

03063

12



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERIA DE LA
COMPUTACIÓN

DESARROLLO E INTERPRETACIÓN DE LA
RELACIÓN ENTRE LAS REDES NEURONALES Y
LOS SISTEMAS CANONICOS DE POST

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

PRESENTA:

ERNESTO ALEJANDRO LIMA SOLANA

DIRECTOR DE LA TESIS: M. EN C. ALBERTO ARTURO
HERRERA BECERRA

MÉXICO D. F.

TESIS CON
FALLA DE ORIGEN

2002



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

DEDICATORIA:

A MI MADRE MARGARITA, que con su ejemplo siempre me enseñó el camino del éxito tomando bases justas, sólidas y honestas.

A MI ABUELA SARA, que siempre se desveló por mí, creando cada día las bases necesarias para poder dedicarme sólo a mis estudios.

A MIS TÍOS MARY Y PEPÉ, que siempre me imprimieron ejemplo fiel ante el estudio, la superación y las metas que se deben alcanzar en la vida.

A ALONSO, que siempre me dio fuerzas para terminar la Tesis, preocupándose de cada detalle, permitiendo su ayuda que hoy esta tesis sea una realidad. Su ejemplo ante el estudio y la superación fue fuente de inspiración.

A mi asesor ALBERTO HERRERA que siempre estuvo al tanto de mi preparación docente para este trabajo.

Al INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y SISTEMAS(IIMAS), que me dio la posibilidad de cursar la maestría.

A TODOS MIS AMIGOS CUBANOS Y MEXICANOS: a Ginny Figueroa, a Casandra Hernández, a Eduardo y Ariadna Álvarez-Guerra, , a Normita y Orlando Romero, a Alejandro Ainslie y Familia, a Ernesto Linares, a Esteban y Antonia Vargas, y a todos que de una forma u otra me dieron ímpetu para concluir mis estudios de posgrado.

AGRADECIEMIENTOS:

En especial mucho agradezco a Alonso, que siempre estuvo al pendiente de la terminación de mis estudios de maestría, su apoyo fue significativo para poder llevar las materias y la tesis sin preocupación alguna.

A CONACyT que me dio la posibilidad gracias a su programa de intercambio de comenzar los estudios de maestría.

A Jorge Sánchez que me apoyó docente y profesionalmente permitiéndome fusionar los conocimientos teóricos con la práctica.

ÍNDICE

PÁGINA

INTRODUCCIÓN	1
CAPÍTULO 1 "ANTECEDENTES"	3
1.1 Introducción	3
1.2 Máquinas de Turing	4
1.3 Sistemas de Post	7
1.4 Ejemplos de sistemas de dos etiquetas	10
1.5 Equivalencia Turing-Post	14
CAPÍTULO 2 "REDES NEURONALES"	22
2.1 Introducción	22
2.2 Construcción de una neurona artificial	22
2.2.1 Morfología de una neurona	23
2.2.2 Modelo de neurona artificial	25
2.2.3 Modelo matemático de neurona artificial	26
CAPÍTULO 3 "PROPUESTA DE LA RED DE NEURONAS"	29
3.1 Introducción	29
3.2 Modelo de neurona	29
3.3 Modelo de red	30
3.3.1 Estímulos	31
3.3.2 Operación	32
3.3.3 Funcionalidad	36
CAPÍTULO 4 "SIMULACIÓN"	38
4.1 Introducción	38
4.2 Codificación	38
4.3 Simulación	39
4.3.1 Condiciones Iniciales	39
4.3.2 Representación gráfica de resultados	41
4.3.3 Análisis de resultados	41
4.4 Interpretación	46
4.4.1 Primer ciclo de cómputo	46
4.4.2 Segundo ciclo de cómputo	49
4.4.3 Tercer ciclo de cómputo	51
DISCUSIÓN Y CONCLUSIONES	55
REFERENCIAS BIBLIOGRÁFICAS	58

INTRODUCCIÓN

El presente trabajo describe las redes de neuronas artificiales en términos de los preceptos de la computación clásica. Se utilizarán modelos clásicos de computación como referencia para mostrar la equivalencia existente entre esos modelos clásicos de cómputo y las redes de neuronas artificiales; explicando cómo y por qué son consideradas un modelo de computación.

Cabe señalar que es importante considerar las redes de neuronas como un modelo de computación puesto que éstas representan una nueva y prometedora generación de procesamiento de información. Las redes neuronales son eficaces en tareas tales como el acoplamiento de patrones y la clasificación, la aproximación de funciones, la optimización, la cuantificación de vectores y para el agrupamiento de datos, mientras que las computadoras tradicionales, debido a su arquitectura, son ineficaces para estas tareas, específicamente para las de acoplamiento de patrones. Sin embargo, las computadoras tradicionales son más rápidas en tareas de computación algorítmicas que requieren de operaciones aritméticas precisas.

Las redes de neuronas artificiales, como las aquí propuestas, pueden simular un sistema de dos etiquetas; éste sistema de dos etiquetas está incluido dentro de la familia de los sistemas canónicos de Post y se caracterizan por borrar los dos primeros símbolos de una cadena. Para este fin se mostrará un ejemplo ilustrativo de la simulación y por tanto de la equivalencia existente entre los resultados obtenidos en la red de neuronas artificiales y el sistema de dos etiquetas. Es también sabido que entre la máquina de Turing y los sistemas canónicos de Post, y en especial los sistemas de dos etiquetas, existe una relación de equivalencia.

Entre los aspectos relativos al problema planteado y discutidos en la tesis figuran: la simulación del sistema de dos etiquetas por medio de una red neuronal, en la que se logra mantener los preceptos de un modelo de computación y la interpretación de los resultados finales derivados del proceso de cómputo, tanto desde el prisma del sistema de dos etiquetas como desde la red de neuronas.

Son cuatro los capítulos que conforman esta tesis. El primero está relacionado con los Antecedentes; en el mismo se explican las bases de la problemática a seguir. Se incluyen varias secciones como: una sobre máquinas de Turing, otra sobre los sistemas de Post, otra de ejemplos de sistemas de dos etiquetas y al final del capítulo se incluye una sobre la equivalencia entre Turing y Post.

El segundo capítulo presenta las redes de neuronas, éste explica los conceptos fundamentales sobre las redes de neuronas artificiales, morfología de las mismas, modelo de neurona artificial, entre otros.

En el tercer capítulo se hace la propuesta de la red de neuronas con la que se realizará el trabajo, se ofrece una caracterización de las mismas, explicándose los modelos de neuronas a usar, las señales de entradas en la red y las gráficas correspondientes a éstas.

En el cuarto capítulo se trata el tema de la Simulación, en el cual se muestra mediante tablas y gráficas la manera equivalente en que la red de neuronas trabaja y simula el sistema de dos etiquetas de Post. También se incluye en este capítulo una explicación de la codificación empleada, lo que permite una cómoda comprensión del análisis de equivalencia entre los sistemas de dos etiquetas de Post y la red de neuronas propuesta. Terminado el capítulo cuatro se muestran las conclusiones derivadas del presente trabajo.

El alcance del trabajo de tesis consiste en mostrar que la red de neuronas propuesta es equivalente al sistema de Post de dos etiquetas, éstos sistemas de Post también tienen una relación de equivalencia con las máquinas de Turing, por lo que se establece un vínculo entre la mencionada red de neuronas y la propia máquina de Turing, por medio de los sistemas de Post de dos etiquetas.

CAPÍTULO 1: ANTECEDENTES

1.1 Introducción

La teoría clásica de la computación tuvo su origen en los trabajos en los años '30s de los lógicos: Gödel, Turing, Church, Kleene, Post, entre otros. En esa época todavía no había computadoras como las que se conocen en la actualidad. Aunque el trabajo de estos pensadores, particularmente el de Turing, claramente anticipó el desarrollo de la moderna computadora digital de propósito general, la motivación principal que guió el desempeño de todos ellos fue formular y comprender el concepto de decisión o de un conjunto decisorio.

Cualquier sistema de cómputo creado por el hombre se estructurará tomando como componente principal las clases de funciones que el proceso de cómputo define. Desde un punto de vista formal, si una red de neuronas artificiales es una clase de sistemas de cómputo, entonces es importante estructurarla exactamente en los mismos términos utilizados para otros sistemas computacionales. Para ello es conveniente establecer una relación entre las redes de neuronas y los modelos clásicos de computación, a saber: sistemas canónicos de Post, y máquinas de Turing.

En 1956, S. C. Kleene mostró que la operación de una clase particular de redes de neuronas artificiales, basadas en el modelo de neurona propuesto por McCulloch y Pitts en 1943, puede ser descrita por medio del concepto, ahora clásico, de evento regular [1]. Este importante resultado señala que dicha clase de redes de neuronas pueden ser entendidas como una clase especial de autómatas finitos, los cuales M. Minsky denomina autómatas en partes [2] y, por consecuencia, estas redes pueden ser entendidas como una clase especial de computadora.

Estos trabajos fueron pioneros ya que después de ellos aparecieron una serie de reportes que señalan la relación de otras clases de redes de neuronas artificiales con los modelos clásicos de computación: los autómatas finitos y las máquinas de Turing.

Por ejemplo, se ha demostrado que las redes cuya estructura está dada en términos de pesos enteros tienen las mismas capacidades de cómputo que los autómatas finitos, es decir, estas dos clases de sistemas de procesamiento son equivalentes. Por el contrario, diversos estudios sugieren que redes con estructura basada en pesos racionales pueden ser equivalentes a la máquina de Turing [3].

Se ha mostrado que las redes equivalentes a las máquinas de Turing son, en general, muy grandes, estando formadas por cientos de neuronas.

Resulta interesante ahondar en el concepto de un método efectivo, o procedimiento efectivo. Se dice que un método o procedimiento llamado *M*, para poder llevar a cabo algún resultado deseado, se le denomina 'efectivo' o 'mecánico' cuando:

1. *M* se presenta en términos de un número finito de instrucciones exactas (cada instrucción se expresa por medio de un número finito de símbolos);
2. *M* siempre producirá, siempre y cuando se realice sin errores, el resultado deseado en un número finito de pasos;
3. *M* puede, en práctica o en principio, ser ejecutada por un ser humano sin ayuda de máquina alguna, salvo lápiz y papel;
4. *M* no exige conocimiento previo por parte del ser humano que la ejecute.

Un ejemplo conocido de método efectivo lo es la prueba de la tabla de verdad usada en tautología (en el campo de la lógica matemática). En la práctica, por supuesto, esta prueba no funciona para fórmulas con gran número de variables proposicionales. Aunque en principio puede utilizarse con éxito en cualquier fórmula del cálculo proposicional, siempre y cuando se cuente con el tiempo suficiente, una buena dosis de tenacidad, algo de papel y lápices.

1.2 Máquina de *Turing*

La máquina de *Turing* es una máquina formal, es decir sin cables ni componentes físicos. Su finalidad es definir los cálculos a partir de las operaciones más sencillas posibles y, utilizándola de forma adecuada, calcular de manera efectiva las funciones dadas. A las funciones que así pueden realizarse se les llaman funciones calculables o computables.

Este tipo de máquina consta hipotéticamente de una unidad de control capaz de interpretar las instrucciones recibidas, y de una cabeza lectora que permite leer el contenido de las casillas en que está dividida una memoria lineal, ilimitada en ambas direcciones de sus extremos.

El contenido de una celda de la cinta puede obtenerse por medio de una cabeza de lectura/escritura. Al inicio de la operación de la máquina, la cinta contiene una secuencia o cadena de entrada, escrita en algún código discreto de manera tal que sólo un símbolo se encuentre en una celda, rodeada en ambos lados, de secuencias infinitas de símbolos blancos. La cabeza se localiza en el símbolo del extremo izquierdo de la cadena de entrada.

En cada paso de su operación, la máquina lee el símbolo en la celda bajo la cabeza, verifica la situación del control finito y ejecuta tres operaciones: escribe un nuevo símbolo en la celda bajo la cabeza, mueve la cabeza hacia la celda de la

izquierda o hacia la de la derecha y cambia la situación del control finito. Estas transiciones se describen de manera determinante.

Cuando la operación alcanza una situación especial, denominada situación de parada, la máquina se detiene. El resultado del cómputo realizado corresponde a la secuencia de símbolos no blancos que se encuentren en la cinta.

La figura 1.1, que se muestra a continuación, representa la máquina de Turing hipotética.

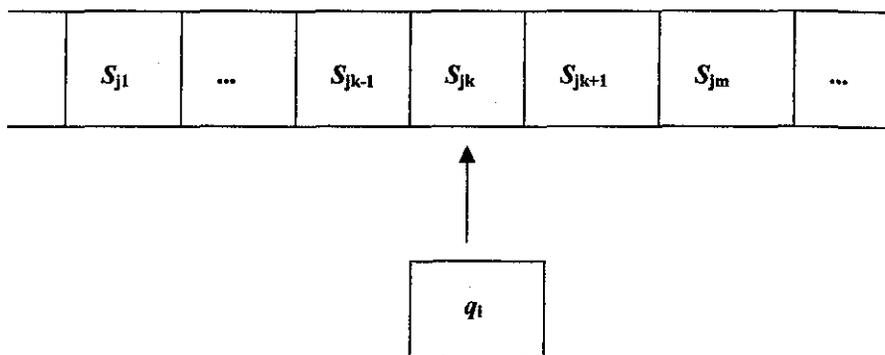


Figura 1.1 Máquina de Turing

La máquina en su funcionamiento pasa por diferentes estados denotados q_j en instantes t sucesivos. El argumento de la función que quiere calcularse, estará almacenado previamente en la memoria, y en el instante inicial, antes de que la máquina comience a funcionar, la cabeza lectora apuntará al símbolo más a la izquierda del argumento. A partir de ese momento la máquina realizará operaciones, que dependerán del estado en que ella se encuentre, y del símbolo que en ese momento lea la cabeza lectora. Con las operaciones mencionadas anteriormente, se podrán realizar las siguientes tareas: sustituir el símbolo leído por otro, pasar a leer el símbolo que está en memoria a la derecha del símbolo leído, pasar a leer el símbolo que está en memoria a la izquierda, o saltar directamente a ejecutar otra instrucción si se cumple una condición especificada; en todos los casos, y una vez ejecutada la tarea indicada, se pasa al estado que también se indica en la propia instrucción.

Estas instrucciones, o cuádruplas, pueden representarse e interpretarse como a continuación se muestra, con la suposición que la máquina de Turing M siempre está en movimiento. El carácter q se usa para indicar el estado de la máquina de Turing, s se utiliza para indicar el símbolo; el subíndice j es usado tanto en los estados q como en los símbolos s .

$q_i s_j R q_l$ si desde q_i se lee s_j , entonces se pasa a leer la casilla situada a su derecha y se cambia al estado q_l

$q_i s_j L q_l$ si desde q_i se lee s_j , entonces se pasa a leer la casilla a su izquierda y se cambia al estado q_l

donde: q_i indica un estado de la máquina tomado del conjunto de estados Q ; a su vez q_l es usado para indicar el estado siguiente, s_j indica uno de los símbolos que pueden aparecer en la memoria tomado del alfabeto S ; R es un símbolo que indica pasar a leer la casilla de memoria situada a la derecha; L es un símbolo que indica pasar a leer la casilla de memoria situada a la izquierda.

Se define una máquina de Turing M como un conjunto finito, no vacío, estructurada por cuádruplas, de forma que no contenga dos de ésta con los dos primeros símbolos correspondientes iguales. Indicaremos por $Q = \{q_i\}$ al conjunto de estados de la máquina, por $S = \{s_j\}$ el alfabeto de símbolos de la memoria.

En Q siempre existe un estado particular que suele indicarse por q_0 y se nombra estado inicial, en el que se supone que está la máquina al comenzar a operar. A las cadenas formadas por símbolos de este alfabeto se las denota como expresiones de memoria. Si en una expresión de memoria se incluye un único símbolo q_i , el cual indica un estado de Q , siempre y cuando no se sitúe a la derecha de cualquier otro símbolo, se obtendrá una expresión llamada descripción instantánea de la máquina M .

La descripción instantánea de la máquina M en el instante inicial podría representarse por:

$$\alpha = q_0 s_{i_1} s_{i_2} s_{i_3} \dots s_{i_n}$$

α : es una cadena de longitud arbitraria. Se usa para indicar la descripción instantánea, a su vez, q_0 representa el estado inicial, y $s_{i_1}, s_{i_2}, \dots, s_{i_n}$ representan los diferentes símbolos de la cinta. El estado donde se encuentra la cabeza de lectura y escritura es q_i , el símbolo que se ha leído es s_j .

Esto quiere decir que en la cinta encuentran un conjunto de símbolos, pero hay un estado particular donde se halla la cabeza de lectura y la unidad de control.

En cualquier otro instante una descripción instantánea sería de la forma:

$$\alpha = s_{j_1} \dots s_{j_k} q_i s_j s_{j_1} \dots s_{j_m}$$

Para entender cómo funciona una máquina de Turing, se analizará cómo se realizan las transiciones de unas descripciones instantáneas a otras.

Dada una descripción instantánea, en la que aparezcan el par de símbolos $q_i s_j$ (dado un estado y un símbolo, el movimiento puede ser a la derecha o la izquierda), se busca la cuaterna de la máquina que comience con estos dos símbolos y se opera en consecuencia, es decir, para cada uno de los tipos de cuádruplas, se construye la descripción instantánea siguiente, β , como se indica a continuación:

para $q_j s_i R q_i$	sería $\beta = s_{j_1} \dots s_{j_{k-1}} s_{j_k} q_i s_{j_{k+1}} \dots s_{j_m}$
para $q_j s_i L q_i$	sería $\beta = s_{j_{k+1}} \dots q_i s_{j_k} s_{j_{k-1}} \dots s_{j_1}$

Si no existe en M ninguna cuaterna que comience por $q_i s_j$, se diría que la descripción instantánea α es terminal. En cualquiera de los casos anteriores se dice que de la descripción instantánea α se pasa a la β mediante la máquina de Turing M , lo que se denota: $\alpha = \beta(M)$. Se le llama computación o cálculo de una máquina de Turing M a la sucesión finita de descripciones instantáneas $\alpha_1, \alpha_2, \dots, \alpha_n$, de forma que para $1 < i < n$, sea $\alpha_i = \alpha_{i+1}(M)$, y sea α_n , una descripción instantánea terminal.

Debido a la naturaleza dinámica de las redes, el modelo de la máquina de Turing parece ser el más adecuado para establecer la relación de equivalencia. Sin embargo, esto no es fácil de llevar a cabo de manera directa. Así conviene buscar el vínculo deseado de forma indirecta, por medio de modelos de computación equivalentes al de Turing. De éstos los más prometedores resultan ser las máquinas de registros y los sistemas canónicos de Post.

Lo interesante de la máquina de Turing es que para cualquier procedimiento computacional bien definido existe una máquina universal Turing capaz de simular una máquina que ejecute dichos procedimientos[2].

1.3 Sistemas de Post

Los sistemas lógicos se valen de un método para llevar algo a cabo. Por medio del método y de los símbolos matemáticos se lleva una lógica específica en la obtención de resultados.

Dentro de los sistemas lógicos se encuentran los sistemas formales de Post, dentro de los sistemas formales de Post se hallan los sistemas canónicos de Post y dentro de éstos se encuentran los sistemas de dos etiquetas. Los sistemas de dos etiquetas, por tanto, son sistemas canónicos de Post que varios libros definen como: sistemas tag (en lengua inglesa).

Algunos Sistemas Lógicos pueden representarse por medio de sistemas de Post; es decir, los que cumplan con la definición de sistemas canónicos de Post y estos sistemas lógicos, a su vez, pueden representar la máquina de Turing. Mientras que Turing y Church hacen énfasis en la función y en las transiciones, Post se va al lenguaje y lo hace con un método lógico.

Un sistema canónico es un sistema lógico. A su vez, un sistema de dos etiquetas es un sistema canónico de Post, por lo que puede decirse que una producción es una regla que transforma cadenas, una regla que dice textualmente cómo extraer una cadena y reorganizar sus partes. Quizás borrando algunas y agregando otras.

Un sistema canónico es un sistema lógico especificado de la siguiente manera:

- (1) Un alfabeto A .
- (2) Algunos axiomas (cadenas de A).
- (3) Algunas producciones cuyas cadenas constantes son cadenas en A .

Cualquier sistema formal puede reducirse a un sistema canónico de Post con un simple axioma y sólo producciones de la forma simple: $g\$ \rightarrow \h , conocida como producción normal [2]. El símbolo $\$$ será usado para representar cadenas.

Un sistema de dos etiquetas es un sistema canónico normal de Post que satisface las siguientes condiciones: Si $A = \{a_1, a_2, \dots, a_n\}$ es el alfabeto del sistema y $g_i\$ \rightarrow \h_i ($i=1 \dots n$), el conjunto de producciones, P se define como un número entero positivo, g_i es la cadena antecedente, en cambio h_i es la cadena consecuente, entonces:

- (1) Todas las cadenas antecedentes constantes g_i tienen la misma longitud P .
- (2) La cadena consecuente h_i depende sólo de la primera letra de la g_i asociada.

Debido al hecho de que h_i del consecuente es determinada sólo por la primera letra de g_i y de que el número de las letras en las g_i es una constante de P , los sistemas de dos etiquetas tienen que operar así:

- (1) Leer la primera letra de g_i .

(2) Borrar las dos primeras letras de inicio de la cadena.

(3) Agregar la cadena al consecuente asociado h_i al final de cada cadena.

Los sistemas de dos etiquetas son un tipo de sistemas de reescritura.

Un sistema z de dos etiquetas, con z como entero positivo fijo, es un cálculo asociado con una transformación del alfabeto A a A^* ; el conjunto de palabras de longitud finita trabaja en A . La notación A indica el alfabeto y la notación A^* indica el lenguaje universal. P_x representa una producción de la forma: $g_i\$ \rightarrow \$ h_i$.

Un paso en el cómputo consiste en realizar las siguientes tres operaciones, ilustradas abajo para $z=2$.

- a , primera letra de la palabra antecedente ofrecida al sistema de dos etiquetas, se memoriza;
- las primeras letras zetas de la palabra son borradas;
- P_x , la producción asociada con " a ", se adhiere al final de lo que queda de la palabra antecedente.

Este proceso es repetido para la palabra así obtenida, hasta que la palabra antecedente tenga menos letras que las zetas, o la primera letra de la palabra antecedente sea una letra de parada.

En el caso de los sistemas de Post el resultado del cómputo se traduce como impotencia o imposibilidad para seguir aplicando más producciones, por lo que el proceso de transformación de cadenas se detiene, pudiendo establecer una relación entre la cadena inicial y la resultante, y poder así ver el proceso de transformación.

Para Post fue muy importante la forma que se le daba al antecedente, por lo que estas producciones terminaron por denominarse producciones de Post.

Un sistema canónico de Post es monogénico si para cada cadena S existe como máximo una nueva cadena S' que pueda producirse a partir de ella en un solo paso. Sin duda alguna, un sistema de dos etiquetas es monogénico, ya que para cualquier cadena, lo que suceda con ésta, depende sólo de su primera letra[2].

La máquina, en este caso, sistema de dos etiquetas, representa el proceso de cómputo por medio de producciones

Las máquinas programables con dos registros pueden computar cualquier función recursiva al ejecutar los programas[2].

Es vital aclarar en el presente escrito que la formulación más general de Post permitía que cada producción tuviese varios antecedentes[2]. De igual manera, en el caso de las formulaciones más generales, Post permitió que dos de las cadenas en el antecedente fuesen iguales. El resultado fue que la utilización de la regla sólo se restringía a una cadena o a un teorema en los cuales hubiese una repetición especial de alguna subcadena o variable en dos lugares del antecedente.

Es preferible prohibir los antecedentes de esta naturaleza, no porque se pretenda restringir la generalidad de los sistemas, sino porque está de la parte del ejecutor del problema establecer qué es permisible y que no lo es, y para los fines del presente trabajo y de las características del sistema de dos etiquetas no es conveniente este tipo de antecedentes.

1.4 Ejemplos de sistemas de dos etiquetas

A continuación se muestran las producciones asociadas al sistema de dos etiquetas-2 con que se trabajará en este escrito de tesis:

$a \rightarrow b$

$b \rightarrow bc$

$c \rightarrow !$

La razón por la que se seleccionó el sistema de dos etiquetas mostrado arriba es porque resulta fácil de comprender. Éste consta de tres elementos o caracteres en su alfabeto, a saber: a, b y c y tres reglas de producción. Como se detalla en las producciones anteriores, a produce b, posteriormente b produce bc y c detiene el proceso, siendo ésta la condición de parada.

Después de un análisis con varias cadenas de entrada se llega a las siguientes conclusiones y reglas generales para este sistema de dos etiquetas (se denota a w como la cadena residual obtenida o también suele llamársele sufijo de la cadena, una vez que se eliminen los dos primeros elementos, a saber: a y x):

1- $axw, |w| > 0, x \text{ es } a, b \text{ o } c$

$axw \rightarrow wb$, pues como se conoce en el sistema de dos etiquetas $a \rightarrow b$ y w será la palabra restante o que quede, pues son borrados los dos primeros caracteres.

La norma en w significa que, el sufijo de la cadena siempre deberá ser mayor que cero, pues de lo contrario la cadena estaría formada por dos caracteres únicamente: a y x , los cuales serían borrados y no habría cadena alguna; por esta razón se necesita trabajar con cadenas formadas con al menos tres caracteres. A los símbolos se le puede asociar un valor numérico. La norma es unívoca, se obtiene sumando los símbolos (considerando cada símbolo con un valor numérico asociado de uno).

- 2- $bxw \rightarrow wbc$, como ya se conoce en el sistema de dos etiquetas $b \rightarrow bc$ y w será la letra restante, al ser borradas las dos primeras.
- 3- $cxw \rightarrow w$, se detiene el proceso, pues como se sabe en el sistema de dos etiquetas el comenzar con c en la palabra antecedente provoca que se detenga el proceso.

El sistema de dos etiquetas recién presentado, será la guía para la simulación.

Si se recuerdan las condiciones y características del sistema de dos etiquetas, al menos se tendrán para el análisis cadenas formadas por tres letras[2].

Ejemplos con diferentes cadenas de entrada únicamente con tres caracteres. El Símbolo " \rightarrow " se usa como indicador de transformación de cadenas. El conjunto de producciones del sistema de dos etiquetas ofrecido como ejemplo, se toma como guía o referencia, como ya se planteó.

$bbb \rightarrow bbc \rightarrow cbc \rightarrow c!$ (Se detiene el proceso pues c es la letra de parada)

$bba \rightarrow abc \rightarrow cb$

$baa \rightarrow abc \rightarrow cb$

$aaa \rightarrow ab$

$aab \rightarrow bb$

$bbc \rightarrow cbc \rightarrow c!$ (Se detiene el proceso al encontrar c , letra de parada)

A continuación se proporcionan ejemplos de cadenas de entradas con más de tres caracteres. Para propósitos de la presente se ejemplifican cadenas de tres elementos para facilitar la explicación.

En el modelo Post puede haber una palabra en el sistema de reescritura que no garantice que se aplique la producción de parada. Es decir, que nunca se llegue a esa producción, lo cual hace que no pueda decirse nada al respecto, pues sería

riesgoso vaticinar algo y luego llegar a la producción de parada. La experiencia y la práctica hacen posible, aunque de manera empírica, predecir algunas de éstas cadenas de entrada que a la postre provoquen este desenlace.

En la máquina de Turing se presentan situaciones similares, donde el cómputo no se detiene, lo cual ya se conoce, cayendo en la contradicción de: si para entonces no para y viceversa.

Cuando la máquina de Turing para completamente puede entonces verificarse el contenido de la cinta, para analizar los valores y ver si el cómputo es válido.

Como resultado de las reglas obtenidas antes, efecto del análisis de las diferentes cadenas de entrada, se muestran las siguientes transformaciones:

$bbb \rightarrow bbc \rightarrow cbc \rightarrow c$ (ejemplo ya discutido)

$bbbb \rightarrow bbbc \rightarrow bcbc \rightarrow bcbc \rightarrow \dots$ (se cae en un ciclo sin fin)

$bbbbb \rightarrow bbbbc \rightarrow bcbcb \rightarrow cbcbc \rightarrow cbc!$ (se topa con c como primera letra en la última cadena obtenida; como se sabe, c es la letra de parada en el sistema de dos etiquetas ejemplo)

$bbbbbb \rightarrow bbbbbc \rightarrow bbbcbcb \rightarrow bcbcbcb \rightarrow bcbcbcb \rightarrow \dots$ (se cae en ciclo sin fin, no se detiene el proceso de reescritura de cadenas)

$bbbbbbb \rightarrow bbbbbbc \rightarrow bbbcbcbcb \rightarrow bcbcbcbcb \rightarrow cbcbcbc \rightarrow cbc!$

$bbbbbbbb \rightarrow bbbbbbcb \rightarrow bbbcbcbcbcb \rightarrow bcbcbcbcbcb \rightarrow bcbcbcbcbcb \rightarrow \dots$ (se cae en un ciclo sin fin, no se detiene el proceso de reescritura de cadenas)

Cada vez que la cadena de entrada se componga por un número par de bes entonces se estará en un ciclo sin fin de reescritura de cadenas, por las características propias del sistema de producción ofrecido como ejemplo o guía; es decir, esta situación sucederá cuando: $(b)^n$ donde $n \geq 4$, siendo n par.

En la última cadena de bes ofrecida como ejemplo se cae en un ciclo infinito, según las producciones del sistema de dos etiquetas proporcionado como ejemplo; de hecho salta a la vista que la cadena inicial compuesta por ocho bes caerá directamente en un ciclo sin fin de reescritura de cadenas, pues al borrarse las primeras dos bes que componen la cadena se obtiene la cadena compuesta por seis bes, la cual ya se conoce.

Partiendo del caso más sencillo del ejemplo, se sabe que la cadena formada por un número par de bes como lo es bbbb, llevará a un resultado de ciclo sin fin de reescritura y por añadidura todas las cadenas pares que se formen a partir de la raíz bbbb llevarán a un ciclo infinito de reescritura y por tanto a un cómputo infinito.

A continuación se dan ejemplos de transformación de cadenas, éstas se encuentran formadas sólo por la letra "a" (al ser presentada la cadena por primera vez), tomando como referencia el sistema de dos etiquetas del inicio de esta sección.

aaa → ab

aaaa → aab → bb

aaaaa → aaab → abb → bb

aaaaaa → aaaab → aabb → bbb → bbc → cbc → c!

aaaaaaa → aaaaab → aaabb → abbb → bbb → bbc → c!

aaaaaaaa → aaaaaab → aaaabb → aabbb → bbbb → bbbc → bcbc → bcbc

La cadena formada solamente por letra a puede llevar a un ciclo infinito, si por ejemplo se da el caso de obtener alguna cadena intermedia en el proceso de conversión formada por las letras bbbb, o cualquier combinación de cadenas formadas solamente por un número par de bes mayor o igual que cuatro. A continuación se ejemplifica la transformación de cadenas, éstas se forman por combinación tanto de letras aes como de bes, según el sistema de dos etiquetas tomado como referencia.

abab → abb → bb

ababa → abab → abb → bb

ababab → ababb → abbb → bbb → bbc → c!

abababa → ababa → abab → abb → bb

abababab → abababb → ababbb → abbbb → bbbb → bbbc → bcbc (No se detiene el procedimiento de reescritura de cadenas)

ababababa → abababa → ababa → abab → abb → bb

En el caso de cadenas iniciales formadas por pares de combinaciones ab, conducen a un proceso de reescritura de cadenas que no se detiene cuando esta combinación es mayor a cuatro; es decir a partir de seis (ababab). Las cadenas impares de combinaciones ab, como por ejemplo: ababa, abababa, ..., y que empiezan siempre por a, conducen a cadenas finales bb. Es decir, todos los palíndromos formados en su inicio por la letra a y seguidos por "bes" producen siempre bb.

$bab \rightarrow bbc \rightarrow c!$

$baba \rightarrow babc \rightarrow bcbc \rightarrow bcbc$ (No se detiene el procedimiento de reescritura de cadenas)

$babab \rightarrow babbc \rightarrow bbcbc \rightarrow cbcbc \rightarrow cbc!$

$bababa \rightarrow bababc \rightarrow babcbc \rightarrow bcbbcb \rightarrow$ (No se detiene el procedimiento de reescritura de cadenas)

$bababab \rightarrow bababbc \rightarrow babbcbc \rightarrow bbcbbcb \rightarrow cbcbbcb \rightarrow cbcbc!$

Las cadenas iniciales formadas por pares de combinaciones ba conducen a un proceso de reescritura de cadenas que no se detiene. Los palíndromos formados por cadenas que empiezan por ba , compuestos solamente por estas dos letras y que terminen en la letra b , siempre logran detener el proceso.

Es decir, los palíndromos generados a partir de $b(ab)^n$ siendo n impar, garantizan la detención del proceso. En cambio las cadenas generadas a partir de $(ba)^n$ con $n > 2$ y con repeticiones de la secuencia $babababa\dots$, siendo n par, llevarán a un proceso infinito de reescritura de cadenas.

1.5 Equivalencia Turing-Post

Todo intento por proporcionar un análisis exacto de la noción intuitiva de una función efectivamente calculable resulta ser equivalente, en el sentido estricto de que cada análisis selecciona la misma clase de funciones. Como ya se mencionó, ha habido otros intentos esporádicos para definir los procedimientos o método efectivo, pero todos se dirigen a:

- La definición formal del tema efectivo u otras nociones formales de la idea de un método efectivo.
- Las clases de funciones a evaluar con los métodos efectivos.

Un sistema formal con axiomas y producciones puede modelar un proceso. A simple vista, un sistema formal parecería permisivo en lugar de imperativo. Igualmente pareciera no existir nada que correspondiera al control del proceso en una máquina, nada como algún mecanismo obvio que dicte 'qué es lo que hay que hacer a continuación'. De hecho, no hay noción de tiempo o secuencia excepto, quizás, la secuencia de pasos de prueba de un teorema[2].

En términos generales, hay muchas pruebas diferentes de un teorema y no se espera poder utilizar la secuencia de pasos en la prueba como mecanismo de control de proceso. Puede hacerse, sin embargo, usando algunas estrategias como: el uso de símbolos de puntuación especial de varias formas. Puede, de

hecho; plasmarse la noción de máquina dentro del concepto de un sistema formal con sólo axiomas y producciones.

Es posible construir, como lo señala Minsky en su libro *Computación finita e infinita*, un sistema formal de dos etiquetas, que de manera muy clara simule la actividad de una máquina de Turing. Los sistemas de dos etiquetas (sistemas de reescrituras) son equivalentes a un tipo de máquinas de Turing con la cabeza de lectura y escritura separadas, moviéndose en una sola dirección con la cinta entre ellas[2].

La figura 1.2 muestra cómo la máquina de estructura definida (máquina de Turing), el sistema de dos etiquetas y la red de neuronas, obtienen la misma familia de funciones como resultado del proceso computacional).

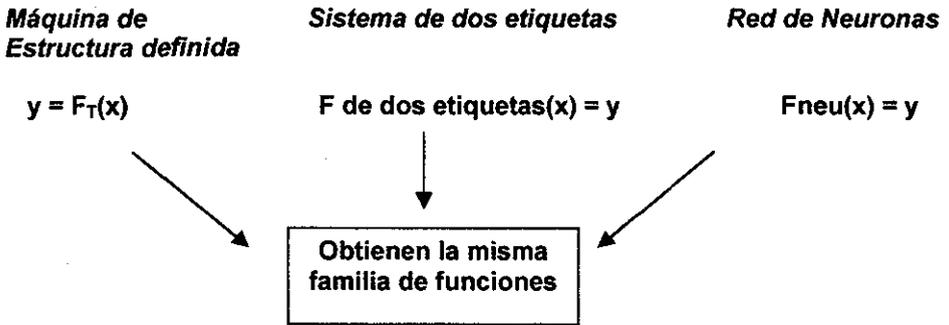


Figura 1.2 Familia de funciones obtenida

En cualquier sistema logístico siempre es posible crear un procedimiento que busque todas las posibles pruebas.

Al final se obtiene un proceso efectivo que genere todas las secuencias finitas en el alfabeto dado. El orden de generación de estas secuencias puede servir como numeración para las pruebas.

A continuación se muestra un ejemplo de cómo pasar de una representación de la máquina de Turing a un sistema de dos etiquetas como prueba de equivalencia: ahora se usan quintuplas porque se ha agregado la dirección.

Representación de Quintuplas

q_i	s_j	q_{ij}	s_{ij}	d_{ij}
0	0	0	0	1
0	1	1	0	1
0	B	H	0	-

q_i	s_j	q_{ij}	s_{ij}	d_{ij}
1	0	1	0	1
1	1	0	0	1
1	B	H	1	-

La máquina requiere de dos estados, uno para los números pares y otro para los impares, puesto que las quintuplas se representarán para un contador de paridad. La Tabla de la izquierda representa la quintupla q_0 (estado asociado a números pares) y la tabla de la derecha la quintupla q_1 (estado asociado a números impares). El símbolo \$ es usado para representar cadenas, el subíndice 1 se emplea para indicar una subcadena anterior a la que se está analizando y se usa el subíndice 2 cuando se usa una subcadena posterior a la cadena objeto de análisis.

Como se conoce, las quintuplas especifican las acciones que la máquina realiza.

Las producciones son:

$${}_1q_00\$_2 \rightarrow {}_1q_0\$_2$$

$${}_1q_01\$_2 \rightarrow {}_1q_1\$_2$$

$${}_1q_0B\$_2 \rightarrow q_0H$$

$${}_1q_10\$_2 \rightarrow {}_1q_1\$_2$$

$${}_1q_11\$_2 \rightarrow {}_1q_0\$_2$$

$${}_1q_1B \rightarrow q_1H$$

Axiomas: $q_0: 0, 1$ (con el formato): q_00, q_01

La máquina opera entre la configuración inicial q_0 y B , q_0 puede ser 1 ó 0.

B le indica a la máquina dónde finaliza la secuencia. H es el símbolo de parada.

Como conclusión de la sección 1.5, puede decirse que si se encuentra un modelo de cómputo que sea equivalente a los sistemas de dos etiquetas de Post, entonces puede también decirse que será equivalente a la máquina de Turing, pues como se comentó en las secciones 1.2 y 1.3, existe una equivalencia entre la máquina de Turing y los sistemas de Post. El propio Minsky en su corolario sobre el teorema conocido con el nombre Cocke-Minsky, lo plantea [2].

Es muy importante recordar el Teorema de Cocke y Minsky, a saber dice así:

“Para cualquier máquina de Turing T se tendrá un sistema de dos etiquetas (conocido en inglés como sistema tag), que se comporte de manera similar a T , en el sentido de (m,n) , cuando un axioma que codifica las cintas como: $Aa \quad aa \quad aa \dots aa \quad Bb \quad bb \quad bb \quad bb$ con m' aes y n bes. El sistema de dos etiquetas tiene un número de borrado $P=2$. [2]

El teorema de Cook y Minsky indica que cualquier máquina de Turing determinante puede simularse por medio de un sistema de dos etiquetas.

Minsky plantea el siguiente corolario:

“La computabilidad con sistemas normales monogénicos equivale a la computabilidad con funciones recursivo-generales, máquinas de Turing, sistemas canónicos generales, etc.”

Resulta posible obtener una máquina de Turing a partir de un sistema de dos etiquetas, si se toma en cuenta la equivalencia que existe entre ambos. Pueden llegar a construirse sistemas de dos etiquetas por separado para cada uno de los estados de la máquina de Turing, es decir: representar cada estado de la máquina de Turing, por medio de producciones; así se prueba el corolario recién presentado. El próximo paso es unir e identificar ciertas letras o símbolos asociados a cada estado de la máquina de Turing, hasta formar un sistema de dos etiquetas simple que se comporte como la máquina completa de Turing.

Se conoce que la máquina de Turing opera según el esquema de la figura 1.3. Es conveniente ahora pensar que dicha máquina operará según el esquema de la figura 1.4. Visto de esta manera, la máquina de Turing tendrá muchos estados con estas características: q_{i0} y q_{i1} . Siempre existirá un par de estados secretos que recordar, mismos que tuvieron que ser leídos mientras estaba escribiendo.

Si se continua esta figura se generará un árbol, ya que si se sigue el proceso de descomposición funcional se obtendrán máquinas de Turing para cada estado en cuestión. Esta figura es de gran ayuda para entender que es posible no sólo descomponer la máquina de Turing, sino también representar un sistema de dos etiquetas para cada uno de estos estados.

Los estados en el nuevo sistema no corresponden exactamente a los estados del sistema viejo. Los nuevos estados son también llamados quintuplas como los que a continuación se ofrecerán de ejemplo:

$(q_i, s_j, d_i, q_b, q_h)$: las que significan en el mismo orden: estado, símbolo, movimiento, si se lee un cero se va a q_b y si se lee un 1 se va a q_h .

Sólo existe una quintupla para cada estado, en vez de dos. Para llevar esto a cabo, gracias a un sistema de dos etiquetas, se exhibirá un conjunto de producciones que tendrá el efecto de un estado que se mueve hacia la derecha. Las siguientes representaciones de m y n ya fueron explicadas en el libro de Minsky: Computación Finita e Infinita.

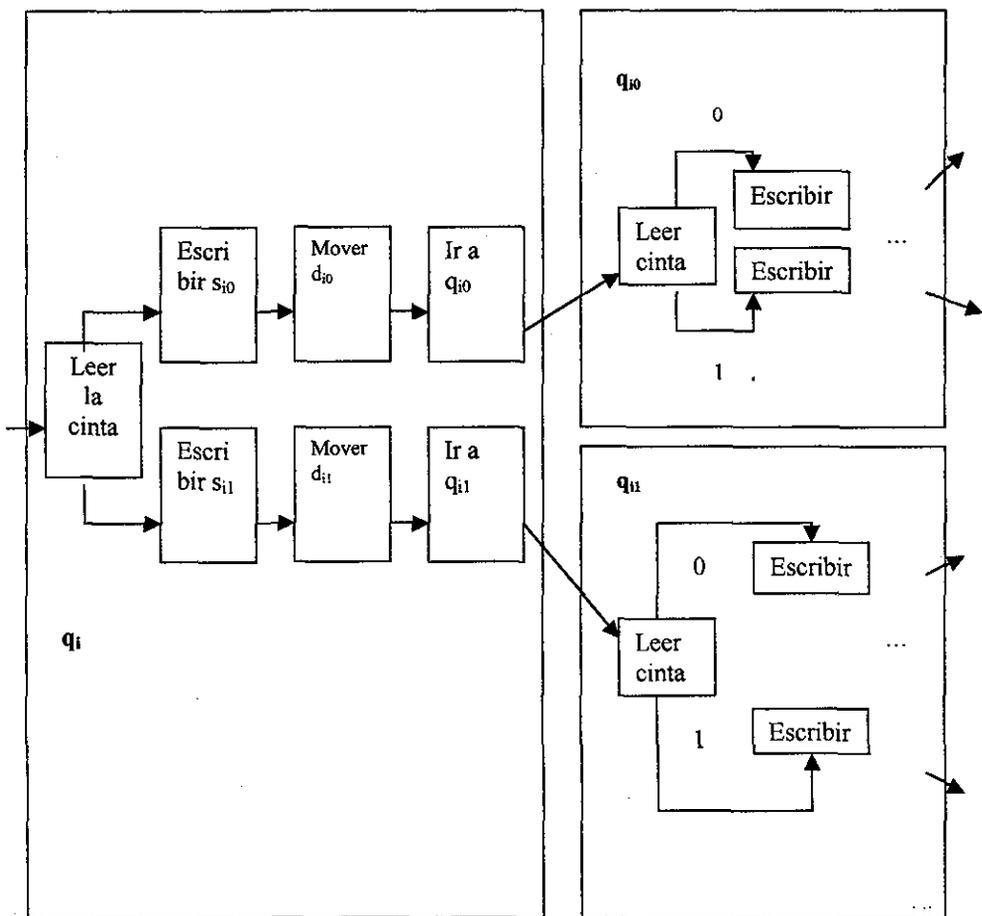


Figura 1.3 Descomposición Funcional de la Máquina de Turing en dos Máquinas (Tomada de la referencia [2])

Conjunto de producciones:

$$m \rightarrow 2m + S_i$$

$$n \rightarrow H(n) = \begin{cases} n/2 & \text{si } n \text{ es par} \\ (n-1)/2 & \text{si } n \text{ es impar} \end{cases}$$

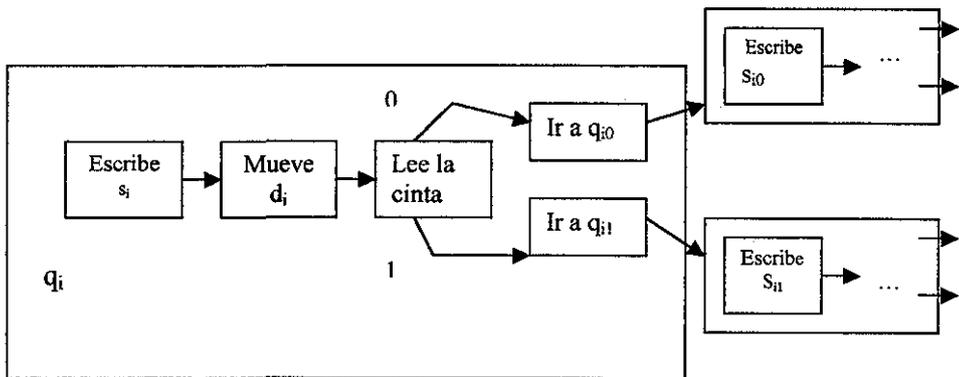


Figura 1.4 Descomposición de un estado q_i

Será necesario cambiar m y n , por lo que tendrá que prepararse el sistema de tal forma que seguirá al estado q_{i0} , si n es par, y hacia q_{i1} , si n es impar.

Se comienza con la cadena:

$$Aa(aa)^m Bb(bb)^n$$

Las primeras producciones serán:

Si s_i es 0:

$$A \rightarrow Cc$$

$$a \rightarrow cccc$$

Si s_i es 1:

$$A \rightarrow Cccc$$

$$a \rightarrow cccc$$

Para abreviar las producciones del sistema de dos etiquetas se mostrará sólo la primera letra del antecedente y la cadena consecuente, como: $ax \$ \rightarrow \$ h_i$. En cada caso z , el número de letras borradas es dos, aplicando las producciones de arriba se llega a $Bb(bb)^n Cc(cc)^m$, donde m es $2m$ o $2m+1$, dependiendo de si el estado requerido por la máquina sobre el cuadro que se aleja es uno o cero. Es importante conocer si n es par o impar.

El procedimiento para hacer esto se inicia por las producciones:

$B \rightarrow S$ por ejemplo sería $Bx\$ \rightarrow \S
 $b \rightarrow s$ $bx\$ \rightarrow \s

Al aplicar las producciones anteriores a la cadena: $Bb(bb)^n Cc(cc)^m$, se obtiene la cadena resultante: $Cc(cc)^m Ss^n$.

Ahora se tienen las siguientes producciones:

$C \rightarrow D_1 D_0$
 $c \rightarrow d_1 d_0$

Obteniéndose la cadena:

$Ss^n D_1 D_0 (d_1 d_0)^m$

Lo importante de los resultados finales obtenidos es que éstos se encuentran en alfabetos enteramente separados, lo cual significa que ahora es posible escribir producciones diferentes para determinar lo que sucederá a continuación con la cadena en el caso que n fuese par y en el caso que n fuese impar. Como conclusión deberá escribirse un conjunto de producciones como las de arriba para cada estado q_i de la máquina de Turing, usando alfabetos enteramente diferentes para cada uno. Después se unen, siempre que un estado de salida q_{i1} venga seguido de un estado q_j (q_j será definido como el estado siguiente), posteriormente se hace la homologación entre las letras del estado de salida y las letras asociadas en el estado de entrada.

Esta explicación más detallada puede ser vista en el libro de Minsky titulado: "Computación Finita e Infinita", en el cual se muestra una versión pormenorizada de la explicación con un ejemplo claro del proceso de homologación. Resulta muy importante entender esto pues una vez que se concluya exitosamente la homologación de las letras del estado de salida y las letras del estado de entrada se podrá dar la continuidad adecuada al proceso.

La conclusión importante de esta sección es que pueden simularse, por interconexión de estados de una máquina de Turing, combinadas en un gran sistema de dos etiquetas, todas las producciones del sistemas de dos etiquetas arriba descritas. Esto prueba el teorema de Cocke y Minsky.

CAPÍTULO 2: REDES NEURONALES

2.1 Introducción

Las redes neuronales constituyen una nueva y prometedora generación de sistemas de procesamientos de información que demuestran la habilidad de aprender y reconocer patrones o datos de entrenamiento.

Para describir de manera formal los procesos de cómputo que ocurren en las neuronas, ya sea biológicas, o artificiales, se han propuesto dos enfoques diferentes, aunque complementarios. En el primero de ellos se intenta describir y comprender los procesos biofísicos que ocurren en las neuronas biológicas e interpretarlos como si fuesen procesos de cómputo para poder diseñar y construir posteriormente los elementos procesadores fundamentales que vayan a integrarse en los sistemas neuronales artificiales. En el enfoque alternativo, los sistemas neuronales naturales se describen e interpretan asumiendo *a priori* que éstos son una clase especial, todavía no bien caracterizada por completo, de sistema de cómputo.

De esta manera, en el primer enfoque se espera que todo sistema neuronal artificial, en un mayor o menor grado, debe ser plausible desde un punto de vista biofísico. Por el contrario, en el segundo enfoque se está considerando que todo sistema neuronal ya es un sistema de cómputo, por lo que sólo es necesario descubrir cómo funciona. Los modelos artificiales que derivan del primer enfoque normalmente tienden a ser muy complejos para poder garantizar una descripción biofísica adecuada, aunque tal vez simplificada. A diferencia de esto, en el segundo enfoque se permite el uso de modelos simples de neurona artificial, siempre y cuando éstos permitan proporcionar descripciones de los comportamientos neuronales como si fuesen sistemas de cómputo.

En este último contexto, resulta atractivo considerar modelos de neurona artificial más complejos pues prometen presentar comportamientos más ricos, los cuales pueden ser interpretados como procesos de cómputo también más complejos. Esta perspectiva, sin duda atractiva para los dos enfoques de estudio, tiene, sin embargo, que enfrentar serias dificultades. La principal de ellas es que deben analizarse modelos mucho más complejos de neurona artificial. Éstos pueden resultar tan complejos que una descripción matemática precisa de ellos puede resultar extremadamente difícil o, de hecho, imposible de realizar.

2.2 Construcción de una Neurona Artificial

Una neurona artificial, puede ser entendida como un dispositivo que realiza una transformación no lineal de un conjunto de señales externas a una señal de respuesta. Así, una red neuronal realiza, en paralelo, una transformación no lineal de un conjunto de señales externas a otro conjunto de señales de respuesta.

La clase particular de transformación no lineal realizada por una red neuronal depende, tanto de las interconexiones que se establecen en la red; como de las características propias de las neuronas artificiales que la forman. Dichas características, a su vez, dependen de las propiedades biológicas en las que se inspiró el modelo de neurona artificial.

A continuación se propone un modelo de neurona artificial susceptible de ser realizado en una computadora analógica. Para esto, primero presentamos una descripción breve de las propiedades de las neuronas biológicas en las que se basará el modelo de neurona artificial.

2.2.1 Morfología de la Neurona

El bloque básico de construcción del sistema nervioso central de los animales es la célula nerviosa o neurona. Si bien los neurobiólogos han descrito diferentes clases de neuronas, tanto por su forma como por su funcionamiento, también han observado que, en general, estas células se caracterizan por presentar cuatro regiones morfológicamente diferentes [1]:

- a) el cuerpo celular,
- b) la arborización dendrítica,
- c) el axón o cilindroeje y
- d) las terminales sinápticas.

Adicionalmente, la neurona posee sistemas bioquímicos especializados en la generación de señales electroquímicas en cada una de las regiones anteriores.

El cuerpo celular o soma constituye el centro metabólico de la neurona, pues es donde se localizan los principales sistemas bioquímicos de síntesis, secreción y transporte de sustancias.

Del cuerpo celular se originan diversas extensiones o protuberancias que se bifurcan repetidas veces hasta formar una arborización fina, la cual se denomina arborización dendrítica, y una protuberancia especial de forma tubular que sólo se divide en su parte terminal, la cual se denomina el axón o cilindroeje. Las dendritas constituyen el principal centro receptor de señales de la neurona, mientras que el axón constituye la unidad conductora de las señales procesadas por la neurona. El axón se caracteriza por carecer de sistemas complejos de síntesis de sustancias, aunque contiene diversos sistemas especializados en el transporte de sustancias.

La parte terminal del axón se divide repetidas veces hasta formar una gran cantidad de ramas finas que poseen propiedades funcionales especializadas. Dichas ramas se denominan terminales o botones sinápticos y son el medio por el cual una neurona establece puntos de contacto con otras neuronas, inclusive consigo misma.

Los puntos de contacto entre neuronas se denominan sinapsis, las cuales están formadas por la terminal de la célula presináptica, la superficie receptora de la célula postsináptica y un espacio que las separa.

Las sinapsis se establecen principalmente en las dendritas de la neurona postsináptica, aunque también se encuentran en el cuerpo celular y, ocasionalmente, en las partes iniciales y terminales del axón de la neurona postsináptica. En la figura 1 se muestra una representación esquemática de una neurona típica.

Ahora bien, una neurona puede ser descrita como un dispositivo procesador de señales caracterizado por las cuatro partes fundamentales siguientes:

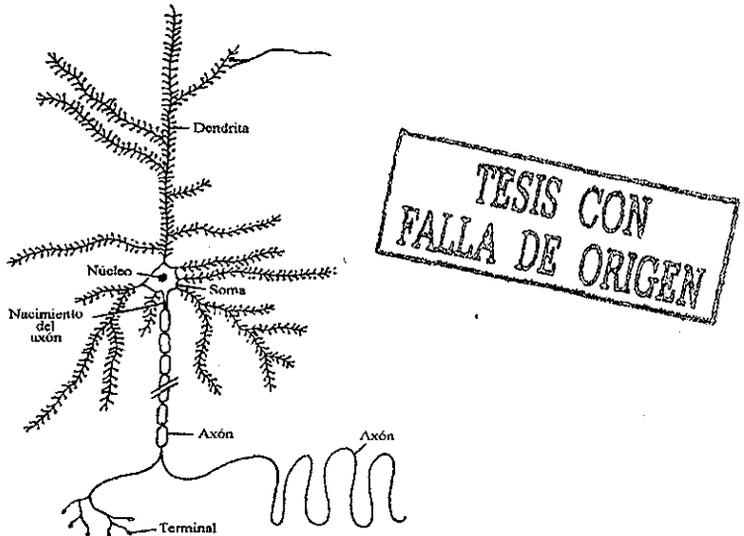


Figura 2.1 Diagrama esquemático para una neurona típica.

- a) Las dendritas, que constituyen el área donde se reciben las señales provenientes de otras neuronas.
- b) El soma, que constituye la región celular donde se congregan y combinan las señales recibidas de otras neuronas.
- c) El axón, que es la región especializada en transmitir el resultado del procesamiento de las señales externas.
- d) La sinapsis, que es la región donde dos neuronas establecen contacto.

Adicionalmente, es importante mencionar que la respuesta neuronal puede multiplicarse tantas veces como se ramifica el axón sin que sus características se alteren significativamente. De esta forma, una neurona puede influir sobre muchas otras neuronas, inclusive sobre ella misma.

2.2.2 Modelo de Neurona Artificial.

Con base a lo presentado en la sección anterior, una neurona artificial puede entenderse como un elemento procesador de señales que es estimulado con n señales aferentes diferentes, x_1, \dots, x_n , agrupadas en un vector $x = [x_1, \dots, x_n]^T \in R^n$ de dimensión n , y genera una única señal eferente o de respuesta $y \in R$. Las señales aferentes agrupadas en el vector de entrada x se considera que provienen de n neuronas vecinas, incluyendo posiblemente a la propia neurona, o que provienen de su medio externo.

La capacidad de procesamiento de señales de la neurona artificial puede expresarse en términos de su transferencia característica, F_{NEU} , la cual es una transformación no lineal del vector de entrada x a la respuesta escalar y ; esto es, $F_{NEU}: R^n \rightarrow R$. De esta manera, una neurona artificial puede ser entendida como un dispositivo procesador de señales elemental del tipo de entradas múltiples y una salida única (MISO por sus siglas en inglés). En particular, la neurona artificial es un procesador cuya operación puede dividirse en dos partes [2]: a) la *operación de confluencia* y b) la *operación de activación*, como se ilustra en la figura 2.

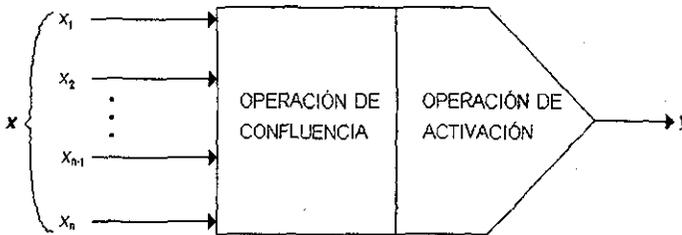


Figura 2.2 Una neurona artificial como un dispositivo tipo MISO.

La operación de confluencia se usa para modelar las llamadas operaciones somáticas observadas en la neurona biológica. Estas operaciones se construyen a partir de los fenómenos sinápticos y de los de agregación. Por otro lado, la operación de activación puede construirse a partir de los fenómenos biológicos que ocurren en el axón (i. e., generación y transporte de potenciales de acción) y en las terminales sinápticas de una neurona biológica. Adicionalmente, en este modelo debe considerarse que la respuesta de una neurona artificial puede multiplicarse tantas veces como sea necesario sin que se altere.

Es importante señalar que el modelo anterior no tiene como objetivo inmediato la descripción de alguna clase de fenómenos biológicos o el emular de manera precisa el comportamiento de un sistema biológico en particular. Dicho modelo tiene como propósito principal el servir como base para el diseño de sistemas neuronales artificiales procesadores de señales. De esta manera, la descripción matemática de las dos operaciones básicas de la neurona artificial no tienen por qué reflejar de manera fiel los procesos fisiológicos observados en las neuronas biológicas.

2.2.3 Modelo matemático para la neurona artificial

A partir del modelo de neurona artificial recién descrito se establece, de inmediato, que las propiedades de una clase particular de neurona artificial están dadas por las características de las operaciones neuronales que la forman. A su vez, las características de estas operaciones dependen de la realización particular que se haya establecido para ellas. Aquí es conveniente recordar que las realizaciones físicas y matemáticas de una misma operación no coinciden en lo general, ya que las últimas son exactas, mientras que las primeras necesariamente son limitadas.

En general, la mayoría de los modelos de neurona artificial que se encuentran en la literatura del campo son matemáticos. De estos, una gran mayoría corresponde a los que podemos llamar modelos invariables en el tiempo. En cambio, si se considera que las señales aferentes a la neurona son dependientes del tiempo, entonces el comportamiento de la neurona artificial debe establecerse en términos de un sistema dinámico $\Sigma = (T, S, X, \phi, Y, h)$, donde T es el dominio temporal, X y Y son los espacios de valores de entrada y de salida, respectivamente, S es el espacio de estados, $\phi: T \times S \times X \rightarrow S$ es la función de transición de estados y $h: T \times S \rightarrow Y$ es la función de salida. El comportamiento de la neurona artificial, es decir, su salida o respuesta $y \in Y$ a un cierto valor de entrada $x \in X$, está gobernado por las funciones ϕ y h , las cuales, a su vez, dependen del estado $s \in S$ en particular en el que se encuentra la neurona. En este caso, el estado s de la neurona resume toda la información necesaria para determinar su evolución futura.

Sin embargo, el desarrollo de modelos variables en el tiempo, discretos o continuos, no es tan amplio como podría esperarse, a pesar de que estos modelos son los que permiten, de manera natural, procesar señales dependientes del tiempo. Una explicación para esto es que un modelo de neurona artificial variable en el tiempo es, desde un punto de vista matemático, un sistema dinámico no lineal, cuyas propiedades no son fáciles de establecer. A pesar de esta dificultad intrínseca, actualmente se reconoce que el estudio de los modelos de redes neuronales variables en el tiempo será de gran importancia para el desarrollo futuro de los sistemas neuronales de procesamiento de señales.

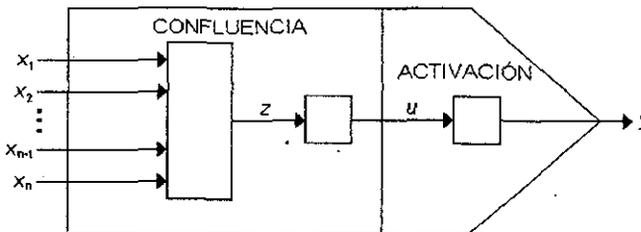


Figura 2.3 Neurona artificial propuesta como un dispositivo tipo MISO.

En la figura 2.2 se produce primero una operación dendrítica, luego otra operación somática y al final una operación axónica o de respuesta. La operación dendrítica está relacionada con las entradas x_1, x_2, \dots, x_n ; la operación somática está ligada a la función identidad y , por último, la operación axónica está relacionada con la comparación contra un umbral.

La primera operación dendrítica está asociada a la función $Z = F(I)$, la segunda operación somática está relacionada con la función identidad: $U = I(z)$, y seguidamente la operación axónica se representa por medio de $y = \sigma(u) = \sigma(z)$, la cual se relaciona con la función identidad.

Las sinapsis son excitatorias si permiten que los impulsos que pasan produzcan el disparo de la neurona receptora, o inhibitoria si dejan que los impulsos que pasan obstaculicen el disparo de la neurona.

Existe un modelo matemático simple de las neuronas biológicas arriba descritas y propuestas por McCulloch y Pitts [1943], generalmente conocidas como *neuronas M-P*. En este modelo, el elemento de procesamiento i -ésimo computa una suma de los pesos de sus entradas y salidas $y_i = 1$ (disparo) ó 0 (sin disparo), según si la suma de los pesos de las entradas está por arriba o por debajo de un cierto umbral ϕ_i .

A continuación se muestra la ecuación:

$$Y_i(t+1) = a\left(\sum_{j=1}^n w_{ij} x_j(t) - \phi_i\right)$$

donde la función de activación $a(f)$ es una función escalón:

$$a(f) = \begin{cases} 1 & \text{si } f \geq 0 \\ 0 & \text{si } \text{otra} \end{cases}$$

El peso w_{ij} representa la fuerza de la sinapsis (llamada la conexión o eslabón) que une la neurona j (fuente) a la neurona i (destino). Un peso positivo corresponde a una sinapsis excitatoria y un peso negativo a una sinapsis inhibitoria. Si w_{ij} es igual a cero, entonces no hay conexión entre las dos neuronas. w_{ij} es un producto interno, Z (de la figura 2.2) representa la función identidad, u (en la misma figura) representa a la función de comparación con el umbral. McCulloch-Pitts es un caso particular de neurona artificial, pues ha muchos casos de neuronas artificiales.

Aun cuando los modelos simples presentan la neurona biológica como una unidad de umbral binaria, una neurona McCulloch-Pitts posee un potencial computacional importante. Realiza las operaciones lógicas básicas NOT, OR y AND cuando los pesos y umbrales se seleccionan en consecuencia. Ya que cualquier función de combinaciones multivariantes puede implementarse a partir de estas operaciones lógicas básicas, una unión sincronizada de dichas neuronas puede realizar computaciones universales, como las realizadas por una computadora digital ordinaria. Varias generalizaciones o variaciones de la neurona M-P, que es el componente básico de las redes neuronales, se presentará posteriormente

CAPITULO 3: PROPUESTA DE LA RED DE NEURONAS

3.1 Introducción

El objetivo del presente capítulo es mostrar y explicar la clase particular de red neuronal que se va a usar para establecer la equivalencia entre los ya mencionados modelos clásicos de computación y la red de neuronas. Primero se explicará en modelo de neurona empleado y seguidamente el modelo de la red.

3.2 Modelo de Neurona

El modelo de neurona propuesta para la red neuronal, con la cual se trabajará en el presente trabajo de Tesis, consta de dos clases diferentes de neuronas:

- Neurona Analógica: Tiene respuesta en el intervalo $y \in [0,1]$
- Neurona Binaria: Tiene dos valores posibles de respuesta: 0 o 1, es decir $y \in \{0,1\}$

En este trabajo el modelo es una red análoga recurrente, donde cada neurona x_i computa una P_i polinomial de sus entradas con coeficientes racionales y donde la escalar resultante pasa por una simple no linealidad de piezas. La actualización de una neurona analógica puede escribirse así:

$Y_a = \sigma (P_i(x))$ Función de Respuesta de Neurona Analógica.

Donde P_i es una polinomial y σ es la función de saturación lineal descrita por:

$$\sigma(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 \leq x \leq 1 \\ 1 & \text{si } x > 1 \end{cases}$$

Las neuronas binarias serán representadas por la siguiente ecuación Y_b , la cual tendrá también asociada la función σ de saturación, siendo:

$Y_b = \sigma_i(P_i(x))$ la función de confluencia de Neurona Booleana,

$$\sigma_i(x) = \begin{cases} 0 & \text{si } x < 1 \\ 1 & \text{si } x \geq 0 \end{cases}$$

La función P (polinomial), descrita al inicio, es una suma de términos reales, como $P(x) = P(x_1) + P(x_2) + \dots + P(x_n)$. Es decir cada uno es una suma de monomios de neurona. Las neuronas pueden considerarse *booleanas* o variables de *interrupción* y de ahí el nombre de *interrupciones* (valor 0 ó 1).

3.3 Modelo de Red

Para establecer el modelo de red se tomó en cuenta la caracterización funcional de cada neurona, en base a su ecuación, así como las señales de entradas. En el capítulo cuatro, se cuenta con mayor detalle el funcionamiento de esta red de neuronas.

La propuesta se basa en tres neuronas continuas o analógicas y seis neuronas booleanas o binarias. La motivación vino inducida por las características propias del problema planteado y del alcance del trabajo, necesitándose contar con neuronas que pudieran llevar a cabo (en conjunto, es decir como red), la simulación del sistema de dos etiquetas propuesto y cualquier sistema de dos etiquetas. Para este fin fue útil nombrar a una neurona que almacenara el valor de la palabra de entrada, también fue útil nombrar una neurona que acumulara la longitud de la palabra de entrada. Al mismo tiempo se requirió de una neurona que llevara las reglas de producción.

Por las características de estas tres neuronas que se requerían, se decidió que su funcionamiento se hiciera en base a una función analógica; para lo cual fue necesario, dada la importancia de la exactitud en los cálculos, realizar una codificación que permitiera un fácil entendimiento.

Se decidió codificar la neurona W en base-4, lo cual permite y favorece representar los tres pasos cíclicos de trabajo del sistema de dos etiquetas, a saber: leer, borrar y agregar, los cuales se explican y detallan en el capítulo cuatro; las propias gráficas de las neuronas C_1 y C_2 muestran estas etapas que se dan por ciclos, una vez que se produce la etapa inicial de lectura de datos. Era necesario representar estos tres pasos del sistema de dos etiquetas en la red de neuronas propuestas por lo que fue necesario el poder observar en W todos y cada unas de sus variaciones, las cuales en notación decimal sólo representarían números fríamente.

La neurona K fue codificada en base seis, pues como representa las producciones, contendrá representaciones de palabras provenientes de W , y no pueden estar en la misma base, pues de lo contrario se presta a confusión, ya que debe haber un separador o delimitador a la hora de escribir una producción en donde sus símbolos o letras ya tienen de hecho una codificación asociada, el usar la misma codificación conduce a no distinguir cuando termina una letra y cuando se termina una producción completa.

Se pudo haber usado otras bases para representar las neuronas W y K , sólo es importante señalar que no deben de ser iguales dichas bases, pues la neurona K engloba a la neurona W , pues K contendrá las producciones, mismas que se hacen a partir de W .

La neurona C_1 y C_2 son dos neuronas que se definieron con el objetivo de marcar etapas dentro de un ciclo de computación. Estas neuronas en sus propias ecuaciones y por ende en sus gráficas muestran y revelan el nexo entre el sistema de dos etiquetas y la red de neuronas propuestas, ello permite observar que si bien es cierto que se está en presencia de un proceso de simulación, en donde el sistema de dos etiquetas está siendo simulado por la mencionada red de nueve neuronas, también es cierto que siempre se dan tres pasos, es decir tres etapas en un ciclo, estas tres etapas son los tres pasos que definieron el funcionamiento del sistema de dos etiquetas.

El propósito con éstas neuronas (C_1 y C_2), es también mostrar en qué parte de la simulación se encuentra el proceso, el mismo análisis de estas neuronas marcó la necesidad de controlar cada ciclo computacional, pues cada uno de éstos, se divide en etapas que se repiten de manera cíclica al comenzar un ciclo; pero para esto fue necesario indicar mediante una neurona, la detención del proceso, para este fin se define la neurona de parada H .

E_w se diseñó con el objetivo de avisar, que la última producción de dígitos consecutivos a partir de W , pronto formará un código de una letra; es la neurona auxiliar de la neurona W . La neurona E_k se propuso con el fin de indicar que la red terminará de leer una letra codificada a partir de la regla de producción, en cambio E_r se incluyó con el objetivo de informar cuando la producción forme una regla de producción entera.

Tres neuronas son analógicas, a saber: W , L y K .

Seis neuronas son binarias, de ellas: dos de control, tres auxiliares y una es usada como neurona de parada, para indicar que el proceso de cómputo se detuvo.

Las neuronas binarias de control son: C_1 y C_2 .

Las neuronas binarias auxiliares son: E_w , E_k y E_r .

La neurona binaria de parada es: H .

3.3.1 Estímulos

La red usa cuatro canales de entrada binaria: I_w, I_v, I_b, I_o . Estos canales también se conocen como estímulos.

El canal I_w transfiere la palabra antecedente binaria marcada por W como una secuencia de *bits* y se halla en 0 cuando no se encuentra transmitiendo. El canal I_v ('v' es de validación), se ubica en 1 cuando I_w empieza a enviar información y permanece en 1 hasta que el último dígito de entrada de la palabra antecedente haya alcanzado la red, luego I_v se convierte en 0 y permanece para siempre con dicho valor.

Los canales I_b e I_e están en la forma de respuesta de impulso: I_b (inicio de la entrada) es 1 sólo cuando I_v va de 0 a 1. Ambas, I_b e I_e son 0 en todas las demás ocasiones.

3.3.2 Operación

Como parte clave de la operación se mostrará a continuación las ecuaciones de actualización de las neuronas que forman la Red. Dada la funcionalidad esperada las ecuaciones tienen que ser recursivas para reflejar el proceso. La constante *Table* define un valor de la tabla de producciones.

La neurona W al codificar la palabra antecedente, está relacionada con la señal de entrada I_w y con la señal de entrada I_v , al mismo tiempo dada la función con que se diseñó E_w , ésta neurona fue incluida en la ecuación de W . Las neuronas C_1 y C_2 están presentes en todas las ecuaciones, por la función que éstas desempeñan.

La neurona L , está relacionada con las neuronas E_w y E_r , pues si se analiza las funciones de éstas últimas, es claro su inclusión en la ecuación de L . La señal de entrada I_w , al transferir la palabra antecedente binaria marcada por W , jugó un papel importante en la ecuación de L .

La ecuación de la neurona K , involucra la señal de entrada I_e , por la función que tiene esta señal en el proceso. También fue necesario incluir las neuronas auxiliares E_w , E_k y E_r , ya que los avisos de cuando se termina de leer una letra, cuando se termina de formar una regla de producción, son importantes para esta ecuación.

A continuación se muestran las ecuaciones correspondientes a la red de neuronas (aunque es bueno aclarar que no es objetivo del presente trabajo el detallar el por qué de cada uno de los coeficientes de cada ecuación):

Existen cuatro condiciones particulares para las ecuaciones de la red de neuronas, a saber: (*Table* es una constante usada en la ecuación de K)

$$C_1 = 0, C_2 = 0$$

Siendo el desarrollo de las ecuaciones como sigue:

$$W = \sigma[(2I_w + 1 + W)I_v / 4](3.a.1)$$

$$L = \sigma[I_w / 4 + I_v / 4L](3.b.1)$$

$$K = \sigma[(I_e)Table](3.c.1)$$

$$C_1 = \sigma[I_e](3.d.1)$$

$$C_2 = 0(3.e.1)$$

$$E_w = \sigma_i[16W - 6](3.f.1)$$

$$E_k = \sigma_i[216K - 57 - 216(E_k + E_r)](3.g.1)$$

$$E_r = \sigma[216K - 57 - 216(E_k + E_r)](3.h.1)$$

$$H = 0(3.i.1)$$

$$C_1=1, C_2=0$$

$$W = \sigma[(2I_w + 1 + W)I_v / 4 + (1 - E_r)W + (1)E_r[(4W - 1)(1 - E_w) + (16W - 7)E_w]](3.a.2)$$

$$L = \sigma[I_w / 4 + I_v / 4L + (1 - E_r)L + E_r[(1 - E_w)4L + E_w 16L]](3.b.2)$$

$$K = \sigma[(I_e)Table + [(6K - 1)(1 - E_k) - 3E_k](1 - E_r) + (36K - 23)E_r](3.c.2)$$

$$C_1 = [I_e + 1(1 - E_w E_r)](3.d.2)$$

$$C_2 = \sigma_i[E_w E_r](3.d.2)$$

$$E_w = \sigma_i[16W - 6](3.f.2)$$

$$E_k = \sigma_i[216K - 57 - 216(E_k + E_r)](3.g.2)$$

$$E_r = \sigma[216K - 57 - 216(E_k + E_r)](3.h.2)$$

$$H = 0(3.i.2)$$

$$C_1=0, C_2=1$$

$$W = \sigma[(2I_w + 1 + W)I_v / 4](3.a.3)$$

$$L = \sigma[I_w / 4 + I_v / 4L + 1/4L](3.b.3)$$

$$K = \sigma[(I_o)Table + [6K - 1(1 - E_k) - 3E_k](1 - E_r) + (36K - 23)E_r]](3.c.3)$$

$$C_1 = \sigma[I_o](3.d.3)$$

$$C_2 = \sigma_i[1] (3.e.3)$$

$$E_w = \sigma_i[16W - 6 - 16E_r](3.f.3)$$

$$E_k = \sigma_i[216K - 57 - 216(E_k + E_r)](3.g.3)$$

$$E_r = \sigma[216K - 57 - 216(E_k + E_r)](3.h.3)$$

$$H = 0(3.i.3)$$

$$C_1=1, C_2=1$$

$$W = \sigma[(2I_w + 1 + W)I_v / 4 + C_1(1 - C_2)E_r + [4W - 1](1 - E_w) + (4W - 3)E_w](3.a.4)$$

$$L = \sigma[I_w / 4 + I_v / 4L + E_w 16L] + 4L(3.b.4)$$

$$K = \sigma[(I_o + E_w)Table + (2)][(6K - 1(1 - E_k) - 3E_k)(1 - E_r) + (36K - 23)E_r] - (1 - E_w)](3.c.4)$$

$$C_1 = \sigma[I_o + 1](3.d.4)$$

$$C_2 = \sigma_i[(1 - E_w)] (3.e.4)$$

$$E_w = \sigma_i[16W - 6](3.f.4)$$

$$E_k = \sigma_i[216K - 57 - 216(E_k + E_r)](3.g.4)$$

$$H = 0(3.i.4)$$

Ecuaciones generales de actualización de las neuronas:

$$W = \sigma[(2I_w + 1 + W)I_v / 4 + C_1(1 - C_2)(1 - E_r)W + C_1(1 - C_2)E_r[(4W - 1)(1 - E_w) + (16W - 7)E_w] + (1 - C_1)C_2[W + (1 - E_k)1/4L + E_k 3/4L] + C_1C_2[4W - 1](1 - E_w) + (4W - 3)E_w](3.a)$$

$$L = \sigma[I_w / 4 + I_v / 4L + C_1(1 - C_2)(1 - E_r)L + C_1(1 - C_2)E_r[(1 - E_w)4L + E_w 16L] + (1 - C_1)C_2 \cdot 1/4L + C_1C_2 4L](3.b)$$

$$K = \sigma[(I_e + C_1C_2E_w)Table + (C_1 + C_2 - 2C_1C_2)[6K - 1(1 - E_k) - 3E_k](1 - E_r) + (36K - 23)E_r] - C_1C_2(1 - E_w)(3.c)$$

$$C_1 = \sigma_i[I_e + C_1(1 - C_2)(1 - E_w E_r) + (1 - C_1)C_2 E_r + C_1C_2](3.d)$$

$$C_2 = \sigma_i[C_1(1 - C_2)E_w E_r + C_2(1 - C_1E_w)] (3.e)$$

$$E_w = \sigma_i[16W - 6 - 16E_w C_1C_2 - 16E_w E_r(1 - C_2) - 16E_r C_2(1 - C_1)](3.f)$$

$$E_k = \sigma_i[216K - 57 - 216(E_k + E_r)](3.g)$$

$$E_r = \sigma[216K - 57 - 216(E_k + E_r)](3.h)$$

$$H = \sigma[(1 - C_1)C_2(6K - 4)(1 - I_v)](3.i)$$

Es conveniente aclarar que las ecuaciones anteriores fueron resultado de trabajo continuo y de experimentación tomando como base las necesidades que se tenían y el propósito final de cada neurona en sí.

En la ecuación de la neurona W al estar expresado su resultado en base-4, se requirió de $4W-1$, para controlar el máximo valor posible (tres). Se usaron las expresiones: $(4W-3)$ y $(16W-7)$, ya que era necesario controlar los casos: $3/4 = 0.75$ y $7/16 = 0.4375$, pues en estos valores la red debe hacer ceros ciertas sumas de términos, ya que W en estas situaciones tiene menos de tres letras.

También I_w es usada en la ecuación de W por la misma función que tiene esta señal de entrada en el contexto de la red, función que ya se comentó. De forma similar se incluye I_v para el control del envío de información.

E_w se diseñó con el objetivo de avisar, que la última producción de dígitos consecutivos a partir de W, pronto formará un código de una letra, por lo que fue necesario asegurar esto y contrarrestar tal situación de aviso de la manera: $(1 - E_w)$ también reflejado en la ecuación de W. Se usó el término $C_1(1 - C_2)$ para controlar la toma de ciertos valores bajo específicas condiciones de estas variables, fuentes de control de etapas de computación.

Esto es extensivo para todas las ecuaciones de la red. En todas se incluyeron C_1 y C_2 , para el control de etapas, como ya se mencionó. Las neuronas auxiliares E_w , E_k y E_r , son añadidas en las ecuaciones de las neuronas, en dependencia de la función de la neurona objetivo y de las funciones propias de estas neuronas auxiliares.

3.3.3 Funcionalidad

Como ya se ha dicho la Red de Neuronas propuesta está basada en seis neuronas continuas y tres neuronas binarias. Cada una tendrá una función en la Red. Estas funciones fueron ya comentadas cuando se explicó la motivación para la definición de cada neurona. Pero de manera resumida se dejará claro la función de cada neurona en la red.

La neurona W codifica la palabra antecedente, la neurona L contendrá la longitud l del de dos etiquetas codificado en W . De hecho almacena el valor 4^l . A su vez, la neurona K sostiene una tabla, codificando las reglas de producción.

Dos de las seis neuronas binarias: C_1 y C_2 cumplen su función de neuronas de control e indican la etapa del ciclo computacional. E_w , E_k y E_r , por su parte, ayudan a reconocer el estado de la secuencia saliente.

E_w indica que la última producción de dígitos consecutivos leídos a partir de W pronto formará el código de una letra. E_k y E_r se relacionan a K de una manera muy similar.

E_k indica que la red terminará de leer una letra codificada a partir de una regla de producción, mientras que E_r muestra que la producción en breve formará una codificación de una regla de producción entera. H es la neurona de parada.

Es durante la simulación que las redes neuronales pueden desempeñar funciones específicas y que en un final también serán utilizadas para definir procesos computables como los descritos por *Turing*. Ciertas familias de Redes Neuronales pueden ser descritas como sistemas de cómputo que representan un proceso efectivo.

Entradas/Canales

I_w I_v I_b I_c C_1 C_2 E_w E_k E_r W L K H

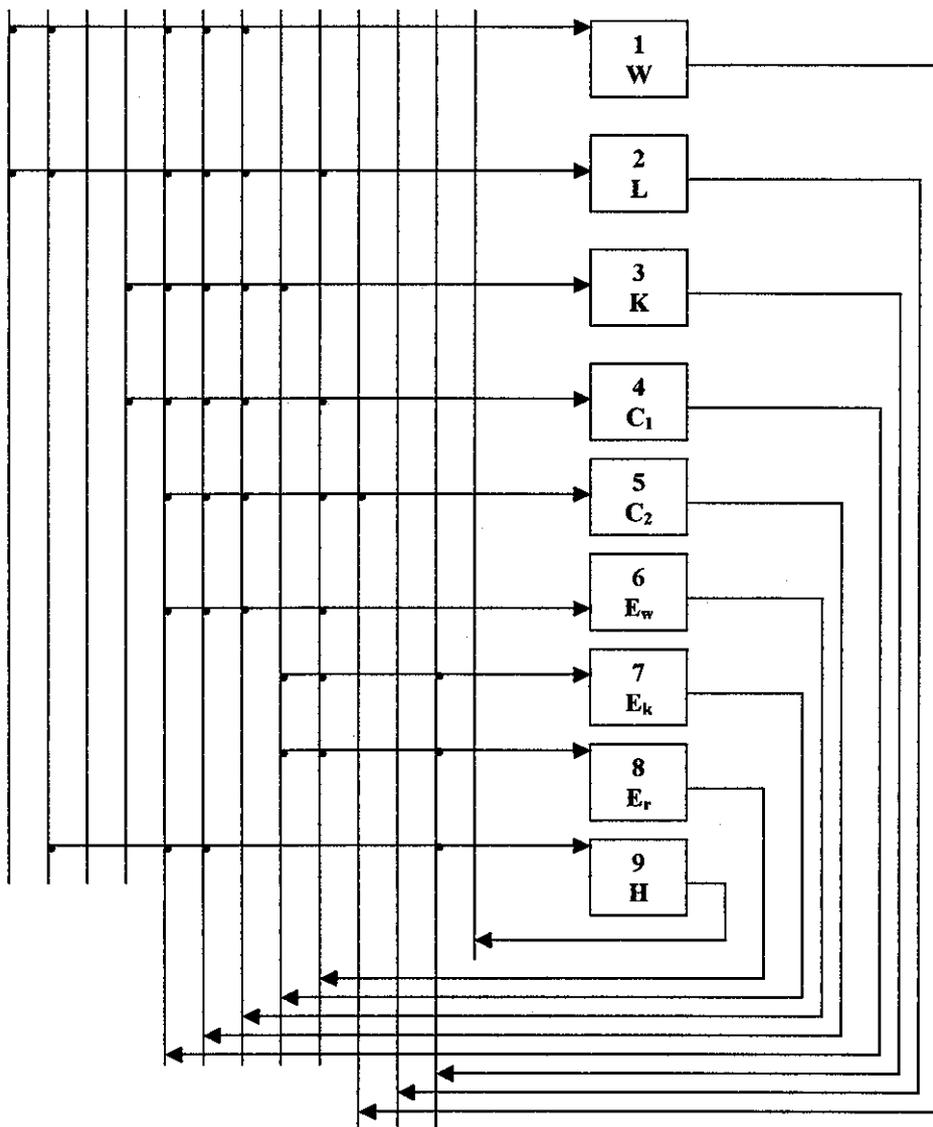


Figura 3.1 Conexión entre neuronas.

CAPITULO 4: SIMULACION

4.1 Introducción

En este capítulo se mostrará la operación de la red de Nueve Neuronas, comprobándose que el sistema de dos etiquetas y esta red operan diferentes, pero sus resultados son equivalentes.

Primero se mostrará la codificación empleada en este trabajo, seguidamente se explicará el comportamiento del circuito neuronal ante las señales de entradas o estímulos, finalizando con el análisis de la simulación, detallando cada uno de los ciclos de cómputos.

4.2 Codificación

Las neuronas artificiales son dispositivos que procesan señales y pueden generar fenómenos de memoria a corto plazo, pero no tienen la capacidad de almacenar grandes secuencias de señales o datos de entrada. Por esta razón, toda la información referente al sistema de Post que se va a simular se establece en términos la codificación numérica que se describe a continuación.

Consideremos que el alfabeto del sistema de Post es $L = \{a_1, \dots, a_N\}$, donde N es el número de letras del alfabeto, siendo a_N la letra de parada del sistema. Cada letra a_i de L , $1 \leq i \leq N$, se codifica como $\text{cod}(a_i) = 1^i 3$ (repetir cadenas de unos), donde el símbolo "3" actúa como delimitador derecho para distinguir las diferentes letras de L . Así, una palabra sobre L , digamos $m = b_1 \dots b_k$, donde $b_j = a_{i_j}$, queda codificada de la manera siguiente:

$$\text{cod}(m) = \text{cod}(b_1) \dots \text{cod}(b_k) = \text{cod}(a_{i_1}) \dots \text{cod}(a_{i_k}) = 1^{i_1} 3 \dots 1^{i_k} 3.$$

Para el conjunto de producciones se establecerá una codificación similar a la de la palabra de entrada. Así, sean P_1, \dots, P_{N-1} las producciones del sistema de Post que se va a simular, considerando que P_N , la única producción de parada, es vacía. Este conjunto se codifica de la manera siguiente:
 $135\text{cod}(P_1)5 \dots \text{cod}(P_{N-1})55$.

En otras palabras, las palabras de entrada se van a codificar en una representación en base 4, mientras que las producciones se representarán en base 6. Adicionalmente, la longitud de la palabra de entrada también se representará en base 4, mientras que la señal constante *table* se representará en base 6.

4.3 Simulación

Esta sección tendrá por objetivo, mostrar la simulación de la red de neuronas tomando como referencia al sistema de dos etiquetas de la sección 1.4. Para lograr este fin se mostrarán las condiciones iniciales, las gráficas de las nueve neuronas según las condiciones iniciales, para seguir más tarde con los ciclos de cómputos y la interpretación de los mismos.

4.3.1 Condiciones Iniciales

Las señales de entrada o estímulos que se usaron en la simulación son hallados en la siguiente tabla.

t	=	0	1	2	3	4	5	6	7	8	9
I_w	=	1	0	0	1	0	0	1	0	0	0
I_v	=	1	1	1	1	1	1	1	1	1	0
I_b	=	1	0	0	0	0	0	0	0	0	0
I_e	=	0	0	0	0	0	0	0	0	1	0

Tabla 4.1 Señales de Entrada

Como ya se mencionó en la sección 3.3.1 del pasado capítulo, la señal I_w se usa para estimular a la neurona W de forma tal que su respuesta corresponda a la codificación de la palabra de entrada ω . La señal I_v se usa, por un lado, para validar a la señal I_w , y por otro lado, para estimular a la neurona L de forma tal que su respuesta corresponda a la longitud $|\omega|$ de la palabra de entrada. Por último, las señales impulso I_b e I_e se usan para indicar los momentos en los que inicia y termina, respectivamente, la transmisión de valores significativos.

La etapa de lectura de datos ocurre una sola vez, por lo que la red neuronal tiene valor cero en todas las neuronas. La red es activada por las señales de entrada. Esta etapa ocurre cuando las neuronas de control toman valor $(C_1, C_2) = (0, 0)$. Esta situación inicial, en la que las neuronas de control C_1 y C_2 toman valor cero será argumentada en la sección siguiente con la ayuda de las gráficas.

La actualización, en la etapa de lectura de datos, se da por medio de las siguientes ecuaciones, las cuales salen de las ecuaciones anteriores, planteadas en el capítulo 3, siendo condiciones particulares de las misma, a saber: etapa de lectura de datos.

$$W = (2I_w + 1 + W)I_v / 4 \quad (\text{ecuación 4.a}) \quad L = I_b / 4 + (I_v / 4)L \quad (\text{ecuación 4.b})$$

$$K = I_e \cdot \text{Table} \quad (\text{ecuación 4.c}) \quad C_1 = I_e \quad (\text{ecuación 4.d})$$

$$C_2 = 0$$

(ecuación 4.e)

La primera ecuación (4.a) muestra la acumulación en línea de $\text{cod}(w)$ en la neurona W . Al mismo tiempo, la longitud l almacenada en L se incrementa en $1/4$ en cada paso, siempre y cuando $l_v = 1$. Cuando este proceso se completa, l_v se convierte en 0. Simultáneamente l_e se ubica en 1. Esto produce y carga la tabla a K y la neurona C_1 se ubica en 1 como preparación a la siguiente etapa. Es decir, la primera etapa del ciclo de computación.

A continuación serán mostradas las gráficas correspondientes a los estímulos o señales de entradas, reflejados en la tabla expuesta inicialmente en la sección anterior. Se obtienen cuatro estímulos consecutivos porque hay cuatro canales de entrada diferentes.

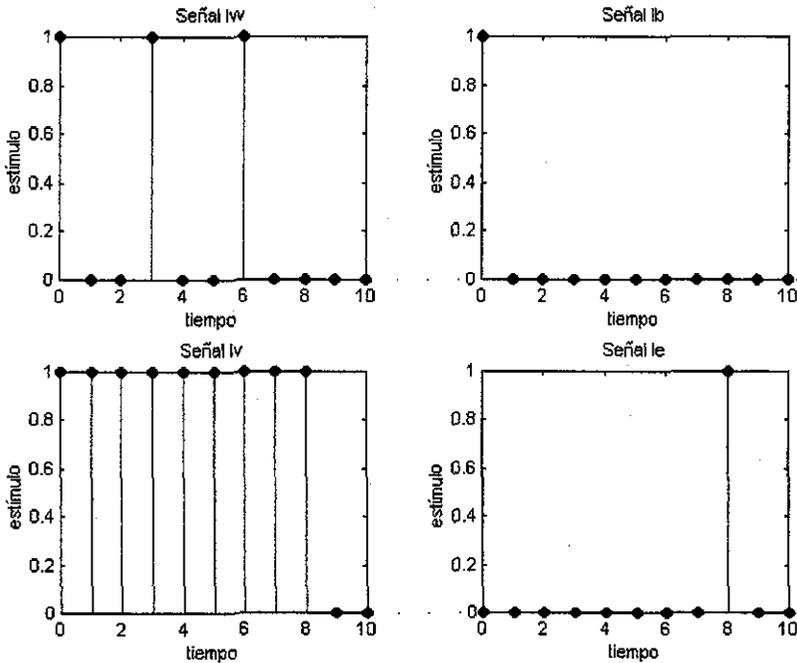


Figura 4.1 Señales de entrada

Para las señales anteriores, la red de neuronas propuestas, exhibe el comportamiento que se ilustra en la Figuras 4.2, 4.3, 4.4 y 4.5, a partir de condiciones iniciales nulas, es decir, a partir de:

$$y_W(0) = y_L(0) = y_K(0) = y_{C_1}(0) = y_{C_2}(0) = y_{E_w}(0) = y_{E_x}(0) = y_E(0) = y_H(0).$$

(ecuación 4.e)

4.3.2 Representación gráfica de resultados

Lectura de Datos, comportamiento de la red neuronal a partir de las condiciones iniciales:

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
0	0.00000000000000	0.00000000000000	0.00000000000000	0	0	0	0	0	0
1	0.75000000000000	0.25000000000000	0.00000000000000	0	0	0	0	0	0
2	0.43750000000000	0.06250000000000	0.00000000000000	0	0	1	0	0	0
3	0.35937500000000	0.01562500000000	0.00000000000000	0	0	1	0	0	0
4	0.83984375000000	0.00390625000000	0.00000000000000	0	0	0	0	0	0
5	0.45996093750000	0.00097656250000	0.00000000000000	0	0	1	0	0	0
6	0.36499023437500	0.00024414062500	0.00000000000000	0	0	1	0	0	0
7	0.84124755859375	0.00006103515625	0.00000000000000	0	0	0	0	0	0
8	0.46031188964844	0.00001525878906	0.00000000000000	0	0	1	0	0	0

Tabla 4.2 Comportamiento cuantitativo de la red de nueve neuronas

4.3.3 Análisis de resultados

En la etapa de lectura de datos, sólo cambia la respuesta de las neuronas analógicas W y L ; la primera acumula el valor codificado de la palabra de entrada ω , mientras que la segunda acumula su longitud, $|\omega|$. El proceso anterior concluye cuando se termina de transmitir las señales significativas, lo cual ocurre en el instante de tiempo $t = 8$ unidades.

A partir de las ecuaciones mostradas en la sección 4.3.1, se puede establecer que la respuesta neuronal a los estímulos ya comentados durante la lectura de los datos, es la siguiente:

t	W	L	K
0	0	0	0
1	.3 ₄	.1 ₄	0
2	.13 ₄	.01 ₄	0
3	.113 ₄	0.001 ₄	0
4	.3113 ₄	0.0001 ₄	0
5	.13113 ₄	0.00001 ₄	0
6	.113113 ₄	0.000001 ₄	0
7	.3113113 ₄	0.0000001 ₄	0
8	.13113113 ₄	0.00000001 ₄	0

Tabla 4.3 Respuesta neuronal a los estímulos, codificados en base-4, asociados a la tabla anterior

Siguiendo la tabla anterior, la primera fila, en $t=0$, $W=0$ y la longitud L será cero, pues se calcula como 4^{-L} , K es cero, pues todavía se está en la etapa de lectura de datos. En la segunda fila, $t=1$, W vale 0.3_4 , que no es más que el valor de 3 delimitador usado para la base 4; a saber sería: $0.3_4 = 3 \times 4^{-1}$; la neurona L almacena $0.1_4 = 1 \times 4^{-1}$. En $t=2$, la neurona W tiene el valor de $0.13_4 = 1 \times 4^{-1} + 3 \times 4^{-2}$, la neurona L en este caso toma el valor de $0.01_4 = 0 \times 4^{-1} + 1 \times 4^{-2}$.

Seguidamente se mostrarán las gráficas de las nueve neuronas propuestas, dichas gráficas serán resultado de los estímulos aplicados, tomando en consideración la tabla 4.2 antes expuesta y las ecuaciones .

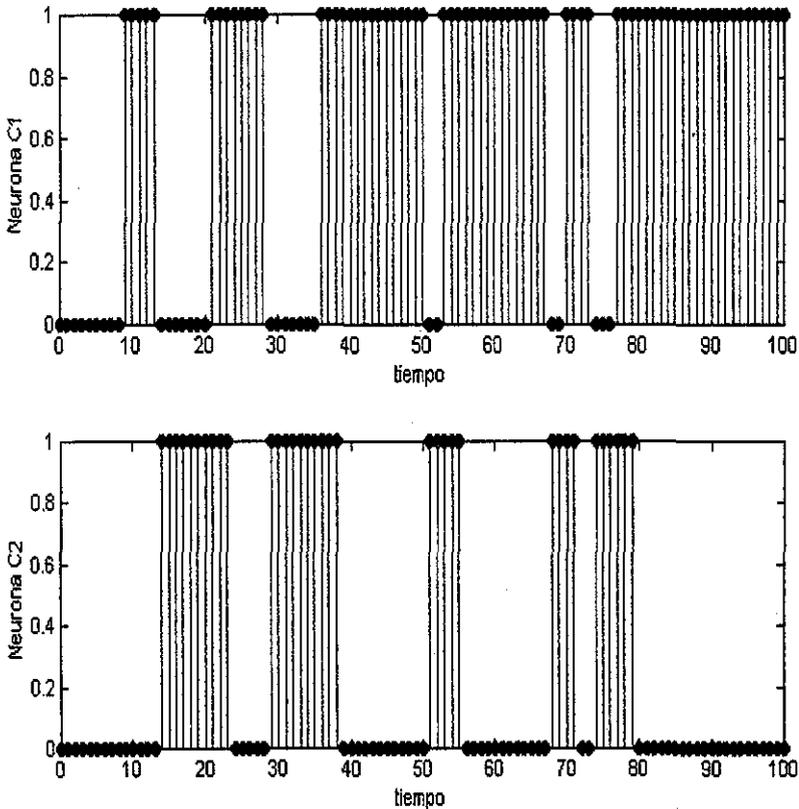


Figura 4.2

Comportamiento de las neuronas C_1 y C_2 como respuesta a los estímulos I_w , I_v , I_b e I_e

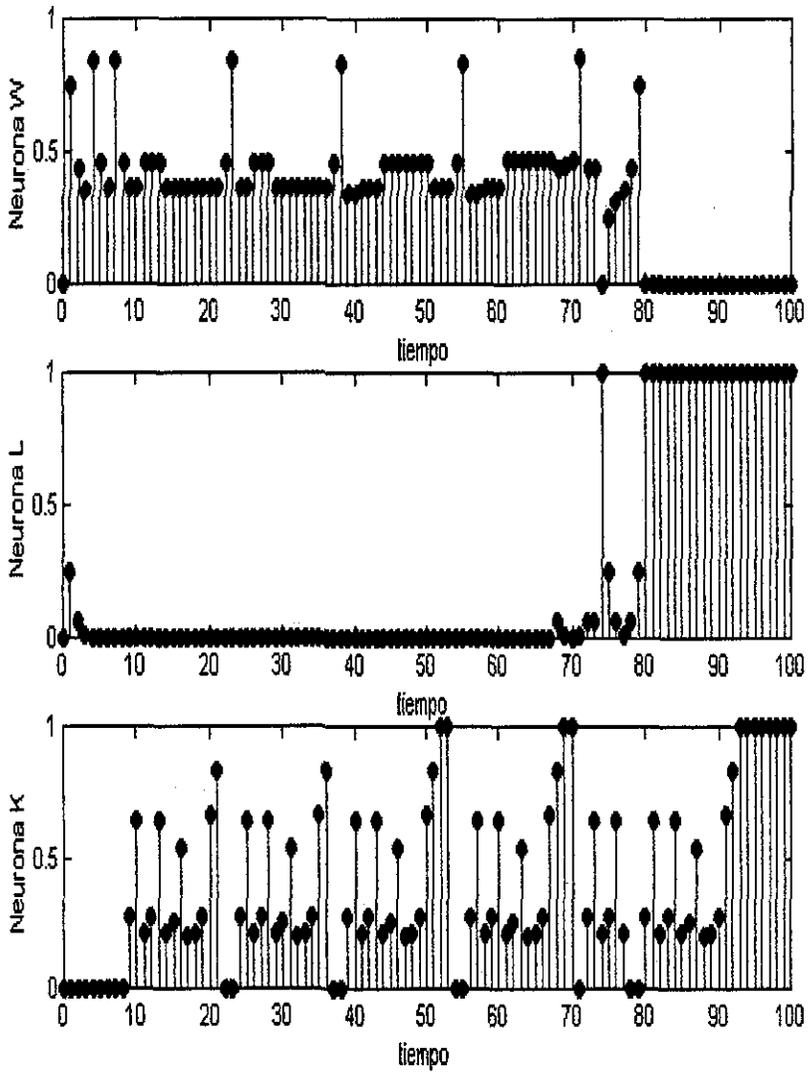


Figura 4.3

Comportamiento de las Neuronas W,K y L como respuesta a los estímulos I_w , I_v , I_b e I_o .

TESIS CON
FALLA DE ORIGEN

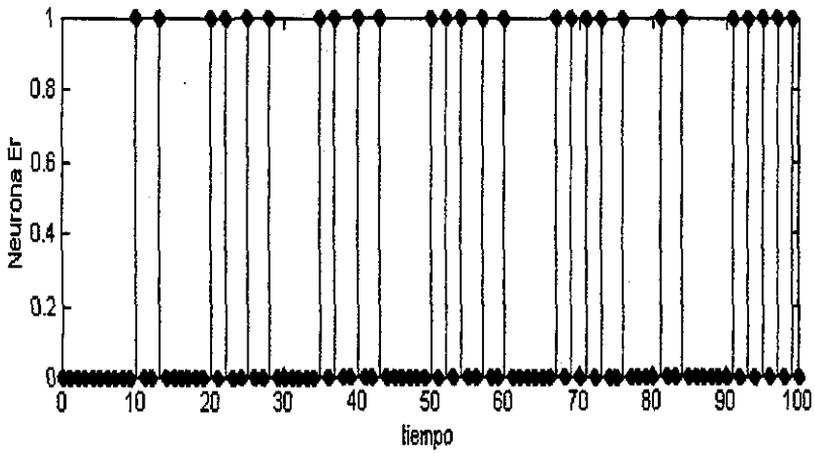
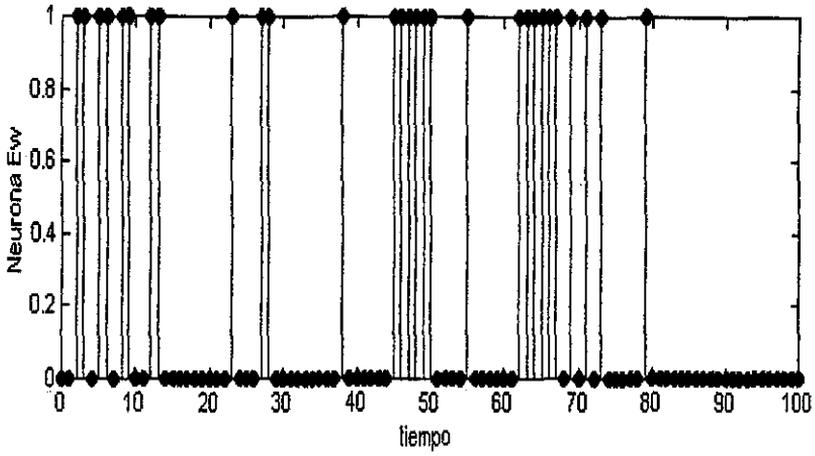


Figura 4.4

Comportamiento de las neuronas E_w y E_r como respuesta a los estímulos I_w , I_v , I_b e I_o .

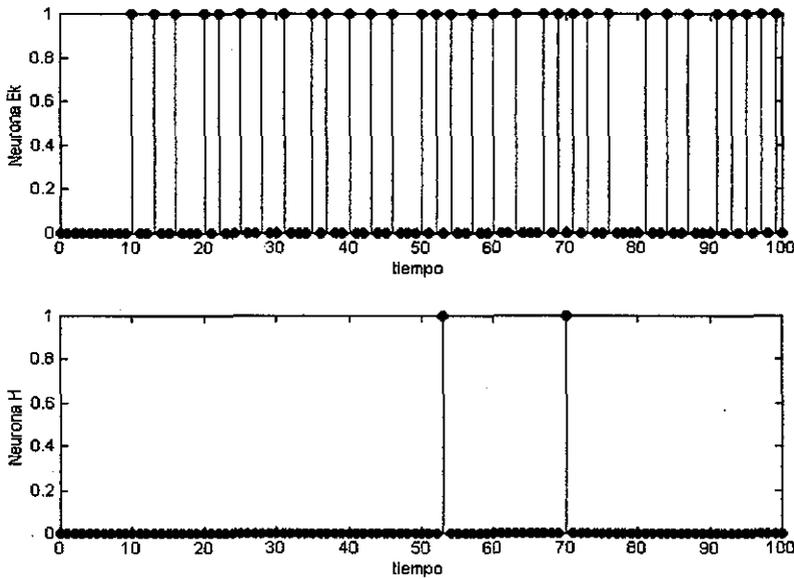


Figura 4.5

Comportamiento de las Neuronas E_k y H como respuesta a los estímulos I_w , I_v , I_b e I_o

Es claro que el siguiente dígito binario de los estímulos completa la codificación de la palabra de entrada, establece su longitud y permite que la neurona K adquiera el valor constante *table*, el cual corresponde a la codificación, en base 6, del conjunto de producciones del sistema de Post que se va a simular con la red neuronal.

Desde la etapa de lectura de datos C_1 y C_2 juegan un papel de control, complementándose en algunos casos y siendo iguales en otros, es decir según la gráfica representada por medio de la figura 4.2, se nota los siguientes ciclos: $(C_1, C_2) \rightarrow (0,0), (1,0), (0,1), (1,1)$. Las neuronas C_1 y C_2 se destinan a marcar etapas dentro del ciclo computacional. La Figura 4.2 se observa la función tipo interruptor que desempeñan las mencionadas neuronas.

Además cabe señalar que las neuronas C_1 y C_2 muestran la consecutividad de los tres pasos de trabajo del sistema de dos etiquetas, a saber: $(1,0)$, equivale al paso uno explicado en la sección 1.3 de Antecedentes, $(0,1)$ al paso dos, $(1,1)$ equivale al paso tres. Esta secuencia se repetirá hasta que se encuentre la condición o letra de parada, mismas que tuvieron que ser definidas en el sistema de dos etiquetas.

Una vez concluida la etapa de lectura de datos, el Primer Ciclo Computacional ocurre comenzando con la etapa (1,0). El estímulo impulso I_e activa a la neurona de control C_1 , lo cual es la indicación de que va a dar comienzo el primer ciclo de cómputo. Éste se describe a continuación. Tomando C_1 el valor de uno.

Los ciclos computacionales se sucederán en el orden: $(1,0) \rightarrow (0,1) \rightarrow (1,1)$



En la etapa (1,0) de la red, borra la primera letra codificada de W y prepara a K para que inicie con la regla de producción asociada con la letra removida. En la etapa (0,1) las letras de la regla asociada con la letra recientemente removida se agregan a W . En la etapa (1,1) la red remueve la segunda letra de W . Estas tres etapas son los tres pasos propios e inherentes al funcionamiento del sistema de dos etiquetas explicados en el capítulo uno y en el capítulo tres.

El hecho que se detengan los ciclos computacionales estará relacionado con la condición de parada del sistema de dos etiquetas que la red de nueve neuronas propuesta esté simulando. La red que se señala en el presente trabajo puede simular a cualquier sistema de dos etiquetas, especificando siempre la condición de parada, la cual debe ser única.

4.4 Interpretación

Es importante recordar que los resultados obtenidos son fruto de aplicar las producciones del sistema de dos etiquetas puesto de ejemplo en la sección 1.4

4.4.1 Primer ciclo de cómputo

Etapa $(C_1, C_2) = (1, 0)$, Determinación y eliminación del primer símbolo de la palabra de entrada codificada.

La primera línea de la tabla siguiente corresponde al momento en el que la respuesta de las tres neuronas analógicas poseen la codificación completa. A partir de este momento, la actividad neuronal se conduce de manera libre, es decir, sin estímulos externos. Esta etapa en particular es controlada exclusivamente por la neurona C_1 .

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
9	0.36507797241211	0.00000381469727	0.27413126310715	1	0	1	0	0	0
10	0.36507797241211	0.00000381469727	0.64478757864288	1	0	0	1	1	0
11	0.46031188964844	0.00001525878906	0.21235283114375	1	0	0	0	0	0
12	0.46031188964844	0.00001525878906	0.27411698686252	1	0	1	0	0	0
13	0.46031188964844	0.00001525878906	0.64470192117514	1	0	1	1	1	0

Tabla 4.4 Valores de la red de nueve neuronas en etapa (1,0) del primer ciclo de cómputo.

Al interpretar los resultados anteriores en términos de las codificaciones establecidas se obtienen los valores siguientes:

t	W	L	K
10	0.113113113 ₄	0.00000001 ₄	0.351135113111355 ₆
11	0.13113113 ₄	0.0000001 ₄	0.1135113111355 ₆
12	0.13113113 ₄	0.0000001 ₄	0.135113111355 ₆
13	0.13113113 ₄	0.0000001 ₄	0.35113111355 ₆

Tabla 4.5 Resultados en base-4 asociados a la tabla 4.4, de la red de nueve neuronas en la etapa (1,0) del primer ciclo de cómputo.

Como se puede observar en la tabla anterior, durante esta etapa se elimina el primer símbolo de la palabra de entrada (lo cual corresponde al cambio en la respuesta de la neurona W), se ajusta la longitud de la nueva palabra codificada (esto corresponde al cambio en la respuesta de L) y, finalmente, se localiza la producción asociada al símbolo reconocido (lo cual corresponde a los cambios en K).

Es conveniente remarcar los cambios que ocurren en la respuesta de la neurona K , ya que en este caso es necesario retirar el marcador inicial de la codificación, la secuencia inicial 135, y localizar la posición en la que se encuentra la producción buscada. En este caso, tal producción es $b \rightarrow bc$, lo cual corresponde a la segunda producción en la codificación, la secuencia 1131113, por lo que se debe retirar la primera producción. En el siguiente instante de tiempo, se completa la eliminación del primer símbolo de la cadena de entrada, con el consecuente ajuste en la longitud de la nueva cadena codificada y se termina de eliminar la primera producción. Esto coincide con el cambio a la siguiente etapa del primer ciclo de cómputo, lo cual se señala con la activación simultánea de las neuronas binarias E_w y E_r . Esta nueva etapa del proceso se describe a continuación.

Etapa (C_1, C_2) = (0, 1): Aplicación de la producción determinada.

Como se acaba de mencionar, la primera línea de la tabla siguiente corresponde a la palabra de entrada 0.113113₄, su longitud 0.000001₄ y el conjunto de producciones 0.113111355₆. A partir de este momento, el comportamiento neuronal es controlado por la neurona C_2 .

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
14	0.36499023437500	0.00024414062500	0.20926916230508	0	1	0	0	0	0
15	0.36505126953125	0.00006103515625	0.25561497383046	0	1	0	0	0	0
16	0.36506652832031	0.00001525678906	0.53368984298277	0	1	0	1	0	0
17	0.36507797241211	0.00000381469727	0.20213905789660	0	1	0	0	0	0
18	0.36507892608643	0.00000095367432	0.21283434737961	0	1	0	0	0	0
19	0.36507916450500	0.00000023841858	0.27700608427767	0	1	0	0	0	0
20	0.36507922410965	0.00000005960464	0.66203650566604	0	1	0	1	1	0

Tabla 4.6 Valores de la red de nueve neuronas en etapa (0,1) del primer ciclo de cómputo

La interpretación del comportamiento anterior es la siguiente:

t	W	L	K
15	0.1131131 ₄	0.0000001 ₄	0.13111355 ₆
16	0.11311311 ₄	0.00000001 ₄	0.3111355 ₆
17	0.113113113 ₄	0.000000001 ₄	0.111355 ₆
18	0.1131131131 ₄	0.0000000001 ₄	0.11355 ₆
19	0.11311311311 ₄	0.00000000001 ₄	0.1355 ₆
20	0.113113113111 ₄	0.000000000001 ₄	0.355 ₆

Tabla 4.7 Resultados en base-4 asociados a la tabla 4.6, de la red de nueve neuronas en la etapa (0,1) del primer ciclo de cómputo.

En esta etapa se retira el símbolo codificado en la producción aislada y se concatena al final de la cadena de entrada, con el consiguiente ajuste en su longitud. Esta acción es equivalente a la aplicación de la producción del sistema de Post al final de la cadena de entrada.

Una vez más, en el siguiente instante de tiempo se completa la aplicación de la producción a la cadena de entrada, coincidiendo con el cambio de etapa. Este cambio se promueve en el momento en el que la respuesta de la neurona E_r se hace no nula. La nueva etapa debe concluir el proceso de re - escritura, lo cual se muestra a continuación.

Etapa $(C_1, C_2) = (1, 1)$: Eliminación del segundo símbolo inicial de la cadena de entrada.

En esta etapa se completa la aplicación de la producción a la cadena de entrada. La etapa comienza con la palabra de entrada 0.1131131131113₄, la cual corresponde, según la codificación establecida, a la palabra *bbbc*. El conjunto de producciones se ha reducido ahora a simplemente 0.5₆, la cual corresponde a la producción de parada o producción vacía según la codificación establecida. Resta por eliminar la primera *b* de la palabra actual, símbolo que corresponde al segundo símbolo de la palabra original $\omega = bbb$. Esta nueva etapa está controlada por las dos neuronas de control C_1 y C_2 .

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
21	0.36507926881313	0.00000001490116	0.83331420397735	1	1	0	0	0	0
22	0.46031707525253	0.00000005960464	0.00000000000000	1	1	0	1	1	0
23	0.84126830101013	0.00000023841858	0.00000000000000	1	1	1	0	0	0

Tabla 4.8 Valores de la red de nueve neuronas en etapa (1,1) del primer ciclo de cómputo

La interpretación es la siguiente:

t	W	L	K
22	0.131131131113_4	0.00000000001_4	0_6
23	0.31131131113_4	0.0000000001_4	0_6

Tabla 4.9 Resultados en base-4 asociados a la tabla 4.8, de la red de nueve neuronas en la etapa (1,1) del primer ciclo de cómputo.

La eliminación del último dígito de la codificación se da en el siguiente instante de tiempo, coincidiendo con el cambio de etapa. Este cambio es promovido ahora por la activación de la neurona E_w . Durante las tres etapas que conformaron a este primer ciclo de cómputo, la neurona de parada H nunca se activó, lo cual indica que el proceso de cómputo iniciado con la presentación de los estímulos todavía no concluye. La siguiente etapa nuevamente es la etapa (1, 0), pero ahora considerando que la nueva palabra de entrada es 0.1131131113_4 , código correspondiente a *bbc*. Los siguientes ciclos de cómputo se describen, a continuación.

4.4.2 Segundo ciclo de cómputo

Etapa (C_1, C_2) = (1, 0): Eliminación del primer símbolo de la nueva cadena de entrada.

Comportamiento neuronal iniciando con la palabra codificada *bbc*. Nótese que el cambio de etapa promovió además que la neurona K reajustará su actividad al valor constante *table*. Esto es, la respuesta de la neurona K representa nuevamente al conjunto de producciones del sistema de Post.

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
24	0.36507320404053	0.00000095367432	0.27413126310715	1	0	0	0	0	0
25	0.36507320404053	0.00000095367432	0.64478757864288	1	0	0	1	1	0
26	0.46029281616211	0.00000381469727	0.21235283114375	1	0	0	0	0	0
27	0.46029281616211	0.00000381469727	0.27411698686252	1	0	1	0	0	0
28	0.46029281616211	0.00000381469727	0.64470192117514	1	0	1	1	1	0

Tabla 4.10 Valores de la red de nueve neuronas en etapa (1,0) del segundo ciclo de cómputo

Interpretación:

t	W	L	K
25	0.1131131113_4	0.0000000001_4	0.351135113111355_6
26	0.131131113_4	0.000000001_4	0.1135113111355_6
27	0.131131113_4	0.000000001_4	0.135113111355_6
28	0.131131113_4	0.000000001_4	0.35113111355_6

Tabla 4.11 Resultados en base-4 asociados a la tabla 4.10, de la red de nueve neuronas en la etapa (1,0) del segundo ciclo de cómputo.

En el siguiente instante de tiempo se termina de eliminar el primer símbolo de la cadena *bbc* —en este caso *b*— se ajusta la longitud de la nueva cadena y se procede a localizar la producción asociada al símbolo eliminado —en este caso *b* → *bc*. Este proceso se lleva a cabo en la etapa siguiente.

Etapa $(C_1, C_2) = (0, 1)$: Aplicación de la producción determinada.

En esta etapa se deben eliminar los símbolos de la producción determinada, *bc* en este caso, y concatenarlos al final de la cadena de entrada para obtener la nueva cadena *bcbc*.

<i>t</i>	<i>W</i>	<i>L</i>	<i>K</i>	<i>C</i> ₁	<i>C</i> ₂	<i>E</i> _w	<i>E</i> _k	<i>E</i> _r	<i>H</i>
29	0.36468505859375	0.00006103515625	0.20926916230508	0	1	0	0	0	0
30	0.36470031738281	0.00001525878906	0.25561497383046	0	1	0	0	0	0
31	0.36470413208008	0.00000381469727	0.53368984298277	0	1	0	1	0	0
32	0.36470699310303	0.00000095367432	0.20213905789660	0	1	0	0	0	0
33	0.36470723152161	0.00000023841858	0.21283434737961	0	1	0	0	0	0
34	0.36470729112625	0.00000005960464	0.27700608427767	0	1	0	0	0	0
35	0.36470730602741	0.00000001490116	0.66203650566604	0	1	0	1	1	0

Tabla 4.12 Valores de la red de nueve neuronas en etapa (0,1) del segundo ciclo de cómputo

Interpretación:

<i>t</i>	<i>W</i>	<i>L</i>	<i>K</i>
30	0.11311131 ₄	0.00000001 ₄	0.13111355 ₆
31	0.113111311 ₄	0.000000001 ₄	0.3111355 ₆
32	0.1131113113 ₄	0.0000000001 ₄	0.111355 ₆
33	0.11311131131 ₄	0.00000000001 ₄	0.11355 ₆
34	0.113111311311 ₄	0.000000000001 ₄	0.1355 ₆
35	0.1131113113111 ₄	0.0000000000001 ₄	0.355 ₆

Tabla 4.13 Resultados en base-4 asociados a la tabla 4.12, de la red de nueve neuronas en la etapa (0,1) del segundo ciclo de cómputo.

En el siguiente instante de tiempo se completa la generación de la nueva cadena, *bcbc*, mientras que el conjunto de producciones ha quedado vacío. Todo esto coincide con el cambio de etapa.

Etapa $(C_1, C_2) = (1, 1)$: Eliminación del segundo símbolo inicial.

En esta etapa se completa la transformación de la cadena original *bbc* en la cadena *cbc*.

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
36	0.36470731720328	0.00000000372529	0.83331420397735	1	1	0	0	0	0
37	0.45882926881313	0.0000001490116	0.00000000000000	1	1	0	1	1	0
38	0.83531707525253	0.00000005960464	0.00000000000000	1	1	1	0	0	0

Tabla 4.14 Valores de la red de nueve neuronas en etapa (1,1) del segundo ciclo de cómputo

Interpretación:

t	W	L	K
37	0.1311131131113 ₄	0.0000000000001 ₄	0 ₆
38	0.311131131113 ₄	0.000000000001 ₄	0 ₆

Tabla 4.15 Resultados en base-4 asociados a la tabla 4.14, de la red de nueve neuronas en la etapa (1,1) del segundo ciclo de cómputo.

La eliminación del último dígito de la codificación se da en el siguiente instante de tiempo, coincidiendo con el cambio de etapa. De nueva cuenta, durante las tres etapas del segundo ciclo de cómputo, la neurona de parada H nunca se activó, lo cual indica que el proceso de cómputo todavía no concluye.

4.4.3 Tercer ciclo de cómputo

Etapa (C_1, C_2) = (1, 0): Eliminación del primer símbolo de la cadena de entrada.

En esta etapa se elimina el primer símbolo de la cadena cbc , el cual es precisamente el símbolo de parada.

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
39	0.34126830101013	0.00000023841858	0.27413126310715	1	0	0	0	0	0
40	0.34126830101013	0.00000023841858	0.64478757864288	1	0	0	1	1	0
41	0.36507320404053	0.00000095367432	0.21235283114375	1	0	0	0	0	0
42	0.36507320404053	0.00000095367432	0.27411698686252	1	0	0	0	0	0
43	0.36507320404053	0.00000095367432	0.64470192117514	1	0	0	1	1	0
44	0.46029281616211	0.00000381469727	0.20926916230508	1	0	0	0	0	0
45	0.46029281616211	0.00000381469727	0.25561497383046	1	0	1	0	0	0
46	0.46029281616211	0.00000381469727	0.53368984298277	1	0	1	1	0	0
47	0.46029281616211	0.00000381469727	0.20213905789660	1	0	1	0	0	0
48	0.46029281616211	0.00000381469727	0.21283434737961	1	0	1	0	0	0
49	0.46029281616211	0.00000381469727	0.27700608427767	1	0	1	0	0	0
50	0.46029281616211	0.00000381469727	0.66203650566604	1	0	1	1	1	0

Tabla 4.16 Valores de la red de nueve neuronas en etapa (1,0) del tercer ciclo de cómputo

Interpretación:

t	W	L	K
40	0.11131131113 ₄	0.0000000001 ₄	0.351135113111355 ₆
41	0.1131131113 ₄	0.0000000001 ₄	0.1135113111355 ₆
42	0.1131131113 ₄	0.0000000001 ₄	0.135113111355 ₆
43	0.1131131113 ₄	0.0000000001 ₄	0.35113111355 ₆
44	0.131131113 ₄	0.000000001 ₄	0.113111355 ₆
45	0.131131113 ₄	0.000000001 ₄	0.13111355 ₆
46	0.131131113 ₄	0.000000001 ₄	0.3111355 ₆
47	0.131131113 ₄	0.000000001 ₄	0.111355 ₆
48	0.131131113 ₄	0.000000001 ₄	0.11355 ₆
49	0.131131113 ₄	0.000000001 ₄	0.1355 ₆
50	0.131131113 ₄	0.000000001 ₄	0.55 ₆

Tabla 4.17 Resultados en base-4 asociados a la tabla 4.16, de la red de nueve neuronas en la etapa (1,0) del tercer ciclo de cómputo

En el siguiente instante de tiempo se elimina completamente el primer símbolo de la cadena de entrada, obteniéndose la respuesta neuronal 0.1131113₄, y el conjunto de producciones se reduce al valor neuronal 0.5₆, el corresponde a la producción vacía.

Etapa (C_1, C_2) = (0, 1): Aplicación de la producción determinada.

Durante esta etapa se establece que el proceso de cómputo está por concluir. Se debe aclarar, sin embargo, que esto no implica que el comportamiento neuronal va a concluir, esto es, que de alguna manera las neuronas pierdan bruscamente su actividad, ya que esto es imposible. Sólo es una indicación de que se está por concluir la parte significativa, para fines de la equivalencia, del comportamiento neuronal.

t	W	L	K	C_1	C_2	E_w	E_k	E_f	H
51	0.364688505859375	0.00006103515625	0.83331420397735	0	1	0	0	0	0
52	0.36470031738281	0.00001525878906	1.00000000000000	0	1	0	1	1	0

Tabla 4.18 Valores de la red de nueve neuronas en etapa (0,1) del tercer ciclo de cómputo

Interpretación:

t	W	L	K
51	0.1131113 ₄	0.0000001 ₄	0.5 ₆
52	0.11311131 ₄	0.00000001 ₄	1 ₆

Tabla 4.19 Resultados en base-4 asociados a la tabla 4.18, de la red de nueve neuronas en la etapa (0,1) del tercer ciclo de cómputo

En el siguiente instante de tiempo se debe activar la neurona de parada H coincidiendo con el cambio de etapa.

Etapa $(C_1, C_2) = (1, 1)$: Conclusión.

El comportamiento neuronal que se ha establecido desde la presentación de los estímulos hasta el momento presente pudo ser interpretado con éxito de manera similar a como se lleva a cabo el proceso de la cadena de entrada bbb por el sistema de Post de la sección 1.4.

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
53	0.36471176147461	0.00000381469727	1.00000000000000	1	1	0	0	0	1
54	0.45884704589844	0.00001525878906	0.00000000000000	1	1	0	1	1	0
55	0.83538818359375	0.00006103515625	0.00000000000000	1	1	1	0	0	0

Tabla 4.20 Valores de la red de nueve neuronas en etapa (1.1) del tercer ciclo de cómputo

Nótese que la neurona de parada H responde con una señal impulso. Esta neurona sólo se usa para señalar que el comportamiento neuronal ha sido equivalente al proceso de re - escritura del sistema de Post.

A partir del momento en el que tal neurona muestra por primera vez una respuesta no nula, el comportamiento neuronal debe decaer gradualmente hasta que, primero, se genera la codificación de la palabra resultante final, c en este caso, como respuesta de la neurona K y, en segundo término, la actividad neuronal termina por decaer hacia las condiciones de reposo, en este caso condiciones nulas, ya que el circuito neuronal ya no recibe nuevos estímulos. En la tabla siguiente se muestra, a manera de ejemplo, la parte inicial del decaimiento de la actividad neuronal.

t	W	L	K	C_1	C_2	E_w	E_k	E_r	H
56	0.34155273437500	0.00024414062500	0.27413126310715	1	0	0	0	0	0
57	0.34155273437500	0.00024414062500	0.64478757864288	1	0	0	1	1	0
58	0.36621093750000	0.00097656250000	0.21235283114375	1	0	0	0	0	0
59	0.36621093750000	0.00097656250000	0.27411698686252	1	0	0	0	0	0
60	0.36621093750000	0.00097656250000	0.64470192117514	1	0	0	1	1	0

Tabla 4.21 Valores de la red de nueve neuronas posteriores a la activación de la neurona H .

La simulación que se acaba de describir ilustra la forma en la que el comportamiento del circuito neuronal es equivalente funcionalmente al sistema de Post mostrado en la sección 1.4. A partir del comportamiento observado se puede deducir que para todo teorema del sistema de Post, la red neuronal, partiendo de condiciones iniciales nulas y con los estímulos adecuados, debe mostrar un comportamiento en el que eventualmente la neurona de parada se activa. De manera similar, todo comportamiento neuronal que parta de condiciones iniciales nulas y en el que la neurona de parada se activa debe ser conducido por un conjunto de estímulos cuya interpretación de acuerdo a la codificación propuesta corresponde a un teorema del sistema de Post.

Cuando la primera letra de W es la letra de parada, por construcción de la codificación, la producción localizada se encontrará vacía --ello se interpretará como parada sobre la producción-- que se indica por el hecho de que la tabla está entonces simplemente en 5 en base-6. La ecuación de actualización de la neurona H indica que la situación se detecta en el paso inicial del estado $(0, 1)$.

A la red neuronal puede considerársele como una fórmula matemática de neuronas que cuenta con muchos valores de activación posibles. Las neuronas de alto orden se restringen a las del tipo de Neurona Interruptora. Es esa la razón por la cual tarda más que el sistema de dos etiquetas. Es decir, la estructura de la red neuronal es completamente diferente a la del sistema de dos etiquetas.

La red neuronal requiere de toda la serie de tiempo así como el estar pendiente de la activación de la neurona H , pues los valores obtenidos en ese momento son los que interesan, aunque más tarde la neurona vuelva de nuevo a cero o su actividad decaiga.

Resulta interesante analizar qué se entiende por cada modelo que detiene el proceso, así como sus características particulares.

En el caso de la red neuronal, si la neurona H se enciende se está en presencia de un proceso exitoso, pero si no se enciende no se puede decir nada al respecto. Lo que sí es casi seguro es que una vez que la neurona H (conocida en la red de nueve neuronas como neurona de parada) se enciende, entonces la actividad de la red neuronal decae, siendo importante para el análisis los valores obtenidos en el momento que la neurona H se enciende. Al encenderse H las demás neuronas se desactivan, quedando activada sólo la neurona H . En la sección de gráficas se puede observar esta descripción de decaimiento en el proceso computacional una vez que H se activa.

DISCUSIÓN Y CONCLUSIONES

El presente trabajo pretende aportar a los especialistas de redes neuronales artificiales:

- 1- El establecimiento de los conceptos de computabilidad con base en los sistemas de neuronas artificiales.

Este punto es de suma importancia ya que demuestra que las redes neuronales no deben verse como piezas aisladas de un rompecabezas. Ciertamente, la noción actual de la computación se basa en los modelos de las tesis de Turing y de la de Church. La red de neuronas artificiales es una clase de sistema de cómputo. A la red neuronal debiera considerársele como una fórmula matemática de neuronas con varios valores de activación posibles. Aunque cada neurona es simple y finita, es posible que compute con precisión; por lo tanto, es importante estructurarlas exactamente en los mismos términos utilizados por otros sistemas de cómputo. Para ello es conveniente establecer una relación entre las redes neuronales y los modelos clásicos de computación. Debido a la naturaleza dinámica de las redes, el modelo de la máquina de Turing es el más adecuado para establecer dicha relación. Ya que esto no es fácil de llevar a cabo de manera directa, conviene buscar el vínculo deseado de manera indirecta, por medio de modelos de computación equivalentes a los de Turing. De éstos los más prometedores resultan ser las máquinas de registros y los sistemas canónicos de Post. En el presente trabajo se explica a manera de ejemplo cómo pasar de una quintupla de la máquina de Turing a un Sistema canónico de Post, estableciendo las producciones y axiomas que se derivan. Un sistema de dos etiquetas es un sistema canónico normal de Post que satisface ciertas condiciones, por lo que resulta fácil establecer la relación de especificidad con los Sistemas Canónicos de Post. Por otro lado, el teorema de Cook y Minsky indica que para cualquier máquina de Turing existe un sistema de dos etiquetas que la simula, además la red de nueve neuronas propuesta simula el sistema de dos etiquetas.

- 2- La presentación de la ventaja de contar con modelos de neurona artificial cuya función de agregación no es lineal.

Este aspecto es fundamental. La neurona W es un ejemplo de cómo trabajar con modelos continuos pero no lineales. De igual manera las neuronas booleanas son un ejemplo de modelo no lineal. Estas representaciones permiten contar con alternativas útiles que no se encasillan a un modelo lineal puro. En el presente trabajo se demuestra que una neurona x_i computa una P_i polinomial con mucha precisión.

Además la red de nueve neuronas permite ver la mezcla de varios tipos de neuronas en un modelo; a saber las continuas: W , K y L y las booleanas: E_k , E_r , E_w , C_1 , C_2 y H .

- 3- La ilustración de la capacidad de cómputo de las redes con un número pequeño de procesadores neuronales.

Este punto es crucial. En la actualidad tienden a optimizarse los recursos y los tiempos de los cálculos. Cuán interesante resulta poder representar un modelo mínimo de neuronas que realice las funciones requeridas y que a la vez se relacione con los modelos clásicos. Es decir, los sistemas de dos etiquetas y las máquinas de *Turing*. La primera vez el número de neuronas solicitadas fue de 886, según *Siegelmann*, mientras que en artículos posteriores se redujo a 96. Aunque cada neurona es simple y finita es capaz de computar con precisión.

La neurona W es un ejemplo de cómo el cálculo en base-4 es computado sin problema alguno de capacidad. La neurona K refleja también los cálculos necesarios en base-6 para realizar este proceso. Éste soporta las producciones sin menor inconveniente matemático.

- 4- La ilustración de los problemas numéricos que tienen que ser considerados y resueltos al realizar simulaciones confiables con la clase de redes consideradas.

En la práctica se ha comprobado que no siempre todos los cálculos y representaciones deben de hacerse en decimal; hay ocasiones que por cuestiones de representatividad y claridad se debe usar otra base que permita ver como en los casos específicos de las neuronas W y K , los propios resultados obtenidos. Para comprender los tres pasos de trabajo del sistema de dos etiquetas, fue necesario usar en W base-4, (pudo haber sido otra base, distinta a la decimal). Las operaciones de localización, borrado y agregación fueron representadas muy bien con puros unos y tres como delimitador. Lo mismo sucedió con la neurona K , con la salvedad que en este caso no se debe usar la misma base que en la neurona W , puesto que K engloba a W desde el punto de vista de representación de producciones.

Es muy importante considerar cualquier situación numérica, pues las neuronas no almacenan secuencias sino números. Esa es la razón por la cual la neurona W mantendrá una representación en base-4 de la codificación de la palabra antecedente y K constituirá una representación en base-6 de las producciones codificadas. W almacena la codificación de la palabra antecedente.

5-. El presente trabajo deja claro el poder establecer la simulación de sistemas de dos etiquetas por medio de una red de nueve neuronas, estos sistemas de dos etiquetas pueden tener un número variable de producciones y siempre una condición de parada bien definida.

Referencias Bibliográficas.

[1] S. C. Kleene, "*Representation of events in nerve nets and finite automata*". In Automata Studies (Annals of Mathematical Studies 34). Princeton University Press, Princeton, N. J., USA, 1956.

[2] M. Minsky, "*Computation: Finite and infinite machines*". Prentice - Hall, Inc., Englewood Cliffs, N. J., USA, 1967.

[3] H. T. Siegelmann, "Neural networks and analog computation: Beyond the Turing limit". Birkhäuser, Boston, USA, 1999.

[4] H. T. Siegelmann and M. Margenstern, "*Nine switch - affine neurons suffice for Turing universality*". Neural Networks 12 (1999) 593 - 600.

Bibliografía adicional

[1] L. Bumm, F. Cucker, M. Schub and S. Smale, "*Complexity and real computation*". Springer - Verlag, New York, USA, 1998.

[2] R. G. Taylor, "*Models of computation and formal languages*". Oxford University Press, New York, USA, 1998

[3] Turing, A. 1950. Computing machinery and intelligence. Mind 59:433-60.

[4] Schweizer, P. 1998. The truly total Turing Test. Minds and Machines 8:263-272.

[5] Smart, J. J. C. 1961. Godel's theorem, Church's theorem, and mechanism. Synthese 13:105-10.

[6] Robinson, W. S. 1995. Brain symbols and computationalist explanation. Minds and Machines 5:25-44.