la primera capa se asigna a una entrada de cada neuroide de la segunda capa. La convergencia radica en que, entre cada dos capas neuronales, cada neuroide de la segunda capa neuronal recibe las salidas de todos los neuroides de la primera capa. La topología reverberante es producida por una función de interconexión recurrente. Es importante observar que las redes neuronales con topología reverberante pueden tardar ciclos adicionales en obtener una salida coherente debido a la recurrencia existente en las capas neuronales. La topología reverberante es usada frecuentemente por modelos de redes neuronales artificiales que desean simular procesos de competencia y de inhibición entre neuroides.

3.5 APRENDIZAJE

Una red aprende a través de la modificación adaptativa de sus capas neuronales. A saber, las modificaciones a las capas neuronales pueden consistir en incorporar o eliminar neuroides, incorporar o eliminar conexiones entre los neuroides, alterar los umbrales de los neuroides y alterar los valores de los vectores de pesos sinápticos de los neuroides.

Tanto eliminar o incorporar neuroides, como eliminar o incorporar conexiones entre neuroides, son modificaciones adaptativas reducibles a la alteración de los vectores de pesos sinápticos de los neuroides. La incorporación o eliminación de un neuroide puede simularse incorporando o eliminando todas sus conexiones con el resto de la red. Para eliminar la conexión de un neuroide U_{ij} hacia otro neuroide U_{ox} basta con hacer nula su entrada $v_{(o,p)-(i,j)}$, lo que se simula asignando 0 al peso correspondiente $w_{(o,p)-(i,j)}$. Si se desea incorporar la conexión de un neuroide U_{ij} hacia otro neuroide U_{ox} basta con hacer excitatoria o inhibitoria su entrada $v_{(o,p)-(i,j)}$, lo que se simula asignando un valor positivo o negativo al peso correspondiente $w_{(o,p)-(i,j)}$.

Debido a lo expuesto en el párrafo anterior, los modelos clásicos de redes neuronales artificiales usan generalmente solo la modificación a los vectores de pesos sinápticos para buscar el aprendizaje de la red. Sin embargo, este puede considerarse un error grave, pues los modelos clásicos han olvidado la modificación de los valores umbrales para regular la excitabilidad de los neuroides. De hecho, se puede demostrar que existen funciones que un neuroide puede realizar modificando tanto el valor umbral como los pesos sinápticos y que no son realizables con la sola modificación a los pesos sinápticos. Tal demostración es reservada a la sección 6.3.

Una función de aprendizaje, denotada como λ , es una función que asigna un valor a cada ω_{ijk} de acuerdo a la respuesta dada y a la requerida de la red. La forma general de la función de aprendizaje es:

$$\lambda(\omega_{ijk}, \Gamma) = \bar{\omega}_{ijk} + (\Xi(\Phi(\partial(\Gamma)), \Phi(\partial(\Gamma))) \times \rho(\tau_{ij}(\alpha_{ij}(\Gamma)), \bar{\omega}_{ijk}) \times \eta)$$

donde:

- $\Xi(\Phi(\partial(\Gamma)), \Phi(\partial(\Gamma)))$ es la función de errata de la red ocurrida en el tiempo Γ
- $\partial(\Gamma)$ es el patrón de entrada presentado en el tiempo Γ
- $\Phi(\partial(\Gamma))$ es la salida deseada de la red para el patrón presentado en el tiempo Γ
- $\phi(\partial(\Gamma))$ es la salida de la red para el patrón presentado en el tiempo Γ
- $\rho(\tau_{ij}(\alpha_{ij}(\Gamma)), \bar{\omega}_{ijk})$ es la función de proporción de errata de $\bar{\omega}_{ijk}$ ocurrida en el tiempo Γ
- η es la tasa de aprendizaje de la red

La función de errata de la red ocurrida en el tiempo Γ , denotada como $\Xi(\Phi(\partial(\Gamma)), \phi(\partial(\Gamma)))$, es una función cuyo valor retornado es una medida de disimilitud entre la respuesta dada por la red $\phi(\partial(\Gamma))$ y la respuesta deseada por la red $\Phi(\partial(\Gamma))$ para el patrón de entrada $\partial(\Gamma)$. La función de proporción de errata de $\bar{\omega}_{ijk}$ ocurrida en el tiempo Γ , denotada como $\rho(\tau_{ij}(\alpha_{ij}(\Gamma)), \bar{\omega}_{ijk})$, es una función cuyo valor retornado representa la proporción o porcentaje en la que ha contribuido el peso $\bar{\omega}_{ijk}$ en el error global de la red. La tasa de aprendizaje, denotada como η , es una variable o una constante que regula que tan drástica es la modificación a los pesos sinápticos de los neuroides. Generalmente entre mayor sea la tasa de aprendizaje, el aprendizaje será más rápido pero las respuestas serán menos exactas; de la misma forma, entre menor sea la tasa de aprendizaje, el aprendizaje será más lento pero las respuestas serán más exactas.

Algunas de las funciones de aprendizaje tradicionales de los modelos neuronales son la función Hebb y la función Widrow-Hoff. La función Hebb es la forma general de la función de aprendizaje en la que definen a Ξ , ρ como $\Xi(\Phi(\partial(\Gamma)), \phi(\partial(\Gamma))) = \phi(\partial(\Gamma))$ y $\rho(\tau_{ij}(\alpha_{ij}(\Gamma)), \bar{\omega}_{ijk}) = \tau_{ij}(\alpha_{ij}(\Gamma))$. Por otra parte, la función Widrow-Hoff o función delta, es la forma general de la función de aprendizaje que define Ξ , ρ como $\Xi(\Phi(\partial(\Gamma)), \phi(\partial(\Gamma))) = \Phi(\partial(\Gamma)) - \phi(\partial(\Gamma))$ y $\rho(\tau_{ij}(\alpha_{ij}(\Gamma)), \bar{\omega}_{ijk}) = \tau_{ij}(\alpha_{ij}(\Gamma))$. Ningún modelo de red clásico ha demostrado formalmente que cierta función de aprendizaje garantizará la convergencia a la solución de un problema.

Un algoritmo de entrenamiento es una secuencia de pasos que busca el aprendizaje de la red. No hay un solo modelo de red clásico haya propuesto un algoritmo de entrenamiento que garantice la convergencia a la solución de un problema. De hecho, existe el consenso de que no se sabe aún a ciencia exacta cómo es que

ocurre el aprendizaje de una red. El algoritmo de entrenamiento para una red consta generalmente de los siguientes pasos:

1. Inicializar el vector de pesos sinápticos de todos neuroides a través de la función de inicialización
2. Haz
 - 2.1. Presentar un patrón de entrada a la capa neuronal de aferente
 - 2.2. Calcular las funciones de transferencia de todos los neuroides de la red
 - 2.3. Obtener el patrón de salida de la red de la capa neuronal eferente
 - 2.4. Aplicar la función de aprendizaje a cada uno de los neuroides de la red
3. Hasta que se cumpla el criterio de parada

Los criterios de parada de un algoritmo de entrenamiento son variados y dependientes de la naturaleza del problema a resolver. Los más comúnmente empleados son:

- Cuando se ha llegado a un número de iteraciones.
- Cuando se han presentado todos los patrones de entrada a la red en cierto número de ocasiones.
- Cuando se ha logrado minimizar cierta función de error.
- Cuando la red responde correctamente a un porcentaje determinado de los patrones de entrada.

De los criterios de parada anteriores, el más usado es el último. En algunos casos se exige que la red termine su entrenamiento cuando haya sido capaz de responder correctamente a todos los patrones de entrada.

3.4 GESTA DE LAS REDES NEURONALES ARTIFICIALES

Los intentos por comprender el funcionamiento del cerebro se remontan a siglos atrás. Existen referencias a los trabajos realizados por Edward Smith Papyrus (3000 A.C.) e Hipócrates (377 A.C.) acerca del cerebro [Fischler, Firschein 1987]. Presuntamente, el de Smith es un trabajo que intenta describir la localización de ciertas áreas de control motor y sensitivo en el cerebro. Es más reciente el trabajo de William James, que en 1890 publica "Psychology (Brief Course)" que contiene algunos presagios de las teorías actuales acerca del funcionamiento del cerebro.

En 1943, Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático de dieciocho años, escribieron un artículo acerca del funcionamiento de las neuronas a través de circuitos eléctricos [McCulloch, Pitts 1943]. El de McCulloch y Pitts era el modelo de una neurona como unidad de procesamiento con varias entradas y una sola salida, semejando un interruptor digital. Pocos años después, en 1949, Donald Hebb publica "The organization of Behavior", el cual sugiere que una ruta en una red neuronal es reforzada cada vez que es usada, lo que se conoce como la Regla de Aprendizaje de Hebb y que sigue siendo usada en algunas redes en nuestros días.

En los 50's el desarrollo de la tecnología en el hardware y software hizo posible la simulación de las teorías acerca del funcionamiento del sistema nervioso. Nathaniel Rochester y otros de los laboratorios de

investigación de IBM crearon una red basada en el trabajo de Hebb. El primer intento falló, sin embargo con la colaboración de Hebb y otros investigadores se hicieron adaptaciones exitosas.

Fue en 1956, durante el Dartmouth Summer Research Project on Artificial Intelligence, el momento que puede considerarse como el nacimiento de las redes neuronales artificiales como campo de estudio de la inteligencia artificial. Frank Rosenblatt, en 1957, inició su trabajo del perceptrón, un modelo de red construido en hardware. El perceptrón es de hecho la red construida más antigua y se sigue usando en algunas aplicaciones. En 1958, John von Neuman publicó su libro "The Computer and the Brain" [Newman 58], donde sugiere imitar las funciones de una neurona usando relevos telegráficos o tubos de vacío.

En 1959, Bernard Widrow y Marcian Hoff, de Stanford en los Estados Unidos de América (EUA), desarrollaron ADALINE (ADaptive LINear Elements). Es esta la primera red aplicada a un problema real, la creación de filtros adaptativos para eliminar ecos en las líneas telefónicas. Esta aplicación ha sido comercialmente empleada por décadas.

Stephen Grossberg, de la Boston University, crea en 1967 su red llamada Avalanche, usada para realizar actividades de reconocimiento de lenguaje y enseñanza de comandos para mover los brazos de robot a través de motores.

En 1969 Marvin Minsky y Seymour Papert publicaron "Perceptrons" [Minsky, Papert 69], un libro de gran influencia que mostraba limitantes en el Perceptron de Roseblatt. Los inconvenientes mostrados por Minsky y Papert hacían parecer que la línea de investigación en redes neuronales artificiales no llevaría a resultados interesantes. Muchos investigadores decidieron entonces buscar en otros campos de la inteligencia artificial que parecían más promisorios, mientras que los obsesionados continuaron esforzándose en el área de redes neuronales artificiales. James Anderson, un neurofisiólogo de la Brown University, desarrolló un modelo llamado Brain-State-in-a-Box (BSB). En Japón, Kunihiko Fukushima desarrolló Neocognitrón, un modelo de red para el reconocimiento de patrones visuales. Teuvo Kohonen un ingeniero eléctrico de la Helsinki University desarrolló un modelo similar al de Anderson pero de manera independiente. Otros como Grossberg, Rumelhart, McClelland, Marr, Poggio, Amari y Cooper continuaron trabajando en sus laboratorios.

Fue hasta 1982, después de la severa crítica de "Perceptrons", cuando se pudo distinguir un resurgimiento en el interés por las redes neuronales artificiales. John Hopfield presentó, a la National Academy of Sciences de los EUA un artículo sobre su modelo de red. El modelo de Hopfield resulta de gran interés pues, a diferencia de los modelos anteriores, se preocupa más por el uso práctico de la red que por la simulación del sistema nervioso de los entes biológicos.

También en 1982 se celebró en Kyoto, Japón, la "U.S.-Japan Joint Conference on Cooperative/Competitive Neural Networks". Poco después Japón anunció su programa "Fronteras Humanas", también conocido como "Sexta Generación". Fijitsu comenzó a desarrollar sus "computadoras pensantes" para aplicaciones robóticas. En 1985 el American Institute of Physics en los EUA inició su encuentro anual "Neural Networks for Computing".

En 1987 el Institute of Electrical and Electronics Engineers' (IEEE) celebró la "First International Conference on Neural Networks". El mismo año se creó la International Neural Network Society (INNS), liderada por Grossberg en los EUA, Kohonen en Finlandia, y Amari en Japón.

Para 1989 había por lo menos 160 proyectos de redes neuronales artificiales aplicados a sistemas militares de defensa apoyados por distintos gobiernos en el mundo. En 1990 el Departamento de Defensa de los EUA anunció su "Small Business Innovation Research Program" con 16 proyectos dedicados a las redes neuronales artificiales, mientras que otros 13 mencionaban la posibilidad de usar redes.

Nombre	Autor	Año	Descripción
Supresor de ruido adaptativo	Bernard Widrow	1960-62	Elimina ecos en las líneas telefónicas y reduce los errores de transmisión en los modems [Anderson, Rosenfeld 1988].
Reconocedor de palabras	Intel	1983	Reconoce ciertas palabras o frases expresadas verbalmente por un humano. El sistema distingue a un solo parlante a la vez y está limitado a aproximadamente un ciento de palabras o frases aisladas [DARPA 1988].
NETtalk	Sejnowsky, Rosenberg	1986	Lee textos introducidos en voz alta [Sejnowsky, Rosenberg 86].
Clasificador de sonar	Paul Gorman de Bendix Aerospace	1987	Detecta minas, rocas o blancos bajo el mar a través de las señales emitidas por un sonar [Kinoshita, Palevsky 1987].
Analizador térmico de neutrones	Science Applications International Corporation	1989	Detecta bombas ocultas en equipaje. Se le ha entrenado para detectar materiales explosivos a través de la identificación de patrones de emisiones de rayos gamma. Ha sido usado por la Administración Federal de Aviación en el Aeropuerto Internacional JFK [Johnson 1989].
Revisor de motores	Siemens	1989	Escucha el funcionamiento de motores para detectar aquellos que resultan demasiado ruidosos [O'Reilly 1989], [Schwartz 1989].
Estrategia mercadotécnico	BehavHeuristics	1989	Da consejos a la administración de una línea aérea en base a predicciones en el comportamiento de la empresa [Green Lee 1988], [Edventures 1988].

Tabla 3.1 Algunas aplicaciones comerciales exitosas de las redes neuronales artificiales

La investigación existente al rededor del sistema nervioso de los seres biológicos sigue acrecentándose día a día, de manera que se ha ido agrupando a diversas áreas con puntos de interés diferentes pero complementarios entre sí. Los médicos y neurólogos se interesan en la fisiología, composición química y la biología molecular. Los biólogos se inclinan por saber más acerca de los sistemas adaptativos y los métodos de auto-organización de los seres vivos. Por último, los investigadores de la Inteligencia Artificial desean encontrar mecanismos de aprendizaje y creatividad que sean aplicables a la resolución de un problema del mundo real.

No son muchas las aplicaciones comerciales basadas en redes neuronales al momento de escribir estas líneas, algunas de las más exitosas son presentadas de manera resumida en la Tabla 3.1. Entre las áreas que se han señalado como promisorias para la aplicación de las redes neuronales artificiales están el reconocimiento de imágenes, reconocimiento continuo de voz, procesamiento de lenguaje natural, transformaciones de texto a voz, reconocimiento de caracteres hechos a máquina y a mano simples o complejos, aplicaciones militares, compresión de datos y optimización problemas NP (No Polinomiales), es decir, de explosión combinatoria.

Para los problemas NP una red tal vez no halle la mejor solución, pero se han mostrado proyectos en los que al menos sí puede encontrar una solución bastante sensata.

3.5 REDES NEURONALES ARTIFICIALES EN COMPARACIÓN A OTRAS TÉCNICAS

En la Inteligencia Artificial (IA), área de las ciencias de la computación concerniente al diseño de sistemas computarizados que exhiben alguna forma de inteligencia, pueden distinguirse dos vertientes principales para la representación del conocimiento, el desarrollo de sistemas de representación simbólica y el desarrollo de sistemas de representación biótica. Ambas áreas tienen como punto en común su interés en modelar artificialmente el razonamiento y el deseo de conocer la naturaleza de la inteligencia humana. Dentro del campo de los sistemas bióticos se encuentran las redes neuronales artificiales y los algoritmos genéticos, mientras que en el campo de sistemas simbólicos están representaciones del conocimiento bastante estudiadas como redes semánticas, frames, guiones, lógica y otras. La diferencia entre las dos áreas no solo radica en la forma en que se representa el conocimiento sino también en la metodología de programación. Una descripción más detallada acerca del funcionamiento de los sistemas de representación simbólica y su diferencia con respecto a las redes neuronales artificiales en específico puede encontrarse en la Tabla 3.2.

Sistemas de representación simbólica	Redes Neuronales
Ponen más atención a la conducta cerebral	Ponen más atención a la fisiología cerebral
Están más fundamentadas en la psicología	Están más fundamentadas en la neurología
Se basan más en qué es lo que el cerebro hace	Se basan más en cómo es que el cerebro hace algo
El aprendizaje es didáctico, es decir, por reglas	El aprendizaje es socrático, es decir, por ejemplos.
Generalmente datos incompletos no dan un resultado	Generalmente datos incompletos dan una salida coherente
Generalmente se procesa una sola hipótesis a la vez	Pueden procesarse múltiples hipótesis simultáneamente
Su memoria es direccionada	Su memoria es asociativa
Su elemento de procesamiento es la máquina de Von Neuman	Su elemento de procesamiento es un neuroide
La tolerancia a fallas de un elemento de procesamiento es baja o nula	La tolerancia a fallas de un elemento de procesamiento es alta
Son programados de manera explícita, esto es, por el programador	Son programadas de manera implícita, esto es, la red se autoprograma
Se expresa cómo es que debe hacerse cada tarea para obtener un resultado	Se expresa cuál es el resultado que se desea y no cómo debe obtenerse
Una aplicación representa la lógica y conocimientos adquiridos por programador	Una aplicación representa la lógica y conocimientos adquiridos por la red
Generalmente se deben especificar todas las situaciones a las que se puede enfrentar un programa.	No es necesario especificar todas las situaciones a las que se enfrentará una red.
El procesamiento es inherentemente serial aunque muchas tareas pueden paralelizarse.	El procesamiento es inherentemente paralelo aunque muchas tareas pueden serializarse.
Representa grandes ventajas cuando se trata con problemas que requieren análisis numérico, se conocen exactamente las reglas que deben realizarse para resolver una tarea, el conjunto de reglas no es significativamente grande y el problema a resolver no tiene una explosión combinatoria.	Representa grandes ventajas cuando se trata con problemas que requieren de visión, reconocimiento continuo de voz, síntesis de voz, aprendizaje de máquinas, reconocimiento de objetos. También si existen límites teóricos de velocidad para resolver un problema impuestos por un solo procesador, si es difícil establecer las reglas para resolver una tarea; si los algoritmos para resolver un problema son demasiado complejos para realizarlos en la práctica; si el conjunto de reglas para resolver un problema es demasiado grande.

Tabla 3.2 Funcionamiento y diferencias entre las técnicas de representación simbólica y las redes neuronales artificiales.

El que existan diferencias entre dos técnicas de la IA no significa que sean excluyentes una a la otra. Un proyecto de la IA puede enriquecerse definitivamente de los múltiples enfoques que pueden usarse sabiéndolos conjugar de manera que se logre que el producto final sea verdaderamente poderoso. Las redes neuronales artificiales y las demás técnicas de la IA no son rivales sino complementos uno del otro. Algunas características observadas en proyectos de investigación como inherentes a las redes neuronales artificiales y que pueden ser de gran interés para el desarrollo de sistemas inteligentes son:

- **Procesamiento paralelo.** Todos los neuroides de una red son susceptibles de trabajar en forma paralela al igual que sucede en las redes neuronales.
- **Procesamiento distribuido.** La computación está distribuida en más de un elemento de procesamiento que tiene un número potencialmente elevado de canales de comunicación con otros elementos de procesamiento.
- **Almacenamiento distribuido del conocimiento.** Aunque pueden destinarse premeditadamente ciertas áreas dedicadas a almacenar determinados datos, el conocimiento puede verse no como almacenado en un solo neuroide, sino más bien como diseminado en la totalidad de la red.
- **Acceso asociativo a la memoria.** Los métodos digitales convencionales de acceso a memoria usados en las computadoras se alimentan de direcciones retornando los datos que son almacenados en un lugar específico y único correspondiente a tal dirección. Por el contrario, de acuerdo a la forma en que trabaja una red se le podría clasificar como asociativa, es decir direccionada por el contenido, pues se alimenta de datos mediante los cuales se retornan otros datos.
- **Tolerancia a fallas.** Si un neuroide falla, los neuroides restantes de la red pueden seguir funcionando y emitiendo respuestas, aunque obviamente tal vez no con la misma fiabilidad o grado de certeza.
- **Generalización.** Una red puede responder a entradas parecidas a las presentadas anteriormente, característica que las hace realmente eficientes para la generalización de respuestas a patrones que incluso pudieron jamás haber sido vistos antes por la red.
- **Intuición.** Las redes neuronales artificiales son capaces de responder a patrones incompletos, difusos, ambiguos o con gran cantidad de ruido.
- **Adaptabilidad.** En ocasiones el entorno de trabajo cambia de manera que se debe responder a un patrón de entrada de manera diferente a la que se había respondido antes. Una red es capaz de olvidar ciertas respuestas aprendidas y volver a aprender a responder correctamente a un patrón de entrada.

3.6 MODELOS CLÁSICOS

De los modelos de redes neuronales artificiales que han sido creados hasta la fecha, solo unos cuantos se usan comúnmente. A estos les hemos llamado modelos clásicos y aparecen mencionados en la Tabla 3.3.

Nombre	Autor	Año	Descripción
Perceptron	Frank Rosenblatt	1957	Usado para reconocimiento de caracteres. No puede reconocer patrones complejos. Es la red más antigua. Fue hecho en hardware.
MADALINE	Bernard Widrow	1960-62	Elimina interferencias producidas en transmisiones de radares, modems y teléfonos. Ha sido comercializado por más de veinte años.
Avalanche	Stephen Grossberg	1967	Utilizado para el reconocimiento continuo de voz y control de brazos de robot
Cerebelatron	David Marr, James Albus, Andres Pellionez	1969-82	Aplicado en control de robots.
Back Propagation	Paul Werbos, David Parker, David Rumelhart	1974-85	Empleado en control de robots y síntesis de voz a partir de texto. Es la red más popular. Aprende fácil.
Brain State in a Box	James Anderson	1977	Extrae conocimiento en bases de datos.
Neocognitron	Kunihiko Fukushima	1978-84	Reconocimiento de caracteres escritos por humanos. Requiere muchos neuroides y sinapsis. Bastante complejo.
Adaptive Resonance Theory	Gail Carpenter, Stephen Grossberg	1978-86	Reconocimiento de patrones. Es sofisticado y no muy aplicado hoy en día.
Self Organizing Map	Teuvo Kohonen	1980	Requiere de mucho entrenamiento. Usado en estudios de flujo aerodinámico donde ha probado ser mejor que muchos algoritmos
Hopfield	John Hopfield	1982	Recupera datos a través de fragmentos de información. No aprende.
Bidirectional Associative Memory	Bart Kosko	1985	Utilizado como memoria asociativa, es decir, direccionable por el contenido. Puede almacenar una gran cantidad de datos. Los datos aprendidos deben ser previamente codificados. Aprende fácilmente.
Boltzmann & Cauchy Machines	Jeffrey Hinton, Terry Sejnowsky, Harold Szu	1985-86	Reconoce de imágenes de sonar y radar. Requiere un largo entrenamiento.
Counter Propagation	Robert Hecht-Nielsen	1986	Aplicado en la compresión de imágenes y análisis estadístico. Usa una gran cantidad de neuroides y sinapsis.

Tabla 3.3 Principales modelos clásicos.

El Perceptron fue el primer modelo de red, creado por Frank Rosenblatt en 1958, un neurofisiólogo que había sido antes psicólogo. En términos generales el modelo tiene dos capas totalmente conectadas, sus salidas asumen únicamente valores 0 o 1. Su función de transferencia es lineal, por lo que otros investigadores demostraron que las dos capas podrían convertirse en una sola capa. Sus principales aplicaciones son clasificación de figuras, reconocimiento de caracteres y sistemas de visión de robots. Su gran desventaja es que solo reconoce patrones linealmente separables, situación conocida también como el problema XOR, precisamente porque no podía resolver esta función lógica. Originalmente el Perceptron era un dispositivo físico, es decir hardware, que intentaba modelar computacionalmente las funciones de la retina para el reconocimiento de patrones visuales. Rosenblatt creía que el sistema debería auto-organizarse, manejar entradas variables con ruido y asociar estímulos con respuestas. El Perceptron podía generalizar y responder a estímulos que no había visto antes si el patrón de entrada era similar a algún patrón aprendido anteriormente. Sin embargo, los problemas debían ser linealmente separables para ser clasificados. Fue fuertemente criticado por inflexible hasta el punto de ser llamado el modelo de un "cerebro dañado" que no podía hacer nada interesante. El perceptron es un modelo con limitaciones descritas detalladamente por Minsky y Papert [Minsky, Papert 1969], pero con el gran mérito de ser un trabajo pionero que acaparó la atención de una gran cantidad de investigadores y que sirvió de inspiración a otros nuevos modelos de redes neuronales artificiales.

El **ADaptive LINear Elements (ADALINE)** y el **Multiple ADALINE's (MADALINE)** es un modelo creado por **Bernard Widrow** y **Marcian Hoff**. **Widrow** y **Hoff** introdujeron la idea de un umbral en todas las capas. Su modelo constaba de una red de tres capas. Cada neurona computaba la suma pesada de sus entradas multiplicada por los pesos sinápticos correspondientes más un término de tendencia. Su función de transferencia es una función duramente limitada. La regla de aprendizaje se aplicaba aún cuando la respuesta era correcta. El algoritmo de detección y corrección de errores es conocido como la **Regla Widrow-Hoff** o el algoritmo **LMS (Least Mean Squares)**.

El modelo de red **Brain-State-in-a-Box (BSB)**, propuesto por **James Aderson** un psicólogo con orientaciones neurofisiológicas, usa retroalimentación con corrección de errores. Se ha usado para clasificaciones de aves y diagnósticos médicos de enfermedades en base a síntomas adolecidos por el paciente. Sus capas no son de conexión total. La salida usa una función de transferencia de rampeo. El **BSB** es tolerante a fallas y ruido pero requiere mucho tiempo de supervisión para el entrenamiento.

La red de **Hopfield**, creada por **Jhon Hopfield**, un físico de los laboratorios **Bell** de **AT&T**, es la conceptualización de una red en términos de su energía y de sistemas físicos de dinámica. Un elemento de procesamiento puede cambiar su estado si haciéndolo reduce el "nivel de frustración" de la red. La salida de cada neuroide es binaria. La red es totalmente conectada, con recurrencias, y funciones de transferencia no lineales. La capacidad de almacenamiento es aproximadamente 15% menor al número de elementos, lo que es visto como una limitante importante del modelo. La red de **Hopfield** fue creada en un circuito integrado, lo que hizo que fuera bien recibida por la comunidad.

La red de **Back Propagation** fue originada de manera independiente y aproximadamente al mismo tiempo por **Werbos**, **Parker**, **Rumelhart**, **Hinton** y **Williams**. Es el modelo de red más popular y que ha probado mayor efectividad. Aunque se trabaja aún en hacerla veloz, resulta generalmente más rápida que otros modelos. Sin embargo, curiosamente el procedimiento de aprendizaje de la **Back Propagation** no está aún bien entendido y no hay garantía de convergencia de la red a la solución del problema. Se ha usado para leer de texto, control de brazos de robot y pronósticos. Esencialmente es un modelo refinado **Widrow-Hoff** que calcula la diferencia entre las salidas actuales y las deseadas. Usando este error los pesos son cambiados en proporción al error multiplicado por la señal de entrada. Si un nodo no estuvo activo durante el ciclo significa que no contribuyó al error por lo que no es necesario cambiar sus pesos. Cada ciclo tiene dos pasadas, una pasada hacia adelante para estimar el error y entonces una pasada hacia atrás para modificar los pesos de manera que el peso se decremente.

Los **Mapas Auto-organizativos** fueron creados por **Tuevo Kohonen** un ingeniero electromecánico de la **Universidad de Helsinki** de **Tecnología**. Su red aprende sin una señal de supervisión. Modela muy de cerca al sistema neurobiológico. Los pesos sinápticos deben ser inicializados y las entradas deben ser normalizadas. Los neuroides compiten por el privilegio de aprender, el ganador toma todo y los vecinos son ajustados. Es muy rápido, pues sus respuestas son en tiempo real, puede aprender constantemente adaptándose a los cambios. Esta red tiene una sola capa recurrente y altamente conectada. Su rendimiento a demostrado ser mejor que otros algoritmos para realizar cálculos acerca del flujo aerodinámico.

4. EL NEUROIDE LÓGICO

4.1 MODELO

Un **neuroide lógico** (NL) es un neuroide que cumple con el siguiente modelo de neuroide:

1. Tiene un vector de entradas \mathbf{t} cuyos n elementos son variables booleanas, es decir, $t_i \in \{0, 1\}$.
2. Tiene un vector de pesos sinápticos \mathbf{w} de n elementos en los que cada w_i puede asumir solo los valores de -1 para hacer un valor t_i inhibitorio, 0 para hacer un valor t_i nulo, o bien 1 para hacer un valor de t_i excitatorio.
3. Su valor umbral es un número entero.
4. Su función de integración se define como $\alpha(\Gamma) = \sum_{i=1}^n t_i w_i + \zeta(\Gamma)$, donde ζ es una función de sumación temporal.
5. Su función de transferencia es una función lógica, es decir, $\tau: (\alpha(\Gamma)) \rightarrow \{0, 1\}$, que se define como

$$\tau(\alpha(\Gamma)) = \begin{cases} 0 & \text{para } \alpha(\Gamma) < \Theta \\ 1 & \text{de otra forma} \end{cases}$$

La función de sumación temporal $\zeta(\Gamma)$, tal como se describió en la sección 3.1, es esencial para el modelado de los fenómenos de plasticidad neuronal. Aunque se tienen en este momento propuestas de funciones a usar como $\zeta(\Gamma)$, se ha optado dejar los detalles de $\zeta(\Gamma)$ para futuros reportes de investigación, de manera que en este texto se resalte ante todo el mecanismo del NL. Por ello, en lo siguiente se asumirá que $\zeta(\Gamma) = 0$ por lo que habrá solo una sumación espacial.

Con respecto al valor arrojado por la función de transferencia $\tau(\alpha(\Gamma))$, para fines de este trabajo se ha asumido que sus valores resultantes solo pueden ser 0 o 1, es decir valores discretos. Sin embargo, tal y como se describirá en la sección 6.1, próximas investigaciones estarán relacionadas a comprobar que el comportamiento y propiedades del neuroide lógico permanecen sin cambio si la $\tau(\alpha(\Gamma))$ retorna valores reales comprendidos entre el 0 y el 1, lo que puede resultar valioso para el tratamiento directo de problemas analógicos.

Un NL puede verse de manera global como una función lógica la cual recibe como parámetros las n variables booleanas de entrada, a partir de las cuales se retorna un valor booleano. Las n variables booleanas son captadas por los n elementos de \mathbf{v} , para ser entonces procesadas por las funciones de activación y transferencia dando como resultado final un valor booleano.

En adelante \uparrow denotará al valor máximo de integración que puede retornar la función de integración $\alpha(\Gamma)$, mientras que \downarrow denotará al valor mínimo de integración que puede retornar $\alpha(\Gamma)$. También se denotará como Υ al conjunto de todos los v_i de un NL que tienen un peso sináptico $\omega_i = 1$, mientras que $\neg\Upsilon$ denota al conjunto de todos los v_i de un NL que tienen un peso sináptico $\omega_i = -1$. Observe que Υ es el conjunto de todas las entradas excitatorias a un NL, mientras $\neg\Upsilon$ es el conjunto de todas las entradas inhibitorias a un NL. Finalmente, Υ' denotará al conjunto formado por los elementos de Υ que son iguales a 1, Υ^0 denotará al conjunto formado por los elementos de Υ que son iguales a 0, $\neg\Upsilon'$ denotará al conjunto formado por los elementos de $\neg\Upsilon$ que son iguales a 1, $\neg\Upsilon^0$ denotará al conjunto formado por los elementos de $\neg\Upsilon$ que son iguales a 0. Es obvio que $\Upsilon' \subseteq \Upsilon$, $\Upsilon^0 \subseteq \Upsilon$, $\neg\Upsilon' \subseteq \neg\Upsilon$ y $\neg\Upsilon^0 \subseteq \neg\Upsilon$.

Podemos ahora especificar a la función de integración de un NL también como $\alpha(\Gamma) = |\Upsilon'| - |\neg\Upsilon'|$, que es equivalente a $\alpha(\Gamma) = \sum_{i=1}^n v_i \omega_i$ debido a la definición de un NL. Con la especificación anterior, queda claro que el valor máximo de integración que es retornado por la función de integración, se alcanza cuando todas las entradas excitatorias son iguales a 1 y todas las entradas inhibitorias iguales a 0. En otras palabras, $\uparrow = |\Upsilon'|$, pues el valor máximo que puede asumir $|\Upsilon'|$ es $|\Upsilon|$ cuando $\Upsilon' = \Upsilon$, es decir, cuando todas las entradas excitatorias son iguales a 1; y por el otro lado, el valor mínimo que puede asumir $|\neg\Upsilon'|$ es 0 cuando $\neg\Upsilon' = \emptyset$ porque $\Upsilon^0 = \Upsilon$, esto es, cuando no existen entradas inhibitorias iguales a uno porque todas las entradas inhibitorias son iguales a 0. Además, es evidente que el valor mínimo de integración que es retornado por la función de integración $\alpha(\Gamma)$ se alcanza cuando todas las entradas inhibitorias son iguales a 1 y todas las entradas excitatorias iguales a 0. En otras palabras, $\downarrow = -|\neg\Upsilon'|$, pues el valor máximo que puede asumir $|\neg\Upsilon'|$ es $|\neg\Upsilon|$ cuando $\neg\Upsilon' = \neg\Upsilon$, es decir, cuando todas las entradas inhibitorias son iguales a 1; y por el otro lado, el valor mínimo que puede asumir $|\Upsilon'|$ es 0 cuando $\Upsilon' = \emptyset$ porque $\Upsilon^0 = \Upsilon$, esto es, cuando no existen entradas excitatorias iguales a 1 porque todas las entradas excitatorias son iguales a cero.

4.2 PROPIEDADES CON RESPECTO A ∇

Un neuroide lógico ∇ o neuroide lógico disyuntivo, es un neuroide lógico en el que se cumple que $\Theta = \downarrow + 1$, es decir, que el valor umbral sea igual a uno menos el número de entradas inhibitorias al NL. Observe que si la función de integración $\alpha(\Gamma)$ de un ∇ es menor que el valor umbral Θ entonces significa que $\alpha(\Gamma)$ es igual a su valor mínimo de integración \downarrow , esto es debido a que por su definición el umbral es más grande por 1 que el valor mínimo de la función de integración. La propiedad anterior se demuestra formalmente en el Lema 4.1.

Lema 4.1 Integración de ∇ . Si $\alpha(\Gamma) < \Theta$ entonces $\alpha(\Gamma) = \downarrow$

Demostración.

1. $\alpha(\Gamma) < \Theta$
2. $\Theta = \downarrow + 1$
3. $\alpha(\Gamma) < \downarrow + 1$
4. $\alpha(\Gamma) = \downarrow$

Hipótesis

Por definición de ∇

De (2) substituyendo en (1)

De (3) pues \downarrow es el valor mínimo de $\alpha(\Gamma)$

En adelante a un neuroide lógico disyuntivo con n entradas se le denotará como $\bigvee_{i=1}^n v_i$, donde obviamente cada v_i es una variable booleana por definición de NL. Un ∇ es capaz de simular el funcionamiento de un operador \vee sobre un conjunto de literales creadas a partir de sus variables booleanas de entrada v_i . Para que un ∇ procese una literal creada a partir de v_i , simplemente se asigna -1 al peso sináptico \bar{w}_i si está negada la variable booleana v_i , y 1 al peso sináptico \bar{w}_i si no está negada la variable booleana v_i . Formalmente, si ($\bar{w}_i = -1$ si $l_i = \neg v_i$; y $\bar{w}_i = 1$ si $l_i = v_i$) entonces $\bigvee_{i=1}^n l_i = \bigvee_{i=1}^n v_i$. Lo planteado anteriormente es demostrado en el Teorema 4.1, en el que se usará Θ' , $\alpha(\Gamma)'$, y \downarrow' para denotar respectivamente al umbral, función de integración y valor mínimo de integración en $\bigvee_{i=1}^n l_i$.

Teorema 4.1 Transferencia de ∇ . Si ($\bar{w}_i = -1$ si $l_i = \neg v_i$; y $\bar{w}_i = 1$ si $l_i = v_i$) entonces $\bigvee_{i=1}^n l_i = \bigvee_{i=1}^n v_i$

Demostración.

1. $\bar{w}_i = -1$ si $l_i = \neg v_i$; y $\bar{w}_i = 1$ si $l_i = v_i$
2. si $l_i = \neg v_i$, entonces $\bar{w}_i = -1$
3. si $l_i = v_i$, entonces $\bar{w}_i = 1$
4. $\bigvee_{i=1}^n l_i = \bigvee_{i=1}^n v_i$

Hipótesis

De (1) de la conjunción de la parte

De (1) de la conjunción de la parte

Por inducción sobre n

- 4.1. Si $n = 1$ entonces $\bigvee_{i=1}^n l_i = \bigvee_{i=1}^n v_i$

4.1.1. $n = 1$

Hipótesis

4.1.2. $l_1 = v_1 \circ l_1 = \neg v_1$

Por definición de l

4.1.3. Si $l_1 = v_1$ entonces $\bigvee_{i=1}^1 l_i = \bigvee_{i=1}^1 l_i$:

4.1.3.1. $l_1 = v_1$

Hipótesis

4.1.3.2. $\bar{\omega}_1 = 1$

De (4.1.3.1) por (3)

4.1.3.3. $\downarrow = 0$

De (4.1.1) y (4.1.3.2) por definición de \downarrow

4.1.3.4. $\Theta = 1$

De (4.1.3.3) por definición de ∇

4.1.3.5. $\bigvee_{i=1}^1 l_i = \bigvee_{i=1}^1 l_i$

Por Tabla 4.1

4.1.4. Si $l_1 = \neg v_1$ entonces $\bigvee_{i=1}^1 l_i = \bigvee_{i=1}^1 l_i$:

4.1.4.1. $l_1 = \neg v_1$

Hipótesis

4.1.4.2. $\bar{\omega}_1 = -1$

De (4.1.4.1) por (2)

4.1.4.3. $\downarrow = -1$

De (4.1.1) y (4.1.4.2) por definición de \downarrow

4.1.4.4. $\Theta = 0$

De (4.1.4.3) por definición de ∇

4.1.4.5. $\bigvee_{i=1}^1 l_i = \bigvee_{i=1}^1 l_i$

Por Tabla 4.2

4.1.5. $\bigvee_{i=1}^1 l_i = \bigvee_{i=1}^1 l_i$

De (4.1.2), (4.1.3) y (4.1.4) por casos

4.1.6. $\bigvee_{i=1}^1 l_i = \bigvee_{i=1}^1 l_i$

De (4.1.1) substituyendo en (4.1.5)

4.2. Si $n=j+1$ entonces $\bigvee_{i=1}^n l_i = \bigvee_{i=1}^n l_i$:

4.2.1. $n=j+1$

Hipótesis

4.2.2. $\bigvee_{i=1}^j l_i = \bigvee_{i=1}^j l_i$

Hipótesis inductiva

4.2.3. $\Theta' = \downarrow' + 1$

Definición de Θ en $\bigvee_{i=1}^j l_i$

4.2.4. $l_{j+1} = v_{j+1} \circ l_{j+1} = \neg v_{j+1}$

Por definición de l

4.2.5. Si $l_{j+1} = v_{j+1}$ entonces $\bigvee_{i=1}^j l_i \vee l_{j+1} = \bigvee_{i=1}^j l_i \vee l_{j+1}$:

4.2.5.1. $l_{j+1} = v_{j+1}$

Hipótesis

4.2.5.2. $\bar{\omega}_{j+1} = 1$

De (4.2.5.1) por (3)

4.2.5.3. $\downarrow = \downarrow'$

De (4.2.1), (4.2.2) y (4.2.5.2) por definición de \downarrow

4.2.5.4. $\Theta = \downarrow' + 1$

De (4.2.5.3) por definición de ∇

4.2.5.5. $\Theta = \Theta'$

De (4.2.3) substituyendo en (4.2.5.4)

4.2.5.6. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' < \Theta$

De (4.2.5.5)

4.2.5.7. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' \geq \Theta$

De (4.2.5.5)

4.2.5.8. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' + 1 = \Theta$:

4.2.5.8.1. $\alpha(\Gamma)' < \Theta'$

Hipótesis

- 4.2.5.8.2. $\alpha(\Gamma)' = \downarrow'$ De (4.2.5.8.1) por Lema 4.1
- 4.2.5.8.3. $\alpha(\Gamma)' + 1 = \downarrow' + 1$ De (4.2.5.8.2) por paso algebraico
- 4.2.5.8.4. $\alpha(\Gamma)' + 1 = \Theta$ De (4.2.5.4) substituyendo en (4.2.5.8.3)
- 4.2.5.9. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' + 1 > \Theta'$:
- 4.2.5.9.1. $\alpha(\Gamma)' \geq \Theta'$ Hipótesis
- 4.2.5.9.2. $\alpha(\Gamma)' + 1 > \Theta'$ De (4.2.5.9.1) por paso algebraico
- 4.2.5.9.3. $\alpha(\Gamma)' + 1 > \Theta$ De (4.2.5.5) substituyendo (4.2.5.9.2)
- 4.2.5.10. $\bigvee_{i=1}^j l_i \vee l_{j+1} = \bigvee_{i=1}^j l_i \nabla l_{j+1}$ Por Tabla 4.3
- 4.2.6. Si $l_{j+1} = \neg l_{j+1}$ entonces $\bigvee_{i=1}^j l_i \vee l_{j+1} = \bigvee_{i=1}^j l_i \nabla l_{j+1}$:
- 4.2.6.1. $l_{j+1} = \neg l_{j+1}$ Hipótesis
- 4.2.6.2. $\omega_{j+1} = -1$ De (4.2.6.1) por (2)
- 4.2.6.3. $\downarrow = \downarrow' - 1$ De (4.2.1), (4.2.2) y (4.2.6.2) por definición de \downarrow
- 4.2.6.4. $\Theta = \downarrow'$ De (4.2.6.3) por definición de ∇
- 4.2.6.5. $\Theta = \Theta' - 1$ De (4.2.3) substituyendo en (4.2.6.4)
- 4.2.6.6. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' = \Theta$:
- 4.2.6.6.1. $\alpha(\Gamma)' < \Theta'$ Hipótesis
- 4.2.6.6.2. $\alpha(\Gamma)' = \downarrow'$ De (4.2.6.6.1) por Lema 4.1
- 4.2.6.6.3. $\alpha(\Gamma)' = \Theta$ Substituyendo (4.2.6.4) en (4.2.6.6.2)
- 4.2.6.7. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' > \Theta$:
- 4.2.6.7.1. $\alpha(\Gamma)' \geq \Theta'$ Hipótesis
- 4.2.6.7.2. $\alpha(\Gamma)' > \Theta' - 1$ De (4.2.6.7.1) por paso algebraico
- 4.2.6.7.3. $\alpha(\Gamma)' > \Theta$ De (4.2.6.5) substituyendo en (4.2.6.7.2)
- 4.2.6.8. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' - 1 < \Theta$:
- 4.2.6.8.1. $\alpha(\Gamma)' < \Theta'$ Hipótesis
- 4.2.6.8.2. $\alpha(\Gamma)' - 1 < \Theta' - 1$ De (4.2.6.8.1) por paso algebraico
- 4.2.6.8.3. $\alpha(\Gamma)' - 1 < \Theta$ De (4.2.6.5) substituyendo en (4.2.6.8.2)
- 4.2.6.9. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' - 1 \geq \Theta$:
- 4.2.6.9.1. $\alpha(\Gamma)' \geq \Theta'$ Hipótesis
- 4.2.6.9.2. $\alpha(\Gamma)' - 1 \geq \Theta' - 1$ De (4.2.6.9.1) por paso algebraico
- 4.2.6.9.3. $\alpha(\Gamma)' - 1 \geq \Theta$ De (4.2.6.5) substituyendo en (4.2.6.9.2)

$$4.2.6.10. \sum_{i=1}^j l_i \nabla l_{j+1} = \sum_{i=1}^j l_i \nabla l_{j+1}$$

$$4.2.7. \sum_{i=1}^j l_i \nabla l_{j+1} = \sum_{i=1}^j l_i \nabla l_{j+1}$$

$$4.2.8. \sum_{i=1}^{j+1} l_i = \sum_{i=1}^{j+1} l_i$$

$$4.2.9. \sum_{i=1}^j l_i = \sum_{i=1}^j l_i$$

Por Tabla 4.4

De (4.2.4), (4.2.5) y (4.2.6) por casos

De (4.2.7) por definición de $\sum_{i=1}^j$ y ∇ De (4.2.1) y (4.2.8) pues $n=j+1$

l_i	$l_i = l_i$	$\sum_{i=1}^j l_i$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	0	0	0	0	De (4.1.3.1), (4.1.3.2), (4.1.3.3) y (4.1.3.4)
1	1	1	1	1	De (4.1.3.1), (4.1.3.2), (4.1.3.3) y (4.1.3.4)

Tabla 4.1 Cumplimiento de $\sum_{i=1}^j l_i = \sum_{i=1}^j l_i$ para $l_i = l_i$

l_i	$l_i = -l_i$	$\sum_{i=1}^j l_i$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	0	0	0	0	De (4.1.4.1), (4.1.4.2), (4.1.4.3) y (4.1.4.4)
1	1	0	-1	1	De (4.1.4.1), (4.1.4.2), (4.1.4.3) y (4.1.4.4)

Tabla 4.2 Cumplimiento de $\sum_{i=1}^j l_i = \sum_{i=1}^j l_i$ para $l_i = -l_i$

l_{i+1}	$l_{i+1} = l_{i+1}$	$\sum_{i=1}^j l_i$	$\sum_{i=1}^j \nabla l_{i+1}$	$\alpha(\Gamma)'$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	0	0	0	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)'$	0	De (4.2.5.6), (4.2.5.1), (4.2.5.2)
0	0	0	1	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)'$	1	De (4.2.5.7), (4.2.5.1), (4.2.5.2)
1	1	0	1	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)' + 1$	1	De (4.2.5.8), (4.2.5.1), (4.2.5.2)
1	1	1	1	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)' + 1$	1	De (4.2.5.9), (4.2.5.1), (4.2.5.2)

Tabla 4.3 Cumplimiento de $\sum_{i=1}^j \nabla l_{i+1} = \sum_{i=1}^j \nabla l_{i+1}$ para $l_i = l_i$

l_{i+1}	$l_{i+1} = -l_{i+1}$	$\sum_{i=1}^j l_i$	$\sum_{i=1}^j \nabla l_{i+1}$	$\alpha(\Gamma)'$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	1	0	1	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)'$	1	De (4.2.6.6), (4.2.6.1), (4.2.6.2)
0	1	1	1	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)'$	1	De (4.2.6.7), (4.2.6.1), (4.2.6.2)
1	0	0	0	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)' - 1$	0	De (4.2.6.8), (4.2.6.1), (4.2.6.2)
1	0	1	0	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)' - 1$	1	De (4.2.6.9), (4.2.6.1), (4.2.6.2)

Tabla 4.4 Cumplimiento de $\sum_{i=1}^j \nabla l_{i+1} = \sum_{i=1}^j \nabla l_{i+1}$ para $l_i = -l_i$

A manera de resumen, un ∇ es un NL con $\Theta = \downarrow + 1$ en el que se cumple que $\bigvee_{i=1}^n l_i = \bigwedge_{i=1}^n l_i$, siempre y cuando cada l_i sea creada a partir de una v_i bajo las siguientes reglas:

- Si una entrada v_i tiene un peso sináptico \bar{w}_i , inhibitorio de -1 entonces v_i es una variable booleana negada, es decir, $l_i = \neg v_i$,
- Si una entrada v_i tiene un peso sináptico \bar{w}_i , excitatorio de 1 entonces v_i es una variable booleana no negada, es decir, $l_i = v_i$,
- Si una entrada v_i tiene un peso sináptico \bar{w}_i , nulo de 0 entonces v_i es una variable booleana no tomada en cuenta, es decir, es como si la variable booleana v_i no existiera.

4.3 PROPIEDADES CON RESPECTO A \wedge

Un neuroide lógico Δ o neuroide lógico conjuntivo, es un neuroide lógico en el que se cumple que $\Theta = \uparrow$, es decir, que el valor umbral sea igual al número de entradas excitatorias al NL. Observe que si la función de integración $\alpha(\Gamma)$ de un Δ es mayor o igual que el valor umbral Θ entonces significa que $\alpha(\Gamma)$ es igual a su valor máximo de integración \uparrow , esto es debido a que por su definición el umbral es exactamente el valor máximo de la función de integración. La propiedad anterior se demuestra formalmente en el Lema 4.2.

Lema 4.2 Integración de Δ . Si $\alpha(\Gamma) \geq \Theta$ entonces $\alpha(\Gamma) = \uparrow$

Demostración.

1. $\alpha(\Gamma) \geq \Theta$
2. $\Theta = \uparrow$
3. $\alpha(\Gamma) \geq \uparrow$
4. $\alpha(\Gamma) = \uparrow$

Hipótesis

Por definición de Δ

De (2) substituyendo en (1)

De (3) pues \uparrow es el valor máximo de $\alpha(\Gamma)$

■

En adelante a un neuroide lógico conjuntivo con n entradas se le denotará como $\overset{\Delta}{\Delta} v_i$, donde obviamente cada v_i es una variable booleana por definición de NL. Un Δ es capaz de simular el funcionamiento de un operador \wedge sobre un conjunto de literales creadas a partir de sus variables booleanas de entrada v_i . Para que un Δ procese una literal creada a partir de v_i , simplemente se asigna -1 al peso sináptico \bar{w}_i si está negada la variable booleana v_i , y 1 al peso sináptico \bar{w}_i si no está negada la variable booleana v_i . Formalmente, si $(\bar{w}_i = -1$ si $l_i = \neg v_i$; y $\bar{w}_i = 1$ si $l_i = v_i$) entonces $\bigwedge_{i=1}^n l_i = \overset{\Delta}{\Delta} l_i$. Lo planteado anteriormente es demostrado en el Teorema 4.2, en el que se usará Θ' , $\alpha(\Gamma)'$, y \downarrow' para denotar respectivamente al umbral, función de integración y valor mínimo de integración en $\overset{\Delta}{\Delta} l_i$.

Teorema 4.2 Transferencia de Δ . Si $(\omega_i = -1$ si $l_i = \neg v_i$; y $\omega_i = 1$ si $l_i = v_i$) entonces $\overset{\omega}{\Delta} l_i = \overset{\omega}{\Delta} l_i$

Demostración.

1. $\omega_i = -1$ si $l_i = \neg v_i$; y $\omega_i = 1$ si $l_i = v_i$

2. si $l_i = \neg v_i$, entonces $\omega_i = -1$

3. si $l_i = v_i$, entonces $\omega_i = 1$

4. $\overset{\omega}{\Delta} l_i = \overset{\omega}{\Delta} l_i$:

4.1. Si $n=1$ entonces $\overset{\omega}{\Delta} l_i = \overset{\omega}{\Delta} l_i$:

4.1.1. $n=1$

4.1.2. $l_i = v_i$ o $l_i = \neg v_i$

4.1.3. Si $l_i = v_i$, entonces $\overset{1}{\Delta} l_i = \overset{1}{\Delta} l_i$:

4.1.3.1. $l_i = v_i$

4.1.3.2. $\omega_i = 1$

4.1.3.3. $\uparrow = 1$

4.1.3.4. $\Theta = 1$

4.1.3.5. $\overset{1}{\Delta} l_i = \overset{1}{\Delta} l_i$

4.1.4. Si $l_i = \neg v_i$, entonces $\overset{-1}{\Delta} l_i = \overset{-1}{\Delta} l_i$:

4.1.4.1. $l_i = \neg v_i$

4.1.4.2. $\omega_i = -1$

4.1.4.3. $\uparrow = 0$

4.1.4.4. $\Theta = 0$

4.1.4.5. $\overset{-1}{\Delta} l_i = \overset{-1}{\Delta} l_i$

4.1.5. $\overset{1}{\Delta} l_i = \overset{1}{\Delta} l_i$

4.1.6. $\overset{-1}{\Delta} l_i = \overset{-1}{\Delta} l_i$

4.2. Si $n=j+1$ entonces $\overset{\omega}{\Delta} l_i = \overset{\omega}{\Delta} l_i$:

4.2.1. $n=j+1$

4.2.2. $\overset{j}{\Delta} l_i = \overset{j}{\Delta} l_i$

4.2.3. $\Theta^* = \uparrow^*$

4.2.4. $l_{j+1} = v_{j+1}$ o $l_{j+1} = \neg v_{j+1}$

4.2.5. Si $l_{j+1} = v_{j+1}$ entonces $\overset{j}{\Delta} l_i \wedge l_{j+1} = \overset{j}{\Delta} l_i \Delta l_{j+1}$:

4.2.5.1. $l_{j+1} = v_{j+1}$

Hipótesis

De (1) de la conjunción de la parte

De (1) de la conjunción de la parte

Por inducción sobre n

Hipótesis

Por definición de l

Hipótesis

De (4.1.3.1) por (3)

De (4.1.1) y (4.1.3.2) por definición de \uparrow

De (4.1.3.3) por definición de Δ

Por Tabla 4.5

Hipótesis

De (4.1.4.1) por (2)

De (4.1.1) y (4.1.4.2) por definición de \uparrow

De (4.1.4.3) por definición de Δ

Por Tabla 4.6

De (4.1.2), (4.1.3) y (4.1.4) por casos

De (4.1.1) substituyendo en (4.1.5)

Hipótesis

Hipótesis inductiva

Definición de Θ en $\overset{j}{\Delta} l_i$

Por definición de l

Hipótesis

- 4.2.5.2. $\bar{\omega}_{j+1} = 1$ De (4.2.5.1) por (3)
- 4.2.5.3. $\uparrow = \uparrow + 1$ De (4.2.1), (4.2.2) y (4.2.5.2) por definición de \uparrow
- 4.2.5.4. $\Theta = \uparrow + 1$ De (4.2.5.3) por definición de Δ
- 4.2.5.5. $\Theta = \Theta + 1$ De (4.2.3) substituyendo en (4.2.5.4)
- 4.2.5.6. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' < \Theta$:
- 4.2.5.6.1. $\alpha(\Gamma)' < \Theta'$ Hipótesis
- 4.2.5.6.2. $\alpha(\Gamma)' < \Theta' + 1$ De (4.2.5.6.1) por paso algebraico
- 4.2.5.6.3. $\alpha(\Gamma)' < \Theta$ De (4.2.5.5) substituyendo en (4.2.5.6.2)
- 4.2.5.7. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' < \Theta$:
- 4.2.5.7.1. $\alpha(\Gamma)' \geq \Theta'$ Hipótesis
- 4.2.5.7.2. $\alpha(\Gamma)' = \uparrow'$ De (4.2.5.7.1) por Lema 4.2
- 4.2.5.7.3. $\alpha(\Gamma)' < \uparrow' + 1$ De (4.2.5.7.2) por paso algebraico
- 4.2.5.7.4. $\alpha(\Gamma)' < \Theta$ De (4.2.5.4) substituyendo en (4.2.5.7.3)
- 4.2.5.8. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' + 1 < \Theta$:
- 4.2.5.8.1. $\alpha(\Gamma)' < \Theta'$ Hipótesis
- 4.2.5.8.2. $\alpha(\Gamma)' + 1 < \Theta' + 1$ De (4.2.5.8.1) por paso algebraico
- 4.2.5.8.3. $\alpha(\Gamma)' + 1 < \Theta$ De (4.2.5.5) substituyendo en (4.2.5.8.2)
- 4.2.5.9. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' + 1 = \Theta$:
- 4.2.5.9.1. $\alpha(\Gamma)' \geq \Theta'$ Hipótesis
- 4.2.5.9.2. $\alpha(\Gamma)' = \uparrow'$ De (4.2.5.9.1) por Lema 4.2
- 4.2.5.9.3. $\alpha(\Gamma)' + 1 = \uparrow' + 1$ De (4.2.5.9.2) por paso algebraico
- 4.2.5.9.4. $\alpha(\Gamma)' + 1 = \Theta$ De (4.2.5.4) substituyendo en (4.2.5.9.3)
- 4.2.5.10. $\bigwedge_{\Delta} l_i \wedge l_{j+1} = \bigwedge_{\Delta} l_i \Delta l_{j+1}$ Por Tabla 4.7
- 4.2.6. Si $l_{j+1} = \neg v_{j+1}$ entonces $\bigwedge_{\Delta} l_i \wedge l_{j+1} = \bigwedge_{\Delta} l_i \Delta l_{j+1}$:
- 4.2.6.1. $l_{j+1} = \neg v_{j+1}$ Hipótesis
- 4.2.6.2. $\bar{\omega}_{j+1} = -1$ De (4.2.6.1) por (2)
- 4.2.6.3. $\uparrow = \uparrow'$ De (4.2.1), (4.2.2) y (4.2.6.2) por definición de \uparrow
- 4.2.6.4. $\Theta = \uparrow'$ De (4.2.6.3) por definición de Δ
- 4.2.6.5. $\Theta = \Theta'$ De (4.2.3) substituyendo en (4.2.6.4)
- 4.2.6.6. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' < \Theta$ De (4.2.6.5)

4.2.6.7. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' \geq \Theta$

De (4.2.6.5)

4.2.6.8. Si $\alpha(\Gamma)' < \Theta'$ entonces $\alpha(\Gamma)' - 1 < \Theta$:4.2.6.8.1. $\alpha(\Gamma)' < \Theta'$

Hipótesis

4.2.6.8.2. $\alpha(\Gamma)' - 1 < \Theta'$

De (4.2.6.8.1) por paso algebraico

4.2.6.8.3. $\alpha(\Gamma)' - 1 < \Theta$

De (4.2.6.5) substituyendo en (4.2.6.8.2)

4.2.6.9. Si $\alpha(\Gamma)' \geq \Theta'$ entonces $\alpha(\Gamma)' - 1 < \Theta$:4.2.6.9.1. $\alpha(\Gamma)' \geq \Theta'$

Hipótesis

4.2.6.9.2. $\alpha(\Gamma)' = \uparrow'$

De (4.2.6.9.1) por Lema 4.2

4.2.6.9.3. $\alpha(\Gamma)' - 1 < \uparrow'$

De (4.2.6.9.2) por paso algebraico

4.2.6.9.4. $\alpha(\Gamma)' - 1 < \Theta$

De (4.2.6.4) substituyendo en (4.2.6.9.3)

4.2.6.10. $\bigwedge_{i=1}^j l_i \wedge l_{j+1} = \bigwedge_{i=1}^j l_i \Delta l_{j+1}$

Por Tabla 4.8

4.2.7. $\bigwedge_{i=1}^j l_i \wedge l_{j+1} = \bigwedge_{i=1}^j l_i \Delta l_{j+1}$

De (4.2.4), (4.2.5) y (4.2.6) por casos

4.2.8. $\bigwedge_{i=1}^n l_i = \bigwedge_{i=1}^{j+1} l_i$ De (4.2.7) por definición de $\bigwedge_{i=1}^m$ y $\bigwedge_{i=1}^n$ 4.2.9. $\bigwedge_{i=1}^n l_i = \bigwedge_{i=1}^n l_i$

De (4.2.1) substituyendo en (4.2.8)

l_i	$l_i = l_i$	$\bigwedge_{i=1}^1 l_i$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	0	0	0	0	De (4.1.3.1), (4.1.3.2), (4.1.3.3) y (4.1.3.4)
1	1	1	1	1	De (4.1.3.1), (4.1.3.2), (4.1.3.3) y (4.1.3.4)

Tabla 4.5 Cumplimiento de $\bigwedge_{i=1}^j l_i = \bigwedge_{i=1}^1 l_i$ para $l_i = l_i$

l_i	$l_i = \neg l_i$	$\bigwedge_{i=1}^1 l_i$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	1	1	0	1	De (4.1.4.1), (4.1.4.2), (4.1.4.3) y (4.1.4.4)
1	0	0	-1	0	De (4.1.4.1), (4.1.4.2), (4.1.4.3) y (4.1.4.4)

Tabla 4.6 Cumplimiento de $\bigwedge_{i=1}^j l_i = \bigwedge_{i=1}^1 l_i$ para $l_i = \neg l_i$

i_{j+1}	$i_{j+1} = i_{j+1}$	$\bigwedge_{\Delta}^j i_j$	$\bigwedge_{\Delta}^j i_{j+1}$	$\alpha(\Gamma)'$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	0	0	0	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)'$	0	De (4.2.5.6), (4.2.5.1), (4.2.5.2)
0	0	1	0	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)'$	0	De (4.2.5.7), (4.2.5.1), (4.2.5.2)
1	1	0	0	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)' + 1$	0	De (4.2.5.8), (4.2.5.1), (4.2.5.2)
1	1	1	1	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)' + 1$	1	De (4.2.5.9), (4.2.5.1), (4.2.5.2)

Tabla 4.7 Cumplimiento de $\bigwedge_{\Delta}^j i_{j+1} = \bigwedge_{\Delta}^j i_j \bigwedge_{\Delta}^j i_{j+1}$ para $i_j = i$.

i_{j+1}	$i_{j+1} = \neg i_{j+1}$	$\bigwedge_{\Delta}^j i_j$	$\bigwedge_{\Delta}^j i_{j+1}$	$\alpha(\Gamma)'$	$\alpha(\Gamma)$	$\tau(\alpha(\Gamma))$	Referencia
0	1	0	0	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)'$	0	De (4.2.6.6), (4.2.6.1), (4.2.6.2)
0	1	1	0	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)'$	1	De (4.2.6.7), (4.2.6.1), (4.2.6.2)
1	0	0	0	$\alpha(\Gamma)' < \Theta'$	$\alpha(\Gamma)' - 1$	0	De (4.2.6.8), (4.2.6.1), (4.2.6.2)
1	0	1	0	$\alpha(\Gamma)' \geq \Theta'$	$\alpha(\Gamma)' - 1$	0	De (4.2.6.9), (4.2.6.1), (4.2.6.2)

Tabla 4.8 Cumplimiento de $\bigwedge_{\Delta}^j i_{j+1} = \bigwedge_{\Delta}^j i_j \bigwedge_{\Delta}^j i_{j+1}$ para $i_j = \neg i$.

A manera de resumen, un ∇ es un NL con $\Theta = \uparrow$ en el que se cumple que $\bigwedge_{\Delta}^n l_i = \bigwedge_{\Delta}^n l_i$, siempre y cuando cada l_i sea creada a partir de una i_i bajo las siguientes reglas:

- ⇒ Si una entrada i_i tiene un peso sináptico $\bar{\omega}_i$ inhibitorio de -1 entonces i_i es una variable booleana negada, es decir, $l_i = \neg i_i$.
- ⇒ Si una entrada i_i tiene un peso sináptico $\bar{\omega}_i$ excitatorio de 1 entonces i_i es una variable booleana no negada, es decir, $l_i = i_i$.
- ⇒ Si una entrada i_i tiene un peso sináptico $\bar{\omega}_i$ nulo de 0 entonces i_i es una variable booleana no tomada en cuenta, es decir, es como si la variable booleana i_i no existiera.

4.4 COROLARIOS

Sabemos por el Teorema 4.1 que $\bigwedge_{\Delta}^n l_i = \bigwedge_{\Delta}^n l_i$, por lo tanto, por definición de término canónico disyuntivo, un ∇ puede simular un término canónico disyuntivo. Por otro lado, debido a que por el Teorema 4.2 sabemos que $\bigwedge_{\Delta}^n l_i = \bigwedge_{\Delta}^n l_i$, podemos afirmar entonces que, por definición de término canónico conjuntivo, un Δ puede simular un término canónico conjuntivo. Lo descrito en este párrafo es resaltado en el Corolario 4.1 y el Corolario 4.2.

Corolario 4.1 (Del Teorema 4.1) Un ∇ es capaz de simular un término canónico disyuntivo.

Corolario 4.2 (Del Teorema 4.2) Un Δ es capaz de simular un término canónico conjuntivo.

Un ∇ canónico es un ∇ que simula un término canónico disyuntivo. Un Δ canónico es un Δ que simula un término canónico conjuntivo. En el Teorema 1.4 se demostró que $|T| = 2^n$, es decir, que el número de términos canónicos conjuntivos distintos que pueden formarse con n variables booleanas es exactamente 2^n . Por lo tanto, podemos afirmar que existen exactamente 2^n neuroides Δ canónicos distintos para n variables booleanas de entrada. De la misma forma, el Corolario 1.3 sostiene que $|\neg T| = 2^n$, esto es, que el número de términos canónicos disyuntivos distintos que pueden formarse con n variables booleanas es exactamente 2^n . Por ello, podemos aseverar que existen exactamente 2^n neuroides ∇ canónicos distintos para n variables booleanas de entrada. Lo expuesto es este párrafo es plasmado en el Corolario 4.3 y el Corolario 4.4.

Corolario 4.3 (Del Teorema 1.4). Existen exactamente 2^n neuroides $\bigwedge_{i=1}^n l_i$ canónicos distintos

Corolario 4.4 (Del Corolario 1.3). Existen exactamente 2^n neuroides $\bigvee_{i=1}^n l_i$ canónicos distintos

Los 2^n neuroides Δ y ∇ canónicos pueden determinarse a partir de las interpretaciones de una función booleana precisamente con el mismo proceso descrito en la sección 1.5 para obtener los 2^n términos canónicos conjuntivos y disyuntivos. Los 2^n neuroides Δ y ∇ canónicos para tres entradas t_1 , t_2 y t_3 son mostrados en la Tabla 4.9. Debido a que son términos canónicos, es evidente que cada Δ_i es la negación de ∇_i y viceversa, lo que se puede corroborar fácilmente observando la Tabla 4.9.

Interpretaciones			Δ canónicos		∇ canónicos	
t_1	t_2	t_3	Denominación	Término	Denominación	Término
ψ_1	0	0	Δ_1	$\neg t_1 \Delta \neg t_2 \Delta \neg t_3$	∇_1	$t_1 \nabla t_2 \nabla t_3$
ψ_2	0	0	Δ_2	$\neg t_1 \Delta \neg t_2 \Delta t_3$	∇_2	$t_1 \nabla t_2 \nabla \neg t_3$
ψ_3	0	1	Δ_3	$\neg t_1 \Delta t_2 \Delta \neg t_3$	∇_3	$t_1 \nabla \neg t_2 \nabla t_3$
ψ_4	0	1	Δ_4	$\neg t_1 \Delta t_2 \Delta t_3$	∇_4	$t_1 \nabla \neg t_2 \nabla \neg t_3$
ψ_5	1	0	Δ_5	$t_1 \Delta \neg t_2 \Delta \neg t_3$	∇_5	$\neg t_1 \nabla t_2 \nabla t_3$
ψ_6	1	0	Δ_6	$t_1 \Delta \neg t_2 \Delta t_3$	∇_6	$\neg t_1 \nabla t_2 \nabla \neg t_3$
ψ_7	1	1	Δ_7	$t_1 \Delta t_2 \Delta \neg t_3$	∇_7	$\neg t_1 \nabla \neg t_2 \nabla t_3$
ψ_8	1	1	Δ_8	$t_1 \Delta t_2 \Delta t_3$	∇_8	$\neg t_1 \nabla \neg t_2 \nabla \neg t_3$

Tabla 4.9 Neuroides ∇ y Δ obtenidos a partir de las interpretaciones de tres variables de entrada

Está demostrado por el Teorema 1.5, que cada Δ_i resulta ser neuroide Δ canónico que evaluando sus entradas hace verdadera a la interpretación ψ_i y falsa a las interpretaciones restantes. De la misma forma, por el Corolario 1.5 se sostiene que ∇_i es el neuroide ∇ canónico que evaluando sus entradas hace falsa a la interpretación ψ_i y verdadera a las interpretaciones restantes. Estas mismas propiedades son expuestas en el Corolario 4.5 y el Corolario 4.6.

Corolario 4.5 (Del Teorema 1.5) Un Δ_i evaluando sus entradas hace cierta a la interpretación ψ_i y falsa a las interpretaciones restantes.

Corolario 4.6 (Del Corolario 1.5) Un ∇_i evaluando sus entradas hace falsa a la interpretación ψ_i y cierta a las interpretaciones restantes.

Por el Corolario 4.5 y el Corolario 4.6 sabemos que para todo i , $\neg \Delta_i = \nabla_i$; por lo tanto podemos inferir que un NL puede simular la negación de la conjunción de un conjunto de literales booleanas. De igual manera, por el Corolario 4.5 y el Corolario 4.6 sabemos que para todo i , $\neg \nabla_i = \Delta_i$; por lo tanto podemos inferir que un NL puede simular la negación de la disyunción de un conjunto de literales booleanas. Lo explicado en este párrafo es resaltado en el Corolario 4.7 y el Corolario 4.8.

Corolario 4.7 (Del Corolario 4.5 y el Corolario 4.6) Un NL es capaz de simular $\neg(\bigwedge_{i=1}^n l_i)$.

Corolario 4.8 (Del Corolario 4.5 y el Corolario 4.6) Un NL es capaz de simular $\neg(\bigvee_{i=1}^n l_i)$.

5. LA RED NEURONAL CANÓNICA

5.1 MODELO

Una red neuronal artificial es una red neuronal canónica si en su modelo de red se cumple que:

1. $|U| = 3$
2. $|U_1| = n$, donde n es el número de entradas a la red neuronal canónica
3. $|U_2| \leq 2^n$
4. $|U_3| = s$, donde s es el número de salidas deseadas de la red neuronal canónica
5. U_1 es una capa neuronal de proyección
6. Para todo $U_{2,j}$, $U_{2,j}$ es un NL
7. Para todo $U_{3,e}$,
si $\omega_{(U_1, U_2, U_3)} \neq 0$ y $U_{2,j}$ es un Δ entonces $U_{3,e}$ es un ∇ , o bien,
si $\omega_{(U_1, U_2, U_3)} \neq 0$ y $U_{2,j}$ es un ∇ entonces $U_{3,e}$ es un Δ
8. La función de interconexión establece una topología divergente/convergente de U_1 a U_2 y de U_2 a U_3 , con posibilidad de reverberancia.

Una red neuronal canónica consta únicamente de tres capas. La capa U_1 es una capa neuronal aferente cuyo cometido es el proyectar cada uno de los datos de entrada sobre todos los neuroides de la segunda capa. Para lograr la proyección, cada neuroide de la capa U_1 tiene un vector de entradas con un solo elemento booleano, un vector de pesos sinápticos con un solo elemento de valor siempre igual a 1, la función producto punto como función de integración y la función identidad como función de transferencia. Dependiendo del problema en particular al que es expuesta una red neuronal canónica, pudiera usarse como función de integración una función que incorporara sumación temporal o alguna normalización de los datos de entrada. Para este trabajo se ha asumido que los valores resultantes de la función de transferencia pueden ser 0 o 1, es

decir, valores booleanos y discretos. Sin embargo, tal y como se describirá en la sección 6.2, futuras investigaciones estarán relacionadas a comprobar que el funcionamiento y propiedades de una red neuronal canónica permanecen sin cambio con funciones de integración que incorporen sumación temporal o normalización, así como con funciones de transferencia cuyos valores retornados sean reales comprendidos entre el 0 y el 1, lo que puede resultar valioso para el tratamiento directo de problemas analógicos. Que el número de elementos en la capa U_1 sea igual al número de entradas a la red es algo evidente por las actividades para las que esta destinada esta capa, concretamente $|U_1| = n$ para n entradas a la red, tal y como se estableció en la especificación de la red neuronal canónica.

La capa U_2 es la capa neuronal asociativa. A ella llegan datos provenientes de U_1 que son procesados y entregados a la capa neuronal U_3 . Finalmente, la capa neuronal U_3 es la capa neuronal eferente que procesa los datos provenientes de la capa U_2 y entrega los resultados al exterior. En la condición impuesta de que solo neuroides Δ se conecten a neuroides ∇ entre las capas U_2 y U_3 , radica la mecánica de la red neuronal canónica. Si se permite que los neuroides Δ se conecten solo a neuroides ∇ entre las capas U_2 y U_3 , entonces sabemos por el Teorema 4.1 y el Teorema 4.2 que es posible hacer la conjunción de términos canónicos disyuntivos o la disyunción de términos conjuntivos. En otras palabras, por definición de la forma canónica booleana, a través de la capa U_2 y la capa U_3 es posible crear polinomios booleanos en forma canónica, donde la capa U_2 puede visualizarse como un conjunto de términos canónicos, mientras que la capa U_3 es la disyunción o conjunción de tales términos canónicos. Esto mismo es resaltado en el Corolario 5.1.

Corolario 5.1 (del Teorema 4.1 y Teorema 4.2) Una red neuronal canónica es capaz de simular un conjunto de polinomios booleanos en forma canónica.

El que se haya definido en la especificación del modelo de una red neuronal canónica que la función de interconexión establezca una topología divergente/convergente de U_1 a U_2 y de U_2 a U_3 , asegura el desempeño de los elementos de la capa U_3 para la disyunción o conjunción de los términos formados en U_2 con las variables de entrada obtenidas en U_1 . Observe que la especificación del modelo de la red neuronal canónica no impide la posibilidad de reverberancias, lo que ofrece flexibilidad para implantar redes neuronales artificiales en las que se requiera de respuestas hechas en base a los datos de entrada actuales y a las decisiones realizadas con anterioridad.

Es claro que habrá tantos polinomios booleanos en una red neuronal canónica como neuroides lógicos existan en la capa U_3 . Cada NL de U_3 será un polinomio formado por los términos representados en los NL de U_2 con los que tiene una conexión con peso sináptico no nulo, es decir diferente de 0. Si un NL de U_3 guarda una conexión con peso sináptico inhibitorio, es decir igual a -1 , significa, por el Teorema 4.1 y el Teorema 4.2, que tal sinapsis es el enlace con un término negado. Si un NL de U_3 guarda una conexión con peso sináptico excitatorio, es decir igual a 1, significa, por el Teorema 4.1 y el Teorema 4.2, que tal sinapsis es el enlace con un término no negado.

En el Corolario 5.1 se afirma que una red neuronal canónica puede simular un conjunto de polinomios canónicos, donde cada NL en la capa U_3 podría verse como un polinomio canónico formado por términos canónicos representados por los neuroides lógicos canónicos de la capa U_2 que pueden crearse de n entradas a

la red. Del Corolario 4.3 y del Corolario 4.4 sabemos que existen exactamente 2^n neuroides lógicos canónicos diferentes, por lo tanto sabemos que el número máximo de elementos deseables en la capa U_2 es 2^n . Es por esta razón que en la especificación del modelo de la red neuronal canónica se estableció que $|U_2| \geq 2^n$.

En el Corolario 5.1 se afirma que una red neuronal canónica puede simular un conjunto de polinomios canónicos, donde cada NL en la capa U_1 podría verse como un polinomio canónico formado por términos canónicos representados por los neuroides lógicos canónicos de la capa U_2 que pueden crearse de n entradas a la red. Del Teorema 1.6 y del Corolario 1.10 sabemos que existen exactamente 2^n polinomios canónicos distintos, por lo tanto los neuroides de U_2 pueden simular hasta 2^n polinomios canónicos distintos. En la práctica, sin embargo, el número de polinomios canónicos distintos que pueden formarse dependerá de la cantidad de elementos disponibles en la capa neuronal U_2 , es decir que el número de polinomios booleanos canónicos distintos es $2^{|U_2|}$.

Observe que, debido a que la capa U_1 no realiza procesamiento alguno, estrictamente hablando una red neuronal canónica requiere solo de las capas U_2 y U_3 para llevar a cabo su cometido. De hecho computacionalmente hablando, la implantación de la capa neuronal U_1 de una red neuronal canónica sería un simple vector en el cual se colocarían los datos de entrada a ser leídos por los neuroides de la capa U_2 . Aún más, la capa U_1 podría ser eliminada escribiendo directamente los datos de entrada a las localidades correspondientes de los vectores de entrada de los neuroides de la capa U_2 . Sin embargo, aunque en la práctica pueda dispensarse el uso de la capa U_1 , se ha decidido mantener al menos de manera lógica la idea de esta capa en el modelo teórico de la red neuronal canónica para por una parte mantener la semejanza con la organización en capas aferente-asociativa-eferente de los sistemas biológicos, mientras que por la otra vislumbrar a la normalización de datos de entrada y a la simulación de los procesos de plasticidad sináptica como posibles empleos de U_1 . Tomando como base las líneas de este párrafo, es que se ha establecido en la especificación del modelo de una red neuronal canónica que sean tres las capas neuronales que la conformen, en otras palabras, $|U| = 3$.

5.2 PROPIEDADES

En lo siguiente a un conjunto no vacío de neuroides Δ se le denominará capa neuronal conjuntiva, o capa Δ . De la misma forma, a un conjunto no vacío de neuroides ∇ se le llamará capa neuronal disyuntiva, o capa ∇ . Una capa neuronal lógica es una capa ∇ o una capa Δ . Una red neuronal canónica universal (RNCU) es una red neuronal canónica en la que se cumple que:

1. U_2 es una capa neuronal lógica
2. $|U_2| = 2^n$, para n entradas a la red
3. Para todo g, j , si $1 \leq g \leq |U_2|$ y $1 \leq j \leq |U_2|$ y $g \neq j$ entonces $U_{2g} * U_{2j}$

4. Si U_2 es una capa Δ entonces U_3 es una capa ∇ , o bien,

si U_2 es una capa ∇ entonces U_3 es una capa Δ

5. Para todo i, j , si $2 \leq i \leq 3$ y $1 \leq j \leq |U_i|$ entonces la cardinalidad de \mathfrak{t} y \mathfrak{w} para la capa U_i es de $|U_{i-1}|$

6. La función de interconexión es la función de interconexión total, es decir,

para todo i, j, k , si $2 \leq i \leq 3$ y $1 \leq j \leq |U_i|$ y $1 \leq k \leq |U_{i-1}|$ entonces $\mathfrak{v}_{j,k} = \tau_{i-1,k}(\alpha_{i-1,k}(\mathbf{I}))$

7. La función de inicialización se define como:

7.1. para todo j, k , si U_2 es una capa Δ y $1 \leq j \leq |U_2|$ y $1 \leq k \leq n$ entonces $\mathfrak{w}_{2,j,k} = P\left(\frac{j-1}{2^{k-1}}\right)$

donde P es una función definida como $P\left(\frac{x}{y}\right) = \begin{cases} -1 & \text{si el cociente de } \frac{x}{y} \text{ es par} \\ 1 & \text{de otra forma} \end{cases}$

7.2. para todo j, k , si U_2 es una capa ∇ y $1 \leq j \leq |U_2|$ y $1 \leq k \leq n$ entonces $\mathfrak{w}_{2,j,k} = I\left(\frac{j-1}{2^{k-1}}\right)$

donde I es una función definida como $I\left(\frac{x}{y}\right) = \begin{cases} -1 & \text{si el cociente de } \frac{x}{y} \text{ es impar} \\ 1 & \text{de otra forma} \end{cases}$

7.3. para todo j, k , si U_3 es una capa ∇ y $1 \leq j \leq |U_3|$ y $1 \leq k \leq |2^n|$ entonces $\mathfrak{w}_{3,j,k} = 0$

para todo j, k , si U_3 es una capa Δ y $1 \leq j \leq |U_3|$ y $1 \leq k \leq |2^n|$ entonces $\mathfrak{w}_{3,j,k} = 1$

Las tres primeras condiciones a cumplir por una RNCU, que U_2 es una capa neuronal lógica, que $|U_2| = 2^n$, y que $U_{2,i} * U_{2,j}$ si $g * j$, así como la sexta condición acerca de la función de interconexión hacen que la capa neuronal U_2 se convierta en el conjunto T de todos los términos canónicos conjuntivos o bien en el conjunto $\neg T$ de todos los términos canónicos disyuntivos que pueden formarse de n variables booleanas, esto por el Corolario 4.1, el Corolario 4.2, el Corolario 4.3, el Corolario 4.4, el Teorema 1.4 y el Corolario 1.3. Lo mismo afirmado en estas líneas es explicitado por su relevancia en el Corolario 5.2.

Corolario 5.2 Composición de la capa U_2 de una RNCU (del Corolario 4.1, el Corolario 4.2, el Corolario 4.3, el Corolario 4.4, el Teorema 1.4, el Corolario 1.3 y la definición de RNCU) $U_2 = T \circ U_1 = \neg T$

La cuarta condición a cumplir de la definición de RNCU, que U_3 sea una capa ∇ si U_2 es una capa Δ , o bien, que U_3 sea una capa Δ si U_2 es una capa ∇ , así como la sexta condición acerca de la función de interconexión son las estipulaciones que hacen posible la disyunción de los términos canónicos conjuntivos, o bien la disyunción de los términos canónicos conjuntivos que están en U_2 .

La razón de la quinta condición a satisfacer para la definición de una RNCU, que la cardinalidad de \mathfrak{v} y \mathfrak{w} para la capa U_i es de $|U_{i,1}|$ para $2 \leq i \leq 3$, era fácilmente derivable para la capa U_3 del Corolario 1.8, el Corolario 1.9, el Corolario 4.1, el Corolario 4.2, pues como el número máximo de términos canónicos que puede tener un polinomio canónico es 2^n , debe asegurarse que tanto los vectores de entradas como los vectores de pesos sinápticos tengan 2^n elementos. De la misma forma, para la capa U_2 sabemos por definición de término canónico, por el Corolario 4.1, y el Corolario 4.2, que todo término canónico debe contener n literales, por lo tanto \mathfrak{v} y \mathfrak{w} en U_2 deben tener n elementos.

La sexta condición a cumplir para la definición de una RNCU referente a la función de interconexión es necesaria en la capa U_2 debido a que, por definición de término canónico, cada neuroide lógico debe contener a cada una de las n variables booleanas de entrada. En tanto que para la capa U_3 , aunque no necesariamente un polinomio canónico contiene todos los términos canónicos posibles, el que exista una función de interconexión total si simplificará en gran medida la exposición acerca del aprendizaje de la RNCU, mismo que es detallado en la sección 5.3. Observe que no existe condición alguna que prohíba la posibilidad de reverberancias en la RNCU, lo que ofrece flexibilidad para implantar redes neuronales artificiales en las que se requiera de respuestas hechas en base a los datos de entrada actuales y a las decisiones realizadas con anterioridad.

La séptima condición a observar para la definición de una RNCU es acerca de una función de inicialización que para la capa U_2 inicializa los valores del vector de pesos sinápticos \mathfrak{w} de cada neuroide lógico a la manera como se llenan las columnas correspondientes a las variables en una tabla de verdad. Más formalmente, la función de inicialización asigna a cada neuroide $U_{2,i}$ la interpretación ψ_i en su vector de pesos sinápticos $\mathfrak{w}_{2,i}$. De esta manera, se asegura que cada NL de la capa U_2 estará preparada desde su concepción para representar al i -ésimo término canónico determinado por ψ_i . La función de inicialización no hace explícita una modificación de Θ pues por definición el valor umbral de un NL es determinado por los valores de su vector de pesos, por lo que cualquier modificación a \mathfrak{w} implica de manera implícita una modificación a Θ .

Por otro lado, para U_3 los vectores de pesos de todos los neuroides son puestos por la función de inicialización en 0 o en 1 dependiendo de si la capa U_3 es una capa ∇ o una capa Δ respectivamente. Esta condición de arranque estabiliza la salida inicial de la RNCU asegurando tenga siempre una salida inicial igual a 0. En otras palabras, la función de inicialización hace que la función de transferencia de todos los neuroides de la capa U_3 sea igual a 0. Esta característica de la RNCU será aprovechada durante el entrenamiento de la red, mismo que es expuesto en la sección 5.3.

Una RNCU disyuntiva es una red neuronal canónica universal en la que la capa U_2 es una capa Δ y la capa U_3 es una capa ∇ . Una RNCU conjuntiva es una red neuronal canónica universal en la que la capa U_2 es una capa ∇ y la capa U_3 es una capa Δ . Debido a la definición de polinomio canónico y a la definición de RNCU, es evidente que una RNCU disyuntiva es un conjunto de polinomios canónicos disyuntivos y que por lo tanto es un subconjunto del conjunto D de todos los polinomios canónicos disyuntivos. De la misma forma, una RNCU conjuntiva es un conjunto de polinomios canónicos conjuntivos, por lo que es un subconjunto del conjunto C de todos los polinomios canónicos conjuntivos. Lo expuesto en este párrafo es plasmado formalmente en el Corolario 5.3.

Corolario 5.3 Composición de la capa U_3 de una RNCU (de la definición de polinomio canónico y la definición de RNCU) $U_3 \subseteq D$ o $U_3 \subseteq C$

Sabemos por el Corolario 5.3 que con una RNCU pueden representarse todos los polinomios canónicos. También conocemos por el Corolario 1.12 y el Corolario 1.13 que toda función booleana es representable por un polinomio canónico, por lo tanto podemos inferir que cualquier función booleana es representable a través de la RNCU. Esto mismo es recalcado en el Corolario 5.4 y el Corolario 5.5.

Corolario 5.4 Universalidad de la RNCU disyuntiva (Del Corolario 5.3 y el Corolario 1.12) para toda f , si f es una función booleana entonces f es representable por una RNCU disyuntiva

Corolario 5.5 Universalidad de la RNCU conjuntiva (Del Corolario 5.3 y el Corolario 1.13) para toda f , si f es una función booleana entonces f es representable por una RNCU conjuntiva

Las propiedades de la RNCU resultan claramente inferibles del Corolario 5.1, con el que se estableció que una red neuronal canónica es capaz de simular un conjunto de polinomios canónicos, por lo que las propiedades de los polinomios canónicos revisadas en la sección 1.5 son válidas en general para cualquier red neuronal canónica y en particular para la RNCU.

5.3 APRENDIZAJE

Puede existir más de un método de aprendizaje para una RNCU. Los algoritmos expuestos en esta sección no intentan ser óptimos, sino más bien ser métodos que permitan exponer fácilmente los mecanismos y propiedades de la RNCU y de las redes neuronales canónicas en general. Bajo este criterio, se propone el siguiente algoritmo de entrenamiento para la RNCU:

1. Inicializar el vector de pesos sinápticos y los valores umbrales de todos los NL de la RNCU a través de la función de inicialización
2. Hacer
 - 2.1. Presentar un patrón de entrada $\partial(\Gamma)$ a la capa neuronal aferente U_1
 - 2.2. Calcular las funciones de transferencia de todos los neuroides de la RNCU
 - 2.3. Obtener el patrón de salida de la red $\Phi(\partial(\Gamma))$ de la capa neuronal eferente U_2
 - 2.4. Aplicar la función de aprendizaje λ a la capa eferente U_3 usando $\Phi(\partial(\Gamma))$, $\partial(\Gamma)$ y el patrón de salida deseado de la red $\Phi(\partial(\Gamma))$

Hasta que la red haya respondido correctamente a todos los patrones de entrada

Los patrones Φ , ϕ y ∂ son funciones cuyos contradominios son n -adas de variables booleanas, por lo tanto en lo siguiente se usará la nomenclatura $\Phi(\partial(\Gamma))_i$, $\phi(\partial(\Gamma))_i$ y $\partial(\Gamma)_i$ para hacer referencia al i -ésimo elemento de las n -adas $\Phi(\partial(\Gamma))$, $\phi(\partial(\Gamma))$ y $\partial(\Gamma)$ respectivamente. Para el caso de ∂ , cada variable booleana $\partial(\Gamma)_i$ representa el valor de entrada al neuroide $U_{1,i}$ en el tiempo Γ . Para el caso de ϕ , cada variable booleana $\phi(\partial(\Gamma))_i$ representa el valor de salida del neuroide $U_{2,i}$ en el tiempo Γ . Finalmente, para el caso de Φ , cada variable booleana $\Phi(\partial(\Gamma))_i$ representa el valor de salida deseado del neuroide $U_{3,i}$ en el tiempo Γ .

La función de aprendizaje λ propuesta para el algoritmo de aprendizaje de la RNCU es definida como:

$$\lambda(\omega_{3,j,i}(\Gamma)_{h_0}) = \begin{cases} (\omega_{3,j,i}(\Gamma)_{h_0} - 1) \times -1 & \text{si } \phi(\partial(\Gamma))_i \neq \Phi(\partial(\Gamma))_i \\ \omega_{3,j,i}(\Gamma)_{h_0} & \text{de otra forma} \end{cases}$$

donde $(\partial(\Gamma))_{i_0}$ es la n -ada $\partial(\Gamma)$ tomada como número binario y convertida a número decimal.

La función de aprendizaje de la RNCU hace únicamente un cambio en el vector de pesos sinápticos de los neuroides lógicos de la capa neuronal U_3 cuyas salidas no correspondan con la salida esperada de la red. Aún más, la función de aprendizaje asegura que la RNCU aprenderá la respuesta correcta $\Phi(\partial(\Gamma))$ al patrón de entrada $\partial(\Gamma)$ en una sola iteración, pues se habilita o inhibe la conexión del neuroide incorrecto de la capa U_3 con el neuroide de la capa U_2 responsable de simular el término canónico que origina la falla. Observe que aunque no se ha explicitado en la función de aprendizaje, un cambio en el vector de pesos de un neuroide lógico trae como consecuencia un cambio en el valor umbral por definición de NL. De esta manera en el mejor de los casos, que ocurre cuando la red ha respondido exactamente como se esperaba, existen cero modificaciones en la RNCU. Por otra parte, en el peor de los casos, que ocurre cuando ninguno de los neuroides de la última capa responde correctamente, existen $\{U_3\}$ modificaciones en los vectores de pesos sinápticos más $\{U_3\}$ modificaciones en los valores umbrales de los neuroides de la capa U_3 de la RNCU. Las

$2 \times |U_3|$ modificaciones del peor de los casos pueden juzgarse relativamente pocas si consideramos que la capa eferente es generalmente la capa de una red que tiene menos neuroides. Lo expuesto en este párrafo es subrayado por su relevancia en el Corolario 5.6.

Corolario 5.6 (De la definición de NL, RNCU y λ) En el peor de los casos se requieren de $2 \times |U_3|$ modificaciones en una iteración del algoritmo de entrenamiento de una RNCU.

Sabemos por el Corolario 5.3 que una RNCU es un conjunto de polinomios canónicos. Por el Corolario 1.15 y el Corolario 1.16 sabemos que una función booleana de n variables es representable por uno y solo un polinomio canónico formado a partir de n variables. Finalmente, sabemos por el Teorema 1.3 que existen exactamente 2^n interpretaciones diferentes para cualquier función booleana de n variables. Podemos concluir entonces que se requieren presentar a lo más $2^n - 1$ patrones de entrada diferentes a una RNCU para que esta aprenda, pues la RNCU al inicializarse representa ya una función booleana. En otras palabras, en el peor de los casos el algoritmo de entrenamiento requiere exactamente de $2^n - 1$ iteraciones para hacer que la RNCU aprenda. Debemos recalcar el evidente hecho de que solo una de las 2^n funciones booleanas que es posible representar requiere de $2^n - 1$ iteraciones para ser aprendida totalmente por una RNCU. Lo expresado en las líneas anteriores es refrendado en el Corolario 5.7.

Corolario 5.7 (Del Corolario 5.1, Corolario 1.15, Corolario 1.16, Teorema 1.3, la definición de RNCU y su algoritmo de entrenamiento) En el peor de los casos se requieren de $2^n - 1$ iteraciones para que una RNCU aprenda.

El estado de un NL en el tiempo Γ es una n -ada (x_1, x_2) en la que x_1 y x_2 son respectivamente el vector de pesos \mathbb{W} y valor umbral Θ del neuroide en el tiempo Γ . El estado de una capa neuronal canónica U_i en el tiempo Γ es una n -ada $(x_1, \dots, x_j, \dots, x_n)$, donde cada x_j es el estado del neuroide $U_{i,j}$ en el tiempo Γ . El conocimiento de una red neuronal canónica en el tiempo Γ es una n -ada (x_1, x_2) , donde x_1 es el estado de la capa neuronal U_i en el tiempo Γ . El adiestramiento de una red neuronal canónica del tiempo Γ_1 al tiempo Γ_n es una n -ada $(x_1, \dots, x_j, \dots, x_n)$ donde cada x_j es el conocimiento de la red neuronal canónica en el tiempo Γ_j .

A través del estado de un NL se puede conocer si el neuroide está trabajando como un ∇ o bien como un Δ . Se puede conocer además sobre qué variables booleanas de entrada opera y si tales variables están negadas o no. Resulta relativamente sencillo especificar una función de descifrado $Descifrar(U_{ij})$ que asigne a cada U_{ij} una cadena de caracteres que contenga una expresión legible que describa el estado del neuroide lógico U_{ij} a través de los símbolos de la lógica de predicados. A continuación se propone, mediante de un algoritmo, una especificación muy simple de $Descifrar(U_{2,j})$ para la capa U_2 de una RNCU:

1. $c \leftarrow ""$, $k \leftarrow 1$
2. Hacer mientras que $k \leq |U_1|$
 - 2.1. Si $k > 1$ entonces $c \leftarrow Concatenar(c, Tipificar(U_{2,k}))$
 - 2.2. Si $\omega_{2,j,k} = -1$ entonces $c \leftarrow Concatenar(c, "\neg")$
 - 2.3. $c \leftarrow Concatenar(c, "U_{1,k}")$
 - 2.4. $k \leftarrow k + 1$
3. $c \leftarrow Concatenar(c, ")"$

Al finalizar el algoritmo la cadena de caracteres c contendrá una expresión booleana que describe el funcionamiento de un neuroide $U_{2,j}$ con símbolos del álgebra de predicados. El paso (1) del algoritmo inicializa c y el índice k . Los pasos (2.1) al (2.4), tal y como lo indica (2), se repetirán para cada uno de los elementos de ω . El paso (2.1) se sirve de la función *Tipificar* para conocer qué tipo de función lógica, \wedge o \vee , está efectuando el NL sobre sus variables de entrada; el resultado se une al final de la cadena c a través de la función *Concatenar*. El paso (2.2) identifica si una variable de entrada está negada o no. El paso (2.3) inserta en c la variable booleana de entrada analizada en turno. Finalmente el paso (3) termina de encerrar entre paréntesis el término canónico booleano creado. La función *Tipificar* es también fácil de especificar:

$$Tipificar(U_{ij}) = \begin{cases} "\wedge" & \text{si } \Theta_{i,j} = \uparrow \\ "\vee" & \text{si } \Theta_{i,j} = \downarrow + 1 \end{cases}$$

Debido a que la red neuronal canónica está bien cimentada en la lógica, es relativamente sencillo pensar en algún algoritmo que permita extraer su conocimiento. Para revelar el conocimiento, el algoritmo debe analizar el estado de cada capa neuronal y su vez el estado de cada neuroide en cada capa. El exhibir el conocimiento de una red neuronal canónica resulta de gran importancia pues al examinarlo se puede explicar el por qué se llegó a cierta solución dadas ciertas condiciones de entrada. A continuación se presenta un sencillo algoritmo de extracción del conocimiento para la RNCU a través del cual se obtienen dos cadenas de caracteres legibles en los símbolos de la lógica de predicados que representan la hipótesis y la tesis que siguió la RNCU para llegar al patrón de salida en U_j :

1. $t = ""$, $h = ""$, $j = 1$
2. Hacer mientras que $j \leq |U_j|$
 - 2.1. Si $j > 1$ entonces t -Concatenar(t , " \wedge "), h -Concatenar(h , " \wedge ")
 - 2.2. Si $\phi_j = 0$ entonces t -Concatenar(t , " \neg "), h -Concatenar(h , " \neg ")
 - 2.3. t -Concatenar(t , " $U_{j,k}$ ")
 - 2.4. h -Concatenar(h , "(")
 - 2.5. $k = 1$
 - 2.6. Haz mientras que $k \leq |U_{2,j}|$
 - 2.6.1. Si $w_{3,j,k} \neq 0$ entonces
 - 2.6.1.1. Si $k > 1$ entonces h -Concatenar(h , Tipificar($U_{3,j}$))
 - 2.6.1.2. h -Concatenar(h , Descifrar($U_{2,j,k}$))
 - 2.6.2. $k = k + 1$
 - 2.7. h -Concatenar(h , ")")
 - 2.8. $j = j + 1$

Al finalizar el algoritmo de extracción la cadena de caracteres h y la cadena de caracteres t contienen respectivamente una expresión booleana con la hipótesis y la tesis que llevaron a la solución actual arrojada por la RNCU. El paso (1) inicializa h , t y el índice j para el neuroide analizado de la capa U_j . En paso (2) analiza cada neuroide de la capa U_j obteniendo de él qué tipo de función lógica realiza, \wedge o bien \vee , y con qué neuroides de la capa U_2 mantiene conexión. El paso (2.1) inserta un símbolo \wedge en caso de que exista más de un neuroide en U_j para encadenar los elementos de la tesis y la hipótesis en una expresión conjuntiva. En (2.2) se analiza la salida de neuroide en turno de la capa U_j y en caso de que la salida sea falsa se niega el elemento de la tesis y la hipótesis correspondiente para que la expresión sea válida con respecto a la interpretación que dio origen la salida ϕ de la RNCU. En (2.3) se añade a la tesis el neuroide analizado de la capa U_j . El (2.4) inicia a encerrar entre paréntesis el elemento de la hipótesis que da origen a la salida del neuroide analizado. En (2.5) se inicializa el índice del vector de pesos sinápticos del neuroide analizado de la capa U_j y que servirá también para conocer con qué neuroides de la capa U_2 se mantiene conexión. El paso (2.6.1) se repite para cada uno de los elementos del vector de pesos sinápticos del neuroide de U_j analizado y se ejecuta solo en caso de que el elemento analizado de la capa U_j mantenga conexión con el k -ésimo elemento de U_2 . En el paso (2.6.1.1) si existe más de un neuroide lógico de U_2 con el que se guarda conexión entonces se inserta ya sea un \wedge o un \vee dependiendo del tipo de neuroide analizado de U_j , esto permite ir

encadenando los términos canónicos del polinomio en la hipótesis. En (2.6.1.2) se descifra el neuroide de U_2 con el que se guarda conexión y el término canónico resultante se une al final de la hipótesis. El paso (2.7) termina de encerrar entre paréntesis el elemento de la hipótesis creado y el (2.8) incrementa el índice para analizar el siguiente neuroide de la capa U_3 .

Una vez obtenida la hipótesis h que da origen a la tesis t de salida de la red, puede especificarse un algoritmo de dilucidación para la RNCU que exponga cómo se llegó a una solución dado un patrón de entrada. El algoritmo de dilucidación puede ser tan sencillo como:

1. Concatenar("Si, si ", h , " entonces ", t , "; y ", Exhibir(U_1), " entonces ", t)

El único paso existente del algoritmo hace una concatenación de la hipótesis h , la tesis t y la entrada a la red a la manera de silogismo en modus ponendo ponens. La función Exhibir asigna a la capa U_1 una cadena de caracteres legible con la descripción del estado de esa capa neuronal usando la simbología de la lógica de predicados. A continuación se propone una sencilla especificación de Exhibir a través de un algoritmo:

1. $e \leftarrow (" , j-1$
2. Hacer mientras que $j \leq |U_1|$
 - 2.1. Si $j > 1$ entonces $e \leftarrow \text{Concatenar}(e, "\wedge")$
 - 2.2. Si $u_{1,j} = 0$ entonces $e \leftarrow \text{Concatenar}(e, "\neg")$
 - 2.3. $e \leftarrow \text{Concatenar}(e, "U_{1,j}")$
 - 2.4. $j \leftarrow j+1$
3. $e \leftarrow ")"$

Al completar el algoritmo la cadena de caracteres e contiene una expresión booleana verdadera bajo la interpretación de entrada actual. El paso (1) inicializa e para encerrar entre paréntesis la expresión booleana, además se inicializa el índice j para denotar al neuroide analizado de la capa U_1 . El paso (2.1) al (2.4) se ejecutan para cada uno de los neuroides de U_1 . En (2.1) si existe más de un neuroide en U_1 se inserta un \wedge para encadenar cada uno de las entradas. En (2.2) se verifica si la entrada al neuroide es 0, en caso de serlo, se introduce un \neg para hacer verdadera la expresión. El paso (2.3) introduce el neuroide analizado a la expresión y el (2.4) incrementa el índice con el fin de analizar el siguiente neuroide. Finalmente, el algoritmo en el paso (3) termina de encerrar la expresión booleana creada.

Así como en esta sección se han propuesto a manera de ejemplificación algunos algoritmos para extraer el estado de un NL, el estado de una capa neuronal canónica, y el conocimiento de una red neuronal canónica, es igualmente factible agrupar el conocimiento de la red a través de varios ciclos de ejecución para observar cómo ha ido ocurriendo el adiestramiento de una red neuronal canónica. El obtener el adiestramiento pudiera resultar valioso si se desean mantener estudios estadísticos o el historial acerca de cómo han ido alterándose los conocimientos de la red. Una vez más, los algoritmos planteados han tenido como único fin el demostrar las propiedades teóricas del modelo de la red neuronal canónica a través de la RNCU. Se espera que el lector habrá hecho ya sus conclusiones al respecto de las bondades del modelo y que adecue los algoritmos a sus propias perspectivas.

6. RED NEURONAL CANÓNICA VS. MODELOS CLÁSICOS

6.1 NEUROIDES

En general, los neuroides de los modelos clásicos han sido inspirados en primer lugar en el aspecto fisiológico y en ocasiones anatómico de las neuronas biológicas. También se han considerado pautas tomadas de otros modelos dinámicos o físicos similares, sin embargo, son mucho más frecuentes los casos en los que el modelo de neuroide es especificado en base a situaciones experimentales, intuitivas y empíricas. Afortunadamente, a pesar de sus bases teóricas no muy sólidas, muchos modelos han constituido sonados éxitos experimentales. Tales éxitos y el convencimiento de que las máquinas podrían autoregularse simulando el sistema nervioso de los seres vivos, han permitido tanto la difusión como la continuidad en la investigación en torno al tema. Desgraciadamente, el empirismo ha provocado que muchos éxitos en los laboratorios no hayan sido aún explotados en la vida cotidiana pues en la gran cantidad de los casos se desconoce o duda del funcionamiento del neuroide como elemento mínimo de procesamiento y de cómo se puede obtener una salida coherente a partir de su conexión con otros neuroides.

El neuroide lógico ha sido inspirado en primer lugar en el aspecto puramente fisiológico de las neuronas biológicas. Debido a que el humano posee el sistema nervioso más desarrollado que conocemos y a que sabemos que sus funciones de razonamiento se expresan directamente a través de procesos lógicos elementales, el neuroide lógico ha sido inspirado en segundo término en la lógica matemática simulando precisamente tales procesos lógicos elementales. Gracias a sus sólidas bases teóricas en la lógica, el mecanismo de trabajo del neuroide lógico puede ser claramente comprendido. Debido a su comprensión, se puede imaginar con relativa facilidad el cómo interconectando a los neuroides lógicos se puede generar una salida coherente. Aún más, sus bases lógicas permiten demostrar formalmente cada una de sus propiedades, lo que da fiabilidad al diseñador

de la red para que lo aplicado en el laboratorio sea usado en lo práctico con la total seguridad de que seguirá funcionando con los mismos resultados.

En cuanto al vector de entradas, los neuroides de los modelos clásicos no han demostrado de manera formal que un determinado número de entradas hará que el neuroide funcione de tal manera que favorezca siempre a la solución de un problema. Por el contrario, al señalar las propiedades del neuroide lógico se ha dejado entrever que puede calcularse el número de entradas sabiendo sobre qué variables de entrada en específico se requiere que ejecute una función lógica determinada.

Otra diferencia con respecto al vector de entradas es que los neuroides de los modelos clásicos proponen que este vector esté formado por elementos que son números reales no acotados, mientras que los neuroides lógicos por definición pueden asumir solo los valores 0 y 1. En este punto es necesario hablar de ventajas y desventajas de ambos enfoques. La desventaja de los neuroides en los modelos clásicos es la complejidad debida al rango de valores de entrada que pueden asumirse al no existir acotaciones, sin embargo esto puede constituir una ventaja pues pueden tratarse problemas analógicos de manera directa. La ventaja de los neuroides lógicos es que al existir solo dos valores posibles de entrada, el modelo se simplifica enormemente, sin embargo la desventaja es que si se sigue este enfoque los problemas analógicos deben discretizarse antes de poder ser tratados. Futuras investigaciones estarán dirigidas a comprobar que el comportamiento y propiedades del neuroide lógico permanecen sin cambio si las entradas pudieran asumir valores reales comprendidos entre 0 y 1. Esto último podría resultar valioso para el tratamiento directo de problemas analógicos, en específico aquellos problemas que caen en el ámbito de la probabilidad, de la estadística y en especial de la lógica difusa.

El vector de pesos sinápticos es un punto de gran relevancia pues a través su modificación es que una red aprende. La mayoría de los modelos de redes neuronales artificiales han propuesto que los valores asumibles por el vector de pesos sean números reales. Esto tiene como desventaja que debido a que se trabaja con números reales la cantidad de cambios posibles que pueden efectuarse es teóricamente infinita. Sin embargo, aún cuando la cantidad de cambios fuera infinita, no existiría problema alguno si el modelo de red demostrara formalmente que determinados cambios asegurarán siempre la convergencia a la solución; desgraciadamente este no es el caso en muchos modelos. El modelo del neuroide lógico simplifica en gran medida la alteración al vector de pesos sinápticos pues por definición cada uno de los elementos del vector puede asumir tan solo tres valores 0 para hacer nula una conexión, 1 para hacerla excitatoria y -1 para hacerla inhibitoria.

En cuanto al valor umbral, la función de integración y la función de transferencia no existe mucha diferencia entre los neuroides usados en los modelos clásicos y la red neuronal canónica. Tal vez lo más significativo es que los neuroides de los modelos clásicos usan generalmente valores reales, mientras que el neuroide lógico usa solo los valores 0 y 1. Como ya se ha mencionado con anterioridad, futuros estudios estarán enfocados a comprobar que el comportamiento y propiedades del neuroide lógico permanecen sin cambio si la función de transferencia asume valores reales comprendidos entre 0 y 1. Recordemos también que el neuroide lógico en su definición contempla el uso de una función de sumación temporal, situación que puede ser muy útil para

el modelado los fenómenos de plasticidad neuronal. Futuros escritos serán dedicados al comportamiento del neuroide lógico en la simulación de la plasticidad neuronal a partir de la función de sumación espacial.

Resumiendo, la diferencia radical entre los neuroides usados en los modelos clásicos y el neuroide lógico es que a través del modelo del neuroide lógico se puede conocer de antemano cual será el comportamiento del neuroide pues sus propiedades están formalmente demostradas, mientras que en gran parte de los modelos clásicos el comportamiento de los neuroides no se ha definido con exactitud debido principalmente a la falta de fundamentos teóricos contundentes.

6.2 MODELO

La red neuronal canónica ha sido sustentada tanto en el aspecto anatómico del sistema nervioso como en los fundamentos teóricos de la lógica matemática. Gracias a sus sólidas bases teóricas en la lógica, el mecanismo de trabajo de una red neuronal canónica puede ser claramente comprendido. Por su comprensión, se pueden diseñar con relativa facilidad redes neuronales artificiales que puedan converger adaptativamente a la solución de problemas. Aún más, sus bases lógicas permiten demostrar formalmente cada una sus propiedades, lo que da fiabilidad al diseñador de la red para que lo aplicado en el laboratorio sea usado en lo práctico con la total seguridad de que seguirá funcionando con los mismos resultados.

En cuanto al número de capas neuronales, los modelos clásicos de redes neuronales artificiales han usado regularmente tres capas tal y como el modelo de la red neuronal canónica. Sin embargo, a diferencia de los modelos clásicos, el modelo de la red neuronal canónica ha demostrado que se requieren exactamente dos capas para lograr la convergencia a la solución de un problema. A estas dos capas se les añade una primera capa cuya función es de proyección y normalización de los datos de entrada. Aún más, en el modelo de la red neuronal canónica se ha demostrado formalmente el número de neuroides que se requieren en cada capa neuronal para asegurar la convergencia a la solución de un problema. Estos datos no han sido concebidos para los modelos clásicos.

Los modelos clásicos en general han propuesto el tipo de neuroides usados en las capas de sus redes basándose en muchas ocasiones, además de los sustentos anatómico-funcionales del sistema nervioso, en nociones intuitivas o experimentales. La red neuronal canónica ha demostrado formalmente el porqué determinado tipo de neuroide es necesario en cierta capa neuronal, haciendo énfasis en que tal configuración asegura siempre la convergencia a la solución de un problema.

Muchos de los modelos clásicos, al igual que la red neuronal canónica, proponen una función de interconexión que establezca una topología divergente/convergente entre las capas neuronales. La mayoría de los modelos clásicos sugieren incluso una función de interconexión total al igual que la red neuronal canónica universal. Sin embargo, los modelos clásicos han establecido esta función de interconexión en base a nociones anatómicas o intuitivas sin responder con una demostración contundente al porqué era necesaria una función de interconexión de tales características. Por el contrario, la formulación del modelo de la red neuronal canónica ha esclarecido todas estas incógnitas a la luz de la lógica matemática.

La diferencia radical entre los modelos clásicos y la red neuronal canónica es que a través esta última se puede predecir cual será el comportamiento de la red pues sus propiedades están formalmente demostradas, mientras que en gran parte de los modelos clásicos el comportamiento de la red no se ha definido con exactitud debido principalmente a la falta de fundamentos teóricos.

Es evidente que la red neuronal canónica es un modelo resultante de la aplicación directa de los resultados obtenidos en la definición del neuroide lógico. De hecho, aunque puede ser usado en la práctica, el modelo propuesto de la red neuronal canónica universal fue más bien desarrollado para demostrar las facultades potenciales del neuroide lógico. Futuros reportes de investigación serán dedicados a describir otros modelos de redes neuronales canónicas con ligeras modificaciones a la red neuronal canónica universal que pudieran incrementar su ámbito de aplicación, o bien optimizar el rendimiento global de la red. Algunas de estas modificaciones son la incorporación de una función de interconexión total-parcial con posibilidad de recurrencias y una cardinalidad variable en la capa U_2 .

El esquema de una función de interconexión total-parcial estaría compuesto de dos fases, en la fase inicial la red trabajaría con una función de interconexión total para aprender a responder a los patrones de entrada. En la segunda fase, una vez que se haya cumplido cierto criterio de aprendizaje, se aplicaría un algoritmo de simplificación de polinomios booleanos a su mínima expresión lo que permitiría reducir las computaciones hechas en la capa U_2 de la red. Observe que la primera fase asegura que la red tendrá la capacidad de aprender a responder bien a cualquier patrón de entrada, mientras que la segunda optimiza la operación de la red pasando de la forma canónica a la forma normal de un polinomio booleano. También será contemplado el uso de negaciones de los términos de la capa U_2 cuando esto optimice el funcionamiento global de la red.

En cuanto a la cardinalidad variable en la capa U_2 , se trata de un esquema en el que inicialmente la capa U_2 no tendría neuroides ya que estos se añadirían dinámicamente conforme se vayan requiriendo para responder correctamente a un patrón de entrada. La cardinalidad variable en la capa U_2 puede ser extremadamente útil para evitar la explosión combinatoria que supone la teoría de la red neuronal canónica debido a que el número máximo de neuroides en U_2 puede llegar a ser 2^n , por lo que evidentemente con una n de magnitud relativa no muy grande pudiera ser imposible con la tecnología actual implantar el modelo.

Finalmente, debido a que la red neuronal canónica prevé la posibilidad de recurrencias, es posible simular funciones lógicas que estén basadas en respuestas obtenidas anteriormente. De esta forma una red neuronal canónica podría ser capaz de realizar trabajos cíclicos, lo que jamás ha sido visto anteriormente por modelo clásico alguno. Por ejemplo, resulta paradójico que una red de los modelos clásicos no pueda contar siendo que el humano puede hacerlo a través de su red neuronal. Una red neuronal canónica es capaz de contar. Nuevamente, futuros reportes estarán enfocados a describir modelos de redes neuronales canónicas que son capaces de realizar trabajos cíclicos.

6.3 APRENDIZAJE

El aprendizaje de los modelos clásicos ha sido considerado en general como un tabú, pues si bien las redes experimentales han aprendido, se desconoce a ciencia cierta cómo es que esto ocurre. Una red neuronal canónica despeja cualquier duda acerca de su proceso de aprendizaje, lo que la hace diferente y ventajosa frente a cualquier otro modelo antes propuesto.

La función de inicialización fue concebida por los modelos clásicos en general como parte de un proceso intuitivo basado en hechos experimentales sin saber con exactitud si tales valores iniciales pudieran o no agilizar el aprendizaje de la red. Por el contrario, la función de inicialización de una red neuronal canónica es diseñada de tal forma que solo sea necesaria la modificación de los neuroides de la tercera capa neuronal para conseguir el aprendizaje. De esta forma la función de inicialización cumple su cometido de ser un agente que propicie la estabilidad operativa de la red durante sus primeras iteraciones.

En los modelos clásicos la función de aprendizaje es aplicada en la mayoría de los casos a todos los neuroides de la red. En la red neuronal canónica solo es necesario aplicar la función de aprendizaje a la capa eferente de la red, lo que implica una gran reducción de esfuerzo computacional requerido así como del tiempo consumido en hacer que la red encuentre una solución correcta. De hecho los cambios en la capa eferente no son necesariamente realizados a todos los neuroides de la capa, sino solo a aquellos en los que la salida no ha sido correcta lo que hace muy eficiente al proceso de aprendizaje. Aún más, las funciones de aprendizaje en los modelos clásicos usualmente requieren de múltiples ciclos de ejecución para hacer que la red responda bien a un solo patrón de entrada, mientras que la función de aprendizaje de la red neuronal canónica garantiza que se requiera de un solo ciclo para aprender la respuesta correcta a un patrón de entrada.

Los modelos clásicos no han demostrado formalmente cuántos ciclos se requieren para que se logre la convergencia a la solución de un problema, peor aún es que no se haya demostrado que se logrará siempre la convergencia. En otras palabras en los modelos clásicos no existe garantía alguna de que se llegará a una respuesta correcta. Por el contrario, al usar la red neuronal canónica se sabe de antemano que siempre se logrará la convergencia a la solución si es que esta existe. Además se conoce con exactitud el número de iteraciones requeridas para que esto suceda en el mejor y el peor de los casos.

Son raros los modelos clásicos que ofrecen alguna estimación acerca de cual es su capacidad máxima de almacenamiento de información. Con el modelo de la red neuronal canónica se sabe exactamente el número de patrones que pueden ser reconocidos y el número de funciones que pueden ser aprendidas.

Aunque existe un amplio esfuerzo de investigación dedicado a la extracción del conocimiento almacenado en las redes neuronales artificiales, ningún modelo clásico ha ofrecido en su definición la descripción de algún método o algoritmo que facilite el saber cómo es que se llegó a la solución de un problema. La extracción del conocimiento de una red neuronal canónica es relativamente sencilla debido a que tiene una representación directa en la lógica. Puede incluso seguirse paso a paso la evolución del aprendizaje de la red a través del almacenamiento del proceso de adiestramiento de la red neuronal canónica. Durante su definición se han propuesto sencillos algoritmos para la red neuronal canónica a través de los cuales se puede conocer con

exactitud desde el papel que desempeña cierto neuroide en la red hasta las reglas de inferencia que usa la red en su totalidad para llegar a una conclusión.

En cuanto al algoritmo de entrenamiento sugerido, tal vez la diferencia más notable entre la red neuronal canónica y los modelos de redes neuronales artificiales clásicos es que la primera no solo modifica los pesos sinápticos sino también el valor umbral para corregir el funcionamiento de los neuroides. Debido a que durante su aprendizaje los modelos clásicos solo modifican los pesos de sinápticos, es evidente por vacuidad que al menos para el caso digital en el que la función de transferencia es una función lógica, un neuroide lógico puede simular cualquier función lógica creada con los neuroides de los modelos clásicos. Esto mismo se enuncia por su relevancia en el Corolario 6.1.

Corolario 6.1 (Por definición de neuroide lógico y de la función de aprendizaje para la red neuronal canónica) Para el caso digital, un neuroide lógico puede simular cualquier función de un neuroide de los modelos clásicos.

Sin embargo, es demostrable que existe siempre al menos una función lógica que es simulable con un neuroide lógico y que no es simulable por un neuroide de los modelos clásicos (NMC) si la función de aprendizaje no altera los pesos sinápticos. Esto mismo es formalmente descrito en el Teorema 6.1 que se vale del Lema 6.1 y del Lema 6.2 para su demostración.

Lema 6.1 si $\Theta \leq 0$ entonces $x_1 \vee x_2$ no es simulable en un NMC

Demostración.

1. $\Theta \leq 0$
2. si $x_1 = 0$ y $x_2 = 0$ entonces $\tau(\alpha(\Gamma)) = 1$:
 - 2.1. $x_1 = 0$ y $x_2 = 0$
 - 2.2. $\alpha(\Gamma) = 0$
 - 2.3. $\alpha(\Gamma) \geq \Theta$
 - 2.4. $\tau(\alpha(\Gamma)) = 1$
3. si $x_1 = 0$ y $x_2 = 0$ entonces $x_1 \vee x_2 = 0$
4. $x_1 \vee x_2$ no es simulable en un NMC

Hipótesis

Hipótesis

De (2.1) por definición clásica de α

De (1) por (2.2)

De (2.3) por definición clásica de τ

Definición de \vee

De (2) y (3)

Lema 6.2 si $\Theta > 0$ entonces $\neg x_1 \vee \neg x_2$ no es simulable en un NMC

Demostración.

- | | |
|---|---|
| 1. $\Theta > 0$ | Hipótesis |
| 2. si $x_1 = 0$ y $x_2 = 0$ entonces $\tau(\Gamma) = 0$: | |
| 2.1. $x_1 = 0$ y $x_2 = 0$ | Hipótesis |
| 2.2. $\alpha(\Gamma) = 0$ | Definición clásica de α |
| 2.3. $\alpha(\Gamma) < \Theta$ | De (1) por (2.2) |
| 2.4. $\tau(\alpha(\Gamma)) = 0$ | De (2.3) por definición clásica de τ |
| 3. si $x_1 = 0$ y $x_2 = 0$ entonces $\neg x_1 \vee \neg x_2 = 1$ | Definición de \vee |
| 4. $\neg x_1 \vee \neg x_2$ no es simulable en un NMC | De (2) y (3) |

Teorema 6.1 existe f , tal f es simulable en un NL y f no es simulable en un NMC

Demostración.

- | | |
|--|-----------------------------|
| 1. $\Theta \leq 0$ o $\Theta > 0$ | Definición de Θ |
| 2. si $\Theta \leq 0$ entonces existe f , tal que f es simulable en un NL y f no es simulable en un NMC: | |
| 2.1. $\Theta \leq 0$ | Hipótesis |
| 2.2. $x_1 \vee x_2$ no es simulable en un NMC | De (2.1) por Lema 6.1 |
| 2.3. $x_1 \vee x_2$ es simulable en un NL | Teorema 4.1 |
| 2.4. $x_1 \vee x_2$ es simulable en un NL y $x_1 \vee x_2$ no es simulable en un NMC | De (2.3) y (2.2) |
| 2.5. existe f , tal que f es simulable en un NL y f no es simulable en un NMC | De (2.4) |
| 3. si $\Theta > 0$ entonces existe f , tal que f es simulable en un NL y f no es simulable en un NMC: | |
| 3.1. $\Theta > 0$ | Hipótesis |
| 3.2. $\neg x_1 \vee \neg x_2$ no es simulable en un NMC | De (3.1) por Lema 6.2 |
| 3.3. $\neg x_1 \vee \neg x_2$ es simulable en un NL | Teorema 4.1 |
| 3.4. $\neg x_1 \vee \neg x_2$ es simulable en un NL y $\neg x_1 \vee \neg x_2$ no es simulable en un NMC | De (3.3) y (3.2) |
| 3.5. existe f , tal que f es simulable en un NL y f no es simulable en un NMC | De (3.4) |
| 4. existe f , tal que f es simulable en un NL y f no es simulable en un NMC | De (1), (2) y (3) por casos |

El interesante resultado del Teorema 6.1 hace visible un pequeño detalle que puede ser considerado una gran limitante de los algoritmos de entrenamiento convencionales hallados de los modelos clásicos. De esta forma quedan manifiestos los inconvenientes del paradigma de la sola alteración de los pesos sinápticos como método de aprendizaje que puede englobar cualquier modificación adaptativa a las capas neuronales.

6.4 CONVERSIÓN A LA RED NEURONAL CANÓNICA

La red neuronal canónica no es un modelo cuya intención sea substituir a cualquier otro modelo propuesto con anterioridad. Al contrario la red neuronal canónica es mas bien un esfuerzo para cambiar el enfoque con

el que se ha visto hasta el momento el área de investigación en redes neuronales artificiales, de manera que sus procesos inherentes pasen de ser un tabú a ser una sólida base teórica con fundamentos en las neurociencias, la lógica y las matemáticas. De hecho, el modelo de la red neuronal canónica trata de destacar a las redes neuronales artificiales como un campo más inmerso en la lógica que en las áreas restantes.

No resulta difícil imaginar el cómo pudieran adecuarse los modelos clásicos al modelo de la red neuronal canónica para aprovechar todos los beneficios que pudiera ofrecer este nuevo enfoque. De hecho no hay mayor esfuerzo durante la conversión de un modelo clásico al modelo de una red neuronal canónica que el cambio en la conceptualización que ha prevalecido hasta el momento acerca de las redes neuronales artificiales.

Con respecto a la conversión de un modelo de neuroide al neuroide lógico, los aspectos más relevantes a tomar en cuenta son la asignación del valor umbral y el acotamiento de los valores que pueden asumir el vectores de entradas, el vector de pesos sinápticos y la función de transferencia. El acotamiento de los valores debe ser bien fundamentado de manera que antes de interconectar a los neuroides en una red se conozca con exactitud cuál será el comportamiento de los neuroides ante cualquier entrada. Lo más sugerible es recurrir a los valores usados por definición en el neuroide lógico, aunque pudiera aplicarse obviamente cualquier magnitud resultase proporcional a ellos.

Tocante a la asignación del valor umbral, ya se ha demostrado en diversos teoremas lo importante que resulta una asignación bien fundamentada y definida del valor umbral. Una asignación incorrecta puede entorpecer el funcionamiento de un neuroide y de la red en su totalidad. Se recomienda seguir la asignación del valor umbral propuesta en la definición del neuroide lógico o cualquier asignación que dé como resultado una magnitud proporcional.

En cuanto a la conversión de un modelo de red a la red neuronal canónica, lo esencial a considerar es la disposición de los neuroides en las capas, la función de interconexión y el algoritmo de aprendizaje. La disposición de los neuroides en las capas debe dejar de ser juzgado como un fenómeno aleatorio para el convertirse en un ordenamiento que facilite lo más posible el aprendizaje de la red. Lo aconsejable es disponer a los neuroides de manera que simulen la forma canónica o la forma normal de la lógica ya que esta disposición puede además facilitar la especificación de los algoritmos para la extracción del conocimiento.

La función de interconexión debe cumplir con dos objetivos asegurar la convergencia a la solución del problema y facilitar la expresión del problema buscando la minimización de uso de los recursos computacionales. En los casos en los que no se tienen nociones exactas acerca de como solucionar un problema lo más recomendable, tal y como ha sido formalmente demostrado en este trabajo, es el uso de una función de interconexión total durante la etapa de aprendizaje para evitar que ciertas variables no sean procesadas por los elementos de las capas neuronales. Sin embargo, funciones de interconexión parcial pueden dar buenos resultados si se conoce de antemano con precisión que el problema puede ser enfrentado procesando ciertas variables con determinados neuroides que cumplen con una función lógica específica.

Finalmente, el algoritmo de entrenamiento debe introducir cambios en la red solo en aquellos neuroides en los que se sabe con total certeza que tales cambios rectificarán la respuesta de la red. Aún más, se debe tener absoluta certeza de que los cambios realizados son los exactamente requeridos para conducir al aprendizaje en una sola iteración. Por lo tanto, el algoritmo de entrenamiento debe asegurar la convergencia a la solución del problema en un número especificado de pasos. Aunque puede usarse el algoritmo de aprendizaje propuesto para la red neuronal canónica, versiones optimizadas o especializadas pueden aplicarse a cada red en particular para ofrecer aún más ventajas.

de una red neuronal, se puede demostrar que el algoritmo de entrenamiento de una red neuronal, puede ser interpretado como un algoritmo de búsqueda en un espacio de estados. Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos. Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos.

Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos. Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos.

Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos. Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos.

Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos. Este algoritmo de búsqueda, se puede interpretar como un algoritmo de búsqueda en un espacio de estados, donde el estado de la red neuronal es el vector de pesos sinápticos.

7. CONCLUSIONES

Ha sido presentado un nuevo modelo de red neuronal llamado la red neuronal canónica que a diferencia de los modelos clásicos ha dejado claro su funcionamiento a través de demostraciones formales. Por sus sólidas bases teóricas, la red neuronal canónica ha respondido a la luz de la lógica a las preguntas fundamentales del diseño de una red neuronal, de qué tipos capas neurales debe constar la red, cuál debe ser la función de cada capa neural, cuántas capas neurales deben existir en la red, cuántos neuroides deben estar en cada capa neural, tales como qué tipo de interconexión debe prevalecer en la red, cómo deben ser inicializados cada uno de los neuroides de la red, qué modelo debe usarse en cada una de las capas de la red, qué tipo de valores debe usar el vector de pesos sinápticos, qué tipo de valores deben emplearse en el vector de entradas, qué tipo de valores debe asumir el valor umbral, qué algoritmo de entrenamiento se debe seguir como estándar, qué modificaciones deben realizarse en cada ciclo del algoritmo de entrenamiento, qué funciones de integración, entrenamiento y aprendizaje deben usarse.

Aún más, a través del modelo de la red neuronal canónica se garantiza que siempre existirá una convergencia a la solución de un problema y se sabe en cuantos pasos ocurrirá la convergencia tanto en el mejor como el peor de los casos. Esto es posible debido a que a través del modelo de la red neuronal canónica se conoce con toda precisión qué neuroides en específico deben modificarse y cómo deben modificarse para llegar a la solución. Además la red neuronal canónica es capaz de aprender la respuesta correcta a un patrón de entrada en una sola iteración.

Debido a que se conocen todas sus propiedades se puede saber exactamente cuántos conceptos pueden ser almacenados en la red. También puede extraerse el conocimiento de la red, de manera que es posible especificar métodos para justificar las respuestas que la red ha arrojado.

En futuros reportes se prevé enriquecer más el sustento teórico de la red neural canónica mostrando variantes en las que los neuroides pueden asumir valores analógicos en las funciones de transferencia y en las que los neuroides pueden tener funciones de integración con sumación tanto espacial como temporal para simular los procesos de plasticidad neuronal observados en los organismos biológicos.

Entre otros objetivos a cumplir en el futuro inmediato también se encuentran el proponer variantes de redes neurales canónicas con capas neurales asociativas de cardinalidad variable para evitar la posible explosión combinatoria a la que está expuesta esta capa de la red neural canónica. Se mostrarán modelos en los que es posible simular procesos cíclicos a través de topologías reverberantes, tales como el proceso de contar, si bien ya se ha dejado entrever en este escrito la forma en que podrían simularse tales procesos. Se buscarán métodos de aprendizaje didáctico, es decir guiados por reglas, lo que puede resultar de utilidad si se cuenta con parte o la totalidad del conocimiento extraído previamente de un experto para la solución del problema al que será enfrentada la red neural. Se propondrán modelos de redes neurales canónicas en los que exista la posibilidad de incluir términos canónicos negados, buscando con esto reducir el procesamiento total necesario en una red neural canónica. Se explorarán algoritmos de aprendizaje optimizado en el que la forma canónica simulada pueda reducirse a la forma normal, de manera que el computo requerido en las iteraciones pueda minimizarse de manera sustancial. Se investigarán algoritmos de creatividad que proporcionen a la red neural la posibilidad de concebir conocimiento por sí misma a partir de cuestionantes surgidas de ella misma. Se tiene pensado además sugerir algún arquetipo que explique la composición de los procesos cognitivos en los seres vivos a partir del modelo de la red neural canónica.

Se ha llegado a resultados interesantes a la par del desarrollo de la teoría de la red neural canónica. Tal vez el más interesante de ellos sea la demostración de que el paradigma de la sola alteración de los pesos sinápticos usado en los modelos clásicos para el adiestramiento pudiera inhibir el aprendizaje de ciertos conceptos por parte una red neural. También se han fundamentado teóricamente algunos resultados obtenidos de manera experimental por los modelos clásicos, tales como el porqué es necesaria una función de interconexión total entre capas neurales para ayudar la convergencia y el porqué son necesarias al menos dos capas neurales asociativas para facilitar también la convergencia.

Por sobre todo la teoría de la red neural canónica intenta ser un nuevo enfoque en el campo de las redes neurales que permita establecer cimientos firmes para la mejor comprensión de esta trascendental disciplina, ya que presumiblemente la exploración de los límites computacionales de las redes neurales puede llegar a tener impactos sin precedentes en las ciencias biológicas, las ciencias de la computación, las matemáticas, la lógica e incluso en nuestra vida cotidiana.

BIBLIOGRAFÍA

- [Anderson, Rosenfeld 1988] James A. Anderson, Edward Rosenfeld; "Neurocomputing: Foundations of Research"; MIT Press, 1988; Boston, USA.
- [Boole 1954] G. Boole; "An investigation of the laws of thought"; Dover Publishers, 1954; New York, USA.
- [Birkhoff, Bartee 1970] G. Birkhoff, T. C. Bartee; "Modern Applied Algebra"; McGraw Hill 1970; New York, USA.
- [Birkhoff, Maclane 1965] G. Birkhoff, S. Maclane; "A survey of modern algebra", 3th edition; The Macmillan Co; New York, USA.
- [Darpa 1988] "DARPA Neural Network Study"; Fairfax, VA: AFCEA International Press, November 1988, appendix E; USA.
- [Eberhart 1990] Russel C. Eberhart, Roy W. Dobbins; "Neural Network PC Tools: A Practical Guide"; Academic Press, 1990; San Diego, USA.
- [Edventures 1988] Edventures Holdings; "Release 1.0 Newsletter"; July-9-1989; New York, USA.
- [Fischler, Firschein 1987] Fischler, Firschein; "Intelligen, 1991; USA.cc: The eye, the Brain, and the Computer"; Addison-Wesley, April 1987; USA.
- [Gallant 1988] Stephen I. Gallant; "Connectionist Expert Systems"; Communications of the ACM, vol 31, no.2, February 1988, 152-69; USA.
- [Green 1988] Lee Green; "Neural Networks Still Waiting on the Brink"; Information Week, issue 186, September-12-1988; USA.
- [Grossberg 1982] Stephen Grossberg; "The adaptive brain" (Vols. I, II); Reidel Press 1982; Boston, USA.
- [Hebb 1949] Donald Hebb; "Organization of Behavior"; John Wiley & Sons 1949; New York, USA.

- [Hecht 1987] Robert Hecht-Nielsen; "Counterpropagation networks"; Proceedings of the IEEE First Conference on Neural Networks 1987.
- [Hohn 1961] F. E. Hohn; "Applied Boolean Algebra"; The Macmillan Company 1966; New York USA.
- [Hopfield 1985] John Hopfield, David Tank; "Neural Computation of Decisions in Optimization Problems"; Biological Cybernetics 52.
- [Huntington 1904] E. V. Huntington; "Sets of Independent Postulates for the Algebra of Logic"; Transactions American Mathematical Society, Vol. 5, 1904, 238-309; USA.
- [Nelson, Illingworth 1990] Marilyn McCord Nelson, W.T. Illingworth; "A practical guide to neural nets"; Addison-Wesley, 1991; USA.
- [Johnson 1989] R. Colin Johnson; "Neural Nose to Sniff Out Explosives at JFK Airport"; Electronic Engineering Times, issue 536, May-1-1989; USA.
- [Khanna 1990] T. Khanna; "Foundations of Neural Networks"; Addison-Wesley 1990; USA.
- [Kinoshita, Palevsky 1987] June Kinoshita, Nicholas G. Palevsky; "Computing With Neural Networks"; High Technology, May-1997; USA.
- [Kohonen 1984] Kohonen; "Self-Organization and Associative Memory"; Springer-Verlag 2nd ed.; Berlin, Germany.
- [Minsky, Papert 69] Marvin Minsky, Seymour Papert; "Perceptrons"; MIT Press, 1969; Cambridge, USA.
- [McCulloch, Pitts 1943] Warren S. McCulloch, Walter Pitts; "A Logical Calculus of Ideas Immanent in Nervous Activity"; Bulletin of Mathematical Biophysics, 1943; USA.
- [Mead 1989] C. Mead; "Analog VLSI and Neural Systems"; Addison-Wesley 1989; USA.
- [Minsky 1969] Marvin Minsky; "Perceptrons"; 1969.
- [O'Reilly 1989] Brian O'Reilly; "Computers That Think Like People"; Fortune, vol 119, no. 5, February-27-1989; USA.
- [Pao 1989] Y. Pao; "Adaptive Pattern Recognition and Neural Networks"; Addison-Wesley, 1989; USA.
- [Parker 1982] Parker; "Lemma Logic"; Paper, Stanford University, 1982; USA.
- [Rosenblatt 1957] Frank Rosenblatt; "The perceptron: A perceiving and Recognition Automaton"; Cornell Aeronautical Laboratory, 1957; USA.
- [Rumelhart, Hinton 1986] David E. Rumelhart, Geoffrey Hinton, Ronald J. Williams; "Learning representations by backpropagation errors"; Nature, 323, 1986.
- [Rumelhart 1989] David E. Rumelhart, James L. McClelland; "Parallel Distributed Processing" vols. I, II; The MIT Press, 1989; USA.
- [Shanon 1938] C. E. Shannon; "A symbolic Analysis of Relay and Switching Circuits"; Transactions of the AIEE, Vol 57, 1938, 713-23.
- [Simpson 1990a] Patrick K. Simpson; "Artificial Neural Systems"; Pergamon Press, 1990; New York, USA.
- [Simpson 1990b] Patrick K. Simpson; "Associative Memory Systems"; Proceedings of the

- [Sejnowsky, Rosenberg 1986] International Joint Conference on Neural Systems, January-1990; USA.
Terrence Sejnowski, Charles Rosenberg; "NETtalk: a Parallel Network that Learns to Read Aloud"; Johns Hopkins University Electrical Engineering and Computer Science Technical Report, JHU/ECS-8601, 1986.
- [Schwartz 1989] Tom J. Schwartz; "IJCNN '89"; IEEE Expert 1989; USA.
- [Tortora 1981] Gerard J. Tortora, Nicholas P. Anagnostakos; "Principles of Anatomy and Physiology", third edition; Harper & Row Publishers Inc, 1981; New York, USA.
- [Wasserman 1990] Philip D. Wasserman; "Neurocomputing: Theory and Practice"; Van Nostrand Reinold, 1990. New York, USA.
- [Wechsler 1991] Harry Wechsler, "Neural Networks for Perception" vol II.; Academic Press, 1991; San Diego, USA.
- [Werbos 1974] Paul Werbos; "Beyond regression: New tools for prediction and analysis in the behavioral sciences"; Ph.D dis., Harvard University, 1974; USA.
- [Whitesitt 1961] J. E. Whitesitt; "Boolean Algebra and its applications"; Addison-Wesley 1961; USA.

ILUSTRACIONES

Ilustración 2-1 Organización del sistema nervioso	26
Ilustración 2-2 Principales partes de la neurona.....	27
Ilustración 2-3 Conducción de un impulso desde un axón presináptico hacia una dendrita postsináptica.....	28
Ilustración 2-4 Circuitos de redes neuronales.....	29
Ilustración 2-5 Voltaje en la neurona durante el desencadenamiento de un potencial de acción.....	31
Ilustración 2-6 Voltaje en la neurona durante un potencial excitatorio postsináptico.....	32
Ilustración 2-7 Voltaje en la neurona durante un potencial inhibitorio postsináptico.....	32
Ilustración 3-1 Representación gráfica del modelo de un neuroide.....	38
Ilustración 3-2 Gráficas del comportamiento de las funciones de transferencia comúnmente empleadas.....	39
Ilustración 3-3 Esquema de una red neural de n capas, cada una de las cuales tiene m neuroídes.....	41
Ilustración 3-4 Red neural de dos capas, las cuales tienen una función de interconexión que conecta totalmente la salida de cada neuroide de la capa uno con todos los neuroídes de la capa dos, pero además, la salida de cada neuroide de la capa dos se conecta a la entrada de un neuroide de la capa uno. Esta función de interconexión es un híbrido entre interconexión total y recurrente.	43

TABLAS

Tabla 1.1 Resumen de postulados y teoremas básicos del álgebra booleana.....	10
Tabla 1.2 Interpretaciones de tres variables	12
Tabla 1.3 Evaluaciones para expresiones de dos variables booleanas con los operadores \wedge , \vee y \neg	13
Tabla 1.4 Cumplimiento del postulado 4.a de Huntington.....	14
Tabla 1.5 Cumplimiento del postulado 4.b de Huntington.....	14
Tabla 1.6 Resumen de postulados y teoremas básicos para el álgebra de predicados.....	14
Tabla 1.7 Términos canónicos conjuntivos y disyuntivos obtenidos a partir de las interpretaciones de tres variables.....	17
Tabla 3.1 Algunas aplicaciones comerciales exitosas de las redes neurales	48
Tabla 3.2 Funcionamiento y diferencias entre las técnicas de representación simbólica y las redes neurales...	49
Tabla 3.3 Principales modelos clásicos.....	51
Tabla 4.1 Cumplimiento de $\prod_{i=1}^j l_i = \prod_{i=1}^j l_i$ para $l_i = v_i$	58
Tabla 4.2 Cumplimiento de $\prod_{i=1}^j l_i = \prod_{i=1}^j l_i$ para $l_i = \neg v_i$	58
Tabla 4.3 Cumplimiento de $\prod_{i=1}^j l_i \vee l_{j+1} = \prod_{i=1}^j l_i \nabla l_{j+1}$ para $l_i = v_i$	58

Tabla 4.4 Cumplimiento de $\sum_{i=1}^j l_i \vee l_{j+1} = \sum_{i=1}^j l_i \nabla l_{j+1}$ para $l_i = \neg v_i$	58
Tabla 4.5 Cumplimiento de $\bigwedge_{i=1}^j l_i = \bigwedge_{i=1}^j l_i$ para $l_i = v_i$	62
Tabla 4.6 Cumplimiento de $\bigwedge_{i=1}^j l_i = \bigwedge_{i=1}^j l_i$ para $l_i = \neg v_i$	62
Tabla 4.7 Cumplimiento de $\bigwedge_{i=1}^j l_i \wedge l_{j+1} = \bigwedge_{i=1}^j l_i \Delta l_{j+1}$ para $l_i = v_i$	63
Tabla 4.8 Cumplimiento de $\bigwedge_{i=1}^j l_i \wedge l_{j+1} = \bigwedge_{i=1}^j l_i \Delta l_{j+1}$ para $l_i = \neg v_i$	63
Tabla 4.9 Neuroides ∇ y Δ obtenidos a partir de las interpretaciones de tres variables de entrada.....	64

TEOREMAS

Teorema 1.1 Principio permutatorio. $ \prod_{i=1}^n A_i = \prod_{i=1}^n A_i $	7
Teorema 1.2 Principio Combinatorio. $ 2^A = 2^{ A }$	8
Teorema 1.3 $ \Psi_{x_1, \dots, x_n} = 2^n$	11
Teorema 1.4 $ T = 2^n$	16
Teorema 1.5 para todo i , si $\xi(t, \Psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 1$ entonces $t = t_i$	19
Teorema 1.6 $ D = 2^n$	20
Teorema 1.7 $t \in \zeta$ si y solo si $\xi(\vee \zeta, \Psi_i) = 1$	20
Teorema 1.8 $\xi(\vee \zeta_1, \Psi_i) = \xi(\vee \zeta_2, \Psi_i)$ si y solo si $\zeta_1 = \zeta_2$	21
Teorema 4.1 Transferencia de ∇ . Si $(\bar{\omega}_i = -1$ si $l_i = \neg t_i$; y $\bar{\omega}_i = 1$ si $l_i = t_i)$ entonces $\prod_{i=1}^n l_i = \prod_{i=1}^n \bar{l}_i$	55
Teorema 4.2 Transferencia de Δ . Si $(\bar{\omega}_i = -1$ si $l_i = \neg t_i$; y $\bar{\omega}_i = 1$ si $l_i = t_i)$ entonces $\prod_{i=1}^n l_i = \prod_{i=1}^n \bar{l}_i$	60
Teorema 6.1 existe f , tal f es simulable en un NL y f no es simulable en un NMC	83

LEMAS

Lema 1.1 si $\xi(t, \psi_l = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 1$ entonces $t = I_{x_1}^n \wedge \dots \wedge I_{x_n}^n$	17
Lema 6.1 si $\Theta \leq 0$ entonces $x_1 \vee x_2$ no es simulable en un NMC	82
Lema 6.2 si $\Theta > 0$ entonces $\neg x_1 \vee \neg x_2$ no es simulable en un NMC	83

COROLARIOS

Corolario 1.1 (del Teorema 1.3) Existen 2^n sustituciones para una misma expresión booleana e.	12
Corolario 1.2 (del Teorema 1.3 y Corolario 1.1) Existen 2^n evaluaciones para una misma expresión booleana e.	12
Corolario 1.3 (del Teorema 1.4) $\neg T = 2^n$	16
Corolario 1.4 (del Lema 1.1). si $\xi(\neg t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 0$ entonces $\neg t = \neg t_1^* \wedge \dots \wedge \neg t_n^*$	18
Corolario 1.5 (del Teorema 1.5 y Corolario 1.4) para todo i, si $\xi(\neg t, \psi_i = \{(x_1, a_1), \dots, (x_n, a_n)\}) = 0$ entonces $\neg t = \neg t_i$.	19
Corolario 1.6 (del Teorema 1.5) $\xi(t, \psi) = 1$	19
Corolario 1.7 (del Corolario 1.5) $\xi(\neg t, \psi) = 0$	19
Corolario 1.8 (del Teorema 1.4). El número máximo de términos canónicos conjuntivos que puede tener un polinomio canónico disyuntivo es 2^n.	19
Corolario 1.9 (del Corolario 1.3). El número máximo de términos canónicos disyuntivos que puede tener un polinomio canónico conjuntivo es 2^n.	19
Corolario 1.10 (del Teorema 1.6) $C = 2^{2^n}$	20
Corolario 1.11 $\neg t_i \in \delta$ si y solo si $\xi(\wedge \delta, \psi) = 0$	21

Corolario 1.12 (del Teorema 1.7) para todo f , si f es una función booleana entonces f es representable por un polinomio booleano canónico disyuntivo..... 21

Corolario 1.13 (del Corolario 1.11) para todo f , si f es una función booleana entonces f es representable por un polinomio booleano canónico conjuntivo. 21

Corolario 1.14 (del Teorema 1.8) $\xi(\wedge\delta_1, \psi_i) = \xi(\wedge\delta_2, \psi_i)$ si y solo si $\delta_1 = \delta_2$ 22

Corolario 1.15 (del Corolario 1.12 y el Teorema 1.8) para todo f , si f es una función booleana entonces f es representable por un y solo un polinomio canónico disyuntivo..... 22

Corolario 1.16 (del Corolario 1.13 y el Corolario 1.14) para todo f , si f es una función booleana entonces f es representable por un y solo un polinomio canónico conjuntivo 22

Corolario 1.17 (del Corolario 1.15 y el Teorema 1.6) Existen exactamente 2^n funciones booleanas diferentes que pueden formarse a partir de n variables booleanas. 22

Corolario 4.1 (Del Teorema 4.1) Un ∇ es capaz de simular un término canónico disyuntivo. 63

Corolario 4.2 (Del Teorema 4.2) Un Δ es capaz de simular un término canónico conjuntivo..... 63

Corolario 4.3 (Del Teorema 1.4). Existen exactamente 2^n neuroides $\overset{\Delta}{\underset{1}{\Delta}} l_i$ canónicos distintos..... 64

Corolario 4.4 (Del Corolario 1.3). Existen exactamente 2^n neuroides $\overset{\nabla}{\underset{1}{\nabla}} l_i$ canónicos distintos 64

Corolario 4.5 (Del Teorema 1.5) Un Δ_i , evaluando sus entradas hace cierta a la interpretación ψ_i y falsa a las interpretaciones restantes. 65

Corolario 4.6 (Del Corolario 1.5) Un ∇_i , evaluando sus entradas hace falsa a la interpretación ψ_i y cierta a las interpretaciones restantes. 65

Corolario 4.7 (Del Corolario 4.5 y el Corolario 4.6) Un NL es capaz de simular $\neg(\overset{\Delta}{\underset{1}{\Delta}} l_i)$ 65

Corolario 4.8 (Del Corolario 4.5 y el Corolario 4.6) Un NL es capaz de simular $\neg(\overset{\nabla}{\underset{1}{\nabla}} l_i)$ 65

Corolario 5.1 (del Teorema 4.1 y Teorema 4.2) Una red neural canónica es capaz de simular un conjunto de polinomios booleanos en forma canónica. 68

Corolario 5.2 Capa U2 de una RNCU (del Corolario 4.1, el Corolario 4.2, el Corolario 4.3, el Corolario 4.4, el Teorema 1.4, el Corolario 1.3 y la definición de RNCU) $U_2 = T \circ U_2 = \neg T$ 71

Corolario 5.3 Capa U3 de una RNCU (de la definición de polinomio canónico y la definición de RNCU) $U_3 \subseteq D$ o $U_3 \subseteq C$ 72

- Corolario 5.4** Universalidad de la RNCU disyuntiva (Del Corolario 5.3 y el Corolario 1.12) para toda f , si f es una función booleana entonces f es representable por una RNCU disyuntiva 72
- Corolario 5.5** Universalidad de la RNCU conjuntiva (Del Corolario 5.3 y el Corolario 1.13) para toda f , si f es una función booleana entonces f es representable por una RNCU conjuntiva..... 72
- Corolario 5.6** (De la definición de NL, RNCU y λ) En el peor de los casos se requieren de $2 \times |U_s|$ modificaciones en una iteración del algoritmo de entrenamiento de una RNCU. 73
- Corolario 5.7** (Del Corolario 5.1, Corolario 1.15, Corolario 1.16, Teorema 1.3, la definición de RNCU y su algoritmo de entrenamiento) En el peor de los casos se requieren de $2^n - 1$ iteraciones para que una RNCU aprenda. 74
- Corolario 6.1** (Por definición de neuroide lógico y de la función de aprendizaje para la red neural canónica) Para el caso digital, un neuroide lógico puede simular cualquier función de un neuroide de los modelos clásicos. 82