



Modelo general de codificación de RNA en secuencias de ADN y su aprendizaje basado en selección natural

Jorge Eduardo Ortiz Triviño. Profesor Asistente, Departamento de Ingeniería de Sistemas, jorgeo@ing.unal.edu.co, Ernesto Acosta Martín, Estudiante VIII semestre de Ingeniería de Sistemas, eam@email.com Facultad de Ingeniería, Universidad Nacional de Colombia,

RESUMEN

En este artículo se presenta un modelo de codificación de Redes Neuronales Artificiales basado en computación con ADN mediante el cual es posible diseñar una red que sea capaz de responder adecuadamente ante cualquier impulso en su entrada. Se hace uso de las operaciones de manipulación de cadenas genéticas que se emplean a nivel de laboratorio en la actualidad y trabajos previos en computación con ADN que permiten una futura implementación de ese esquema. Se propone también una técnica de aprendizaje nueva basada en los conceptos de selección natural que aprovecha la capacidad de cómputo masivamente paralelo inherente a la estructura biológica del ADN.

Palabras Clave: Computación con ADN, Redes Neuronales, Máquina de Turing, Selección Natural, primer, enzimas.

INTRODUCCIÓN

Con el nacimiento de la computación, tal como la conocemos actualmente, se vio la posibilidad de darle solución a problemas de procesamiento de datos críticos de esa época; análisis de trayectorias balísticas y solución de ecuaciones diferenciales de manera automática eran algu-

nas de las aplicaciones que requerían la intervención de algún dispositivo mecánico o electrónico que permitiera a los científicos emplear su tiempo en planteamientos de más alto orden.

De igual forma, con la evolución de la computación, se ha comenzado a ver la necesidad de dispositivos con un poder computacional superior que permitan resolver problemas de mayor complejidad,¹ los cuales, con un computador convencional (aún con uno de los más poderosos) tomaría mucho tiempo el retornar la respuesta correcta. Así, surge en las últimas décadas nuevos modelos que, en principio, son capaces de indagar de manera paralela y no secuencial sobre un conjunto bastante grande (p.e. 10^{20} para la computación con ADN) de posibles soluciones, retornando en un tiempo relativamente corto frente a la computación convencional, la respuesta correcta.

Uno de estos nuevos modelos es la computación cuántica, que surgió en 1969 con los trabajos de Rolf Launder (y que seguramente exploraremos en alguno de nuestros futuros escritos sobre informática en esta misma revista), el cual aprovecha la multilocalidad de un electrón en un instante de tiempo para hacer cómputos complejos. El segundo modelo es la computación molecular, y más

específicamente la computación con ADN, que nace en 1994, cuando Leonard Adleman en los Estados Unidos, planteó un algoritmo a través del cual codificaba un conjunto grande de posibles soluciones al problema del *Camino Hamiltoniano*² (o problema del viajero) en el cual, mediante una serie de pasos lógicos (4 en realidad), rechazaba las cadenas que codificaban soluciones que no satisfacían el problema indicado hasta obtener finalmente la solución correcta.

En este artículo se expone la manera de emplear la computación con cadenas de ADN para resolver el problema de obtener las mejores configuraciones de Redes Neuronales Artificiales para cualquier problema completamente especificado.

Redes neuronales artificiales (RNA).

Una Red Neuronal Artificial es un modelo de computación paralela (capaz de efectuar varios cálculos simultáneamente, posiblemente diferentes, sobre elementos y variables de procesamiento no necesariamente iguales) que se fundamenta en la estructura funcional del cerebro tomando como unidad de procesamiento a la Neurona Artificial, basada en la neurona natural, y asumiendo algunas de sus propiedades simplificando el modelo.

¹Específicamente Problemas del tipo NP

Muchas veces, a los profesionales de las ciencias de la computación se les presenta el interrogante de si se estará yendo en la dirección correcta en cuanto al avance tecnológico frente al comportamiento de los procesos biológicos que dan al ser humano el patrón de un majestuoso modo de operar, y es por ello que surge este nuevo modelo basado en el comportamiento de la masa cerebral del ser humano.

La base central del pensamiento a nivel cerebral está compuesta por una capa de neuronas con un espesor de 2 a 3 milímetros, en un área de aproximadamente 2.200 cm^2 . Dichas neuronas (alrededor de cien mil millones), se conectan entre sí, presentando uniones de entre 1.000 y 10.000 neuronas si, de manera que se pueden tener un total de 10^{14} a 10^{15} conexiones en total (y es por esto que el modelo de Redes Neuronales lleva el nombre de computación masivamente paralela), y (probablemente) gracias a esta complejidad, el cerebro humano es capaz de aprender y reconocer patrones que nunca había visto, asociándolos con unos previamente aprehendidos.

Es así que, de manera análoga a una neurona natural, surge la neurona artificial la cual consta de una o más entradas (salidas de neuronas adyacentes), y un núcleo que se encarga del procesamiento de tales entradas para producir una única salida, como se muestra en la figura 1. Así, interconectando una gran cantidad de estos elementos finitos, se puede obtener una red cuyos elementos de procesamiento se encuentran trabajando simultáneamente. A un dispositivo de este estilo se le conoce como Red Neuronal Artificial.

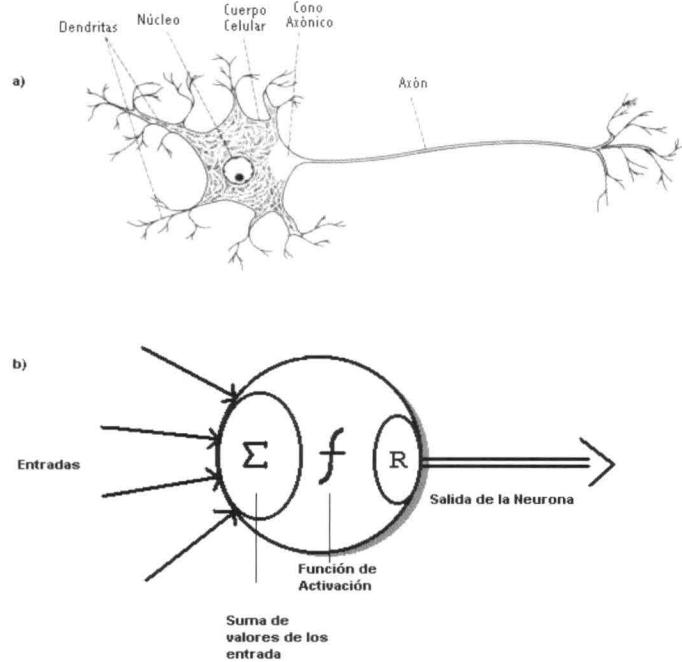


Figura 1 a) Una Neurona Natural, b) Neurona Artificial.

Una red de Neuronas Artificiales intenta simular en su forma y funcionamiento a las redes de Neuronas Naturales aunque aún se encuentra muy lejos de alcanzar tal propósito, se han propuesto muchos tipos de configuraciones de estas con excelentes resultados.

Dentro de las características de diseño de una RNA se pueden mencionar las siguientes:

1. *Topología de la red:* Define el número y distribución de neuronas en la red, y el tipo de conectividad (total o parcial, la conexión de neuronas del mismo nivel, y si se permite o no la realización de bucles).
2. *Conexiones entre pares de neuronas:* Consiste en las ponderaciones asociadas a cada conexión física entre dos neuronas. Es aquí donde reside el conocimiento que adquiere la red a través de su proceso de aprendizaje, en el cual se modifica esta propiedad.
3. *Propiedades de los nodos:* Es la función que hace que la salida de una determinada neurona se active o no. Esta puede ser la función escalón, suma ponderada, gaussiana, sigmoide, arco tangente, etc.
4. Reglas de aprendizaje: es el procedimiento mediante el cual se modifican las ponderaciones de las conexiones entre neuronas de tal forma que al finalizar el proceso de aprendizaje, la red responda de manera correcta ante cualquier impulso. Estas a su vez clasifican a las RNA en dos grupos principales:

- a) *Redes feed-forward:* Son redes que no permiten conexiones en bucles y por lo tanto son redes sin memoria.
- b) *Redes feed-back o recurrentes:* Redes en las que se permiten bucles y por lo tanto presentan uno o más estados de memoria.

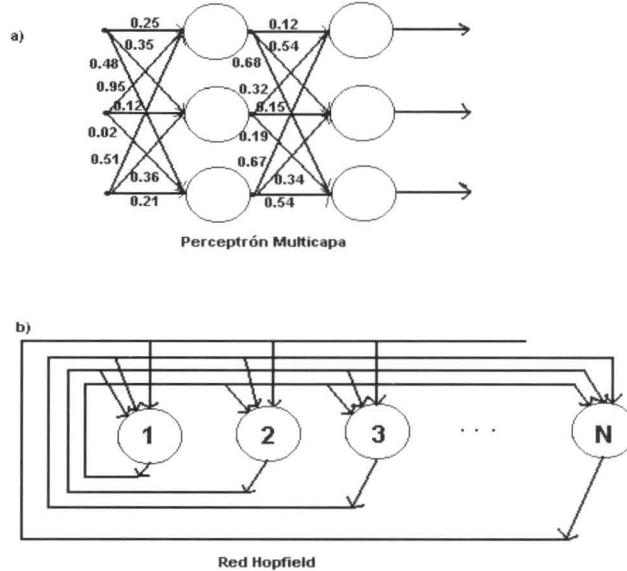


Figura 2 a) Red feed-forward (sin bucles) y b) Red Hopfield (Con bucles hacia nodos distintos).

² El problema del Camino Hamiltoniano (HPP por sus siglas en inglés) consiste en encontrar por lo menos una ruta desde un punto de origen hasta un punto destino pasando por todas las ciudades intermedias una única vez. Ver (1)

Computación Molecular

Observando el comportamiento de la naturaleza, es evidente el perfecto ritmo con el cual millones de células de diversos tipos se organizan y cooperan en un todo armónico para permitir la evolución de un ser biológico (vegetal, animal, etc) y permitir su manifestación. En los organismos vivientes, son muchos los sistemas y subsistemas que controlan funciones vitales en el desarrollo evolutivo. Uno de tales sistemas es el sistema celular, el cual, operando a través de los ácidos nucleicos (ácido desoxirribonucleico, y ácido ribonucleico) residentes en su núcleo, permite el paso por mecanismos hereditarios de las características propias de un grupo, de una generación a la siguiente, determinando las propiedades de cada individuo de manera exclusiva dentro de su género.

El modelo de computación molecular que se ha desarrollado opera sobre moléculas de ADN (ácido desoxirribonucleico) por las características que presenta, las cuales se muestran a continuación. El ADN es un polímero³ compuesto de una unidad fundamental llamada desoxirribonucleótido, la cual es un monómero. Esta molécula presenta dos propiedades fundamentales para las células vivientes: codificación para la producción de proteínas, y autorreplicación. Cada monómero de una molécula de ADN está formado por tres componentes fundamentales: un azúcar, un grupo fosfato, y una base nitrogenada.

El azúcar tiene cinco átomos de carbono numerados de 1' a 5', donde el grupo fosfato está unido al carbono número 5', y la base nitrogenada, o parte variable, está unida al carbono número 1' (figura 3).

Las bases nitrogenadas pueden ser de dos tipos: purinas: Adenina y Guanina, y pirimidinas: Timina y Citosina (ver figura 4), que por comodidad se mencionan mediante las letras A, G, T, y C

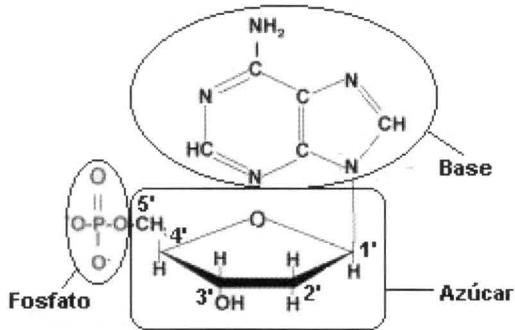


Figura 3 Nucleótido con Base Timina.

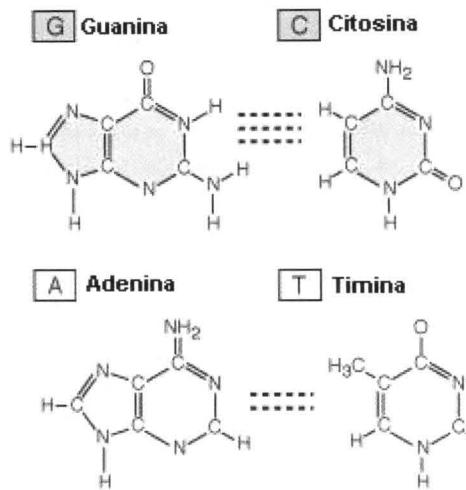


Figura 4 Nucleótidos presentes en una cadena de ADN.

Estos nucleótidos se unen entre sí mediante enlaces fosfodiéster⁴ entre el grupo fosfato de un nucleótido y el grupo hidroxil del otro, formando largas cadenas de aproximadamente 3×10^9 bases nitrogenadas en los seres humanos.

De otro lado, cada nucleótido tiene su complemento dentro del grupo al cual pertenece, así, el complemento de la Adenina es la Timina, el de la Citosina es la Guanina y viceversa. Estas bases complementarias se unen entre sí a través de enlaces de hidrógeno que son más débiles, de tal forma que una larga cadena tiene otra complementaria, y es aquí donde se habla de la complementariedad de Watson-Crick gracias a sus descubridores, quienes en 1953 dedujeron la cadena en forma de doble hélice.

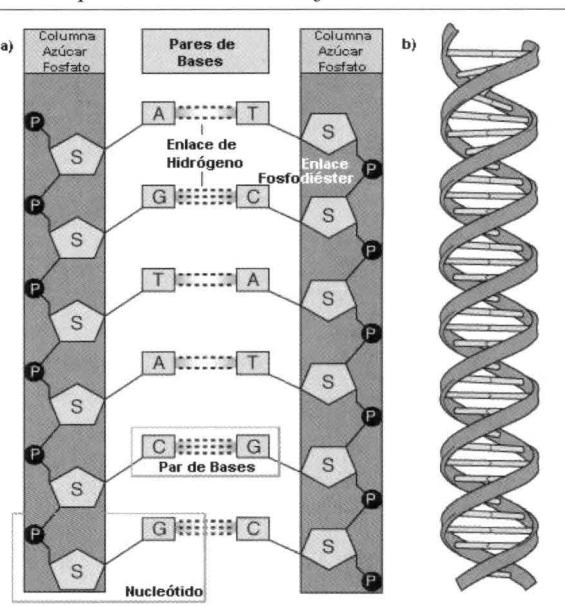


Figura 5 a) Estructura de los enlaces entre Nucleótidos, b) Doble hélice de una cadena de ADN.

³ Compuesto químico que mediante enlaces de moléculas de bajo peso molecular, forma grandes moléculas con un elevado peso molecular

Gracias a estas propiedades, se pueden codificar mediante alguna convención específica, cualquier elemento finito, por ejemplo con la secuencia ATCATG podría representar al mes de enero, la secuencia ATCAAT al mes de Febrero, y así sucesivamente.

Otra propiedad interesante del trabajo con moléculas de ADN es la experiencia que ya se tiene de su manipulación en laboratorio, a través de la cual se pueden realizar operaciones específicas sobre estas, empleando ciertas enzimas para tales fines, las cuales se describen a continuación:

1. *Separación y fusión*: procedimiento a través del cual se pueden separar dobles cadenas de ADN, rompiendo los enlaces de hidrógeno sometiendo la molécula a temperaturas entre 85 y 95°C, obteniendo así dos cadenas sin complementar, y permitieron también unir cadenas simples tras someterlas a temperaturas muy bajas.
2. *Extensión*: empleando la enzima polimerasa, se pueden extender cadenas de ADN, dados (como parte de la cadena que se requiere extender) una secuencia inicial llamada *primer*, para el reconocimiento de la cadena a complementar, y una plantilla (subcadena sin complementar) que describa la secuencia de nucleótidos a añadir.
3. *Reducción*: para esto, se requiere la acción de enzimas Exonucleasas (p.e., la exonucleasa Bal31), que actúa removiendo uno a uno los nucleótidos de cada extremo 3', 5', o de los dos simultáneamente, de la cadena hasta llegar a alguna cadena específica que se deseé obtener.
4. *Corte*: mediante esta técnica se puede separar en dos partes una cadena, destruyendo los enlaces fosfodiéster por la acción de una endonucleasa (p.e. la endonucleasa S1) que actúa en una posición específica de una sencilla o doble cadena.

⁴ Enlace covalente fuerte.

5. *Encadenamiento*: mediante la enzima ligasa, se pueden concatenar dos cadenas reconstruyendo los enlaces de hidrógeno entre los nucleótidos de los extremos de ambas cadenas.
6. *Modificación de Nucleótidos*: las enzimas de modificación (p.e., Metilasa), modifican moléculas de ADN añadiendo o removiendo ciertos componentes químicos. Gracias a esta operación, se pueden remover los grupos fosfato, reemplazándolos por grupos hidroxil, en las terminaciones 5' de una doble cadena para impedir su posible extensión, o remover los grupos hidroxil de una cadena sometida al proceso anterior, reemplazándolos por grupos fosfato, y así, permitir su extensión.
7. *Replicación*: a través del proceso llamado Reacción en Cadena de la Polimerasa (PCR, por sus siglas en inglés), se puede obtener un gran número de copias de una misma cadena, de la cual se dan las secuencias de nucleótidos que conforman sus extremos (primers).
8. *Lectura de una secuencia*: al depositar las moléculas de ADN contenidas en un tubo de ensayo en una solución de poliacrilamid gel en la cual (depositada en una superficie plana y rectangular), en cada extremo se ubican cargas electromagnéticas haciendo que las cadenas se distribuyan a lo largo de toda la superficie por afinidad electrónica, permitiendo leer la cadena subyacente.

Codificación de Redes Neuronales Artificiales (RNA) en cadenas de ADN.

Sin pérdida de generalidad en el modelo se trabajará con *perceptores multicapa* con conectividad total esto restringe un poco el conjunto de las clases de Redes Neuronales pero permite mostrar las bondades del método. Además, la capacidad computacional del modelo tampoco se ve afectada al imponer esta restricción que se hace más por simplicidad que por falta de cualidades inherentes al modelo.

En la computación con secuencias de

ADN se ha de hacer un óptimo uso de las tres características principales que hacen a este modelo superior frente a la computación clásica. Primero, la *capacidad de codificación* en base 4, 16, 64, etc (o diferente de acuerdo a la convención), que aumenta la cantidad de almacenamiento por unidad de volumen, ya que por cada pareja de bases nitrogenadas complementadas se está almacenando una unidad de información en solo 66 átomos, frente a un volumen de varios miles de átomos que componen una compuerta lógica dentro de una unidad de memoria en un circuito integrado. Segundo, el *paralelismo masivo*, dado por la gran cantidad de moléculas diferentes de ADN que pueden estar contenidas en un mismo tubo de ensayo, cada una codificando una solución distinta para el mismo problema, y que son evaluadas de forma simultánea, reduciendo el tiempo de cómputo de un problema que requiera evaluar muchas posibles soluciones a un mismo problema en corto tiempo. Y, tercero, la *complementariedad Watson-Crick*, que permite dada una secuencia de ADN conocer su secuencia complementaria (única), trayendo el lenguaje mixto doble al ámbito computacional⁵.

Con estos principios, se emplea un sistema de codificación en base 4 (y no 6 debido a que se empleará dos símbolos para permitir codificar números Reales) para representar el sistema decimal de numeración. Así, se definen las siguientes equivalencias:

AA	Signo negativo (-)	TA	⇒	6
AC	Separador decimal (, ó .)	TC	⇒	7
AT	0	TT	⇒	8
AG	1	TG	⇒	9
CA	2	GA	⇒	10
CC	3	GC	⇒	11
CT	4	GT	⇒	12
CG	5	GG	⇒	13

De esta forma se puede representar cualquier número Real de la siguiente manera:

$$495.236_{10} = 275.3438_{14} = 2 \times 14^2 + 7 \times 14^1 + 5 \times 14^0 + 3 \times 14^{-1} + 4 \times 14^{-2} + 3 \times 14^{-3} + 8 \times 14^{-4}$$

La cual codificada en una cadena de ADN de acuerdo al método expuesto

corresponde a:

$$495.236_{10} = 275.3438_{14} = \\ \dots\text{ATATCACATCCGACCCCTCCTT} \rightarrow$$

Y para el mismo número pero con signo negativo se obtiene:

$$-495.236_{10} = -275.3438_{14} = \\ \dots\text{ATATAAACATCCGACCCCTCCTT} \rightarrow$$

Codificación en la cual se pueden distinguir seis áreas principales (Figura 6)

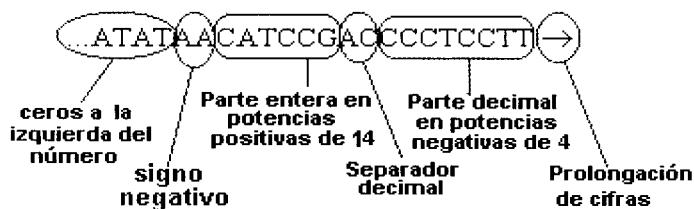


Figura 6 Esquema general de codificación de un número Real.

Para una precisión decimal dada ($1*10^4$, o cuatro cifras significativas, en este caso). La flecha hacia la derecha indica la continuación posiblemente indefinida de cifras en caso de no tener definida tal precisión. Mediante este sistema, es posible codificar en una sola cadena de ADN el conjunto de ponderaciones asociados a las conexiones entre neuronas, tras linealizar y simplificar la matriz de pesos asociada como se ilustra en la figura 7.

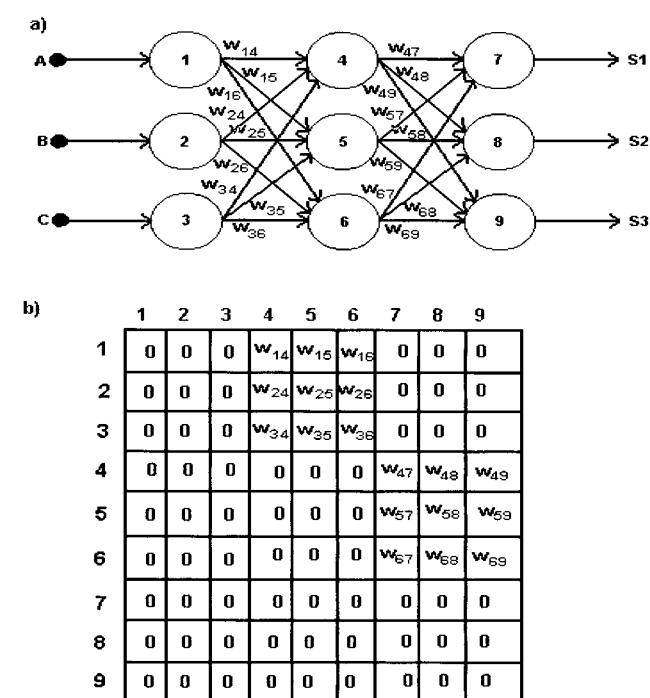


Figura 7 a) Red Neuronal multicapa, feed.forward con una capa de entrada, una capa oculta y una capa de salida y con tres neuronas por capa. b) Matriz de pesos asociada a la red.

Esta matriz de ponderaciones, con las precisiones de diseño anteriormente establecidas, distingue completamente a una RNA, y en este caso es evidente que la mayoría de las posiciones se encuentra ocupada por una ponderación de cero (0) ya que no se permitió la conexión con neuronas anteriores o con la misma neurona (bucles), ni con neuronas semejantes (pertenecientes a la misma capa). Además, para esta Red específica, no se permite un número mayor a una entrada por cada neurona en la capa de entrada, lo cual simplifica para efectos del planteamiento, el modelo propuesto.

A partir de esta matriz, se construye un arreglo unidimensional en el cual se incluyen solo las ponderaciones de las conexiones permitidas, linealizando la matriz, eliminando los elementos nulos, e invirtiendo (para efectos del procesamiento, como se verá más adelante) los elementos del arreglo unidimensional (Figura 8).

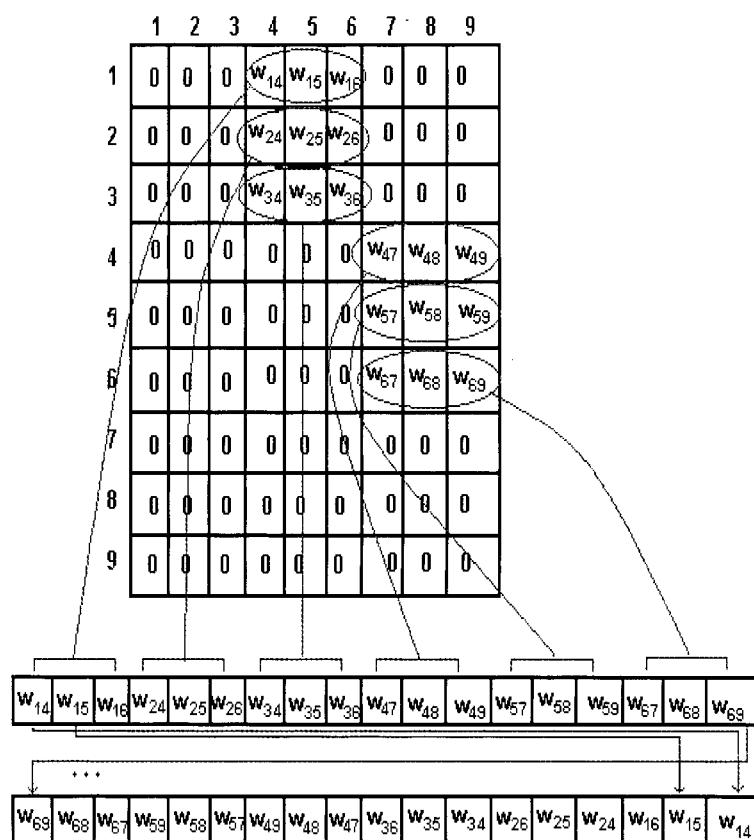


Figura 8. Linealización de la Matriz de ponderaciones y su inversión

De tal forma, que el arreglo lineal se puede llevar de manera directa sobre una cadena de ADN, y así, poder leer la RNA que se encuentra representada en ella, y formada posiblemente al azar, tras definir únicamente la longitud de las subcadenas y cortando la cadena al final de la subcadena número ($N^2*(C-1)$), teniendo N como el número de neuronas

⁵ Ver [2] pag4.

por capa y C como el número de capas incluyendo las capas de entrada y salida, identificando las primeras N subcadenas como las ponderaciones de las conexiones entre la primera neurona de la capa de entrada y cada una de las N neuronas de la capa siguiente, el segundo grupo de N subcadenas como las ponderaciones de las conexiones entre la segunda neurona y las N neuronas de la siguiente capa, el N+1 é-simo grupo de N subcadenas como las ponderaciones de las N conexiones entre la primera neurona de la primera capa oculta y las N neuronas de la segunda capa oculta, el N+2 é-simo grupo de N subcadenas como las ponderaciones asociadas a las N conexiones entre la segunda neurona de la primera capa oculta y las N neuronas de la segunda capa oculta, y así sucesivamente hasta el N*(C-1) éximo grupo de N subcadenas, que definiría los valores de las ponderaciones de las conexiones entre la última neurona de la última capa oculta y las N neuronas de la capa de salida, pudiéndose encontrar valores de 0 (ATATATATAT) indicando que el peso de determinada conexión tiene un valor de cero, o que no existe tal conexión.

Únicamente resta definir la función de activación para cada neurona en particular o para toda la red en general, que en este caso será la función signo, y la regla de actualización o aprendizaje, la cual es realmente una regla de selección como se define a continuación.

Selección de la(s) mejor(es) cadena(s)

El planteamiento de un problema para el planteamiento propuesto sería como sigue: Se desea encontrar la RNA que responda de la mejor manera ante un conjunto dado de pruebas frente a un número determinado de entradas y para determinado número de salidas.

En este planteamiento, si el número de entradas es menor al número de salidas, a las S-E (S: salidas, E: Entradas de la red) entradas se les asigna un valor de cero, y si el número de salidas es menor al número de entradas, solo interesaría leer el valor returnedo por las primeras (o un conjunto arbitrario) E-S salidas, ya que la red esta conformada por el mismo número de neuronas tanto en la capa de entrada, como en las capas ocultas y en la capa de salida.

Cabe aclarar que el número de neuronas por capa en la RNA ha de ser lo suficientemente grande para obtener el número de respuestas deseado, pero aprovechando al máximo la capacidad de obtener salidas analógicas (números Reales) para minimizar el tamaño del problema.

El procedimiento de selección, que como se dijo anteriormente corresponde a la fase de entrenamiento, presenta el siguiente esquema:

1. Generar un número suficientemente grande de cadenas de ADN. Cada subcadena debe ser lo más aleatoria posible.
2. Seleccionar las cadenas con una longitud mayor o igual a $k^*(N^2*(C-1)) + \alpha$, en donde k define la longitud de cada subcadena, determinada por los límites numéricos y la precisión deseada en los valores de las ponderaciones; N determina el número de neuronas por capa, C el número total de capas y α , la longitud en mers (1 mer = 1 nucleótido) del primer de iniciación de cada cadena. Esto es, de las cadenas generadas, seleccionar solo aquellas que por su longitud codifican de manera completa todas las conexiones requeridas.
3. Cortar las cadenas restantes hasta la posición $k^*(N^2*(C-1))$.
4. Por cada una de las N entradas, al-

macenar en un tubo diferente un conjunto de copias de la cadena que codifica el valor de cada entrada para la Red Neuronal (cero para las entradas que no se vayan a emplear), etiquetando cada uno de estos de la forma E_i , en donde i identifica el número de la entrada codificada.

5. Realizar un número P de veces ($P =$ número de pruebas de evaluación):
 - a) Efectuar N veces:

Del valor contenido en el tubo E_i evaluar la función de activación (función signo en este caso) mediante la verificación de la presencia o ausencia de la secuencia AA en la subcadena. En caso de no existir tal secuencia (número positivo), almacenar en un tubo etiquetado S_i una cadena que codifique el valor de uno (1), y en caso contrario, almacenar un cero (0).

- b) C-1 veces:
 - i) N veces:
 - N veces:

Cortar desde la posición long(cad)-k un número de k nucleótidos, y sobre la cadena de longitud k efectuar el producto : $S_i * W_{ij}$, en el cual i denota el número de la neurona cuya salida está siendo evaluada, y j la neurona de la siguiente capa con la cual está conectada. Almacenando el resultado en un tubo etiquetado $S_i W_{ij}$.

- ii) N veces:

Sumar el valor representado en los tubos etiquetados: $S_i W_{ij}$ en los cuales $j=m$, donde m es el número de la neurona actual, y almacenar el resultado en un nuevo tubo etiquetado S_j .

Sobre el valor codificado en las cadenas contenidas en el tubo S_j evaluar la función de activación y almacenar en el mismo tubo una cadena que codifique el valor returnedo por la función (1 o 0).

- c) Rechazar del tubo original aquellas cadenas en las cuales las salidas S_j ,

⁶La operación producto, así como la operación producto empleada más adelante no se explica aquí dado que se mostró en varios trabajos su definición para computación con ADN. Ver (4, 5, 6)

donde j va desde $N^*(C-1)$ hasta N^*C , no correspondan de manera exacta a las salidas proporcionadas en la p -ésima prueba de ejemplo ejecutada.

6. Verificar si existen cadenas en el tubo, y decodificar las RNA codificadas en cada una de ellas.

EJEMPLO (Implementación de la función lógica AND):

Encontrar mediante el modelo propuesto, los valores de las conexiones entre neuronas de una RNA con tres capas (una de entrada, una oculta y una de salida), y con tres neuronas por capa, de tal forma que la o las RNA resultantes muestren en su primera salida, el valor de salida de la función lógica AND.

Para resolver este problema, se ejecuta el algoritmo planteado de la siguiente manera:

1. Sean las cadenas:

•AGTAACAT- CGTTACTG- ATATACAT- ATATACTA- CAGAACGG-
 ATATACAG- AAATACTA- TCGAACAT- AAATACAG- CTTGACTG-
 ATATACAT- ATATACAT- TGGCACCC- AAGTACGA- ATATACCA-
 TCCTACGA- CAATACAT- AACGACTA
 •CACGACAT- ATCCACAG- ATCGACAT- AATTACCC- AGAGACCG-
 AGGAACCC- CAAGACAT- AAGCACAT- ATATACTT- ATATACGC-
 ATATACAT- ATATACAT- AAGAACAT- AATCACCC- AGCAACAG-
 TTGACTG- AGCGACAT- ATGAACGC
 •CACGACAT- ATCCACAG- ATATACAT- AATTACCC- AGAGACCG-
 AGGAACCC- CAAGACAT- AAGCACAT- ATATACTT- ATATACGC-
 ATATACAT- ATATACAT- AAGAACAT- AATCACCC- AGCAACAG-
 TTGACTG- AGCGACAT- AAGAACGC
 •AGTAACAT- CGTTACTG- ATAGACAT- ATATACTA- CAGAACGG-
 ATATACAT- AAATACTA- TCGAACAT- ATATACAT- CTTGACTG-
 ATAGACAT- ATAGACAT- TGGCACCC- AAGTACGA- ATATACCA-
 TCCTACGA- CAATACAT- AACGACTA

generadas aleatoriamente.

2. Son aceptadas todas las cadenas.

3. No se recorta ninguna cadena.

4. Teniendo los siguientes cuatro ejemplos dados en formato: (Entrada_1, Entrada_2, Salida_1), correspondientes a todas las combinaciones posibles de las entradas de una función AND: (0, 0, 0), (0, 1, 0), (1, 0, 0), (1, 1, 1), se rechaza la segunda cadena en el segundo ejemplo por no satisfacer la condición de igualdad, ya que la salida que presenta es 1, siempre que alguna de las entradas sea mayor que 1; y en el cuarto ejemplo se rechaza la cuarta cadena, debido a que su salida es 0 para todos los casos⁷. Debe notarse que la función de activación signo se asumió un umbral igual a cero (en la práctica este supuesto no es general, en la mayoría de los casos los umbrales son diferentes para cada neurona).

5. Las RNA aceptadas (1 y 3) contienen la secuencia:

- (16)-(58.9)-(0)-(0.6)-(210.13)-(0.1)-(-0.6)-(710)-(-0.1)- (49.9)-(0)-(0)-(911.3)-(-12.10)-(0.2)-(74.10)-(20)-(-5.6) y
 - (25)-(3.1)-(0)-(-8.3)-(11.5)-(110.3)-(21)-(-12)-(0.8)-(0.11)-(0)-(0)-(-10)-(-7.3)-(-12.1)-(81.9)-(15)-(-10.11)
- que corresponden a las disposiciones que se muestra en la

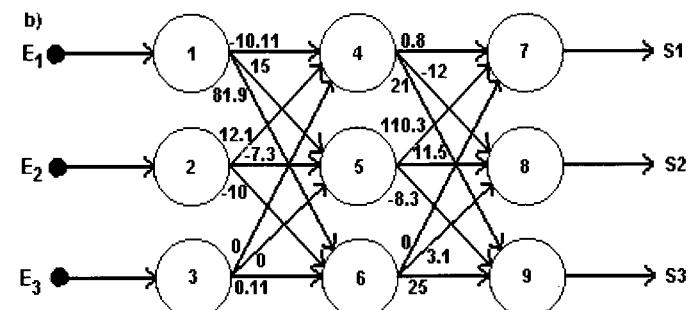
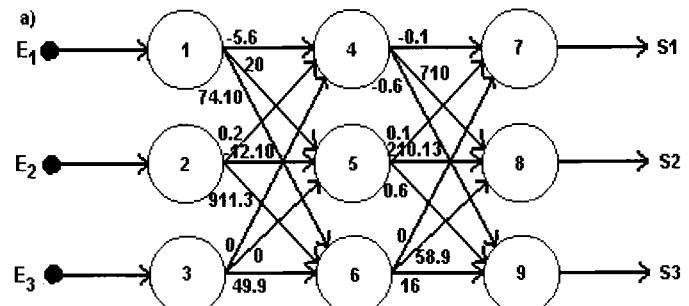


Figura 9 Redes solución del ejemplo.

De otro lado, un caso interesante se obtiene cuando se desea encontrar una red neuronal para implementar la función lógica XOR. ¿Qué cambios se le deberán realizar al modelo para lograr esa meta?. Por tratarse de un caso similar al anterior (no igual) se deja como ejercicio al lector (sugerencia: los valores de los umbrales deben ser diferentes).

CONCLUSIONES

1. Se probó que la computación con moléculas de ADN facilita realmente la obtención de una solución correcta contenida en un conjunto considerablemente grande propuesto al iniciar el proceso de solución de un determinado problema, como lo sugiere el paralelismo masivo natural del modelo.

2. Se mostró una aplicación en la cual se aprovecha al máximo la propiedad de paralelismo masivo ofrecida por el modelo de computación con ADN, cualidad de la cual carecía el modelo de RNA planteado anteriormente, ver (12), para obtener una respuesta (desde luego desconocida a-priori por el usuario) a un problema determinado, sin tener que inter-

venir con el problema más que en el suministro del conjunto de pruebas a realizar como mecanismo de selección.

3.Dada las dificultades actuales (económicas y tecnológicas) para implementar planteamientos de este tipo en laboratorio que permitan realizar todos los pasos propuestos de manera automática con un buen nivel de confianza, no se puede llevar inmediatamente a la práctica el modelo y su demostración es tan solo teórica. Sin embargo se ha mostrado una vez más que para realizar proyectos como este, es necesario

un trabajo interdisciplinario: de profesionales con sólidos conocimientos en biología, genética, química, ciencias de la computación, entre otros.

4.El ejemplo que se implementó para mostrar la flexibilidad del planteamiento es lo suficientemente pequeño para entender el modelo, pero también lo suficiente completo para mostrar la flexibilidad que presenta frente a problemas de mayor magnitud.

5.Se hizo uso de los modelos naturales de

selección, ya que las únicas cadenas solución que permanecen en el tubo inicial son aquellas que por "selección natural" sobrevivieron a todas las pruebas que les fueron impuestas.

6.Este modelo, planteado para redes feed-forward, puede ser extendido a modelos de redes feed-back con conectividad parcial y con funciones de activación específicas para cada neurona, lo cual es tema de futuras profundizaciones de los autores o de los lectores del presente trabajo.

7. El seguimiento riguroso de este paso no se muestra aquí, pero el lector puede realizarlo fácilmente.

REFERENCIAS BIBLIOGRAFICAS

1. **L. Adleman**, Molecular computation of solutions to combinatorial problems, Science, 226 (1994), 1021 - 1024
2. **Paün G., Rozenberg G., Salomaa A.**, DNA Computing, Springer (1998)
3. **Paün G.**, Computing with Bio-Molecules, Springer (1998)
4. **Frank Guarneri, Makiko Fliss**, Carter Bancroft., Making DNA add.
5. **Frisco PierLuigi**, Parallel Arithmetic with Splicing.
6. **Vineet Gupta, Srinivasan Parthasarathy**, **Mohammed J. Zaki**, Arithmetic and Logic Operations with DNA
7. **L. Adleman**, On Constructing A Molecular Computer, (1995)
8. **Ogihara M.**, Biomolecular Computing - A Shape of Computation to Come.
9. **Echeverrieta F.**, Redes Neuronales, algoritmos genéticos y aprendizaje, En Vanguardia (2000)
10. **Rojas M.**, Prototipo de un simulador gráfico de una máquina de cómputo con ADN en un computador digital, Tesis de grado de Ingeniería de Sistemas Universidad Nacional (1998)
11. **Ortiz J.**, Computadores Moleculares: una tecnología prometedora, Revista de la Facultad de Medicina Vol 47 No. 2 (1999) 98-101
12. **Ortiz J., Micán R.**, Un pequeño cerebro artificial basado en ácidos nucleicos, Revista de la Facultad de Medicina Vol 48 No. 2 (2000) 99-103
13. **Mortimer Ch.**, Química, Iberoamérica Ed., (1983) 729-733