



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Predicción de Series Temporales Utilizando
Redes Neuronales
Artificiales

TESIS PROFESIONAL

Que para obtener el título de:

INGENIERO ELÉCTRICO ELECTRÓNICO

P r e s e n t a

FERMIN FLORES BUENDIA

Director de Tesis:

Dr. José Ismael Espinosa Espinosa



México D.F., 2003

ÍNDICE

SINOPSIS

INTRODUCCIÓN

CAPITULO I

" REDES NEURONALES BIOLÓGICAS "

I.1 Objetivo del capítulo	4
I.2 Definición de una neurona y de la red neuronal biológica	4
I.3 Fisiología de la neurona biológica	5

CAPITULO II

" REDES NEURONALES ARTIFICIALES "

II.1 Objetivo de capítulo	8
II.2 Definición de una neurona artificial	8
II.3 Similitud entre una red neuronal artificial y una red neuronal biológica	9
II.4 Elementos que constituyen a una red neuronal artificial	12
II.4.1 Función de entrada	12
II.4.2 Función de activación	14
II.4.3 Función de salida	17
II.5 Tipo de aprendizaje	18
II.5.1 True Time	18
II.6 Aplicaciones, Software y Hardware de las redes neuronales artificiales	27
II.6.1 Áreas de aplicación de las redes neuronales artificiales	27
II.6.2 Software y Hardware de las redes neuronales artificiales	30

CAPITULO III

"PREDICCIÓN: SEÑAL SENOIDAL, NÚMEROS ALEATORIOS"

" PREDICCIÓN DE PUNTOS QUE CONFORMAN UNA SEÑAL SENOIDAL "

III.1 Objetivo del capítulo	34
III.2 Introducción	34
III.3 Desarrollo	35
III.3.1 Preparación de la base de datos	35
III.3.2 Generación del archivo de entrenamiento y verificación (IO)	39
III.3.3 Elaboración de la red neuronal	40
III.3.4 Entrenamiento de la red neuronal	43
III.3.5 Obtención de resultados	45
III.4 Red Neuronal de Ocho Neuronas	48
III.4.1 Preparación de la base de datos	48
III.4.2 Generación del archivo de entrenamiento y verificación (IO)	49
III.4.3 Elaboración de la red neuronal	50
III.4.4 Entrenamiento de la red neuronal	52
III.4.5 Obtención de resultados	54
III.4.6 Utilizando la red neuronal	58
III.4.6.1 Ejercicio I	58
III.4.6.2 Ejercicio II	60

" PREDICCIÓN DE UNA SERIE DE NÚMEROS ALEATORIOS PROVENIENTES DE UNA BASE DE NÚMEROS ALEATORIOS "

III.5 Objetivo del capítulo	64
III. 6 Introducción	64
III.7 Desarrollo	64
III.7.1 Preparación de la base de datos	64
III.7.2 Elaboración de la primera red neuronal	70
III.7.3 Entrenamiento de la red neuronal	71
III.7.4 verificación de la red neuronal	73
III.7.5 Obtención del archivo de resultados	73
III.8.1 Elaboración de la segunda red neuronal	78
III.8.2 Generación del archivo de entradas y salidas (IO)	78
III.8.3 Entrenamiento de la red neuronal	80
III.8.4 Análisis de resultados del archivo de verificación	82
III.8.5 Observaciones	85
III.9.1 Elaboración de la tercera red neuronal	86
III.9.2 Generación del archivo de entradas y salidas (IO)	86
III.9.3 Entrenamiento de la red neuronal	88
III.9.4 Análisis de resultados del archivo de verificación	89
III.9.5 Comentarios generales	91
III.10 Utilizando la primera red neuronal	94
III.10.1 Aplicación I	94
III.10.2 Aplicación II	99

CAPITULO IV

" PREDICCIÓN DE LAS VARIABLES ECONÓMICAS DEL MERCADO BURSÁTIL "

IV .1 Objetivo del capítulo	114
IV. 2 Desarrollo	114
IV.2.1 Verificación de la red neuronal	115
IV.2.2 Graficas de los resultados del archivo de verificación	116
IV.2.3 Comentarios	118
IV.3 Utilizando la red neuronal	118
IV.3.1 Ejercicio I	119
IV.3.2 Ejercicio II	122
IV.3.3 Observaciones	125
IV.4 Resultados finales	125
IV.5 Conclusiones	130

CAPITULO V

Conclusiones	132
--------------	-----

APÉNDICE

" SISTEMA DE DESARROLLO DYNAMIND "

Objetivo	147
Archivos que maneja Dynamind	148
Elaboración de la red neuronal	149

I Elaboración y/o preparación del dato	149
II Elaboración de la red neuronal	153
III Entrenamiento de la red neuronal	154
IV Verificación de la red neuronal	155
V Utilización de la red neuronal	157

BIBLIOGRAFÍA	158
---------------------	-----

SINOPSIS

El objetivo principal de este trabajo de investigación fue emplear **redes neuronales artificiales** para **la predicción de series temporales** o para la **pronosticación de una serie de eventos** que se han presentado dentro de un intervalo de tiempo, por ejemplo el número de nacimientos de seres humanos en un mes o el cambio de temperatura en una habitación en un instante.

Para este trabajo de investigación se empleo el sistema de desarrollo denominado **Dynamind** para predecir los siguientes eventos:

- Los puntos que conforman una señal senoidal
- El siguiente número aleatorio de una base de datos de números aleatorios
- Cinco variables económicas del mercado bursátil

En cada una de estas aplicaciones se generaron diferentes tipos de redes neuronales artificiales debido al tipo de eventos en estudio, pues son fenómenos de naturaleza y comportamiento muy variado.

¿ El porque utilizar redes neuronales artificiales?; por dos razones:

- Por ser una tecnología interesante
- Por ser un camino más en la solución de pronosticación de eventos.

**“ La memoria humana va más
allá de lo que se piensa”
Anónimo**

INTRODUCCIÓN

El presente trabajo de investigación esta conformado por cinco capítulos y un apéndice.

En el primer capítulo se habla sobre la arquitectura y funciones de cada componente que constituyen a una neurona y por ende a la red neuronal biológica.

El segundo capítulo trata sobre la definición de una neurona y de la red neuronal artificial las similitudes entre la neurona biológica y la artificial así como las características que componen a esta ultima; sus aplicaciones, el software y hardware existentes sobre este tipo de tecnología en el mercado; de igual forma, las áreas y campos de aplicación donde mas se utilizan las redes neuronales artificiales como son en Medicina, Medio Ambiente, Biología , Finanzas; siendo esta el área donde más se aplican las redes neuronales artificiales.

En el tercer capítulo; “Predicción de una señal senoidal y números aleatorios”. La importancia de este capítulo radicó en conocer adecuadamente el funcionamiento del sistema de desarrollo denominado Dynaimd en el área de predicción ; no hay que olvidar que el objetivo del trabajo en general es “Explorar la posibilidad de las redes neuronales artificiales para predecir eventos futuros empleando el sistema de desarrollo Dynamind”; efectuando así la primera aplicación de conocer los siguientes puntos que conforman a una señal senoidal, para lo que se generaron dos redes neuronales artificiales , la primera constituida por dos neuronas y la segunda por ocho neuronas.

En la segunda aplicación realizada dentro de este capítulo, “Predicción de una serie de números aleatorios provenientes de una base de números aleatorios”;

se generaron tres redes neuronales artificiales (**RAND1.NET**, **RAND2.NET** Y **RAND3.NET**) cuyas características varían una de la otra con la finalidad de encontrar la red más eficiente para la predicción de eventos.

Para definir cual de las tres redes era la más "efectiva", se elaboró una red más para efectuar una aplicación extra donde el fin es comprobar si la red al contar una base de datos mayor (300 números aleatorios), su comportamiento en predicción fuese mejor.

Con la experiencia adquirida con estas aplicaciones, se procedió a trabajar con una aplicación mas real; "predicción de cinco variables económicas del mercado bursátil".

Como se observa en el capítulo IV, el objetivo en su momento es, pronosticar las siguientes variables económicas: Bono, Divisa, Indbol1, Indbol2 y Valor 1. Variables que se adquirieron del mes de octubre para pronosticar los días 29 de octubre, 30 de octubre, 31 de octubre y 1 de noviembre de 1999.

En el capítulo V, se observan las conclusiones generales que arrojaron todo este trabajo de investigación y en especial todo lo referente a las redes neuronales artificiales en el área de predicción.

Para quienes estén interesados en trabajar en el cautivante mundo de las redes neuronales artificiales y en especial en el área de predicción; al final se anexa un apéndice donde se da a conocer las características principales del sistema de desarrollo Dynamind para generar redes neuronales artificiales

“El aprendizaje es experiencia,
todo lo demás es sólo información”
Albert Einstein

CAPITULO I

“REDES NEURONALES BIOLÓGICAS”

I.1 OBJETIVO DEL CAPITULO: Describir los elementos principales que conforman a la red neuronal biológica y por ende a la neurona.

I.2 DEFINICIÓN DE UNA NEURONA Y DE LA RED NEURONAL.

En realidad una definición exacta de la neurona no existe; sin embargo, se ha encontrado en distintas fuentes que la forma de definir a la neurona ó bien la forma genérica de definir una red neuronal; es que puede considerarse como un modelo artificial y simplificado del cerebro humano.¹

Una red neuronal artificial es “un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento esta inspirada en la célula fundamental del sistema nervioso humano: *la neurona* ”.²

Las neuronas son células especializadas adaptadas específicamente para transmitir mensajes eléctricos a través de todo el cuerpo por medio de ramificaciones.

Debido al entrelazado de estas ramificaciones se van constituyendo grupos de neuronas, que al comunicarse procesan grandes cantidades de información; donde se ejecutan varios procesos, por ejemplo el proceso de aprendizaje.

¹ Martín del Rio, A.Sanz Molina, “Redes Neuronales y Sistemas Borrosos”, pp.10

² ídem

Recapitulando las definiciones anteriores se puede concluir que una neurona es: la unidad funcional básica del sistema nervioso para transmitir mensajes eléctricos (información) de una neurona a otra, hasta constituirse de esta forma la red neuronal.

1.3 FISILOGIA DE LA NEURONA BIOLÓGICA.

Podemos considerar como padre de las redes neuronales al científico aragonés Santiago Ramón y Cajal, descubridor de la estructura neuronal del sistema nervioso. En 1888 Ramón y Cajal demostró que el sistema nervioso estaba compuesto por una red de células individuales: las **Neuronas**.³

Partiendo del hecho de que la **neurona** es la unidad funcional del sistema nervioso; la neurona esta constituida fisiológicamente de la siguiente forma:

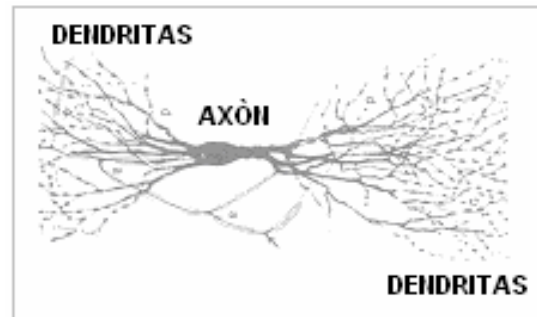
El cuerpo de la neurona o **soma** contiene el núcleo y recibe la información de otras neuronas vecinas a través de conexiones denominadas **dendritas** que son finalmente la entrada a la neurona.

“El número de dendritas varía de doce a miles por célula nerviosa. Cada neurona puede completar desde mil hasta diez mil sinápsis (conexión entre una neurona y otra) y recibir información de otras mil neuronas”⁴

Por otro lado, **El Axón**, se considera como una prolongación muy delgada que lleva la información de salida de la neurona, siendo ésta impulsos o señales a otras células nerviosas; es decir, el axón cuando esta cerca de otras “células destino” la información se convierte en información de entrada para otras células cercanas que se dividen en varias ramificaciones una vez más para conformar otras dendritas con el soma de otra célula. Ver siguiente figura.

³ Ricardo Tapia, “Las Células de la Mente”, Fondo de Cultura Económica, pp.62-69

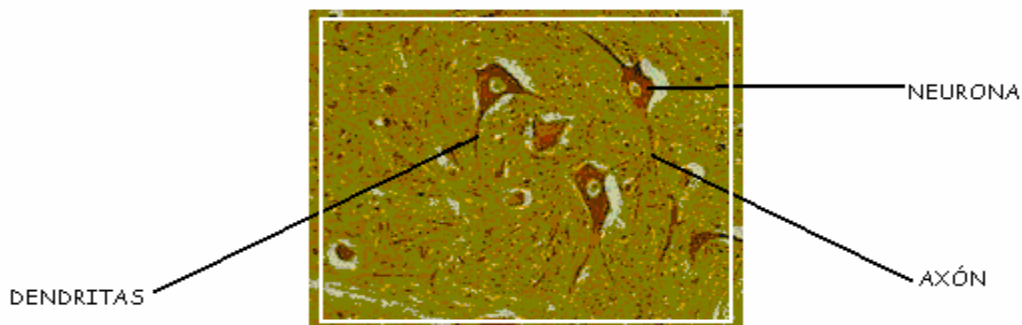
⁴ Charles E Stevens, “The Neuron”, Scientific American, pp.1-2



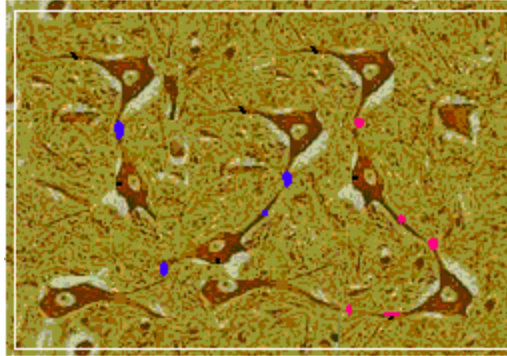
El intercambio de información de una célula a otra está considerada como un proceso químico donde el objetivo es elevar o disminuir el potencial eléctrico dentro del cuerpo de la célula receptora; si su potencial alcanza el umbral se envía un pulso o potencial de acción por el axón, es entonces cuando se dice que la célula se disparó.

Podemos resumir todo lo anterior hasta ahora sobre la forma y estructura de las neuronas de la siguiente manera: todas tienen una porción receptora de señales, formada por las *dendritas*, sumamente ramificadas, un *soma* o cuerpo – en donde se encuentra el núcleo y la maquinaria química para fabricar proteínas y otras moléculas, además es la porción encargada de manejar e integrar la información-y una porción emisora de la información recibida, constituida por el *axón*, el cual también se ramifica considerablemente.

En el siguiente esquema se muestra la fisiología de la neurona.



En la figura de abajo se muestra una red neuronal donde los círculos rojos indican el intercambio de información de una neurona a otra; este cambio de información podría ser el aprendizaje durante un proceso de entrenamiento. Mientras que los círculos azules podrían ser el nuevo aprendizaje (neuronas interconectándose).



“Siempre y cuando nos comprometamos en nuevas conductas,
el cerebro se remodela a sí mismo”
Michael Merzenich

CAPÍTULO II

“REDES NEURONALES ARTIFICIALES”

II. 1 OBJETIVO DEL CAPÍTULO: Describir las características que constituyen a las redes neuronales artificiales, sus aplicaciones y la tecnología existente sobre éstas.

II.2 DEFINICIÓN DE UNA NEURONA ARTIFICIAL.

En 1951 el estudiante Marvin Minsky apoyándose del científico Frederic Skinner dio origen a la primera idea “oficial” sobre inteligencia artificial basándose en la primera red neuronal artificial.

Posteriormente, Minsky y Edmonds, montaron la primera máquina de redes neuronales artificiales, compuesta básicamente de 300 tubos de vacío y un piloto automático de un bombardero B-24, llamando así a su creación “SHARC “. Se trataba nada menos que de una red de 40 neuronas artificiales que imitaban el cerebro de una rata.

Pero entonces, ¿Qué es una red neuronal artificial?

Existen numerosas formas de definir las desde las definiciones cortas y genéricas hasta las que intentan explicar más detalladamente qué son las redes neuronales; por ejemplo:

1. Una nueva forma de computación, inspirada en modelos biológicos.
2. Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles.

3. Un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.
4. Redes neuronales artificiales, son redes interconectadas masivamente en paralelo de elementos simples y organizados jerárquicamente, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.

Y desde el punto de vista de máquinas adaptables una red neuronal artificial puede definirse como un conjunto de procesadores distribuidos en paralelo que guardan conocimientos de la experiencia y los tienen disponibles para usarlos. Las redes neuronales artificiales de cierta forma siguen el mismo procedimiento que el cerebro humano, es decir, el conocimiento es adquirido por la red por medio de un proceso de aprendizaje y las intensidades de conexión entre neuronas conocida como pesos sinápticos son usados para guardar los conocimientos.

II.3 SIMILITUD ENTRE UNA RED NEURONAL ARTIFICIAL Y UNA RED NEURONAL BIOLÓGICA

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar los casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante.

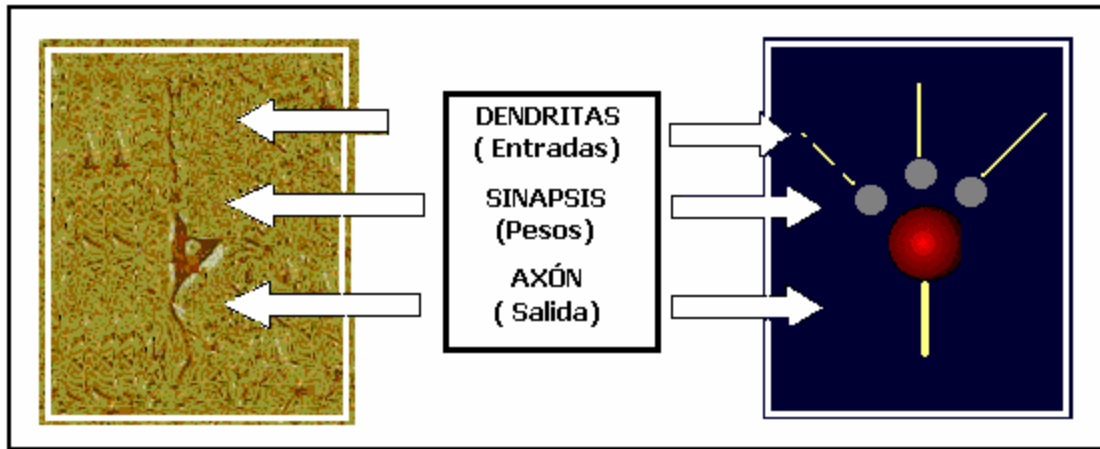
Las redes neuronales artificiales tratan de imitar el hardware del sistema nervioso, con la intención de construir sistemas de procesamiento de la información en paralelo, distribuidos y adaptivos que puedan presentar un cierto comportamiento inteligente.

Recordando y partiendo de las siguientes premisas de que el cuerpo de la neurona biológica denominada **Soma** que contiene el núcleo y que se encarga de todas las actividades metabólicas de la neurona y de recibir toda la información de otras neuronas vecinas a través de conexiones sinápticas, denominadas dendritas y por su parte el axón, que es la salida de la neurona, pues envía impulsos o señales a otras células nerviosas; esa transmisión de una señal de una célula a otra es un proceso químico, en el que se liberan sustancias transmisoras en el lado del emisor de la unión. El efecto es elevar o disminuir el potencial eléctrico dentro del cuerpo de la célula receptora, si ese potencial alcanza un umbral se envía un pulso o potencial de acción por el axón, afirmándose que la célula se disparó, en caso contrario, la célula se encuentra inhibida (reposo).

Bajo este funcionamiento biológico de las redes neuronales se creó el concepto y el funcionamiento de las redes neuronales artificiales, que más que nada es un dispositivo electrónico que responde a señales eléctricas provenientes de otras unidades iguales produciendo una salida que depende de un umbral o de una función de activación.

De igual forma las redes neuronales artificiales están conformadas por varias capas interconectadas entre sí.

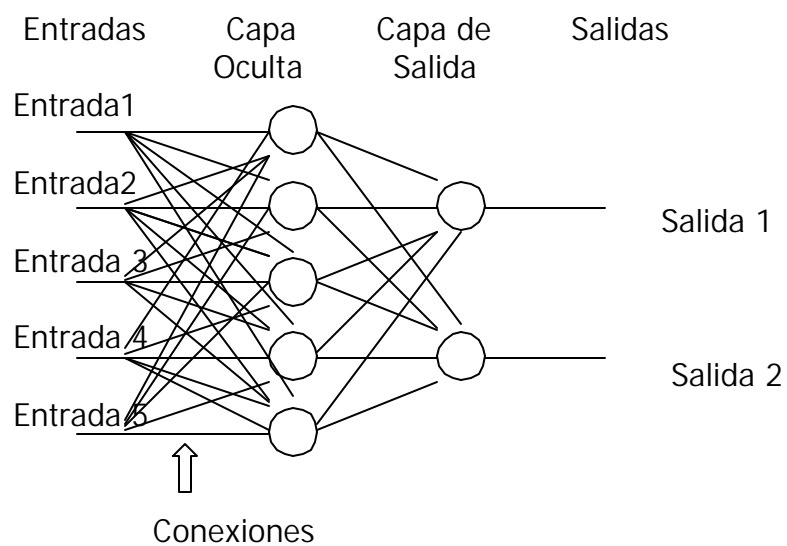
En la siguiente figura se muestra la similitud o comparación de una neurona biológica y una artificial.



Básicamente las características de similitud en funcionamiento y constitución son:

- Conforman diversas capas de neuronas.
- Entradas de información.
- Salidas de información.
- Utilizan pesos.
- Función de activación (disparo)

En la siguiente figura se muestra la “morfología” de una red neuronal artificial.



II.4 ELEMENTOS QUE CONSTITUYEN A UNA RED NEURONAL ARTIFICIAL.

Como se mencionó anteriormente, la constitución esta formada por neuronas interconectadas y arregladas en capas. Los datos ingresan por medio de la "capa de entrada", pasan a través de la "capa oculta" y salen por la "capa de salida".

A continuación y bajo este concepto de funcionamiento, se hablará de cada una de estas etapas que conforman a la red neuronal artificial.

II.4.1 Función de entrada.

La neurona trata a muchos valores de entrada como si fueran uno solo, esto recibe el nombre de entrada global. El problema a discutir es la forma en que esta entrada global (gin_i) se combina para ser simples entradas (in_{i1}, in_{i2}, \dots). Estos se logra a través de la *función de entrada*, la cual se calcula a partir del *vector de entrada*. El vector de entrada se describe de la siguiente forma:

$$Input_i = (in_{i1} \bullet w_{i1}) * (in_{i2} \bullet w_{i2}) * \dots * (in_{in} \bullet w_{in})$$

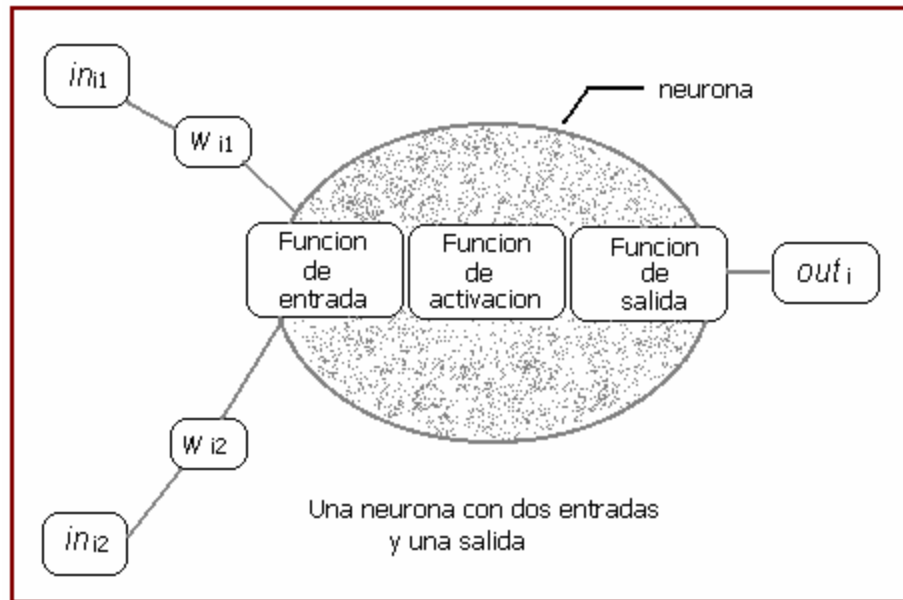
Donde:

El asterisco (*) representa al operador apropiado, como por ejemplo, un máximo, sumatoria, producto, etc.

n: número de entradas a la neurona N_i y w_i al peso.

Los valores de entrada se multiplican por los pesos anteriormente ingresados a la neurona; por ende, los pesos que generalmente no estan restringidos cambian la medida de influencia que tienen los valores de entrada.

En la siguiente figura se explica detalladamente este proceso.



De acuerdo a la figura, in_{i1} es la entrada número 1 a la neurona N_i ; w_{i1} es el peso correspondiente a in_{i1} ; in_{i2} es la entrada número 2 a la neurona N_i ; w_{i2} es el peso correspondiente a in_{i2} y out_i es la salida de la neurona N_i . El conjunto de todas las n entradas $in_i = (in_{i1}, in_{i2}, \dots, in_{in})$ es denominado "vector de entrada". Las salidas de cada neurona son ahora entradas de la neurona N hasta obtenerse un mínimo de salidas o una sola salida final.

Ahora bien, las funciones de entrada más comunes y conocidas son:

1. Sumatoria de las entradas pesadas: es la suma de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos; es decir:

$$\sum_j (in_{ij} w_{ij}), \text{ con } j = 1, 2, \dots, n$$

2. Producto de las entradas pesadas: es el producto de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos, es decir:

$$\prod_j (in_{ij} w_{ij}), \text{ con } j = 1, 2, \dots, n$$

3. Máximo de las entradas pesadas: solamente toma en consideración el valor de entrada mas fuerte, previamente multiplicado por su peso correspondiente, es decir:

$$\text{Max}_j (in_{ij} w_{ij}) \text{ con } j = 1, 2, \dots, n$$

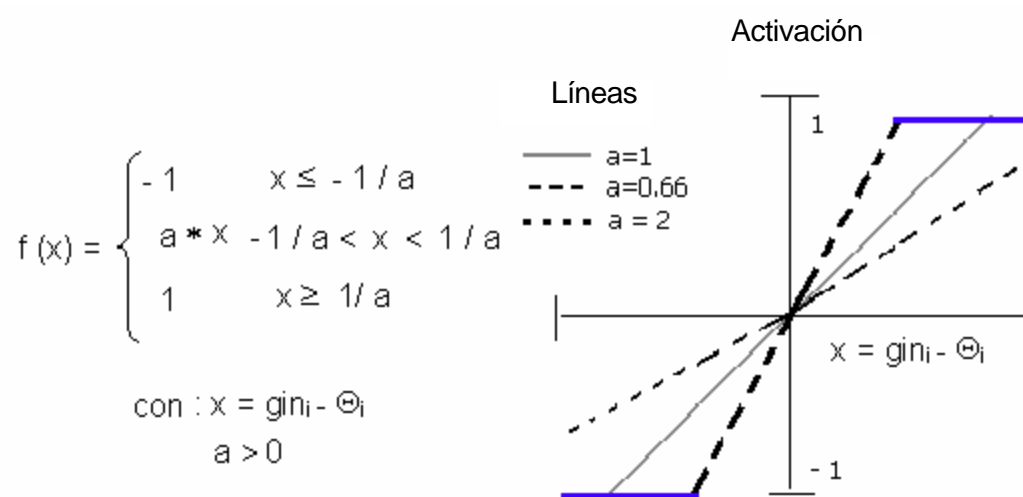
II.4.2 Función de activación:

Una neurona biológica puede estar activada (excitada) o inactiva (no excitada); es decir, que tiene un "estado de activación". Las neuronas artificiales también cuentan con diferentes estados de activación; algunas de ellas solamente cuentan con dos, al igual que las biológicas.

La función de activación calcula el estado de actividad de una neurona; transformando la entrada global (menos el umbral, Θ_i) en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) ó de (-1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 ó -1) ó activa (1).

Las funciones de activación más comunes son:

1. Función lineal:



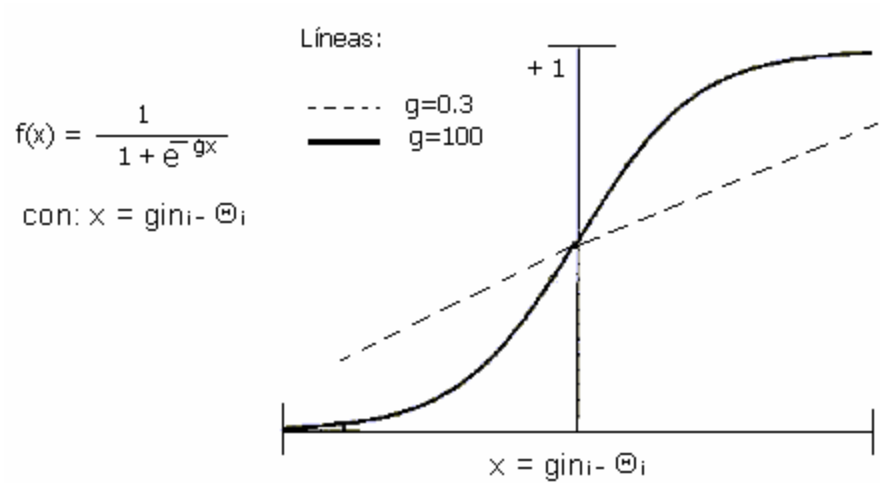
Los valores de salida obtenidos por medio de esta función de activación serán:

$a * (g_i n_i - \Theta_i)$, cuando el argumento ($g_i n_i - \Theta_i$) esté comprendido dentro del rango $(-1/a , 1/a)$.

Por encima o por debajo de esta zona se fija la salida en 1 ó -1, respectivamente.

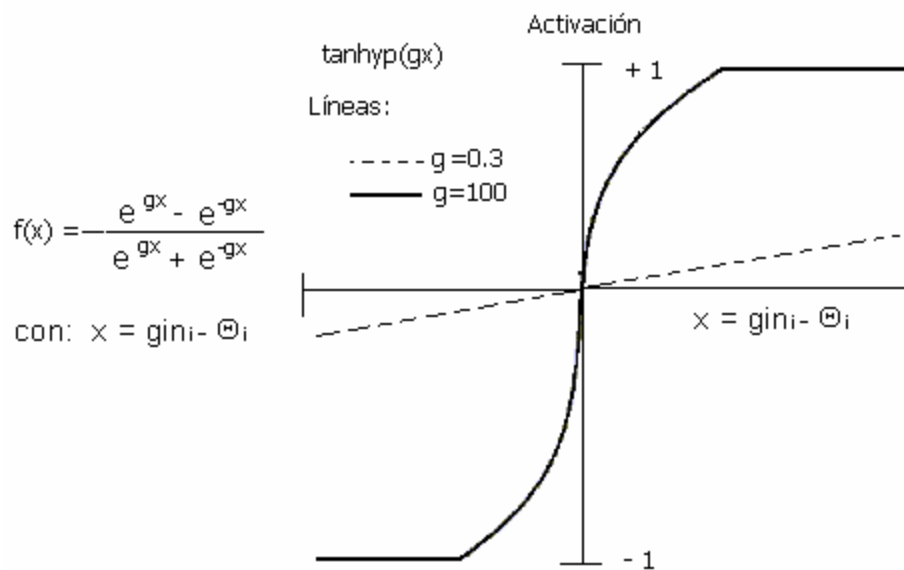
Cuando $a=1$, la salida es igual a la entrada.

2. Función sigmoidea:



Los valores de salida que proporciona esta función están comprendidos dentro de un rango que va de 0 a 1. Al modificar el valor de g se ve afectada la pendiente de la función de activación.

3. Función tangente hiperbólica:

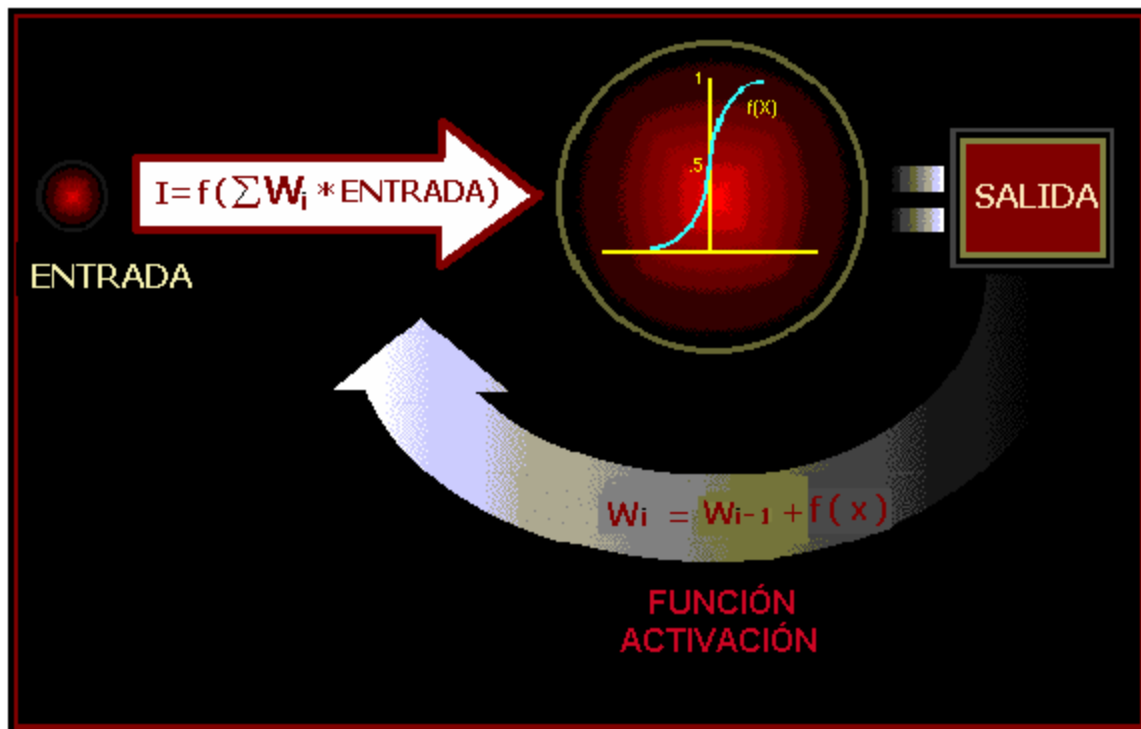


Los valores de salida de la función tangente hiperbólica están comprendidos dentro de un rango que va de -1 a $+1$. Al modificar el valor de g se ve afectada la pendiente de la función de activación.

II.4.3 Función de salida.

El valor resultante proveniente de la neurona i (out_i) es la función de salida; por ende, la función de salida determina qué valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona siguiente. Normalmente, no cualquier valor es permitido como una entrada para una neurona, por tanto, los valores de salida están comprendidos en el rango $[0,1]$ ó $[-1,+1]$.

En la siguiente figura se muestra en resumen el funcionamiento de una neurona con su función de entrada, su función de activación y su función de salida.



II.5 TIPO DE APRENDIZAJE

II.5.1 True Time.

Un parámetro más que determina el funcionamiento final de toda red neuronal artificial es su **algoritmo de aprendizaje**.

Generalmente el algoritmo de aprendizaje va asociado a la topología de la red neuronal.

Existen diversos tipos de algoritmos, por ejemplo:

- Aprendizaje supervisado
- Aprendizaje no supervisado
- Aprendizaje estocástico
- Aprendizaje Hebbiano
- Aprendizaje BackPropagation
- Aprendizaje de Madaline I y III
- Aprendizaje de Hamming
- Aprendizaje de True Time, etc.

Este último es el que se utilizó en cada una de las aplicaciones efectuadas.

A continuación se describe teóricamente y matemáticamente este algoritmo de aprendizaje (TrueTime).

Una de las prioridades principales en su totalidad en áreas de aplicación, es la necesidad de realizar procesos de datos en forma muy rápida. Las redes neuronales se adaptan bien a esto debido a su implementación paralela. La mayoría de las redes puedan operar en un entorno de tiempo real pues existe la necesidad de cambiar por ejemplo los pesos de las conexiones ó de que el entrenamiento sea mínimo.

¿ Pero que es trabajar en Tiempo Real? Estableciendo que cualquier sistema se dice trabajar en tiempo real cuando las operaciones que controlan o mandan al sistema, ya sean correctas o incorrectas, el tiempo en que las ejecuta debe ser inmediatamente presentadas.

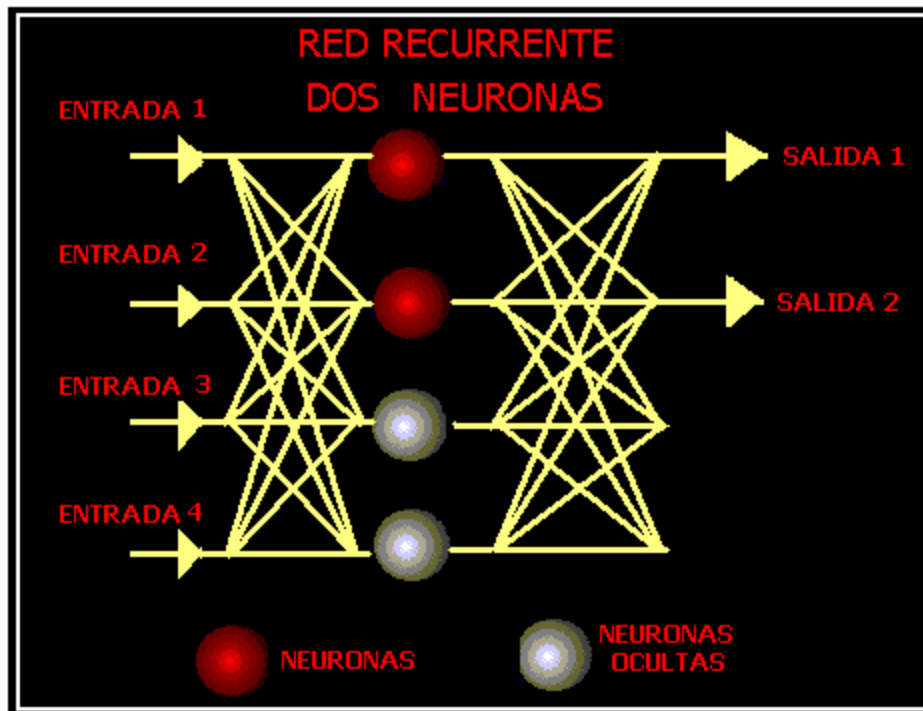
Dynamind trabaja en True Time porque:

1. Maneja tiempos reales, es decir, ejecuta un tiempo para inicializarse la tarea (operación).
2. Presenta un tiempo para efectuar la operación (entrenamiento de la red).
3. El tiempo total para presentar los resultados, que en este caso es evaluar el aprendizaje; es decir, confirmar su capacidad de generalización.

Todo este proceso lo manifiesta Dynamind al emplear el algoritmo de Tiempo Real pues identifica los patrones de aprendizaje dentro de la correlación serial de la cadena de todos los datos (True Time) y son usados al mismo tiempo para pronosticar y/o predecir.

Para efectuar el algoritmo de true time, Dynamind usa conexiones de retroalimentación para el " estado dinámico" de la red.

Por otra parte, el algoritmo de entrenamiento True Time consiste o esta constituido por una simple capa de neuronas, completamente interconectadas teniendo de este modo las conexiones de entrada desde una cadena externa de entradas; es decir, la topología que utiliza este tipo de aprendizaje es de tipo recurrente, ver siguiente figura:



Aunque este tipo de arquitectura es simple, sin embargo, es poderosamente fuerte en el sistema computacional.

Matemáticamente, el algoritmo de aprendizaje True Time se define de la siguiente forma:

Derivando el gradiente basado en la regla de las redes recurrentes, hacemos que la conectividad de la red no sea restringida. Simplemente se considera que se tienen las entradas $I = \{x_k(t), 0 < k < m\}$, y se define otra unidad que pueden ser las capas intermedias (hidden) o simplemente las salidas, denominamos $U = \{y_k(t), 0 < k < n\}$.

$$Z_k(t) = \begin{cases} X_k(t) & ; \text{si } K \in I \\ Y_k(t) & ; \text{si } K \in U \end{cases} \quad (1)$$

Entonces $W_{n,n+m}$ es la matriz de pesos, donde n son los renglones y $n+m$ son las columnas, donde w_{ij} es el peso de la unidad i hasta j las unidades calculadas son las activaciones, es decir, primero se calculan los pesos sumando las entradas; por lo que se tiene lo siguiente:

$$\text{Net}_k(t) = \sum_{I \in U \cup J} W_{kI} Z_I(t) \quad (2)$$

Donde un solo elemento es introducido en la fórmula temporalmente, por lo que las salidas que despliega la función son no lineales.

$$Y_k(t+1) = f_k(\text{net}_k(t)) \quad (3)$$

Usualmente, las salidas y las capas son funciones no lineales.

Nota: La entrada externa en el tiempo t no tiene influencia en la salida o en cualquier unidad de tiempo $t+1$. Entonces se puede decir que la red es a fin de cuentas un sistema dinámico discreto.

Algunas unidades de U son unidades de salidas, para lo cual las "blanco" están definidas. Una "blanco" puede no estar definida para cada entrada. Por ejemplo, se presenta una cadena de red en la que tiene que clasificar si es gramatical o no. Se puede proporcionar una "blanco" solo para el último símbolo de la cadena.

Al definir un error de las salidas es necesario hacer que el error dependa también del tiempo, entonces no puede ser definido con el valor de cero para una salida, la cual no cuenta con su "blanco". Esto permite que $T(t)$ este en la unidad de U con el índice de k para lo que existe un valor de "blanco" $d_k(t)$ en un tiempo t . Esto implica forzar que la notación d_k sea para un instante t , referenciando a t para el tiempo. Por lo que el error para las salidas es:

$$e_k(t) = \begin{cases} d_k(t) - y_k(t) & \text{si } k \in T(t) \\ 0 & \text{otro caso} \end{cases} \quad (4)$$

Y definiendo al error de la función para un simple paso de tiempo como:

$$E(\tau) = \frac{1}{2} \sum_{k \in U} [e_k(\tau)]^2 \quad (5)$$

Para minimizar la sumatoria de este error de la función dentro de un error que contenga todos los pasos de tiempo de la red, es:

$$E_{\text{total}}(t_0, t_1) = \sum_{\tau=t_0+1}^{t_1} E(\tau) \quad (6)$$

Ahora, porque el total del error es la sumatoria total previa de los errores y del error de tiempo, por ende, el gradiente total del error es la sumatoria del gradiente para este tiempo en ese instante y el gradiente para los siguientes instantes, por lo que se tiene lo siguiente:

$$\tilde{\mathbf{N}}_w \mathbf{E}_{\text{total}}(t_0, t+1) = \tilde{\mathbf{N}}_w \mathbf{E}_{\text{total}}(t_0, t) + \tilde{\mathbf{N}}_w \mathbf{E}(t+1) \quad (7)$$

Al mismo tiempo, las series son presentadas a la red, donde se va acumulando los valores del gradiente o en su defecto, los cambios de los pesos. Hasta este momento se ha conservado el valor.

$$\Delta w_{ij}(t) = -\mu \frac{\partial E(t)}{\partial w_{ij}} \quad (8)$$

Después de que la red ha sido presentada con toda las series, se pueden alterar cada peso en w_{ij} .

$$\sum_{t=t_0+1}^{t_1} \Delta w_{ij}(t) \quad (9)$$

Por lo que se necesita un algoritmo para calcular

$$-\frac{\partial E(t)}{\partial w_{ij}} = -\sum \frac{\partial E(t)}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial w_{ij}} = \sum_{k \in U} e_k(t) \frac{\partial y_k(t)}{\partial w_{ij}} \quad (10)$$

Cada paso del tiempo t . Se conoce el error $e_k(t)$ para todos los tiempos (la diferencia entre las "blancos" y las salidas), solo se necesita encontrar la forma para calcular el segundo factor.

$$\frac{\partial y_k(t)}{\partial w_{ij}}$$

Importante:

La medida esencial de la sensibilidad del valor de la salida de unidad k en un tiempo t es un cambio pequeño en el valor de w_{ij} tomando como importancia el

efecto de cada cambio de los pesos sobre la trayectoria de toda la red que va desde t_0 a t .

Nota: el w_{ij} no tiene que estar conectado a la unidad k . Este algoritmo no es local, es decir, es necesario considerar el efecto del cambio en un solo lugar dentro de la red sobre los valores calculados en diferentes lugares de esta.

Hay que estar seguros de entender esto antes de entrar de lleno dentro de la siguiente derivación.

$$\frac{\partial y_k(t)}{\partial w_{ij}}$$

Derivación:

De la ecuación 2 y 3 se obtiene:

$$\frac{\partial y_k(t+1)}{\partial w_{ij}} = f'_k(\text{net } \kappa(t)) \left[\sum_{I \in U \cup J} w_{kI} \frac{\partial z_I(t)}{\partial w_{ij}} + \delta_{iK} z_j(t) \right] \quad (11)$$

donde δ_{ik} es la delta de Kronecker.

$$\delta_{iK} = \begin{cases} 1 & \text{si } i = K \\ 0 & \text{Otro caso} \end{cases} \quad (12)$$

Esto es porque la señal de entrada no depende de los pesos de la red.

$$\frac{\partial z_I(t)}{\partial w_{ij}} = 0 \quad \text{Para } I \in I \quad (13)$$

La ecuación 11 se iguala en:

$$\frac{\partial y_k(t+1)}{\partial W_{ij}} = f'_k(\text{net } \kappa(t)) \left[\sum_{I \in U} w_{kI} \frac{\partial y_I(t)}{\partial W_{ij}} + \delta_{iK} z_j(t) \right] \quad (14)$$

Esta es una ecuación recursiva, esto es, que no se conoce el valor del lado izquierdo para el tiempo 0, para el tiempo 1, para el tiempo 2, etc. Esto es porque se asume que el estado inicial ($t = 0$) es independiente de los pesos, por lo que se tiene:

$$\frac{\partial Y_k(t_0)}{\partial W_{ij}} = 0 \quad (15)$$

$$K \in U, i \in U \text{ y } j \in U \cup I$$

Por lo que es necesario definir los valores

$$p_{ij}^k(t) = \frac{\partial Y_k(t)}{\partial W_{ij}} \quad (16)$$

Para cada paso del tiempo t y propiamente para todos los i, j y k . Para inicializar con la condición inicial.

$$p_{ij}^k(t_0) = 0 \quad (17)$$

y calculando el paso de cada tiempo

$$p_{ij}^k(t+1) = f'_k(\text{net } \kappa(t)) \left[\sum_{I \in U} w_{kI} p_{ij}^I(t) + \delta_{iK} z_j(t) \right] \quad (18)$$

El algoritmo entonces consiste en calcular, cada paso del tiempo t , la cantidad de $p_{ij}^k(t)$ usando las ecuaciones 16 y 17, y usando así los diferentes "blancos" y las actuales salidas para calcular los cambios de los pesos.

$$\Delta w_{ij}(t) = -\mu \sum_{k \in U} e_k(t) p_{ij}^k(t) \quad (19)$$

Es así como se aplica, completamente la corrección para ser aplicado a w_{ij} dando así.

$$\Delta w_{ij} = \sum_{t=t_0+1}^{t_1} \Delta w_{ij}(t) \quad (20)$$

Siendo el incremento de w_{ij} la corrección de los pesos de las neuronas en el tiempo t .

II.6 APLICACIONES, SOFTWARE Y HARDWARE DE LAS REDES NEURONALES ARTIFICIALES

II.6.1 Áreas de aplicación de las redes neuronales artificiales.

Las redes neuronales pueden utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como militares.

Se pueden desarrollar redes neuronales en un período de tiempo razonable con la capacidad de realizar tareas concretas al utilizar otras tecnologías. Cuando se implementa mediante hardware (redes neuronales en chips VLSI), presentan una alta tolerancia a fallos del sistema y proporcionan un alto grado de paralelismo en el procesamiento de datos. Esto posibilita la inserción de redes neuronales de bajo costo.

Dependiendo de los diferentes tipos de redes neuronales y sobre todo del tipo de aplicación que se desee realizar, existen distintos campos de aplicaciones comerciales como por ejemplo:

- Biología:
 - Aprender y / o conocer más acerca del cerebro y otros sistemas.
 - Obtención de modelos de la retina.

- Empresa:
 - Evaluación de probabilidad de formaciones geológicas y petrolíferas.
 - Identificación de candidatos para posiciones específicas.
 - Optimización de plazas y horarios en líneas de vuelo.
 - Reconocimiento de caracteres escritos.
 - Modelado de sistemas para automatización y control.

- Medio Ambiente:
 - Analizar tendencias y patrones.
 - Previsión del tiempo.

- Finanzas:
 - Previsión de la evolución de los precios.
 - Valoración del riesgo de los créditos.
 - Identificación de falsificaciones.
 - Interpretación de firmas.

- Manufacturación:
 - Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc).
 - Control de producción en líneas de procesos.
 - Inspección de calidad.

- Medicina:
 - Analizadores de habla para ayudar en la audición de sordos profundos.
 - Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (electrocardiograma, encefalogramas, análisis sanguíneo, etc).
 - Predicción de reacciones adversas en los medicamentos.
 - Entendimiento de la causa de los ataques cardíacos.

- Militar:
 - Clasificación de las señales de radar.
 - Creación de armas inteligentes.
 - Optimización del uso de recursos escasos.
 - Reconocimiento y seguimiento en el tiro al blanco.
 - Predicción de tormentas de arena en el desierto.

En general, se puede decir que las aplicaciones consisten en realizar un reconocimiento de patrones, como son: buscar un patrón en una serie de patrones, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado. Sin embargo, está creciendo el uso de redes neuronales en distintos tipos de sistemas de control.

Se puede afirmar entonces que el dominio de la aplicación de las redes neuronales se clasifican de la siguiente forma: asociación y clasificación, regeneración de patrones, regresión y generalización, optimización y predicción.

II.6.2 SOFTWARE Y HARDWARE DE LAS REDES NEURONALES ARTIFICIALES.

El modo más habitual de realizar una red neuronal consiste en simularla en un ordenador convencional, como un PC o en una estación de trabajo, haciendo uso de programas escritos en lenguajes de alto nivel.

Esto constituye la manera más barata y directa de realizar una red neuronal, por otra parte, no es necesario que cada diseñador confeccione sus propios simuladores, pues existen sistemas de desarrollo comerciales para redes neuronales artificiales, consistentes en un software de simulación que permite el trabajo con multitud de modelos neuronales.

Una orientación diferente consiste en llevar a cabo la emulación en hardware de la red neuronal mediante el empleo de sistemas de cálculo expresamente diseñados para realizar redes neuronales artificiales basadas en microprocesadores de altas prestaciones (RISC, DSP, etc.), procesadores especialmente diseñados para el trabajo con redes neuronales. Estas estructuras suelen denominarse **aceleradores o neurocomputadores de propósito general**.

La tecnología más habitualmente empleada para ello es la microelectrónica VLSI, denominándose **chips neuronales (Ni1000, de NestorACCESS por ejemplo)** a los circuitos integrados construidos con esta tecnología.

La realización hardware de la red neuronal es la manera de poder aplicar estos sistemas a la resolución de problemas que involucran un gran número de datos y precisan respuestas en tiempo real, por ejemplo, un sistema de detección e interceptación de misiles enemigos.

La realización electrónica de redes neuronales es un campo tremendamente activo que está siendo abordado no sólo por grupos universitarios, sino también por las

grandes empresas de los sectores de la electrónica e informática, como Intel, Siemens, Philips, Hitachi, At&t, entre otras.

El fruto de estas empresas es la aparición de productos comerciales, como los neurocomputadores CNAPS de Adaptive Solutions, y Synapse- X de Siemens. La realización de hardware de redes neuronales constituye un área de trabajo muy joven, en la que la tecnología se explota al límite de sus posibilidades.

Las redes neuronales se utilizan en la resolución de problemas prácticos concretos que normalmente no han sido bien resueltos mediante sistemas más convencionales, como pueda ser el caso del reconocimiento de vehículos en los peajes de las autopistas o la previsión de consumo eléctrico.

Por ejemplo, Sharp ha desarrollado en Japón un sistema de reconocimiento de caracteres para el alfabeto Kanji (japones) utilizando una red neuronal jerárquica basada en LVQ. Por su parte, Synaptics, empresa situada en el Silicon Valley, ha desarrollado un chip neuronal para el reconocimiento de direcciones escritas en los sobres de las cartas.

Un área de intenso trabajo es el tratamiento de la información económica. En este campo, uno de los grupos punteros es el de A.N. Refenses, de la London Business School, que ha desarrollado varios sistemas de predicción de parámetros económicos (orientados a inversión) que superan el rendimiento de los métodos convencionales.

Fujitsu, Kawasaki y Nippon Steel Corp. han desarrollado sistemas neuronales que se emplean en el control de procesos industriales como por ejemplo, en plantas de producción de acero. Citroën emplea redes neuronales en la determinación de la calidad del material utilizado en la confección de los asientos de los vehículos.

Mercedes Benz y Ford estudian el empleo de chips neuronales para ahorrar combustible o predecir fallas; en Renault se emplean mapas autoorganizados para detectar averías en el encendido de los automóviles.

Por ejemplo, el laboratorio de Cibernética de la Facultad de Ciencias, cuenta con diversos sistemas de simulación de redes neuronales como son Cibernet, Neurored y Dynamind; mientras que en cuestión de Hardware cuenta con Nestor ACCESS.

Nestor ACCESS es un componente en hardware capaz de ejecutar las tareas del reconocimiento a muy alta velocidad.

Nestor Ni1000, ejecuta rápidamente el reconocimiento de patrones usando el algoritmo estadístico basado en el Función Básica Radial y Regla de Bayes para determinar la función de densidad de probabilidad y estimaciones.

La virtud del Ni 1000 es que ejecuta el procesamiento de operaciones en paralelo, esos sistemas vienen más integrados y con un costo efectivo.

El Ni 1000 puede ser usado para las siguientes aplicaciones:

- Biométrica o biomédica
- Alta velocidad ORC/ICR
- Diagnósticos y procesamiento de imágenes en medicina
- Inspección Industrial
- Procesos de control
- Procesamiento de señales
- Finanzas.

El Ni 1000 ejecuta 16.5 billones de operaciones especializadas por segundo.

Especificaciones Generales:

- Su sistema trabaja en paralelo
- Arquitectura flexible
- Escalable
- Microcontrolador programable
- Alta velocidad en el aprendizaje de algoritmos
- Es muy rápido, se clasifica en razón de mas de 33,000 patrones por segundo a 33MHZ

Por otra parte, el simulador y emulador **Dynamind** para redes neuronales artificiales; cuenta con tres tipos de algoritmos de aprendizaje, Back Propagation, Madaline III y True Time.

Dynamind es un software que analiza los datos, encuentra patrones o elabora predicciones. El poder de este software se deriva del procesamiento y de su arquitectura ya que esta inspirado en el sistema biológico de las neuronas.

Siendo este uno de los muchos sistemas de desarrollo de redes neuronales artificiales que se encuentran en el mercado.

“Cualquier cosa que la mente puede concebir y creer, puede ser logrado”

Napoleón Hill.

CAPITULO III

“PREDICCIÓN: SEÑAL SENOIDAL Y NÚMEROS ALEATORIOS”

“PREDICCIÓN DE PUNTOS QUE CONFORMAN UNA SEÑAL SENOIDAL”

III.1 OBJETIVO DEL CAPÍTULO: “Elaborar dos redes neuronales en el ambiente de desarrollo Dynamind, capaces de predecir y “reconstruir ” una señal senoidal”.

III.2 INTRODUCCIÓN:

Se generaron dos redes neuronales: una de ellas constituida por ocho neuronas y la segunda por dos, denominadas **SENO8.NET** y **SENO2.NET** respectivamente; el motivo es evaluar en cada una de ellas su capacidad de **pronosticar** una serie de puntos que conforman a la señal senoidal.

Entre otros parámetros que se tomaron en cuenta para determinar cual de ellas es la más eficiente, son por ejemplo, el tiempo de aprendizaje que emplean, los errores que presentan cada una de ellas, y claro está, la que mejor y mayores predicciones cercanas a las reales efectúe.

¿Por qué una red de dos y otra red de ocho neuronas? La elección de elegir una red de 2 y otra de 8 neuronas, es debido a que se toman estas dos opciones como “extremos”, es decir, dos neuronas como la unidad básica para crear la red y la de ocho por considerarla como una red “grande”; aunque puede existir o generarse una de mayor número de neuronas. Por experiencia adquirida se tiene por un lado, que el manejo de datos, en especial los archivos de resultados se vuelve complicado cuando la red es grande, y por el otro, se observa que la diferencia de resultados entre una red de mayor neuronas son similares a los de

una red de escasas neuronas; por lo que se confirmará esta hipótesis en los resultados obtenidos.

Básicamente, la idea es determinar cual de las dos redes neuronales es más eficiente.

Para ello a cada una de las redes se les enseñó 76 puntos de la señal senoidal formando así un ciclo con amplitud unitaria.

III.3 DESARROLLO

III.3.1 PREPARACIÓN DE LA BASE DE DATOS.

El dato o información pertinente para entrenar la red neuronal se obtiene mediante un simple programa en Matlab. Este programa únicamente generará un ciclo de una señal senoidal con amplitud unitaria; cada punto de esta señal esta distribuida cada ocho centésimas dando un total de 76 puntos. A continuación se muestra el programa la señal senoidal y su base de datos que la conforma.

Obtención de la base de datos (Programa Matlab).

```
Clear all; close all; clc;

% Generación de una onda senoidal
% para generar el archivo de datos
% denominado " SENO.DAT "

t = 0:0.08:6;
y=sin (t)
figure (1)
subplot (1,1,1);stem(t,y,'b');
y1 = y';
save ( 'C:/Seno/Matlab/Seno.txt', 'y1', '-ASCII');
```

Explicación del programa:

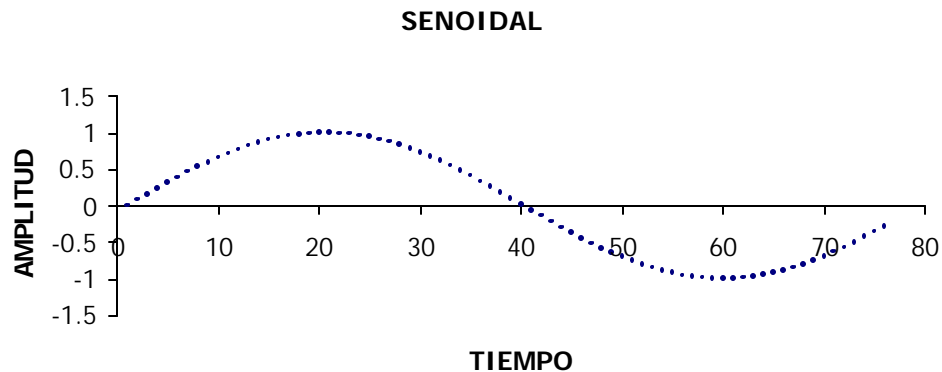
La primera variable es el tiempo (**t**), que es la que va a generar la señal, pues ésta se encuentra dentro de la función "sin" asignada a la variable "**y**". Este tiempo se incrementará cada 8 centésimas hasta llegar a seis unidades.

Con las instrucciones **figure(1)**, **subplot(1,1,1)** y **stem(t,y,'b')** son las que permite graficar la senoidal con las variables **t**, **y**.

Finalmente, la variable "**y'**", nos muestra los datos que conforman a la senoidal en forma de columna y son salvados con el nombre de **SENO.DAT**.

SENO. DAT							
TIEMPO	AMPLITUD	TIEMPO	AMPLITUD	TIEMPO	AMPLITUD	TIEMPO	AMPLITUD
1	0.000	21	1.000	41	0.0584	61	-0.996
2	0.0799	22	0.994	42	-0.138	62	-0.986
3	0.159	23	0.982	43	-0.217	63	-0.970
4	0.238	24	0.964	44	-0.294	64	-0.947
5	0.315	25	0.94	45	-0.369	65	-0.918
6	0.389	26	0.909	46	-0.443	66	-0.883
7	0.462	27	0.873	47	-0.513	67	-0.843
8	0.531	28	0.831	48	-0.580	68	-0.798
9	0.597	29	0.784	49	-0.643	69	-0.747
10	0.659	30	0.732	50	-0.702	70	-0.691
11	0.717	31	0.675	51	-0.757	71	-0.631
12	0.771	32	0.614	52	-0.807	72	-0.567
13	0.819	33	0.549	53	-0.851	73	-0.500
14	0.862	34	0.481	54	-0.890	74	-0.429
15	0.900	35	0.409	55	-0.924	75	-0.355
16	0.932	36	0.335	56	-0.952	76	-0.279
17	0.958	37	0.259	57	-0.973		
18	0.978	38	0.181	58	-0.988		
19	0.991	39	0.101	59	-0.997		
20	0.999	40	0.0216	60	-1.000		

Gráficamente:



Generación de los archivos de entrada y salida (IO).

A continuación se procede a elaborar el archivo "general" de entradas y salidas (IO). En este archivo general denominado **SENO2. IO** su importancia radica en que a partir de él, se obtendrán los archivos de entrenamiento y de verificación al ser particionado.

El procedimiento para la construcción del archivo IO, es el siguiente:

En Ms-Dos, y en la ruta del sistema de desarrollo de Dynamind;

C:\ ndx \ examples

Teclar: **iobuild SENO.DAT SENO2.IO**

Obsérvese que después de la instrucción **iobuild**, enseguida se coloca el nombre del archivo que contiene la información inicial (seno.dat), posteriormente se agrega el nombre del archivo de entradas y salidas (seno2.io).

Inmediatamente, Dynamind despliega la siguiente pantalla:

```

IOBUILDer Version 2.0
[C] 1992 NeuroDynamax, Inc.
Please specify the format for your ASCII file
By choosing a number of the list below.
FEEDFORWARD NETWORKS:
    [1] Inputs followed by outputs
    [2] Inputs and outputs locations in a record to be specified
    [3] Inputs only
RECURRENT NETWORKS (True Time):
    [4] Single or multiple data streams

```

Seleccionar: método cuatro y completar la información pertinente. Recordar que este método trabaja en **tiempo real** o bien, es el método que permite predecir eventos.

Al seleccionarlo, Dynamind va mostrando una serie de preguntas que permitirán generar el archivo IO. Estas preguntas son:

≡ **4 (Selección del método cuatro)**

IOBUILDer - Method 4 (True Time)

Input a one line description of the problem:

≡ **PREDICCIÓN**

Specify the length of a record

≡ **1 (Un solo vector, es decir, una sola columna formada de 1 x 76)**

Specify which fields in a record are inputs,

e.g 1, 8, 3-6 [1 is the first element];

≡ **1 (Al ser una sola columna, por lo tanto una entrada)**

Specify which fields in a record are outputs,

e.g 1, 8, 3-6 [1 is the first element];

≡ **1 (Las entradas son las mismas salidas)**

Specify the total number of network neurons

≡ **2**

Specify the number of records in sequence

≡ **3 (tres vectores por secuencia)**

How many records into the future do you want to predict

≡ **2**

Input the IO sequence label

≡ **SEQ**

Input the IO pair label

≡ **PART**

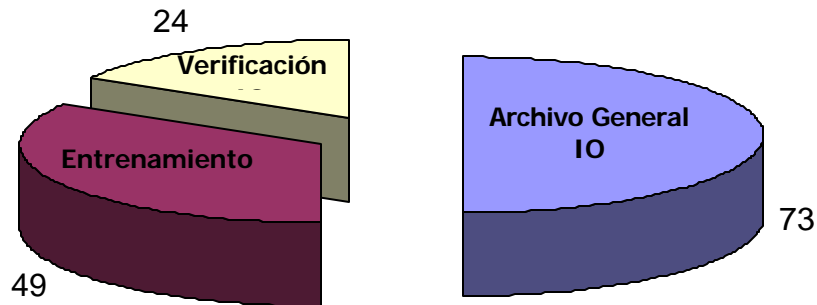
El sistema procesa la información dando como resultado un total de **73 secuencias**.

III.3.2 GENERACIÓN DEL ARCHIVO DE ENTRENAMIENTO Y VERIFICACIÓN

Una vez que se cuenta con el archivo general de entradas y salidas **SENO.IO**, se procede a particionarlo en dos partes, una de ellas corresponderá al archivo de entrenamiento **ESENO2.IO** y la segunda parte para el archivo de verificación **VSENO.IO**.

Para efectuar la partición se toma el siguiente criterio; las dos terceras partes corresponderán al archivo de entrenamiento y el restante para el archivo de verificación; por lo que si se tiene un total de **73 secuencias**, las dos terceras partes de éste son: **49 secuencias** para el **entrenamiento** y **24 secuencias** para la **verificación**.

PARTICIÓN DE DATOS



Específicamente, las secuencias que conforma al archivo de verificación son:

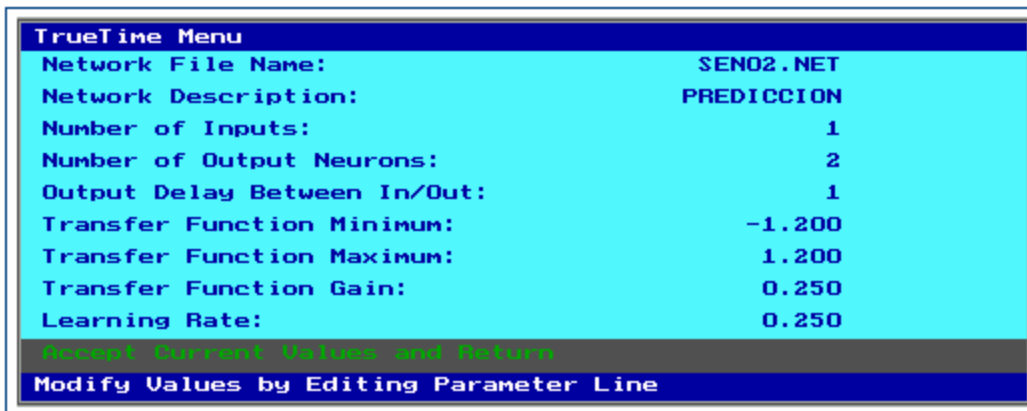
4:6, 16:18, 27:29, 33:35, 42:44, 56:58, 64:66,70:72.

Tres vectores por secuencia (4, 5 y 6).

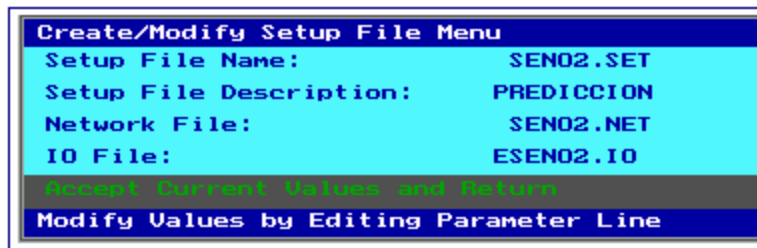
III.3.3 ELABORACIÓN DE LA RED NEURONAL.

Dentro del ambiente de desarrollo de Dynamind del menú principal, seleccionar:
"Edit system configuration / Network file / Create a Network File / True time".

Dynamind despliega la siguiente pantalla mostrando los parámetros a establecer para esta red neuronal; estos parámetros lo conforman desde el nombre de la red, el número de neuronas de entradas, salidas, y los valores correspondientes a la función de transferencia, etc. Tal como se observa en la siguiente figura.



A continuación se genera el archivo setup file **SEN02.SET**. La función de este archivo es guardar el archivo de entradas y salidas (IO) y la red generada. Para ello; del menú principal: *"Edit System Configuration Menu / Setup File / Create Setup File"*; inmediatamente se despliega la siguiente pantalla, donde se tiene que dar de alta el nombre, seleccionar la red indicada y el archivo de entrenamiento.



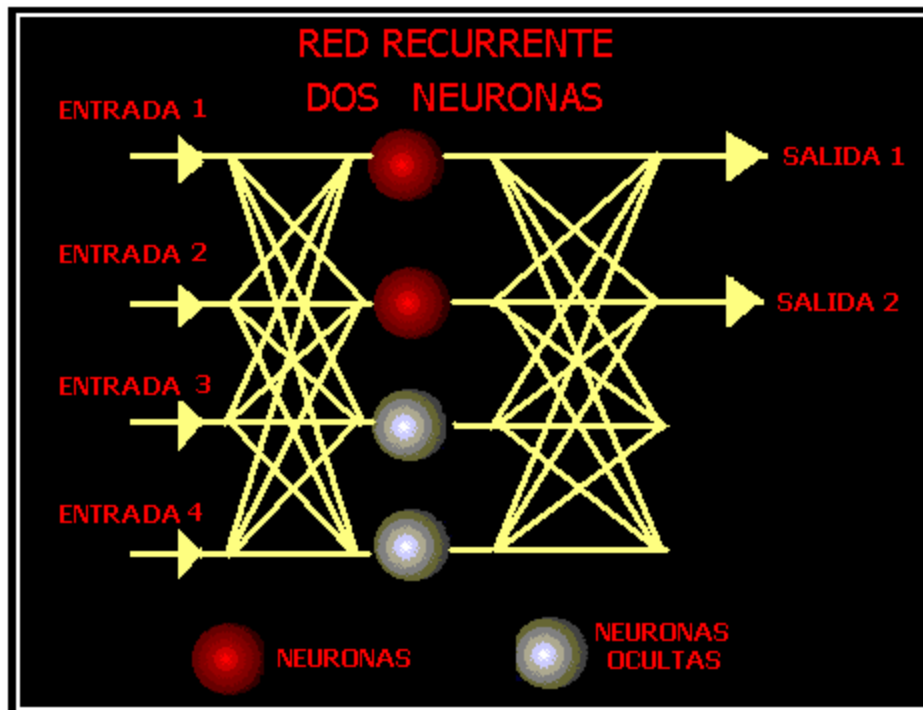
Para confirmar que realmente se han establecido los tres archivos importantes, del menú principal, seleccionar: *"Edit System Configuration / Show network Settings"*. Se desplegará la siguiente ventana donde se confirma la existencia de los archivos y los parámetros de la red establecidos anteriormente.

```
Network Settings
Setup File:          SEN02.SET
Network File:       SEN02.NET
IO File:            SEN0E2.IO

Network Description: PREDICCION
Inputs: 1, Outputs: 2
Output Delay Between Inputs and Outputs: 1 (clock ticks)
Training Algorithm: TrueTime
Transfer Function Min: -1.200, Max: 1.200, Gain: 0.250
Number of Modules: 1
Module 1: Number Neurons: 2 Layer: 1
```

TOPOLOGÍA DE LA RED NEURONAL

La topología de la red neuronal en TrueTime es de tipo recurrente; es decir, consiste de una sola capa de neuronas. Algunas de esas neuronas están definidas en blanco y otras no. Las neuronas no definidas son referenciadas como neuronas "hidden", su utilidad es para complementar el estado dinámico de la red. El número de neuronas ocultas "hidden" van a depender del problema en particular y de las características del dato; es decir, el sistema automáticamente establece el número de neuronas ocultas y estas dependen del número de neuronas establecidas. En este caso, se establecieron dos neuronas, por lo que el sistema estableció dos ocultas automáticamente. Tal como se observa a continuación.



Tipo de aprendizaje utilizado.

El tipo de aprendizaje efectuado por la red es True Time, siendo este el algoritmo de aprendizaje para efectuar predicciones.

III.3.4 ENTRENAMIENTO DE LA RED NEURONAL.

El siguiente paso es entrenar la red neuronal, para entrenar la red del menú principal. Seleccionar:

"Run network file / Train Network / Start Training"

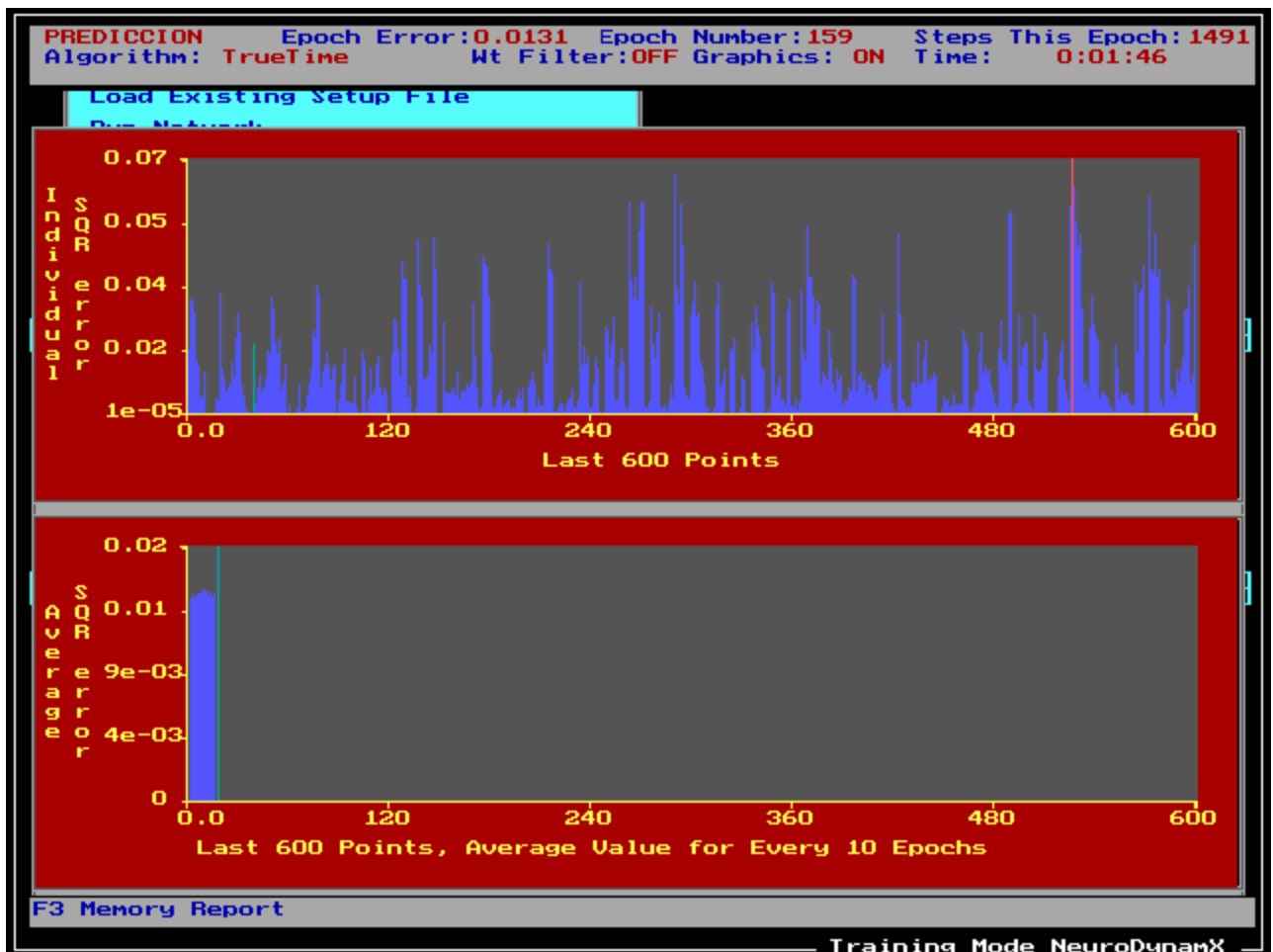
Inmediatamente, la red empieza a efectuar el proceso de entrenamiento.

Este proceso de entrenamiento termina cuando el número de épocas sean 1500. Se adquieren valores de los errores que presentan algunas secuencias antes y después del entrenamiento; la finalidad es saber de cierta forma el aprendizaje correcto o incorrecto de la red.

Para obtener los errores de las secuencias antes y después del menú principal, elegir el modo de "Run/run feedforward/present inputs".

A continuación se presentan los resultados por esta red.

No.	No. SEQ	No. PART		ANTES ENTRENAMIENTO	DESPUÉS ENTRENAMIENTO
1	SEQ 2	PART 3	ERROR	0.0595	0.00227
2	SEQ 12	PART 1	ERROR	0.785	0.000548
3	SEQ 20	PART 2	ERROR	1.060	0.00475
4	SEQ 36	PART 2	ERROR	0.0702	0.0225
5	SEQ 39	PART 2	ERROR	0.000467	0.0096
6	SEQ 48	PART 3	ERROR	0.527	0.000243
7	SEQ 54	PART 2	ERROR	0.910	0.00344
8	SEQ 59	PART 3	ERROR	1.070	0.00716
9	SEQ 67	PART 3	ERROR	0.594	0.0044
10	SEQ 73	PART 2	ERROR	0.200	0.0264



Proceso de entrenamiento de la red neuronal

Al finalizar el entrenamiento, se tiene además los siguientes parámetros:

Epoch Error: **0.0059**
 Time: **1:47**
 Epoch Number: **1499**
 Steps this Epoch: **147**

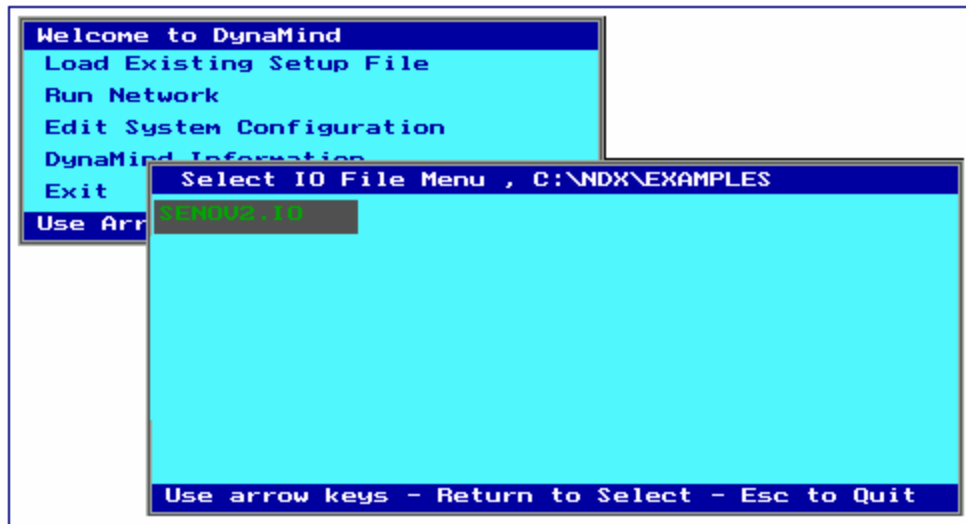
Con estos antecedentes se puede predecir que la red neuronal obtuvo un buen aprendizaje. Los errores que presenta en cada secuencia después del entrenamiento y el error final de época son relativamente bajos.

VERIFICACIÓN DE LA RED NEURONAL.

Cargando el archivo de verificación a la Red Neuronal.

Para verificar el entrenamiento de la red neuronal; es menester cargar el archivo de verificación denominado SENOV2.IO.

Del menú principal elegir: "Edit system configuration/ Select IO file" y elegir el archivo de verificación.



III.3.5 OBTENCIÓN DE RESULTADOS.

Una vez cargado el archivo, se procede a observar los resultados en el modo de feedforward; se ejecuta la corrida y se salvan las salidas de la red. Para ello, seleccionar:

*"Run Feedforward / save out puts to file / write General out put format *.out "*

Automáticamente Dynamind empieza a obtener los resultados y al final se establece el nombre del archivo de resultados que en este caso se llama **SENO2.OUT**.

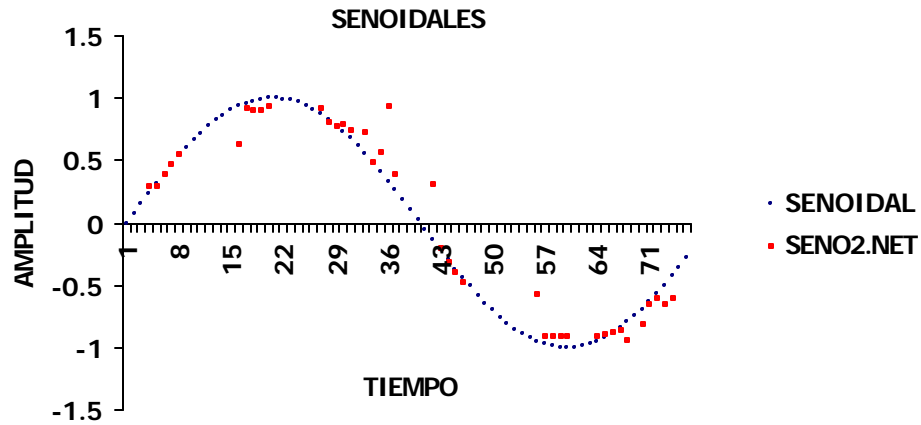
RESULTADOS:

- **Red Neuronal Artificial de Dos Neuronas**

La información que se presenta a continuación son los puntos que conforman a la señal senoidal y los números en negritas son los puntos pronosticados por la red neuronal

No. EVENTO	SEÑAL SENOIDAL	SENOIDAL PREDICIONES	No. EVENTO	SEÑAL SENOIDAL	SENOIDAL PREDICIONES
1	0	0.000	39	0.101	0.101
2	0.0799	0.0799	40	0.0216	0.0216
3	0.159	0.159	41	0.0584	0.0584
4	0.238	0.300	42	-0.138	0.305
5	0.315	0.300	43	-0.217	-0.202
6	0.389	0.388	44	-0.294	-0.305
7	0.462	0.471	45	-0.369	-0.393
8	0.531	0.559	46	-0.443	-0.471
9	0.597	0.597	47	-0.513	-0.513
10	0.659	0.659	48	-0.580	-0.580
11	0.717	0.717	49	-0.643	-0.643
12	0.771	0.771	50	-0.702	-0.702
13	0.819	0.819	51	-0.757	-0.757
14	0.862	0.862	52	-0.807	-0.807
15	0.900	0.900	53	-0.851	-0.851
16	0.932	0.627	54	-0.890	-0.890
17	0.958	0.921	55	-0.924	-0.924
18	0.978	0.916	56	-0.952	-0.564
19	0.991	0.926	57	-0.973	-0.906
20	0.999	0.931	58	-0.988	-0.904
21	1.000	1.000	59	-0.997	-0.906
22	0.994	0.994	60	-1.000	-0.911
23	0.982	0.982	61	-0.996	-0.996
24	0.964	0.964	62	-0.986	-0.986
25	0.940	0.940	63	-0.970	-0.970
26	0.909	0.909	64	-0.947	-0.911
27	0.873	0.929	65	-0.918	-0.888
28	0.831	0.813	66	-0.883	-0.877
29	0.784	0.78	67	-0.843	-0.860
30	0.732	0.813	68	-0.798	-0.939
31	0.675	0.78	69	-0.747	-0.747
32	0.614	0.614	70	-0.691	-0.813
33	0.549	0.773	71	-0.631	-0.657
34	0.481	0.496	72	-0.567	-0.601
35	0.409	0.564	73	-0.500	-0.657
36	0.335	0.941	74	-0.429	-0.596
37	0.259	0.409	75	-0.355	-0.355
38	0.181	0.181	76	-0.279	-0.279

Gráficamente:



NOTAS:

1. Los números que se encuentran en negritas en la tabla son las predicciones elaboradas por la red.
2. En la gráfica, los puntos en rojo son las pronosticaciones, mientras que los azules son los puntos originales de la señal senoidal.

III.4 RED NEURONAL DE OCHO NEURONAS

III.4.1 PREPARACIÓN DE LA BASE DE DATOS.

Generación del archivo de entradas/salidas (IO).

La obtención de la base de datos será a partir del mismo archivo **SENO.DAT**, a partir de éste se genera el archivo "general" denominado **SENO8.IO** del cual saldrán los archivos de entrenamiento y de verificación denominados **SENOE8.IO** y **SENOV8.IO** respectivamente.

Para ello, en Ms-Dos y en la ruta expuesta abajo, teclear:

```
C:\ndx \ examples iobuild SENO.DAT SENO8.IO
```

Inmediatamente, Dynamind despliega la siguiente pantalla:

≡ **4 (Selección del método cuatro)**

IOBUILDer - Method 4 (True Time)

Input a one line description of the problem:

≡ **PREDICCIÓN**

Specify the length of a record

≡ **1 (Un vector de 1X 76)**

Specify which fields in a record are inputs,
e.g 1 ,8, 3-6 [1 is the first element];

≡ **1**

Specify which fields in a record are outputs,
e.g 1 ,8, 3-6 [1 is the first element];

≡ **1 (Las entradas son las mismas salidas)**

Specify the total number of network neurons

≡ **8**

Specify the number of records in sequence

≡ **3 (tres vectores por secuencia)**

How many records into the future do you want to predict

≡ **2**

Input the IO sequence label

≡ **SEQ**

Input the IO pair label

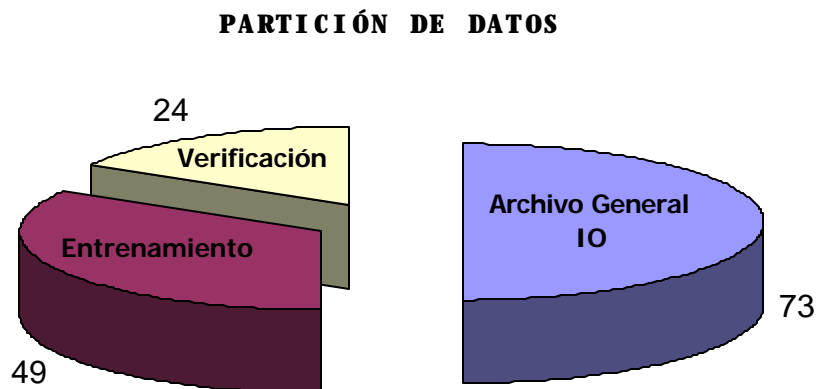
≡ **PART**

El sistema procesa la información dando como resultado un total de **73** secuencias.

Obsérvese que el número de secuencias establecidas en esta red neuronal son idénticas a la red de dos neuronas, esto se debe a que el objetivo es saber cual de las dos redes es más eficiente en la predicción de puntos, es por eso que se manejan parámetros iguales solo que la variable en este caso son el número de neuronas.

III.4.2 GENERACIÓN DEL ARCHIVO DE ENTRENAMIENTO Y VERIFICACIÓN

Una vez que se cuenta con el archivo general de entradas y salidas **SENO.IO**, se procede a particionar del mismo modo que a la primera red neuronal, es decir, las dos terceras partes corresponderán al archivo de entrenamiento y el restante para el archivo de verificación; **49 secuencias** para el entrenamiento y **24 secuencias** para la verificación, denominados **SENOE8.IO** y **SENOV8.IO** respectivamente.



Específicamente, las secuencias que conforma al archivo de verificación son:

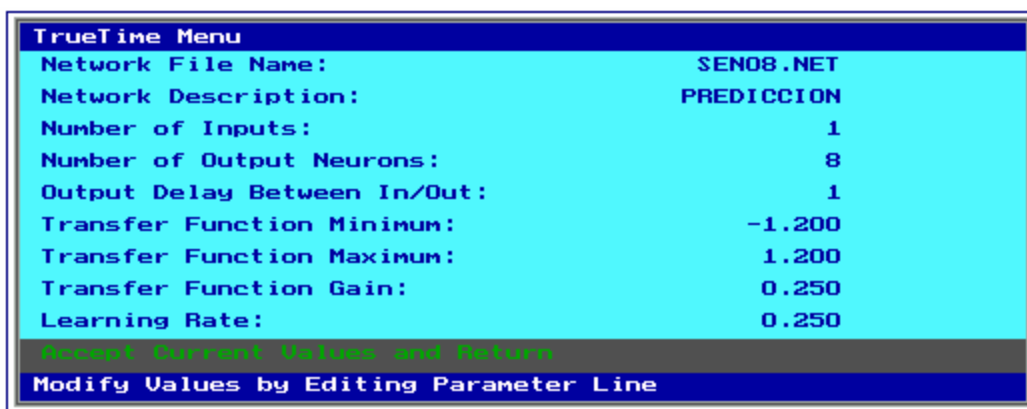
4:6, 16:18, 27:29, 33:35, 42:44, 56:58, 64:66,70:72.

Tres vectores por secuencia.

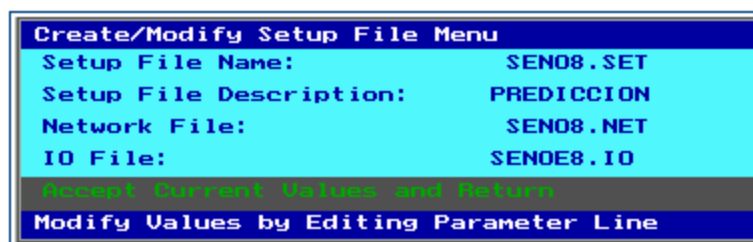
III.4.3 ELABORACIÓN DE LA RED NEURONAL.

Dentro del ambiente de desarrollo de Dynamind del menú principal, seleccionar: *"Edit system configuration / Network file / Create a Network File / True time"*.

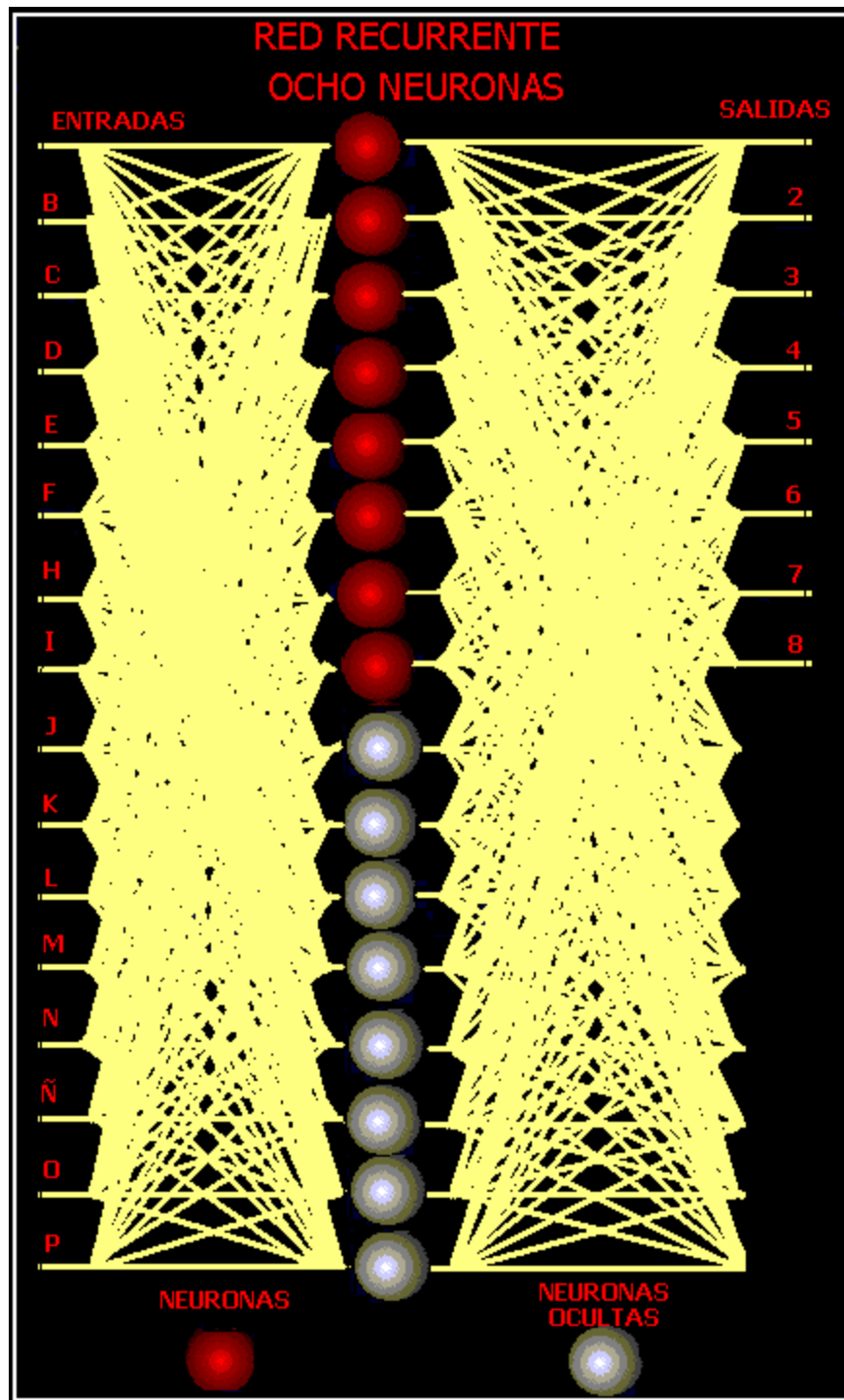
Dynamind despliega la siguiente pantalla; mostrando los parámetros establecidos para esta red neuronal.



A continuación se genera el archivo setup file **SEN08.SET**, del menú principal: *"Edit System Configuration Menu / Setup File / Create Setup File"*; inmediatamente se despliega la siguiente pantalla, donde se da de alta el nombre, seleccionar la red indicada y el archivo de entrenamiento correspondiente.



TOPOLOGÍA DE LA RED NEURONAL.



Tipo de aprendizaje utilizado.

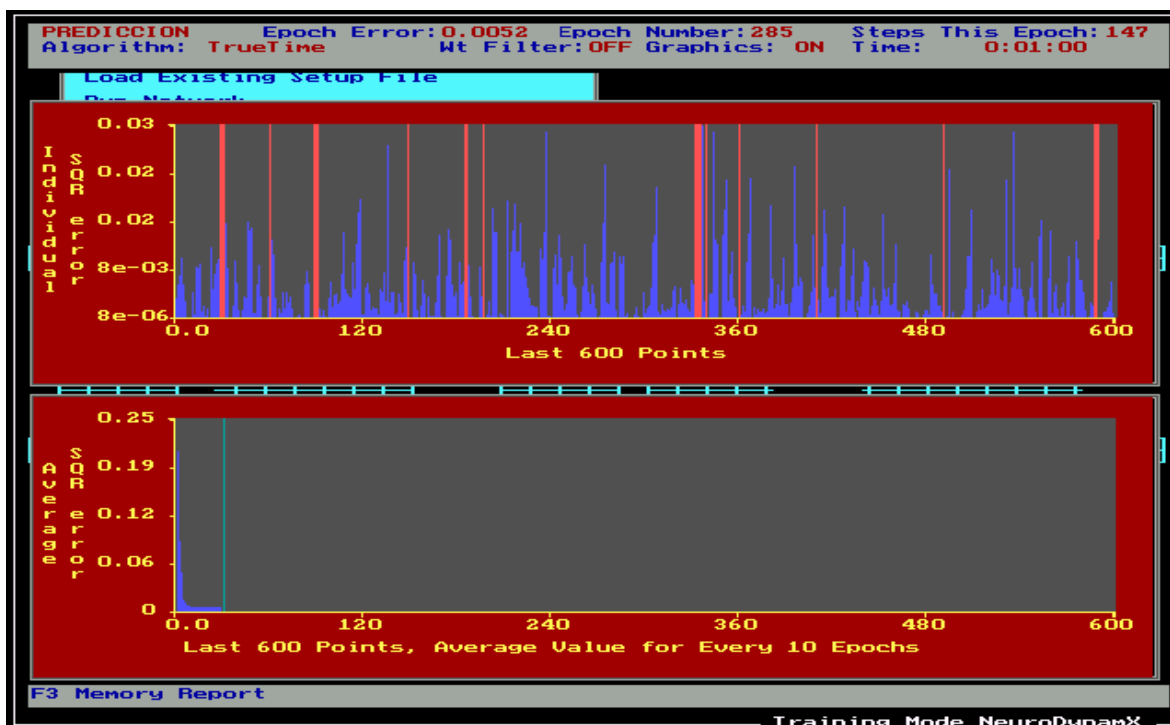
El algoritmo de aprendizaje efectuado por la red es el True Time.

III.4.4 ENTRENAMIENTO DE LA RED NEURONAL.

A continuación se procede a entrenar la red neuronal del menú principal, seleccionar: "Run network file / Train Network / Start Training".

Se adquieren algunos valores de los errores que presentan las secuencias antes y después del entrenamiento, presentándose a continuación.

No.	No. SEQ	No. PART		ANTES ENTRENAMIENTO	DESPUÉS ENTRENAMIENTO
1	SEQ 2	PART 3	ERROR	0.0458	0.00227
2	SEQ 12	PART 1	ERROR	0.723	0.000876
3	SEQ 20	PART 2	ERROR	0.988	0.00475
4	SEQ 36	PART 2	ERROR	0.0552	0.0194
5	SEQ 39	PART 2	ERROR	0.00000574	0.0096
6	SEQ 48	PART 3	ERROR	0.554	0.0000612
7	SEQ 54	PART 2	ERROR	0.933	0.0028
8	SEQ 59	PART 3	ERROR	1.080	0.00486
9	SEQ 67	PART 3	ERROR	0.622	0.00484
10	SEQ 73	PART 2	ERROR	0.216	0.0235



Proceso de entrenamiento

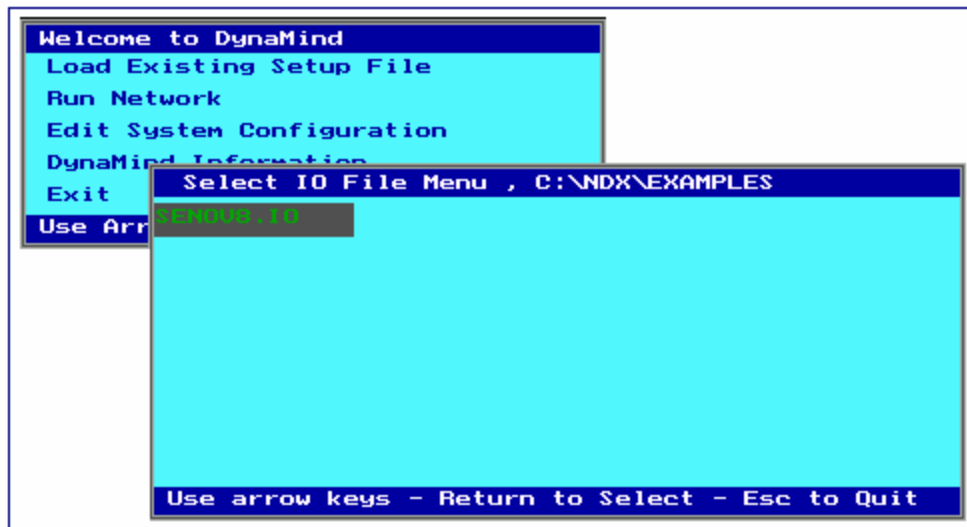
por otro lado, se tiene que:

Epoch Error: **0.0052**
Time: **5:09**
Epoch Number: **1499**
Steps this Epoch: **147**

Con estos antecedentes se puede pronosticar que la red neuronal obtuvo un buen aprendizaje; los errores que se presentan en cada secuencia y el error final de época fueron bajos.

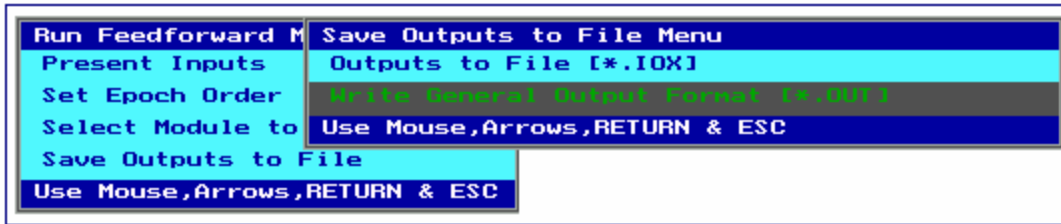
VERIFICACIÓN DE LA RED NEURONAL.

Se procede a cargar el archivo de verificación **SENOV8.IO** a la red neuronal, para ello del menú principal elegir; *"Edit system configuration/ Select IO file"* y elegir el archivo de verificación.



III.4.5 OBTENCIÓN DE RESULTADOS.

Si los resultados obtenidos son satisfactorios, se procede a salvarlos del menú principal: *“Run Feedforward / save out puts to file / write General out put format *.out”*; para este caso el archivo se denomina **SEN08.OUT**.

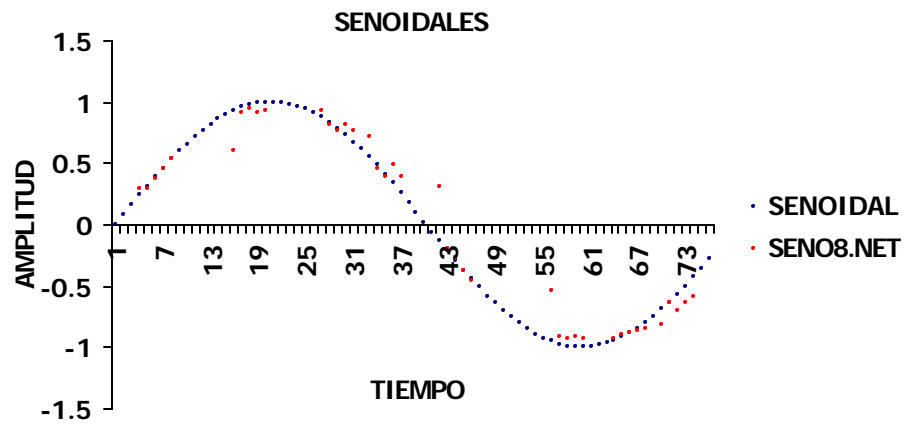


- Red Neuronal Artificial de Ocho Neuronas

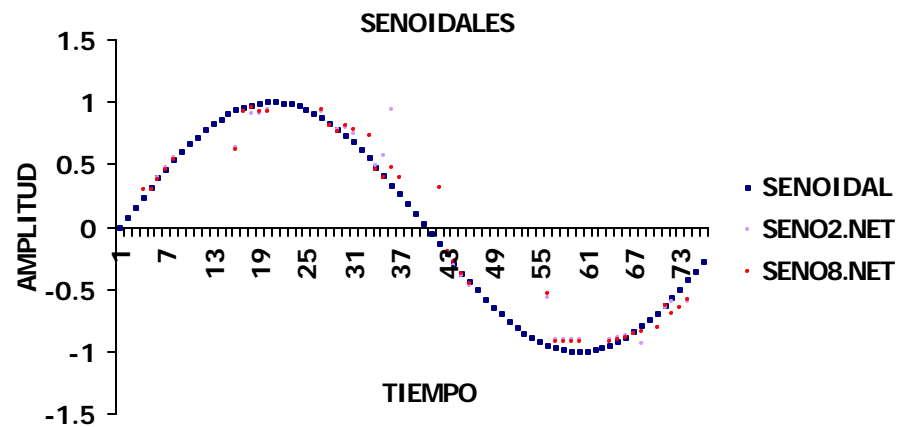
No. EVENTO	SEÑAL SENOIDAL	SENOIDAL PREDICCIONES	No. EVENTO	SEÑAL SENOIDAL	SENOIDAL PREDICCIONES
1	0	0.000	39	0.101	0.101
2	0.0799	0.0799	40	0.0216	0.0216
3	0.159	0.159	41	0.0584	0.0584
4	0.238	0.300	42	-0.138	0.305
5	0.315	0.294	43	-0.217	-0.196
6	0.389	0.377	44	-0.294	-0.283
7	0.462	0.461	45	-0.369	-0.371
8	0.531	0.540	46	-0.443	-0.456
9	0.597	0.597	47	-0.513	-0.513
10	0.659	0.659	48	-0.580	-0.580
11	0.717	0.717	49	-0.643	-0.643
12	0.771	0.771	50	-0.702	-0.702
13	0.819	0.819	51	-0.757	-0.757
14	0.862	0.862	52	-0.807	-0.807
15	0.900	0.900	53	-0.851	-0.851
16	0.932	0.610	54	-0.890	-0.890
17	0.958	0.919	55	-0.924	-0.924
18	0.978	0.926	56	-0.952	-0.540
19	0.991	0.921	57	-0.973	-0.921
20	0.999	0.926	58	-0.988	-0.926
21	1.000	1.000	59	-0.997	-0.921
22	0.994	0.994	60	-1.000	-0.924
23	0.982	0.982	61	-0.996	-0.996
24	0.964	0.964	62	-0.986	-0.986
25	0.940	0.940	63	-0.970	-0.970
26	0.909	0.909	64	-0.947	-0.926
27	0.873	0.931	65	-0.918	-0.901
28	0.831	0.810	66	-0.883	-0.885
29	0.784	0.769	67	-0.843	-0.865
30	0.732	0.810	68	-0.798	-0.845
31	0.675	0.773	69	-0.747	-0.747
32	0.614	0.614	70	-0.691	-0.817
33	0.549	0.725	71	-0.631	-0.640
34	0.481	0.481	72	-0.567	-0.694
35	0.409	0.398	73	-0.500	-0.644
36	0.335	0.481	74	-0.429	-0.587
37	0.259	0.398	75	-0.355	-0.355
38	0.181	0.181	76	-0.279	-0.279

Gráficamente:

Los puntos en rojo muestran las pronosticaciones efectuadas por la red neuronal.



Gráficas de comparación de predicciones entre la red de dos y ocho neuronas.



Comentarios Generales:

Analizando las siguientes premisas:

- Los errores presentados en cada secuencia de ambas redes son similares e iguales.
- El error por época final que presentan ambas solo varía por siete diezmilésimas.
- Recordar que el número de época para el aprendizaje en ambas fue de 1500.
- En general, las gráficas muestran que las predicciones efectuadas por ambas son muy similares y/o cercanas tanto a los puntos que conforman a la señal original como entre ellas mismas.
- De los 76 puntos que conforma la señal original, 40 de ellos se pronosticaron y de estos 40 puntos, el 30 % son buenos ya que solo varían una centésima del punto original.

Por lo que se concluye que la red neuronal **SENO2.NET** y la red neuronal **SENO8.NET** obtuvieron buena capacidad de pronóstico de puntos de una señal senoidal.

Finalmente se confirma la hipótesis de que la diferencia de resultados entre una red de mayor neuronas son similares a los de una red de escasas neuronas, para esta aplicación en particular; siendo la única diferencia el tiempo de aprendizaje ya que entre mayor es la red neuronal, mayor es el tiempo de aprendizaje y viceversa.

III.4.6 UTILIZANDO LA RED NEURONAL.

Una vez comprobado el aprendizaje correcto de la red neuronal, el último paso que se tiene que realizar es **utilizar la red neuronal**.

Como se observó en la parte de resultados, se tiene que tanto la red de dos y de ocho neuronas obtuvieron prácticamente los mismos resultados, por lo que se utilizara la red de dos neuronas (seno2.net), para aplicar este ultimo paso.

Para ello, se aplicarán dos ejercicios. En uno de ellos se le “exigirá” a la red que prediga 15 puntos de la señal senoidal donde abarque tanto valores negativos como positivos.

III.4.6.1 Ejercicio I.

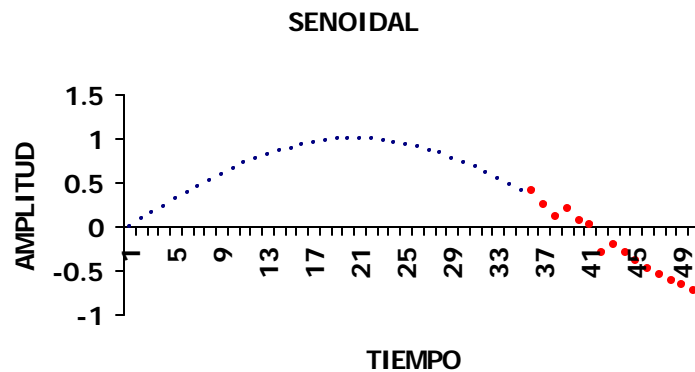
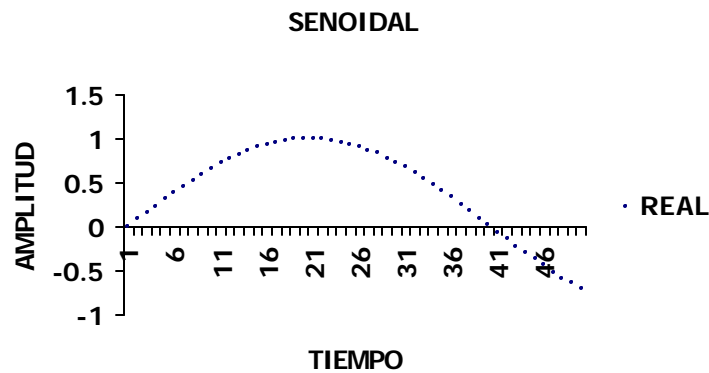
Una vez generado el archivo IO, se procede a cargarlo a la red neuronal y se obtienen los siguientes resultados.

Resultados:

En la siguiente tabla se presentan los 15 puntos pronosticados por la red.

No. EVENTO	TIEMPO	SEÑAL ORIGINAL	PREDICCIONES
1	36	0.335	0.420
2	37	0.256	0.255
3	38	0.181	0.114
4	39	0.101	0.214
5	40	0.0216	0.060
6	41	-0.058	0.018
7	42	-0.138	-0.305
8	43	-0.217	-0.208
9	44	-0.294	-0.300
10	45	-0.369	-0.393
11	46	-0.443	-0.471
12	47	-0.513	-0.545
13	48	-0.58	-0.610
14	49	-0.643	-0.670
15	50	-0.702	-0.718

Gráficamente:



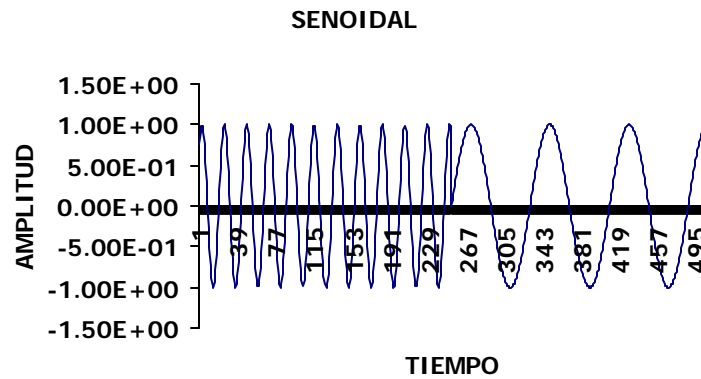
En esta última gráfica se muestran los 15 puntos pronosticados por la red **SENO2.NET**.

Comentario:

Aparentemente, las predicciones realizadas por esta red son relativamente buenas, quizás se deba por un lado, a la compatibilidad de pronosticación (2 puntos) siendo con la que se creo la red. Por otro lado, estos valores son conocidos por la red pues con estos aprendió. Por lo que el siguiente reto para la red neuronal es determinar y/o crear un nuevo archivo de IO pero ahora de una señal muy diferente y evaluar así su comportamiento.

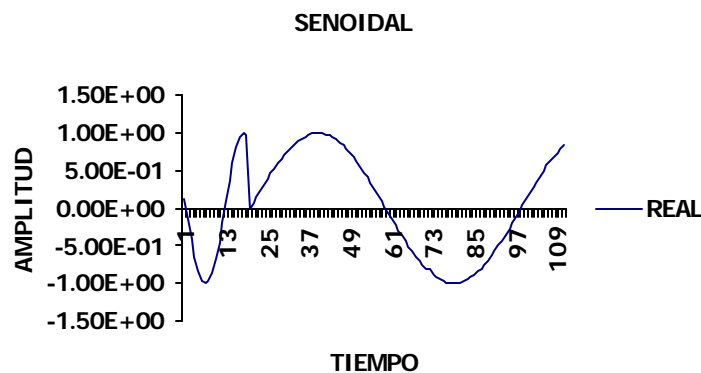
III.4.6.2 Ejercicio II.

Se le introducirá a la red neuronal seno2.net, una nueva señal senoidal con valores distintos; es decir, esta nueva señal senoidal constituida por dos frecuencias diferentes tal como se observa en la siguiente figura.



Como se mencionó anteriormente, el reto es determinar por un lado su capacidad de pronóstico y al mismo tiempo su comportamiento. Para lo que, solo se introducirán la parte donde la senoidal cambia de frecuencia, es decir, a partir del tiempo 235 al 328.

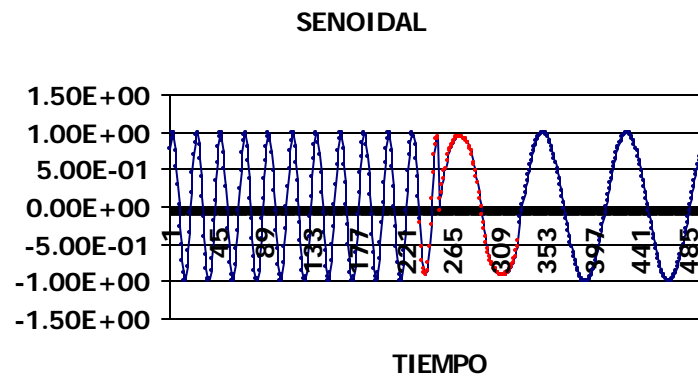
Ver siguiente figura.



Dos ciclos con frecuencia distinta.

RESULTADOS:

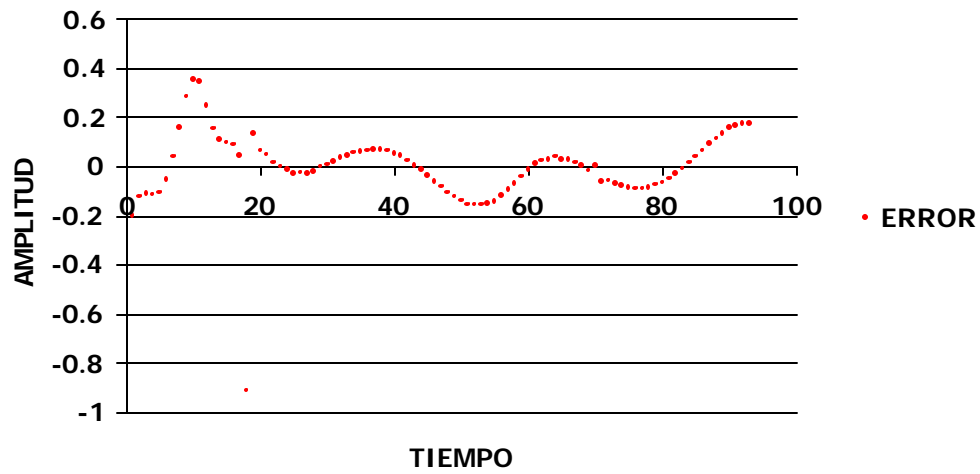
La red pronosticó 93 puntos. Siendo estos donde la senoidal cambia de frecuencia. Los puntos en rojo indican las pronosticaciones realizadas por la red neuronal.

*Comentarios:*

1. A la red neuronal seno2.net aparentemente no le afectó esta señal con frecuencia distinta; pues los 93 puntos que pronosticó son relativamente cercanos a los reales. Excepto en la amplitud que es menor a la original.
2. En la siguiente tabla se muestra los valores reales y pronosticados así como el error existente entre los 92 puntos y la gráfica correspondiente al error.

TIEMPO	VALOR REAL	VALOR PRONOSTICADO	ERROR (VR-VP)	VALOR REAL	VALOR PRONOSTICADO	ERROR (VR-VP)
1	-0.418	-0.22	-0.198	0.732	0.813	-0.081
2	-0.653	-0.531	-0.122	0.675	0.78	-0.105
3	-0.837	-0.725	-0.112	0.614	0.737	-0.123
4	-0.956	-0.842	-0.114	0.549	0.686	-0.137
5	-1	-0.896	-0.104	0.481	0.636	-0.155
6	-0.966	-0.914	-0.052	0.409	0.564	-0.155
7	-0.857	-0.896	0.039	0.335	0.491	-0.156
8	-0.682	-0.842	0.16	0.259	0.409	-0.15
9	-0.453	-0.737	0.284	0.181	0.322	-0.141
10	-0.189	-0.54	0.351	0.101	0.22	-0.119
11	0.089	-0.254	0.343	0.021	0.114	-0.093
12	0.362	0.114	0.248	-0.058	0.012	-0.07
13	0.605	0.451	0.154	-0.138	-0.096	-0.042
14	0.802	0.694	0.108	-0.217	-0.202	-0.015
15	0.936	0.839	0.097	-0.294	-0.305	0.011
16	0.997	0.909	0.088	-0.369	-0.393	0.024
17	0.98	0.933	0.047	-0.443	-0.471	0.028
18	0	0.911	-0.911	-0.513	-0.555	0.042
19	0.079	-0.054	0.133	-0.58	-0.61	0.03
20	0.159	0.096	0.063	-0.643	-0.67	0.027
21	0.238	0.19	0.048	-0.702	-0.718	0.016
22	0.315	0.3	0.015	-0.757	-0.759	0.002
23	0.389	0.388	0.001	-0.807	-0.79	-0.017
24	0.462	0.476	-0.014	-0.815	-0.82	0.005
25	0.531	0.559	-0.028	-0.89	-0.829	-0.061
26	0.597	0.623	-0.026	-0.924	-0.865	-0.059
27	0.659	0.686	-0.027	-0.952	-0.882	-0.07
28	0.717	0.737	-0.02	-0.973	-0.893	-0.08
29	0.771	0.773	-0.002	-0.988	-0.904	-0.084
30	0.819	0.81	0.009	-0.997	-0.906	-0.091
31	0.862	0.842	0.02	-1	-0.911	-0.089
32	0.9	0.865	0.035	-0.996	-0.911	-0.085
33	0.932	0.885	0.047	-0.986	-0.911	-0.075
34	0.958	0.901	0.057	-0.97	-0.906	-0.064
35	0.978	0.916	0.062	-0.947	-0.898	-0.049
36	0.991	0.926	0.065	-0.918	-0.89	-0.028
37	0.999	0.931	0.068	-0.883	-0.874	-0.009
38	1	0.931	0.069	-0.843	-0.86	0.017
39	0.994	0.931	0.063	-0.798	-0.839	0.041
40	0.982	0.931	0.051	-0.747	-0.813	0.066
41	0.964	0.921	0.043	-0.691	-0.783	0.092
42	0.94	0.916	0.024	-0.631	-0.744	0.113
43	0.909	0.904	0.005	-0.567	-0.702	0.135
44	0.873	0.888	-0.015	-0.5	-0.657	0.157
45	0.831	0.868	-0.037	-0.429	-0.596	0.167
46	0.784	0.845	-0.061	-0.355	-0.531	0.176
47				-0.279	-0.456	0.177

Gráficamente:



En esta grafica obsérvese la existencia de pronosticaciones cercanas a los puntos reales ya que los errores presentados son muy cercanos al cero, sin embargo, al transcurrir el tiempo, el error se incrementa.

“PREDICCIÓN DE UNA SERIE DE NÚMEROS ALEATORIOS PROVENIENTES DE UNA BASE DE NÚMEROS ALEATORIOS”

III.5 OBJETIVO DEL CAPÍTULO: Elaborar y encontrar la red neuronal capaz de predecir el siguiente número aleatorio de una secuencia de números aleatorios; para tal efecto, se tendrá que encontrar los parámetros indicados para obtener la eficiencia de la red neuronal.

III.6 INTRODUCCION:

El desempeño principal de la red neuronal es que tenga la capacidad de predecir el siguiente número aleatorio de una base de números aleatorios; es decir, la base numérica original esta compuesta por 50 números del premio mayor de la lotería nacional.

III.7 DESARROLLO:

III.7.1 Preparación de la base de datos.

El formato de estos números no son aceptados por DynaMind, por lo que fue necesario efectuar la normalización de cada uno de ellos empleando la regla de Máximo/ Mínimo (uno de los métodos que propone DynaMind) quedando todos los datos dentro de un rango de ± 1.000 .

A continuación se muestra la base de datos original.

No.	Número Aleatorio	No.	Número Aleatorio
1	39440	26	7660
2	24683	27	23358
3	29882	28	13714
4	7982	29	9060
5	40547	30	36029
6	37580	31	7699
7	10529	32	15799
8	47032	33	6977
9	40767	34	31237
10	2514	35	5359
11	41010	36	29017
12	23495	37	40878
13	35550	38	18229
14	10548	39	7637
15	19445	40	46429
16	5636	41	1424
17	1829	42	40329
18	4275	43	46029
19	5865	44	9540
20	6947	45	10554
21	21429	46	7528
22	28779	47	12529
23	25729	48	27029
24	11429	49	38299
25	44329	50	6557

Normalización de base de datos:

La base de datos debe pasar por el proceso de normalización; esto es por que Dynamid no acepta números de tipo flotante. Para ello se emplea la siguiente fórmula:

$$XS = \left[\left(\frac{Xd - dMin}{dMax - dMin} \right) (TFMax - TFMin) \right] + TFMin$$

donde:

X_d = Dato no escalado

$dMax$ = Valor Maximo del dato a escalar

$dMin$ = Valor Minimo del dato a escalar

$TFMax$ = Valor de la Función de Transferencia Maximo.

$TFMin$ = Valor de la Función de Transferencia Minimo.

A continuación se muestran los valores normalizados.

No.	Número Aleatorio	Número Normalizado	No.	Número Aleatorio	Número Normalizado
1	39440	0.6671	26	7660	-0.7265
2	24683	0.02	27	23358	-0.0382
3	29882	0.2479	28	13714	-0.4611
4	7982	-0.7124	29	9060	-0.6651
5	40547	0.7156	30	36029	0.5175
6	37580	0.5855	31	7699	-0.7248
7	10529	-0.6007	32	15799	-0.3696
8	47032	1.000	33	6977	-0.7565
9	40767	0.7253	34	31237	0.3074
10	2514	-0.9522	35	5359	-0.8274
11	41010	0.7359	36	29017	0.21
12	23495	-0.0321	37	40878	0.7301
13	35550	0.4965	38	18229	-0.2631
14	10548	-0.5999	39	7637	-0.7275
15	19445	-0.2097	40	46429	0.9736
16	5636	-0.8153	41	1424	-1.000
17	1829	-0.9822	42	40329	0.7061
18	4275	-0.875	43	46029	0.956
19	5865	-0.8053	44	9540	-0.6441
20	6947	-0.7578	45	10554	-0.5996
21	21429	-0.1227	46	7528	-0.7323
22	28779	0.1996	47	12529	-0.513
23	25729	0.0658	48	27029	0.1228
24	11429	-0.5613	49	38299	0.617
25	44329	0.8815	50	6557	-0.7749

Una vez normalizados, se procede a generar los diversos archivos de entrada y salida para crear los archivos de entrenamiento y verificación para las tres redes neuronales.

¿Por qué tres redes neuronales? El principal motivo como se indica en el objetivo, es encontrar la red más eficiente; para ello se elaboraron tres redes neuronales donde cada una de ellas varía un parámetro. Este parámetro se fue censando para determinar que tan bueno era para su aprendizaje.

Estos parámetros son: el número de épocas, el número de vectores por secuencia, la ganancia, la razón de aprendizaje, etc.

Generación de los archivos de entradas y salidas (IO) para las distintas redes neuronales.

Antes que otra cosa, se procede a generar el archivo **RANDOM.DAT**; a partir de este, se generaran los diversos archivos IO para cada una de tres redes neuronales.

Para la primera red neuronal denominada **RAND1.NET** el archivo de entradas y salidas (IO) se denominará **RAND1.IO**; a partir de éste se genera el archivo de entrenamiento y de verificación.

Desde Ms-Dos y en la ruta: c: **/ndx/examples**, teclear la siguiente instrucción:

Iobuild RANDOM.DAT RAND1.IO

Presentándose la siguiente imagen.


```

IOBUILDer Version 2.0
[C] 1992 NeuroDynamax,Inc.
Please specify the format for your ASCII file
By choosing a number of the list below.
FEEDFORWARD NETWORKS:
    [1] Inputs followed by outputs
    [2] Inputs and outputs locations in a record to be specified
    [3] Inputs only
RECURRENT NETWORKS (True Time):
    [4] Single or multiple data streams

```

Se selecciona el método cuatro (True Time) y se completa la información pertinente, tal como se muestra a continuación.

>>> **4** (Selección del método cuatro)

IOBUILDer - Method 4 (True Time)

Input a one line description of the problem:

>>> **NÚMEROS ALEATORIOS**

Specify the length of a record

>>> **1 (1 x 50)**

Specify which fields in a record are inputs, e.g 1,8,3-6 [1 is the first element];

>>> **1 (entrada es igual al primer campo)**

Specify which fields in a record are outputs, e.g 1,8,3-6 [1 is the first element];

>>> **1 (las mismas entradas son salidas)**

Specify the total number of network neurons

>>>**2**

Specify the number of records in sequence

>>>**9 (nueve vectores por secuencia)**

How many records into the future do you want to predict

>>> **1 (va a predecir un vector)**

Input the IO sequence label
>>>**SEQ**

Input the IO pair label
>>>**PART**

DynaMind empieza a procesar el archivo de salidas y entradas, presentando **41 secuencias.**

Generación del archivo de Entrenamiento y Verificación (IO).

Ahora se procede a particionar este archivo **RAND1.IO** en dos, es decir, eliminando algunas secuencias de este archivo para asignarlas al archivo de verificación (VRAND1.IO) y las restantes al archivo de entrenamiento (ERAND1.IO).

Para ello, se procede a tomar las 2/3 partes para el archivo de entrenamiento y la tercera parte restante para el archivo de verificación; es decir, 28 secuencias son para el archivo de entrenamiento y 13 secuencias para el de verificación; sin embargo por tener 9 vectores por secuencia se toma el criterio de establecer **33 secuencias para el entrenamiento y 8 secuencias para verificación.**

Las secuencias que a continuación se muestran constituyen el archivo de verificación.

Seq: 3

Seq: 8

Seq: 14

Seq: 17

Seq: 22

Seq: 26

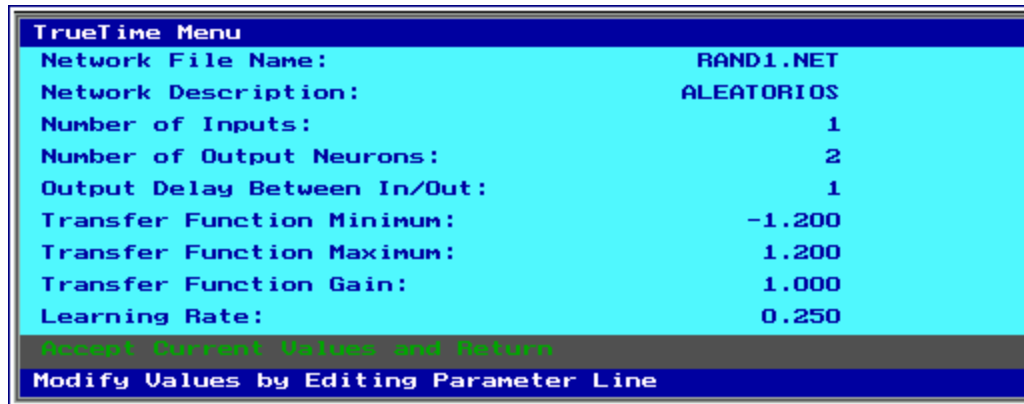
Seq: 35

Seq: 41

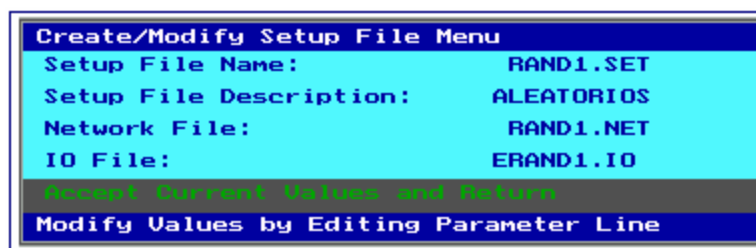
III.7.2 ELABORACIÓN DE LA PRIMERA RED NEURONAL.

En el ambiente de desarrollo de DynaMind del menú principal, seleccionar: "*Edit system / Network file / Create a Network file / True Time*"; mostrándose la siguiente pantalla en la cual se dan de alta los parámetros que se presentan.

El nombre de la red es **RAND1.NET**



A continuación se genera el archivo setup file; para ello, del menú principal "*Edit System Configuration Menu / Setpu File / Create Setup File*"; inmediatamente se desplegará la siguiente imagen donde se tiene que habilitar el nombre del archivo, descripción del archivo, el archivo de la red y el archivo de entradas y salidas (el de entrenamiento).

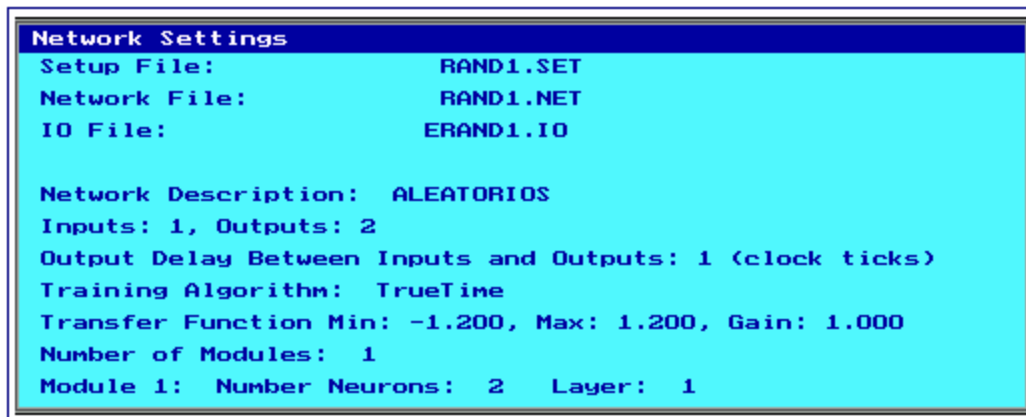


El siguiente paso es habilitar una vez más el archivo **RAND1.SET** y el **ERAND1.IO**; del menú principal elegir: "*Load Existing Setup File*".

Automáticamente se carga la red **RAND1.NET** y el archivo **ERAND1.IO**.

Si se desea corroborar que efectivamente se han cargados estos tres archivos, del menú principal, seleccionar: "Edit System Configuration / Show network Settings".

DynaMind desplegará la siguiente imagen donde además de mostrar los archivos indicados, también despliega los parámetros de la red, como es el número de neuronas, los valores de la función de transferencia, etc.



```
Network Settings
Setup File:          RAND1.SET
Network File:       RAND1.NET
IO File:            ERAND1.IO

Network Description: ALEATORIOS
Inputs: 1, Outputs: 2
Output Delay Between Inputs and Outputs: 1 (clock ticks)
Training Algorithm: TrueTime
Transfer Function Min: -1.200, Max: 1.200, Gain: 1.000
Number of Modules: 1
Module 1: Number Neurons: 2 Layer: 1
```

III.7.3 ENTRENAMIENTO DE LA RED NEURONAL.

El paso siguiente es entrenar la red neuronal.

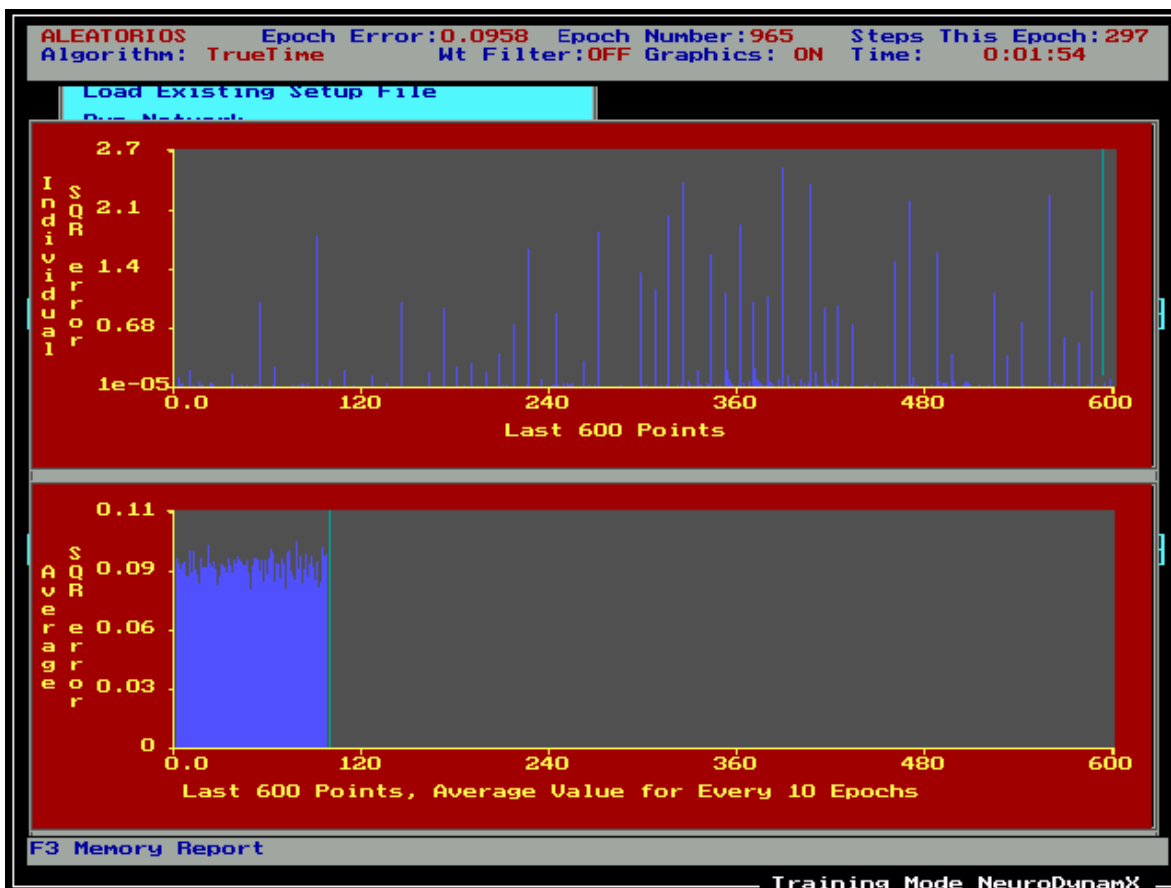
Del menú principal, seleccionar: "Run Network File/ Train Network /Start Training" para inicializar el proceso de entrenamiento.

La red empieza a realizar el proceso de entrenamiento el cual terminará al cumplirse 1000 épocas, que son las que automáticamente Dynamind asigna.

Una vez terminado el proceso se tomaron muestras de error que presentan algunas secuencias ó vectores antes y después de entrenar la red con la finalidad de comparar ó determinar el aprendizaje de la red neuronal.

No.	No. SEQ	No. PART		ANTES ENTRENAMIENTO	DESPUÉS ENTRENAMIENTO
1	SEQ 1	PART 5	ERROR	0.758	0.000465
2	SEQ 2	PART 6	ERROR	0.0687	0.000333
3	SEQ 5	PART 3	ERROR	0.0687	0.000333
4	SEQ 7	PART 1	ERROR	0.0227	0.129
5	SEQ 12	PART 4	ERROR	0.00807	0.000182
6	SEQ 21	PART 2	ERROR	0.185	0.000396
7	SEQ 29	PART 4	ERROR	0.00177	0.0000131
8	SEQ 34	PART 5	ERROR	0.00038	0.000191
9	SEQ 36	PART 5	ERROR	1.18	0.0151
10	SEQ 40	PART 4	ERROR	1.09	0.0169

En la siguiente imagen se observa el proceso de entrenamiento de la red neuronal.



Al finalizar el proceso de entrenamiento, los parámetros que presento al final son:

Epoch Number: **999**

Epoch Error: **0.0649**

Steps This Epoch: **297**

Time: **1:58**

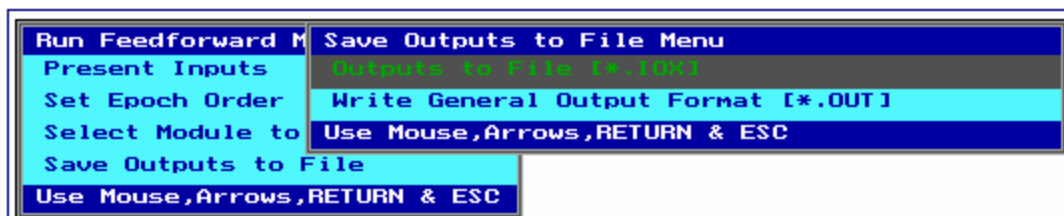
III.7.4 VERIFICACIÓN DE LA RED NEURONAL.

Para cargar el archivo de verificación **VRAND1.IO** a la red neuronal.

Del menú principal, seleccionar: *"Edit System Configuration / Select IO file"* y elegir el archivo correspondiente.

III.7.5 OBTENCIÓN DEL ARCHIVO DE RESULTADOS.

Una vez cargado este archivo de verificación al sistema, se procede a obtener el archivo de resultados que se denomina: **RAND1.OUT**. Este archivo de resultados se obtiene de la siguiente forma; del menú principal seleccionar: *"Run Feedforward / save outputs to file / write General output format *.out"*. Se elige la segunda opción y automáticamente DynaMind empieza a generar el archivo de resultados (ver imagen siguiente).



ANÁLISIS DE RESULTADOS DEL ARCHIVO DE VERIFICACIÓN.

Una vez que se obtienen los resultados tal como los presenta el sistema, se procede a analizarlos.

En la tabla 1 se muestran los datos reales y pronosticados por la red.

- **INTERPRETACIÓN DE RESULTADOS:**

Número	No. Aleatorio Real	No. Aleatorio Real	Número Pronosticado	Número Pronosticado
4	-0.7124	7982	9040.5	-0.666
5	0.7156	40547	40053.9	0.694
6	0.5855	37580	37203.4	0.569
7	-0.6007	10529	10956	-0.582
8	1	47032	44158.6	0.874
9	0.7253	40767	39962.7	0.69
18	-0.875	4275	5825.1	-0.807
19	-0.8053	5865	6760.1	-0.766
20	-0.7578	6947	7512.6	-0.733
23	0.0658	25729	25869.8	0.072
24	-0.5613	11429	11799.8	-0.545
25	0.8815	44329	42562.4	0.804
26	-0.7265	7660	8310.8	-0.698
27	-0.0382	23358	23680.7	-0.024
28	-0.4611	13714	13487.3	-0.471
35	-0.8274	5359	6760.1	-0.766
36	0.21	29017	29495.7	0.231
37	-0.7301	40878	39871.5	0.686
41	-1	1424	4502.5	-0.865
44	-0.6441	9540	9929.89	-0.627
45	-0.5996	10554	11480.5	-0.559
46	-0.7323	7528	8402	-0.694
47	-0.513	12529	12233	-0.526
48	0.1228	27029	27511.7	0.114
49	0.617	38299	37933.2	0.601

TABLA 1

De acuerdo a la tabla anterior y a la tabla 2 que son la base de datos original que a continuación se muestra, se tiene lo siguiente.

No.	NÚMERO ALEATORIO	NÚMERO NORMALIZADO	No.	NÚMERO ALEATORIO	NÚMERO NORMALIZADO
1	39440	0.6671	26	7660	-0.7265
2	24683	0.02	27	23358	-0.0382
3	29882	0.2479	28	13714	-0.4611
4	7982	-0.7124	29	9060	-0.6651
5	40547	0.7156	30	36029	0.5175
6	37580	0.5855	31	7699	-0.7248
7	10529	-0.6007	32	15799	-0.3696
8	47032	1.000	33	6977	-0.7565
9	40767	0.7253	34	31237	0.3074
10	2514	-0.9522	35	5359	-0.8274
11	41010	0.7359	36	29017	0.21
12	23495	-0.0321	37	40878	0.7301
13	35550	0.4965	38	18229	-0.2631
14	10548	-0.5999	39	7637	-0.7275
15	19445	-0.2097	40	46429	0.9736
16	5636	-0.8153	41	1424	-1.000
17	1829	-0.9822	42	40329	0.7061
18	4275	-0.875	43	46029	0.956
19	5865	-0.8053	44	9540	-0.6441
20	6947	-0.7578	45	10554	-0.5996
21	21429	-0.1227	46	7528	-0.7323
22	28779	0.1996	47	12529	-0.513
23	25729	0.0658	48	27029	0.1228
24	11429	-0.5613	49	38299	0.617
25	44329	0.8815	50	6557	-0.7749

TABLA 2

Análisis 1:

Partiendo de la tabla de la base de datos y siguiendo la secuencia se tiene que después de la posición **3**, el dato aleatorio es **29882**, el siguiente en la lista sería la posición **4** que le correspondería el dato aleatorio, **7982**.

Por otro lado, la red **RAND1.NET** pronosticó que en la posición **4**, de acuerdo a la tabla 1 le corresponde el dato aleatorio: **9040.5**

Valor muy alejado del real.

Análisis 2:

La Lotería Nacional presenta los últimos cuatro ganadores del Premio Mayor siendo estos:

Evento	No. Aleatorio
42	40329
43	46029
44	9540
45	10554

La red neuronal pronostica los dos últimos números:

Evento	No. Aleatorio
42	40329
43	46029
44	9929
45	11480

Análisis 3:

Los valores reales son:

Evento	No. Aleatorio
25	44329
26	7660
27	23358
28	13714

La red pronostica los dos últimos como:

Evento	No. Aleatorio
25	44329
26	7660
27	23680
28	13487

Valores muy parecidos a los reales.

Notas:

1. En la información que se presenta en la tabla 1, se encuentra los valores normalizados y no normalizados, tanto de los reales como los pronosticados.
2. Al efectuar el proceso inverso de normalización en la columna de "Número Pronosticado", obsérvese, que los resultados presentan décimas. Por lo que para efecto de comparación se toma la parte entera.
3. La ecuación que se utilizó para efectuar el proceso inverso de normalización es:

$$X_d = \left(\frac{X_s - TF_{\min}}{TF_{\max} - TF_{\min}} \right) (d_{\max} - d_{\min}) + d_{\min}$$

donde:

X_d: Dato a "desnormalizar"

X_s: Dato normalizado

d_{Max}: Valor Máximo del dato a escalar

d_{Min}: Valor Mínimo del dato a escalar

TF_{Max}: Valor de la Función de Transferencia Máximo

TF_{Min}: Valor de la Función de Transferencia Mínimo

III.8.1 ELABORACIÓN DE LA SEGUNDA RED NEURONAL (RAND2.NET)

Esta segunda red neuronal se denomina **RAND2.NET**. El archivo de salidas y entradas (IO), esta constituido ahora por doce vectores por secuencia (tres vectores más que en el caso anterior).

III.8.2 GENERACIÓN DEL ARCHIVO DE ENTRADAS Y SALIDAS (IO).

iobuild RANDOM.DAT RAND2.IO

IOBUILDER - Method 4 (True Time)

Input a one line description of the problem:

>>>**NÚMEROS ALEATORIOS**

Specify the length of a record

>>>**1**

Specify which fields in a record are inputs, e.g 1,8,3-6 [1 is the first element];

>>> **1**

Specify which fields in a record are outputs, e.g 1,8,3-6 [1 is the first element];

>>> **1**

Specify the total number of network neurons

>>>**2**

specify the number of records in sequence

>>> **12 (doce vectores por secuencia)**

How many records into the future do you want to predict

>>> **1**

Input the IO sequence label

>>>**SEQ**

Input the IO pair label

>>>**PART**

DynaMind procesa la información dando como resultado **38 secuencias**.

Generación de los archivos de Entrenamiento y Verificación.

Al contar con el archivo general de entradas y salidas, **RAND2.IO**, se construye el archivo de entrenamiento **ERAND2.IO** y el de verificación **VRAND2.IO**. Para este caso **33 secuencias** constituyen el **archivo de entrenamiento** y **5 secuencias** para el **archivo de verificación** siendo estas las secuencias:

Seq: 8

Seq: 13

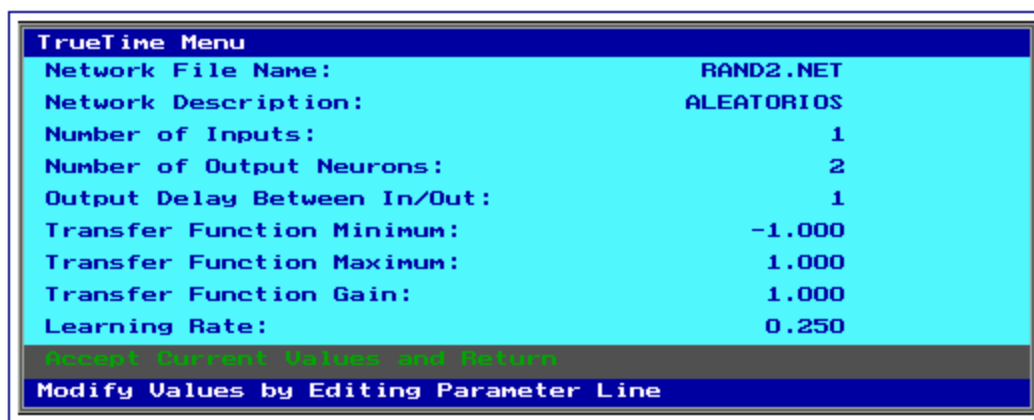
Seq: 18

Seq: 26

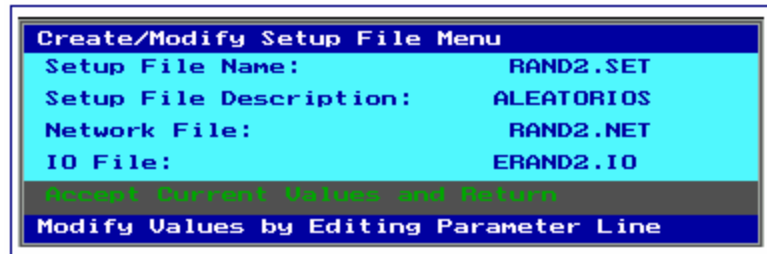
Seq: 38

ELABORACIÓN DE LA RED NEURONAL.

En el ambiente de desarrollo de DynaMind del menú principal, seleccionar: "*Edit system / Network file / Create a Network file / True Time*"; mostrándose la siguiente pantalla en la cual se dan de alta los parámetros que se presentan.



A continuación se genera el archivo setup file; para ello, del menú principal: "*Edit System Configuration Menu / Setpu File / Create Setup File*"; inmediatamente se



desplegará la imagen de lado derecho donde se habilitara el nombre del archivo, descripción del archivo, el archivo de la red y el archivo de entrenamiento.

Ahora se carga al sistema el archivo Setup file, que automáticamente carga la red (**RAND2.NET**) y finalmente el archivo de entrenamiento para efectuar el proceso de entrenamiento.

III.8.3 ENTRENAMIENTO DE LA RED.

Se procede a entrenar la red neuronal, pero primero se adquieren los errores de secuencias antes del entrenamiento y al finalizar este.

Presentando los siguientes resultados:

No.	No. SEQ	No. PART		ANTES ENTRENAMIENTO	DESPUÉS ENTRENAMIENTO
1	SEQ 1	PART 5	ERROR	0.264	0.00296
2	SEQ 1	PART 12	ERROR	0.0549	0.0125
3	SEQ 2	PART 6	ERROR	0.645	0.00163
4	SEQ 5	PART 3	ERROR	0.645	0.00163
5	SEQ 5	PART 12	ERROR	1.03	0.00505
6	SEQ 9	PART 1	ERROR	1.01	1.91
7	SEQ 12	PART 11	ERROR	0.00000663	0.000695
8	SEQ 19	PART 9	ERROR	0.0555	0.000331
9	SEQ 23	PART 8	ERROR	0.102	0.00191
10	SEQ 27	PART 2	ERROR	0.44	0.00127
11	SEQ 31	PART 6	ERROR	0.000159	0.000431

Epoch Number: **1999**

Epoch Error: **0.0655**

Steps This Epoch: **396**

Time: **5:14**

Hay que recordar que las únicas variables que se modificaron para esta red neuronal fueron: El Máximo y Mínimo de la Función de Transferencia y el número de épocas, estableciéndose a 2000.

VERIFICACIÓN DE LA RED NEURONAL

Para evaluar el aprendizaje de la red neuronal se procede a cargar el archivo de verificación (**VRAND2.IO**) a la red neuronal para adquirir posteriormente el archivo de resultados que se denomina **RAND2.OUT**.

III.8.4 ANÁLISIS DE RESULTADOS DEL ARCHIVO DE VERIFICACIÓN.

Una vez que se obtienen los resultados tal como los presenta el sistema, se procede a analizarlos.

- **INTERPRETACIÓN DE RESULTADOS:**

EVENTO	No. ALEATORIO REAL	No. ALEATORIO REAL	NÚMERO PRONOSTICADO	NÚMERO PRONOSTICADO
8	1	47032	41536.2	0.759
9	0.7253	40767	37864.7	0.598
13	0.4965	35550	35219.5	0.482
15	-0.2097	19445	19279.5	-0.217
18	-0.875	4275	5619.9	-0.816
19	-0.8053	5865	6440.8	-0.78
20	-0.7578	6947	7102.1	-0.751
26	-0.7265	7660	7626.6	-0.728
27	-0.0382	23358	23771.9	-0.02
28	-0.4611	13714	12552.3	-0.512
35	-0.8274	5359	6851.3	-0.762
36	0.21	29017	29495.7	0.231
37	0.7301	40878	38503.3	0.626
41	-1	1424	5004.2	-0.843
44	-0.6441	9540	8652.8	-0.683
45	-0.5996	10554	11275.3	-0.568
46	-0.7323	7528	8014.3	-0.711
47	-0.513	12529	11275.3	-0.568
48	0.1228	27029	25938.3	0.075
49	0.617	38299	36884.2	0.555

TABLA 2

En la tabla 2 se muestran los datos reales y pronosticados por la red.

De acuerdo a la tabla 2 y a la tabla original que a continuación se muestra, se tiene lo siguiente.

No.	NÚMERO ALEATORIO	NÚMERO NORMALIZADO	No.	NÚMERO ALEATORIO	NÚMERO NORMALIZADO
1	39440	0.6671	26	7660	-0.7265
2	24683	0.02	27	23358	-0.0382
3	29882	0.2479	28	13714	-0.4611
4	7982	-0.7124	29	9060	-0.6651
5	40547	0.7156	30	36029	0.5175
6	37580	0.5855	31	7699	-0.7248
7	10529	-0.6007	32	15799	-0.3696
8	47032	1.000	33	6977	-0.7565
9	40767	0.7253	34	31237	0.3074
10	2514	-0.9522	35	5359	-0.8274
11	41010	0.7359	36	29017	0.21
12	23495	-0.0321	37	40878	0.7301
13	35550	0.4965	38	18229	-0.2631
14	10548	-0.5999	39	7637	-0.7275
15	19445	-0.2097	40	46429	0.9736
16	5636	-0.8153	41	1424	-1.000
17	1829	-0.9822	42	40329	0.7061
18	4275	-0.875	43	46029	0.956
19	5865	-0.8053	44	9540	-0.6441
20	6947	-0.7578	45	10554	-0.5996
21	21429	-0.1227	46	7528	-0.7323
22	28779	0.1996	47	12529	-0.513
23	25729	0.0658	48	27029	0.1228
24	11429	-0.5613	49	38299	0.617
25	44329	0.8815	50	6557	-0.7749

Análisis 1:

Los valores correspondientes reales son:

EVENTO	No. ALEATORIO
6	37580
7	10529
8	47032
9	40767

La red pronostica los dos últimos valores como:

EVENTO	No. ALEATORIO
6	37580
7	10529
8	41536
9	37864

Valores muy alejados de los reales para este caso.

Análisis 2:

Los valores reales son:

EVENTO	No. ALEATORIO
24	11429
25	44329
26	7660
27	23358

La red pronostica los dos últimos como:

EVENTO	No. ALEATORIO
24	11429
25	44329
26	7626
27	23771

Valores muy cercanos a los reales.

Notas:

1. La información que se presenta en la tabla 1, se encuentran los valores normalizados y no normalizados, tanto de los reales como los pronosticados.
2. Al efectuar el proceso inverso de normalización en la columna de "Número Pronosticado", obsérvese, que los resultados presentan décimas. Por lo que para efecto de comparación se toma la parte entera.

III.8.5 OBSERVACIONES:

Comparando los resultados de verificación de ambas redes neuronales, se observa que la segunda red en realidad no le es de mucha ayuda al modificar el Máximo y Mínimo de la función de transferencia y ni el incremento de número de vectores por secuencia, así como también el incremento del número de épocas en el entrenamiento.

Se tiene que al comparar esta segunda red con la primera, esta última obtuvo iguales o resultados semejantes a la segunda empleando menos tiempo en el aprendizaje.

Se puede afirmar entonces que ambas redes obtuvieron un aprendizaje "aceptable".

Por lo que se generará una tercera red neuronal; en la cual se modificaron tres parámetros (La ganancia de la función de transferencia, el número de épocas en el aprendizaje y el número de vectores por secuencia).

El fin que se pretende es encontrar una tercera red más eficiente que las anteriores.

Una vez que se determine cual de las tres es la efectiva, entonces se procederá a efectuar una aplicación donde la red pronostique el próximo número ganador de la lotería nacional.

III.9.1 ELABORACIÓN DE LA TERCERA RED NEURONAL (RAND3.NET)

Esta tercera red neuronal se denomina **RAND3.NET**. El archivo de salidas y entradas (IO), esta constituido por nueve vectores por secuencia.

III.9.2 GENERACIÓN DEL ARCHIVO DE ENTRADAS Y SALIDAS (IO).

iobuild RANDOM.DAT RAND3.IO

IOBUILDer - Method 4 (True Time)

Input a one line description of the problem:

>>>**NÚMEROS ALEATORIOS**

Specify the length of a record

>>>**1**

Specify which fields in a record are inputs, e.g 1,8,3-6 [1 is the first element];

>>> **1**

Specify which fields in a record are outputs, e.g 1,8,3-6 [1 is the first element];

>>> **1**

Specify the total number of network neurons

>>>**2**

specify the number of records in sequence

>>> **6**

How many records into the future do you want to predict

>>> **1**

Input the IO sequence label

>>>**SEQ**

Input the IO pair label

>>>**PART**

DynaMind procesa la información dando como resultado **44 secuencias**.

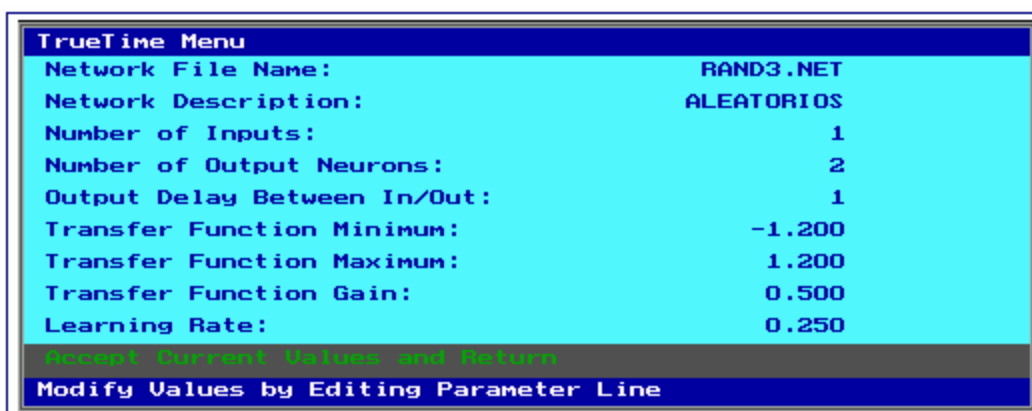
Generación del archivo de Entrenamiento y Verificación.

Al contar con el archivo general de entradas y salidas, **RAND3.IO**, se construye el archivo de entrenamiento **ERAND3.IO** y el de verificación **VRAND3.IO**. Para este caso **35 secuencias** constituyen el **archivo de entrenamiento** y **9 secuencias** para el **archivo de verificación**; siendo estas:

Seq: 3
Seq: 8
Seq: 14
Seq: 18
Seq: 22
Seq: 26
Seq: 35
Seq: 38
Seq: 44

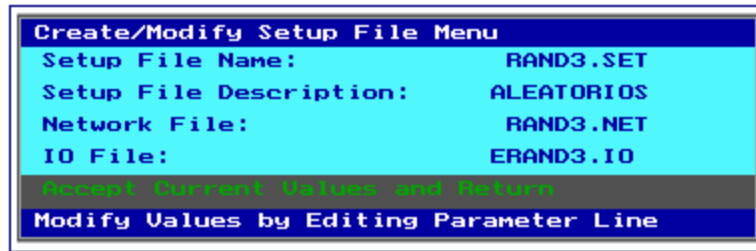
ELABORACIÓN DE LA RED NEURONAL.

En el ambiente de desarrollo de DynaMind del menú principal, seleccionar: "*Edit system / Network file / Create a Network file / True Time*"; mostrándose la siguiente pantalla en la cual se dan de alta los parámetros que se presentan.



Obsérvese la ganancia de la función de transferencia que para esta red es la mitad que tienen las dos redes anteriores.

Generando el archivo **setup file**; del menú principal: *"Edit System Configuration Menu / Setpu File / Create Setup File"*.



Se procede a cargar el archivo Setup file, al mismo tiempo la red, y finalmente se selecciona el archivo de entrenamiento para efectuar el proceso de aprendizaje.

III.9.3 ENTRENAMIENTO DE LA RED NEURONAL.

Se procede a entrenar la red neuronal con **1500 épocas**; pero antes se adquieren una vez más los errores de secuencias antes del entrenamiento y al finalizar este.

Presentando los siguientes resultados:

No.	No. SEQ	No. PART		ANTES ENTRENAMIENTO	DESPUÉS ENTRENAMIENTO
1	SEQ 1	PART 5	ERROR	0.708	0.0112
2	SEQ 2	PART 6	ERROR	0.183	0.000021
3	SEQ 4	PART 5	ERROR	1.25	0.0441
4	SEQ 9	PART 1	ERROR	0.387	0.0108
5	SEQ 12	PART 4	ERROR	0.00237	0.00136
6	SEQ 21	PART 5	ERROR	1	0.0322
7	SEQ 27	PART 2	ERROR	0.0865	0.00132
8	SEQ 31	PART 6	ERROR	0.125	0.00019
9	SEQ 36	PART 5	ERROR	1.2	0.036
10	SEQ 40	PART 5	ERROR	1.16	0.0944

Epoch Error: **0.1174**
 Steps This Epoch: **210**
 Epoch Number: **1499**
 Time: **2:19**

VERIFICACIÓN DE LA RED NEURONAL.

Para evaluar el aprendizaje de la red neuronal se procede a cargar el archivo de verificación **VRAND3.IO**, a la red para adquirir así el archivo de resultados **RAND3.OUT**.

III.9.4 ANÁLISIS DE RESULTADOS DEL ARCHIVO DE VERIFICACIÓN.

Una vez que se obtienen los resultados tal como los presenta el sistema, se procede a analizar.

En la siguiente tabla se muestra algunos de los resultados.

- **INTERPRETACIÓN DE RESULTADOS:**

Número	No. Aleatorio Real	No. Aleatorio Real	Número Pronosticado	Número Pronosticado
4	-0.7124	7982	9336.9	-0.653
5	0.7156	40547	38024.4	0.605
6	0.5855	37580	32825.1	0.377
7	-0.6007	10529	10636.8	-0.596
8	1.000	47032	42243.1	0.79
18	-0.875	4275	5163.8	-0.836
19	-0.8053	5865	5984.8	-0.8
20	-0.7578	6947	6372.4	-0.783
24	-0.5613	11429	12233	-0.526
25	0.8815	44329	40236.4	0.702
26	-0.7265	7660	7854.72	-0.718
27	-0.0382	23358	23133.4	-0.048
35	-0.8274	5359	9245.7	-0.657
36	0.21	29017	25733	0.066
37	0.7301	40878	36542.1	0.54
41	-1.000	1424	4570.9	-0.862
44	-0.6441	9540	9245.7	-0.657
45	-0.5996	10554	13259.2	-0.481
46	-0.7323	7528	8858.1	-0.674
47	-0.513	12529	12233	-0.526
48	0.1228	27029	24501.6	0.012
49	0.617	38299	34854.6	0.466

Tabla 3

De acuerdo a la tabla 3 y a la tabla original de base de datos que a continuación se muestra, se tiene lo siguiente.

No.	NÚMERO ALEATORIO	NÚMERO NORMALIZADO	No.	NÚMERO ALEATORIO	NÚMERO NORMALIZADO
1	39440	0.6671	26	7660	-0.7265
2	24683	0.02	27	23358	-0.0382
3	29882	0.2479	28	13714	-0.4611
4	7982	-0.7124	29	9060	-0.6651
5	40547	0.7156	30	36029	0.5175
6	37580	0.5855	31	7699	-0.7248
7	10529	-0.6007	32	15799	-0.3696
8	47032	1.000	33	6977	-0.7565
9	40767	0.7253	34	31237	0.3074
10	2514	-0.9522	35	5359	-0.8274
11	41010	0.7359	36	29017	0.21
12	23495	-0.0321	37	40878	0.7301
13	35550	0.4965	38	18229	-0.2631
14	10548	-0.5999	39	7637	-0.7275
15	19445	-0.2097	40	46429	0.9736
16	5636	-0.8153	41	1424	-1.000
17	1829	-0.9822	42	40329	0.7061
18	4275	-0.875	43	46029	0.956
19	5865	-0.8053	44	9540	-0.6441
20	6947	-0.7578	45	10554	-0.5996
21	21429	-0.1227	46	7528	-0.7323
22	28779	0.1996	47	12529	-0.513
23	25729	0.0658	48	27029	0.1228
24	11429	-0.5613	49	38299	0.617
25	44329	0.8815	50	6557	-0.7749

Análisis 1:

Los valores correspondientes reales son:

EVENTO	No. ALEATORIO
3	29882
4	7982
5	40547
6	37580

Los dos últimos pronosticados por la red son:

EVENTO	No. ALEATORIO
3	29882
4	7982
5	38024
6	32825

Valores relativamente cercanos a los reales.

Análisis 2:

Los valores reales son:

EVENTO	No. ALEATORIO
25	44329
26	7660
27	23358
28	13714

Y los pronosticados son:

EVENTO	No. ALEATORIO
25	44329
26	7660
27	23133
28	13259

Valores muy cercanos a los reales.

III.9.5 COMENTARIOS GENERALES:

Observando los resultados anteriores que presenta cada una de las redes neuronales, se piensa en realidad que ninguna de ellas le beneficio la modificación de los parámetros, es decir, el máximo y mínimo de la función de transferencia así como la ganancia de la misma, el número de vectores por secuencia y el número de épocas en cada aprendizaje; pues las tres presentan casos en donde sus predicciones son muy cercanas a las reales y otras muy alejadas, en especial en el punto máximo y mínimo de los datos normalizados (± 1.00). Es por eso que no

se sabe a ciencia cierta cual de las tres es la mas " eficiente", afirmando con esto de que no se llegó al objetivo inicial que era el de encontrar una red eficiente.

Por lo que para encontrar cual de las tres es la mejor, se procedió a obtener el error absoluto y por ende el error promedio absoluto que presentan cada una de las redes neuronales al compararlas en el mismo número ó punto de predicción.

Dando como resultado final que red **RAND1.NET** es la mas "efectiva" pues presenta un error promedio menor que **RAND2.NET** y **RAND3.NET** tal como se muestran a continuación y además, de que rand1.net empleo menor tiempo de aprendizaje que las dos ultimas.

Características de las tres redes neuronales.

RAND1.NET

EVENTO	No. ALEATORIO REAL	NÚMERO PRONOSTICADO	ERROR (Real-Pron)	ERROR PROMEDIO
18	4275	5825.1	1550.1	846.918125
19	5865	6760.1	895.1	
20	6947	7512.6	565.6	
26	7660	8310.8	650.8	
27	23358	23680.7	322.7	
28	13714	13487.3	266.7	
35	5359	6760.1	1401.1	
36	29017	29495.7	478.7	
37	40878	39871.5	1006.5	
41	1424	4502.5	3078.5	
44	9540	9929.89	389.89	
45	10554	11480.5	926.5	
46	7528	8402	874	
47	12529	12233	296	
48	27029	27511.7	482.7	
49	38299	37933.2	365.8	

RAND2.NET

EVENTO	No. ALEATORIO REAL	NÚMERO PRONOSTICADO	ERROR (Real- Pron)	ERROR PROMEDIO
18	4275	5619.9	1344.9	1091.54375
19	5865	6440.8	575.8	
20	6947	7102.1	155.1	
26	7660	7626.6	33.4	
27	23358	23771.9	413.9	
28	13714	12552.3	1161.7	
35	5359	6851.3	1492.3	
36	29017	29495.7	478.7	
37	40878	38503.3	2374.7	
41	1424	5004.2	3580.2	
44	9540	8652.8	887.2	
45	10554	11275.3	721.3	
46	7528	8014.3	486.3	
47	12529	11275.3	1253.7	
48	27029	25938.3	1090.7	
49	38299	36884.2	1414.8	

RAND3.NET

EVENTO	No. ALEATORIO REAL	NÚMERO PRONOSTICADO	ERROR (Real -Pron)	ERROR PROMEDIO
18	4275	5163.8	888.8	1816.894667
19	5865	5984.8	119.8	
20	6947	6372.4	574.6	
26	7660	7854.72	194.72	
27	23358	23133.4	224.6	
35	5359	9245.7	3886.7	
36	29017	25733	3284	
37	40878	36542.1	4335.9	
41	1424	4570.9	3146.9	
44	9540	9245.7	294.3	
45	10554	13259.2	2705.2	
46	7528	8858.1	1330.1	
47	12529	12233	296	
48	27029	24501.6	2527.4	
49	38299	34854.6	3444.4	

III.10 UTILIZANDO LA PRIMERA RED NEURONAL.

III.10.1 APLICACIÓN I

Una vez que se ha comprobado el aprendizaje adecuado de la red neuronal, el siguiente y último paso es **utilizar la red**.

En este caso la idea es predecir ó conocer qué también predice la red neuronal los números ganadores de lotería nacional del mes de abril del presente año.

Se escogieron 17 números ganadores que serán los que conformen nuestra base de datos.

Ejercicio 1.

A continuación se muestra la base de datos real y normalizada de los 17 números.

EVENTO	GANADOR	NORMALIZADO
1	27388	0.1385
2	46086	0.9585
3	7416	-0.7372
4	25186	0.042
5	5386	-0.8262
6	33286	0.3972
7	23377	-0.0373
8	21237	-0.1311
9	1912	-0.9786
10	9040	-0.666
11	43009	0.8235
12	32190	0.3491
13	28502	0.1874
14	2341	-0.9609
15	37027	0.5612
16	3918	-0.8906
17	19021	-0.2283

Una vez generado el archivo de entradas y salidas **LOTERIA.IO** se procedió a cargar este archivo a red **RAND1.NET**.

Presentando los siguientes resultados.

Análisis de resultados:

En las siguientes tablas se muestran los resultados por la red; donde:

Primera columna:

- **LOTERIA.IO**

Descripción:

- Es el archivo generado por DynaMind para predecir un solo vector (un número aleatorio).

Segunda columna:

- **LOTERIA.OUT**

Descripción:

- Archivo de resultados generado por DynaMind que presenta desde una hasta ocho salidas para un solo vector (número pronosticado).

Notas:

1. Los números que se encuentran en “negritas” en las distintas salidas son los que se seleccionaron para generar la columna de **números pronosticados**.
2. Las diversas salidas se producen por la misma estructuración del archivo IO ó del número de vectores por secuencia.

Tercera columna:

- **PROMEDIO.OUT**

Descripción:

- Es el promedio de cada una de las salidas del archivo de resultados generado por DynaMind.

LOTERIA.IO			LOTERIA.OUT								PROMEDIO
No.	IN	OUT	OUT1	OUT2	OUT3	OUT4	OUT5	OUT6	OUT7	OUT8	OUT
1	0.1385	0.9585	0.131								0.131
2	0.9585	-0.7372	0.842	0.807							0.8245
3	-0.7372	0.042	-0.721	-0.733	-0.766						-0.727
4	0.042	-0.8262	0.00	-0.006	-0.048	0.012					-0.003
5	-0.8262	0.3972	-0.793	-0.793	-0.8	-0.78	-0.78				-0.793
6	0.3972	-0.0373	0.355	0.355	0.355	0.361	0.366	0.409			0.355
7	-0.0373	-0.1311	-0.078	-0.078	-0.874	-0.078	-0.078	-0.054	-0.114		-0.078
8	-0.1311	-0.9786	-0.078	-0.167	-0.167	-0.167	-0.167	-0.161	-0.196	-0.179	-0.1225
9	-0.9786	-0.666	-0.874	-0.874	-0.874	-0.874	-0.874	-0.874	-0.879	-0.874	-0.874
10	-0.666	0.8235	-0.714	-0.714	-0.714	-0.714	-0.718	-0.714	-0.714		-0.714
11	0.8235	0.3491	0.714	0.714	0.714	0.714	0.714	0.714			0.714
12	0.3491	0.1874	0.344	0.344	0.344	0.344	0.344				0.344
13	0.1874	-0.9609	0.19	0.19	0.19	0.19					0.19
14	-0.9609	0.5612	-0.86	-0.86	-0.86						-0.86
15	0.5612	-0.8906	0.511	0.511							0.511
16	-0.8906	-0.2283	-0.832								-0.832

En la siguiente tabla se muestra los valores reales y pronosticados por la red neuronal; así como el error y error promedio.

Primera columna:

- **NÚMERO (No.)**

Descripción:

- Son los diecisiete números a predecir.

Primera segunda:

- **NÚMEROS REALES**

Descripción:

- Son los números que conforman nuestra base de datos original (LOTERIA. DAT) y los que la red debió de pronosticar.

Tercera columna:

- **NÚMEROS PRONOSTICADOS**

Descripción:

- Son los números pronosticados por la red una vez que han pasado por el proceso inverso de la normalización.

Cuarta columna:

- **ERROR**

Descripción:

- Es el error presentado entre los números reales y los números pronosticados (Número real — Número pronosticado).

Quinta columna:

- **ERROR PROMEDIO**

Descripción:

- Es el promedio del error presentado de los números reales y números pronosticados.

Sexta columna:

- **NÚMEROS PRONOSTICADOS PROMEDIO**

Descripción:

- Es el promedio de las "n" salidas del archivo **LOTERIA.OUT** que presenta la red en cada uno de los 16 números.

Séptima columna:

- **ERROR**

Descripción:

- Es el error presentado por la diferencia existente entre los números reales y números pronosticados promedios.

Octava columna:

- **ERROR PROMEDIO**

Descripción:

- Es el error promedio del error presentado por la diferencia existente entre los números reales y números pronosticados promedios.

No.	NUMEROS REALES	NUMEROS PROSNOTICADOS	ERROR	ERROR PROMEDIO	NUMEROS PROSNOTICADOS	ERROR	ERROR PROMEDIO
				412	PROMEDIO		235.25
1	27388	27215	173		27215	173	
2	46086	42630	3456		43029	3057	
3	7416	7786	-370		7649	-233	
4	25186	23133	2053		24159	1027	
5	5386	6440	-1054		6144	-758	
6	33286	32323	963		32323	963	
7	23377	21628	1749		22449	928	
8	21237	20419	818		21434	-197	
9	1912	4297	-2385		4297	-2385	
10	9040	7945	1095		7945	1095	
11	43009	40510	2499		40510	2499	
12	32190	32072	118		32072	118	
13	28502	28560	-58		28560	-58	
14	2341	4616	-2275		4616	-2275	
15	37027	35880	1147		35880	1147	
16	3918	5255	-1337		5255	-1337	

Comentarios.

Las predicciones elaboradas por **RAND1.NET** son buenos resultados; ya que presento predicciones muy cercanas a las reales. Existen dos predicciones que en realidad se encuentran totalmente "desfasadas" (**No. 9 y No. 14**), esto se debe a que el archivo de entrenamiento (**ERAND1.IO**) de la red no presento con frecuencia números aleatorios cercanos al valor mínimo de la base de datos (**1424 = -1.00**); ocasionando que la red al no tener registrado en los pesos de las neuronas, aquellos valores mínimos, no pronosticará eficientemente.

Por otro lado hay que recordar que el archivo **LOTERIA.IO** no pertenece al archivo de entrenamiento y al de verificación, la única condición fue que estuviese dentro del rango de normalización o bien dentro del rango del máximo y mínimo del valor real [47032, 1424].

III.10.2 APLICACIÓN II

Objetivo: Elaborar una red que sea capaz de predecir el siguiente evento ganador de la lotería nacional.

Introducción:

Con la anterior experiencia, se genero una red neuronal con características similares a la red **RAND1.NET** con la finalidad de predecir el siguiente evento numérico de la lotería nacional, la única diferencia existente entre esta nueva red y rand1.net es la base de datos.

Esta nueva red será entrenada con una base de datos de 300 números, permitiendo con esto que la red tenga mayor información en sus pesos y que los resultados sean mas cercanos a los reales.

DESARROLLO:

La información que conforman a la base numérica, se encuentran en forma de punto flotante, por lo que hay que efectuar el proceso de normalización con la formula ya conocida.

Recordando que para crear una red neuronal con DynaMind son cinco los pasos:

1. Preparación del archivo de entrenamiento (Base de datos)
2. Creación de la red neuronal
3. Entrenamiento de la red neuronal
4. Verificación de la red neuronal
5. Utilización de la red neuronal

PREPARACIÓN DEL ARCHIVO DE ENTRENAMIENTO.

En realidad este es uno de los pasos mas importantes y sobre todo el mas laborioso en el sentido de convertir la información real a una información aceptable para el sistema, por lo cual se emplea uno de los métodos que proporciona el mismo sistema, para este caso se empleará el **Escala Máxima / Mínima.**

En las siguientes tablas se observan los 300 números aleatorios reales y normalizados, empleando la fórmula descrita al inicio de este capítulo.

INFORMACIÓN DE LA BASE DE DATOS

EVENTO	NÚMERO REAL	NÚMERO NORMALIZADO	NÚMERO REAL	NÚMERO NORMALIZADO
1	34017	0.3691	19445	-0.2202
2	41713	0.6804	36598	0.3553
3	32467	0.3065	46903	0.8903
4	43430	0.7498	12959	-0.4825
5	39440	0.5885	41103	0.6557
6	45948	0.8517	14247	-0.4304
7	30450	0.2249	5260	-0.7939
8	29119	0.1711	29041	0.1679
9	24683	-0.0084	41402	0.6678
10	10578	-0.5788	15399	-0.3838
11	44316	0.7857	37058	0.4921
12	9298	-0.6306	25949	0.0428
13	29882	0.2019	41173	0.6585
14	34733	0.3981	7835	-0.6897
15	32090	0.2912	21305	-0.1450
16	41591	0.6754	20914	-0.1608
17	7982	-0.6838	29287	0.1778
18	15058	-0.3976	900	-0.9702
19	33200	0.3361	44112	0.7774
20	12576	-0.4980	16605	-0.3350
21	40547	0.6332	3460	-0.8667
22	34241	0.3782	31225	0.2562
23	11403	-0.5454	30801	0.2391
24	27975	0.1248	11204	-0.5535
25	37580	0.5132	9040	-0.6410
26	45979	0.8529	30448	0.2248
27	41540	0.6734	8609	-0.6584
28	30448	0.2248	20161	-0.1912
29	10529	-0.5808	29104	0.1704
30	23750	-0.0461	15710	-0.3712
31	43017	0.7331	41854	0.6861
32	9730	-0.6131	45620	0.8384
33	47037	0.8957	45681	0.8409
34	37545	0.5118	20747	-0.1675
35	33342	0.3418	9816	-0.6096
36	48093	0.9384	2476	-0.9065
37	40767	0.6421	41456	0.6700
38	19591	-0.2143	45217	0.8221
39	24054	-0.0338	36674	0.4766
40	37389	0.5055	42363	0.7067

EVENTO	NÚMERO		NÚMERO	
	REAL	NORMALIZADO	REAL	NORMALIZADO
1	2514	-0.9049	31563	0.2699
2	19261	-0.2276	4859	-0.8101
3	41010	0.6520	23316	-0.0636
4	11735	-0.5320	43715	0.7613
5	23461	-0.0578	32691	0.3155
6	27927	0.1228	6303	-0.7517
7	18817	-0.2456	19414	-0.2214
8	46670	0.8809	38041	0.5319
9	23495	-0.0564	1436	-0.9485
10	41257	0.6619	26746	0.0751
11	19797	-0.2060	43455	0.7508
12	45789	0.8452	47610	0.9189
13	35550	0.4311	44240	0.7826
14	28828	0.1593	27554	0.1078
15	24179	-0.0287	9969	-0.6034
16	482	-0.9871	18958	-0.2399
17	10548	-0.5800	29985	0.2061
18	30636	0.2324	19178	-0.2310
19	9580	-0.6192	45965	0.8523
20	22483	-0.0973	32341	0.3014
21	20695	-0.1696	31666	0.2741
22	33676	0.3553	22586	0.0932
23	27860	0.1201	24294	-0.0241
24	11074	-0.5587	3978	-0.8457
25	48338	0.9483	2116	-0.9210
26	30934	0.2445	32577	0.3109
27	20927	-0.1603	12314	-0.5086
28	42447	0.7101	23795	-0.0443
29	44620	0.7979	49616	1.0000
30	30060	0.2091	1387	-0.9505
31	22012	-0.1164	17164	-0.3124
32	12681	-0.4937	25997	0.0448
33	11170	-0.5549	37272	0.5008
34	27326	0.0985	25943	0.0426
35	43751	0.7628	48498	0.9548
36	4792	-0.8128	17825	-0.2857
37	15557	-0.3774	163	-1.0000
38	43699	0.7607	34698	0.3967
39	2403	-0.9094	17556	-0.2966
40	7509	-0.7029	2518	-0.9048

EVENTO	NÚMERO	NÚMERO	NÚMERO	NÚMERO
	REAL	NORMALIZADO	REAL	NORMALIZADO
1	17051	-0.3170	30563	0.2295
2	43751	0.7628	586	-0.9829
3	45740	0.8432	2181	-0.9184
4	35516	0.4298	3641	-0.8593
5	13835	-0.4471	45584	0.8369
6	25848	0.0388	28556	0.1483
7	26361	0.0595	26290	0.0566
8	27698	0.1136	18786	-0.2468
9	18163	-0.2720	25527	0.0258
10	35078	0.4120	12410	-0.5047
11	47984	0.9340	5936	-0.7665
12	16895	-0.3233	38787	0.5620
13	33260	0.3385	10724	-0.5729
14	44293	0.7847	4418	-0.8279
15	33200	0.3361	39521	0.5917
16	16658	-0.3329	22517	-0.0959
17	15406	-0.3835	2865	-0.8907
18	45266	0.8241	19712	-0.2094
19	11688	-0.5339	9586	-0.6189
20	23070	-0.0736	794	-0.9745
21	7972	-0.6842	34686	0.3962
22	14644	-0.4144	17686	-0.2913
23	7925	-0.6861	15566	-0.3771
24	33956	0.3667	20455	-0.1793
25	24271	-0.0250	5008	-0.8041
26	36314	0.4620	46343	0.8676
27	33976	0.3675	37287	0.5014
28	15569	-0.3769	30229	0.2159
29	4111	-0.8403	43529	0.7538
30	45961	0.8522	929	-0.9690
31	7001	-0.7235	10575	-0.5789
32	5646	-0.7783	45613	0.8381
33	40730	0.6406	17329	-0.3058
34	12338	-0.5076	36429	0.4667
35	9626	-0.6173	46756	0.8843
36	36942	0.4874	32232	0.2969
37	26480	0.0643	27529	0.1067
38	1929	-0.9286	6971	-0.7247
39	10880	-0.5666	22629	-0.0914
40	27264	0.0960	20329	-0.1844

EVENTO	NÚMERO	NÚMERO	NÚMERO	NÚMERO
	REAL	NORMALIZADO	REAL	NORMALIZADO
1	13729	-0.4514	8250	-0.6729
2	7633	-0.6979	2917	-0.8886
3	40961	0.6500	7758	-0.6928
4	6988	-0.7240	5276	-0.7932
5	3204	-0.8770	9456	-0.6242
6	26595	0.0690	301	-0.9944
7	3321	-0.8723	44392	0.7887
8	26558	0.0675	25025	0.0055
9	3629	-0.8598	4395	-0.8288
10	4529	-0.8234	1609	-0.9415
11	18729	-0.2491	46245	0.8637
12	13725	-0.4515	31335	0.2607
13	41143	0.6573	3039	-0.8837
14	31623	0.2723	43935	0.7702
15	20320	-0.1848	4975	-0.8054
16	22826	-0.0835	25229	0.0137
17	12790	-0.4893	1429	-0.9488
18	40029	0.6123	5325	-0.7912
19	3543	-0.8633	26924	0.0823
20	4471	-0.8258	40129	0.6163
21	31256	0.2575		
22	13045	-0.4790		
23	46778	0.8852		
24	8771	-0.6519		
25	34893	0.4046		
26	26806	0.0775		
27	9700	-0.6143		
28	26508	0.0655		
29	36508	0.4699		
30	15779	-0.3685		
31	19279	-0.2269		
32	14750	-0.4101		
33	7701	-0.6951		
34	3039	-0.8837		
35	6928	-0.7264		
36	14793	-0.4083		
37	9572	-0.6195		
38	30814	0.2396		
39	4341	-0.8310		
40	26257	0.0553		

Generación del archivo IO.

Una vez normalizada la información, se procede a generar el archivo *.dat (en el editor de Ms-Dos, en la ruta: **c: / NDX/EXAMPLES**) que en este caso se denomina **NACIONAL.DAT**; posteriormente se procede a generar el archivo de entradas y salidas **NACIONAL.IO** como se muestra a continuación:

Desde Ms-Dos y en la ruta: **c: /ndx/examples**, teclear la siguiente instrucción:

Iobuild NACIONAL.DAT NACIONAL.IO

Presentándose la siguiente imagen.

```
IOBUILDer Version 2.0
[C] 1992 NeuroDynamax,Inc.
Please specify the format for your ASCII file
By choosing a number of the list below.
FEEDFORWARD NETWORKS:
    [1] Inputs followed by outputs
    [2] Inputs and outputs locations in a record to be specified
    [3] Inputs only
RECURRENT NETWORKS (True Time):
    [4] Single or multiple data streams
```

Se selecciona el método cuatro (True Time) y se completa la información pertinente, tal como se muestra a continuación.

>>> **4** (Selección del método cuatro)

IOBUILDer - Method 4 (True Time)

Input a one line description of the problem:

>>>**LOTERIA NACIONAL**

Specify the length of a record

>>>**1 (1 x 300)**

Specify which fields in a record are inputs, e.g 1,8,3-6 [1 is the first element];

>>> **1 (entrada es igual al primer campo)**

Specify which fields in a record are outputs, e.g 1,8,3-6 [1 is the first element];

>>> **1 (las mismas entradas son salidas)**

Specify the total number of network neurons

>>>**2**

specify the number of records in sequence

>>>**9 (nueve vectores por secuencia)**

How many records into the future do you want to predict

>>> **1**

Input the IO sequence label

>>>**SEQ**

Input the IO pair label

>>>**PART**

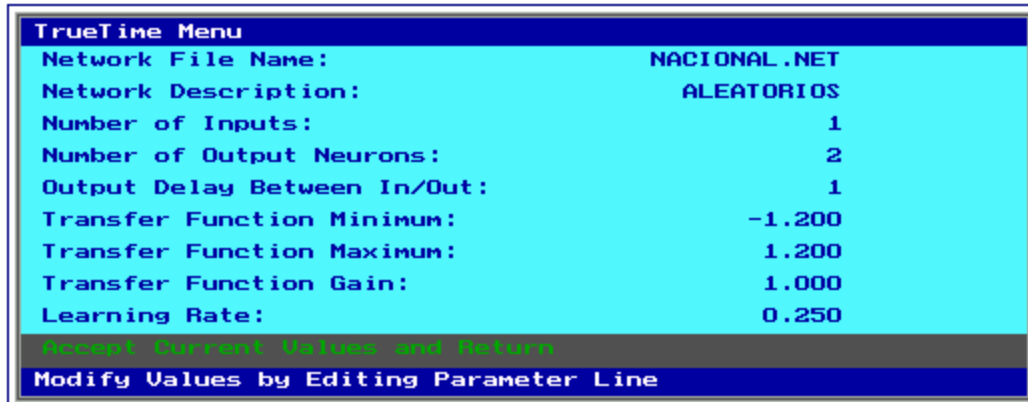
DynaMind empieza a procesar el archivo de salidas y entradas, presentando **291 secuencias.**

GENERACIÓN DE LA RED NEURONAL.

Para la creación de la red neuronal, dentro del sistema se establecen los parámetros necesarios para la red neuronal denominada **NACIONAL.NET**, así como también el archivo **NACIONAL.SET** y **ENACIONAL.IO**.

Recordar que el archivo **ENACIONAL.IO** y **VNACIONAL.IO** son los archivos de entrenamiento y verificación respectivamente; además de que ambos provienen del archivo general **NACIONAL.IO**.

En la siguiente imagen se observan las características de la red **NACIONAL.SET**.



ENTRENAMIENTO DE LA RED NEURONAL.

El entrenamiento de la red neuronal se efectúa a 1000 épocas por medio del archivo **ENACIONAL.IO**; el cual esta constituido por 156 secuencias.

Como en las aplicaciones anteriores, se tomaron los errores de algunas secuencias antes y después del entrenamiento así como los parámetros finales al terminar tal procedimiento. Tal como se muestra a continuación:

NÚMERO	NÚMERO	NÚMERO		ANTES	DESPUÉS
	SEQ	PART		ENTRENAMIENTO	ENTRENAMIENTO
1	Seq 1	part 9	ERROR	0.000207	0.000207
2	Seq 2	Part 9	ERROR	0.4	0.00114
3	Seq 15	Part 1	ERROR	0.272	0.667
4	Seq 25	Part 6	ERROR	0.00411	0.00101
5	Seq 29	Part 8	ERROR	1.03	0.021
6	Seq 53	Part 2	ERROR	0.0293	0.00000297
7	Seq 81	Part 6	ERROR	0.0505	0.00118
8	Seq 89	Part 8	ERROR	1.000	0.316
9	Seq 114	Part 6	ERROR	0.832	0.0214
10	Seq 139	Part 1	ERROR	0.000594	0.64
11	Seq 143	Part 9	ERROR	1.0200	0.0197
12	Seq 158	Part 8	ERROR	0.000961	0.00302
13	Seq 182	Part 5	ERROR	0.1000	0.00206
14	Seq 203	Part 7	ERROR	0.00019	0.00143

Epoch Number: **1999**
 Epoch Error: **0.0742**
 Steps This Epoch: **2304**
 Time: **15:06**

VERIFICACIÓN DE LA RED NEURONAL

Con los antecedentes presentados después del entrenamiento, se puede hablar de un aprendizaje adecuado, sin embargo para afirmar completamente tal hipótesis se procede a cargar a la red el archivo de verificación **VNACIONAL.IO** conformado por 35 secuencias las cuales permiten observar y analizar los resultados que presenta la red.

Únicamente se presentan algunas de las secuencias que se utilizaron para la verificación.

NO. SEQ.	LOTERIA.IO		LOTERIA.OUT		
	No.	IN	OUT	OUT 1	OUT 2
SEQ 9	1	-0.0084	-0.5788	-0.024	
SEQ 10	2	-0.5788	0.7857	-0.526	-0.569
	3	0.7857	-0.6306	0.644	0.666
	4	-0.6306	0.2019	-0.52	-0.531
	5	0.2019	0.3981	0.114	0.126
	6	0.3981	0.2912	0.425	0.414
	7	0.2912	0.6754	0.266	0.271
	8	0.6754	-0.6838	0.623	0.618
	9	-0.6838	-0.3976	-0.592	-0.592
	9	-0.3976	0.3361		-0.425
SEQ 60	1	-0.0973	-0.2202	-0.084	
	2	-0.2202	0.3553	-0.231	
	3	0.3553	0.8903	0.327	
	4	0.8903	-0.4825	0.759	
	5	-0.4825	0.6557	-0.414	
	6	0.6557	-0.4304	0.545	
	7	-0.4304	-0.7939	-0.344	
	8	-0.7939	0.1679	-0.729	
	9	0.1679	0.6678	0.149	
SEQ 119	1	0.8523	0.3014	0.737	
SEQ 120	2	0.3014	-0.1696	0.327	0.294
	3	-0.1696	0.3553	-0.167	-0.161
	4	0.3553	0.1201	0.327	0.322
	5	0.1201	-0.5587	0.137	0.143
	6	-0.5587	0.9483	-0.521	-0.521
	7	0.9483	0.2445	0.759	0.759
	8	0.2445	-0.1603	0.3	0.3
	9	-0.1603	0.7101	-0.179	-0.179
	9	0.7101	0.7979		0.627

NO. SEQ.	LOTERIA.IO		LOTERIA.OUT	
	No.	IN	OUT	OUT 1 OUT 2
SEQ 200	1	-0.2966	-0.9048	-0.26
	2	-0.9048	0.2295	-0.776
	3	0.2295	-0.9829	0.179
	4	-0.9829	-0.9184	-0.783
	5	-0.9184	-0.8593	-0.81
	6	-0.8593	0.8369	-0.737
	7	0.8369	0.1483	0.67
	8	0.1483	0.0566	0.22
	9	0.0566	-0.2468	0.018
SEQ 291	1	0.8637	0.2607	0.714
	2	0.2607	-0.8837	0.3
	3	-0.8837	0.7702	-0.748
	4	0.7702	-0.8054	0.632
	5	-0.8054	0.0137	-0.649
	6	0.0137	-0.9488	-0.078
	7	-0.9488	-0.7912	-0.751
	8	-0.7912	0.0823	-0.748
	9	0.0823	0.6163	0.078

Análisis de resultados.

Ubicando los resultados de estas secuencias en el archivo **NACIONAL.IO**, e interpretando los resultados (efectuando el proceso inverso de normalización), se tienen las siguientes tablas; donde se muestra la información real.

No. EVENTO	LOTERIA	LOTERIA (PRONOSTICACIONES)	
	IN	SEQ 9 OUT 1	SEQ 10 OUT 2
1	24683	24296	
2	10578	11883	10820
3	44316	40813	41357
4	9298	12031	11759
5	29882	28450	28005
6	34733	35398	35126
7	32090	31466	31590
8	41591	40294	40170
9	7982	11809	10251
9	15058		143380

No. EVENTO	LOTERIA IN	LOTERIA (PRONOSTICACIONES)	
		SEQ 119 OUT 1	SEQ 120 OUT 2
1	45965	43112	
2	32341	32975	32159
3	20695	20760	20908
4	33676	32975	32851
5	27860	28277	28425
6	11074	12006	12006
7	48338	43656	43656
8	30934	32307	32307
9	20927	20463	20463
9	42447		40393

No. EVENTO	LOTERIA IN	LOTERIA (PRONOSTICACIÓN)	
		SEQ 60 OUT	
1	22483	22812	
2	19445	19177	
3	36598	32975	
4	46903	43656	
5	12959	14652	
6	41103	38365	
7	14247	16383	
8	5260	6863	
9	29041	28573	

No. EVENTO	LOTERIA	LOTERIA (PRONOSTICACIÓN)
	IN	SEQ 200 OUT
1	2518	5701
2	30563	29315
3	586	5528
4	2181	4861
5	3641	6666
6	45584	41456
7	28556	30329
8	26290	25334
9	18786	19746

No. EVENTO	LOTERIA	LOTERIA (PRONOSTICACIÓN)
	IN	SEQ 291 OUT
1	46245	42544
2	31335	32307
3	3039	6394
4	43935	40516
5	4975	8842
6	25229	22960
7	1429	6319
8	5325	6394
9	26924	26818

Obsérvese que los números en "negritas" en la columna de la salida (out), son los que mas cercanos estuvieron a los reales (IN).

Con estos resultados finales, se comprueba y se afirma que el aprendizaje de la red es relativamente bueno, porque hay secuencias ó eventos no muy cercanos a datos reales.

Una posible solución podría ser, el incrementar la base de datos y seguir variando cada uno de los parámetros que definen a una red neuronal.

UTILIZACIÓN DE LA RED NEURONAL

Una vez que se ha comprobado el aprendizaje de la red se procede a utilizar la red neuronal siendo éste el último de los cinco pasos.

Para ello, la idea es introducirle algunos números ganadores de la lotería nacional del mes de octubre del 2002, donde algunos de ellos nos servirán de referencia para la pronosticación y así conocer finalmente si la red neuronal **NACIONAL.NET** " pronostica" tales valores ganadores en ese año y mes.

Análisis de resultados:

La red presenta los siguientes resultados pronosticados; los números que sobresalen son los que se utilizaron como referencia:

EVENTO	LOTERIA	PREDICCIÓN
1	230	
2	47761	4546
3	8430	
4	32230	32060
5	49530	
6	49445	43904
7	36684	34063
8	2227	
9	19830	19894
10	45309	43286
11	12330	
12	16841	16111
13	26130	26521
14	40764	
15	13455	11883
16	39801	

Análisis 1:

Si el número siguiente del 230 sería el 47761, la red pronostica que le corresponde el 45461.

Los siguientes dos número a partir del 49530, son el 49445 y el 36684; sin embargo, la red pronostica que los siguientes son el 43904 y 34063 respectivamente; y así sucesivamente.

Obsérvese en especial los eventos 9, 12 y 13. Eventos donde la red pronosticó valores muy cercanos a los reales.

¿Porqué la información del año 2002 ?; se adquirió esta información como se ha mencionado líneas arriba, simplemente como referencia para conocer su aprendizaje por un lado y por el otro, introducirle intencionalmente números ganadores que en teoría deben ser distintos a los números ganadores con los que se entreno la red neuronal.

Como información interesante se anexa la siguiente tabla:

LOTERÍA NACIONAL 300 NÚMEROS			
DATOS ESTADÍSTICOS			
MEDIA	MODA	PROMEDIO	DESVIACIÓN ESTÁNDAR
24116.5	43751	23834.463	14359.821

“ El recuerdo es el único paraíso
del cual no podemos ser expulsados ”
“J.P. Richter”

CAPITULO IV

“PREDICCIÓN DE LAS VARIABLES ECONÓMICAS DEL MERCADO BURSÁTIL”

IV.1 OBJETIVO DEL CAPITULO: Predecir cinco variables económicas del mercado bursátil.

IV.2 DESARROLLO

Conociendo de antemano que una de las aplicaciones más utilizadas en predicción es en el área económica, es por eso, que se efectuó un ejemplo más en la pronosticación de las siguientes variables económicas presentadas durante el mes de octubre de 1999.

Variables Económicas: **Bono, Divisa, IndBol1, IndBol2 y Valor1.**

La información de estas variables están referenciadas al cierre que tuvieron al final de cada día. Recordando que la base esta constituida por 19 días.

Como en el caso anterior (números aleatorios), se efectuó la normalización de la base de datos empleando el método de **AutoScaling** para una red de ocho neuronas.

A continuación se observa la base de datos normalizados.

COMPORTAMIENTO BURSATIL					
FECHA	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
01-Oct	0.101	0.242	-0.18	0.558	-0.095
04-Oct	1.077	0.242	0.546	0.024	-0.335
05-Oct	0.244	0.242	0.886	0.02	0.324
06-Oct	0.971	0.242	0.692	1.076	0.119
07-Oct	1.016	0.242	1.284	0.99	0.599
08-Oct	0.91	0.242	1.59	1.421	1.39
11-Oct	0.758	1.21	1.82	1.421	1.846
13-Oct	-0.045	-0.707	0.494	-0.926	0.575
14-Oct	-1.32	-0.707	0.226	-0.62	0.575
15-Oct	-0.819	-1.639	-0.832	-2.12	-1.103
18-Oct	-0.834	-1.639	-1.36	-1.577	-1.342
19-Oct	-0.515	-0.707	-0.606	-1.079	-1.103
20-Oct	-1.411	-0.707	-0.654	-0.025	-1.103
21-Oct	-0.485	-0.707	-1.487	-0.025	-1.582
22-Oct	-0.5	-0.707	-0.754	0.412	-0.623
25-Oct	-1.32	0.242	-1.274	-0.264	-0.623
26-Oct	-0.971	1.21	-0.422	-0.532	0.239
27-Oct	0.136	1.21	-0.571	-0.011	0.575
28-Oct	1.107	2.196	0.603	1.268	1.294

IV.2.1 VERIFICACIÓN DE LA RED NEURONAL.

Los días que permitieron verificar el aprendizaje de la red y que fueron pronosticados fueron:

7 octubre.

13 octubre.

18 octubre.

21 octubre.

25 de octubre.

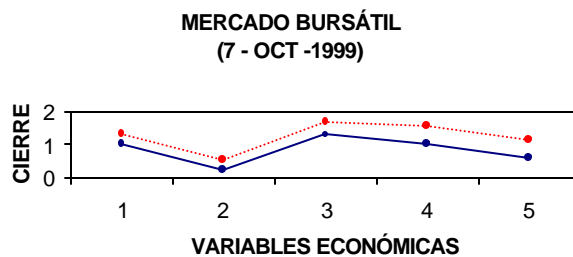
PREDICCIONES BURSATILES					
FECHA	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
01-Oct	0.101	0.242	-0.18	0.558	-0.095
04-Oct	1.077	0.242	0.546	0.024	-0.335
05-Oct	0.244	0.242	0.886	0.02	0.324
06-Oct	0.971	0.242	0.692	1.076	0.119
07-Oct	1.325	0.529	1.674	1.527	1.116
08-Oct	0.91	0.242	1.59	1.421	1.39
11-Oct	0.758	1.21	1.82	1.421	1.846
13-Oct	0.025	-0.928	0.445	-1.095	0.541
14-Oct	-1.32	-0.707	0.226	-0.62	0.575
15-Oct	-0.819	-1.639	-0.832	-2.12	-1.103
18-Oct	-0.819	-1.715	-1.261	-1.565	-1.343
19-Oct	-0.515	-0.707	-0.606	-1.079	-1.103
20-Oct	-1.411	-0.707	-0.654	-0.025	-1.103
21-Oct	-0.409	-0.694	-1.55	-0.1	-1.558
22-Oct	-0.5	-0.707	-0.754	0.412	-0.623
25-Oct	-1.175	0.137	-1.204	-0.249	-0.612
26-Oct	-0.971	1.21	-0.422	-0.532	0.239
27-Oct	0.136	1.21	-0.571	-0.011	0.575
28-Oct	1.107	2.196	0.603	1.268	1.294

Obsérvese en especial los días 18 y 25 de octubre; en estos días la red presentó una mayor capacidad de pronosticación que días restantes.

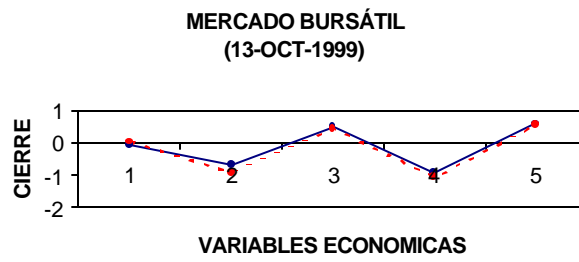
IV.2.2 GRAFICAS DE LOS RESULTADOS DEL ARCHIVO DE VERIFICACION.

Valores pronosticados:
 Valores reales: ———

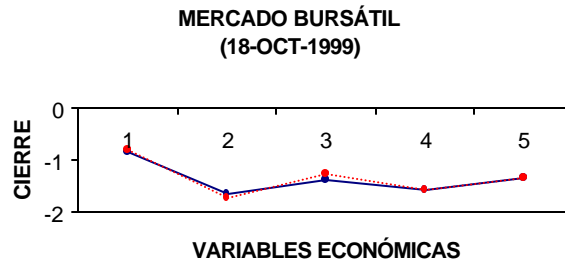
1.



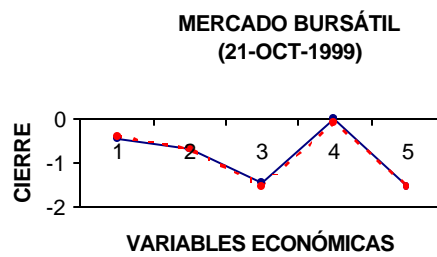
2.



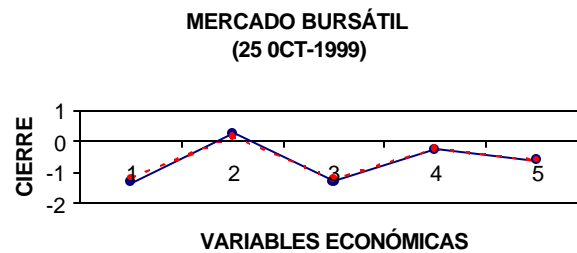
3.



4.



5.



IV.2.3 COMENTARIOS.

Observando los resultados y gráficas de la predicció del cierre al final del día de las cinco variables económicas, se puede concluir que esta red neuronal tuvo un excelente aprendizaje considerando que la información fue mínima.

IV.3 UTILIZANDO LA RED NEURONAL.

Una vez que se ha comprobado el aprendizaje de la red neuronal, el último paso es **utilizar la red**, para ello, se efectuaron dos ejercicios; en ambos se pronosticaron los días 29 de octubre, 30 octubre, 31 octubre y el 1 de noviembre. Para efectuar tales pronosticaciones, en el primer ejercicio, se seleccionó al azar del archivo de entrenamiento y de verificación, una serie de valores numéricos para generar así el archivo de datos (**BURSA1.DAT**), y por ende el archivo de entradas y salidas (**BURSA1.IO**). Obteniendo los resultados que se presentan a continuación:

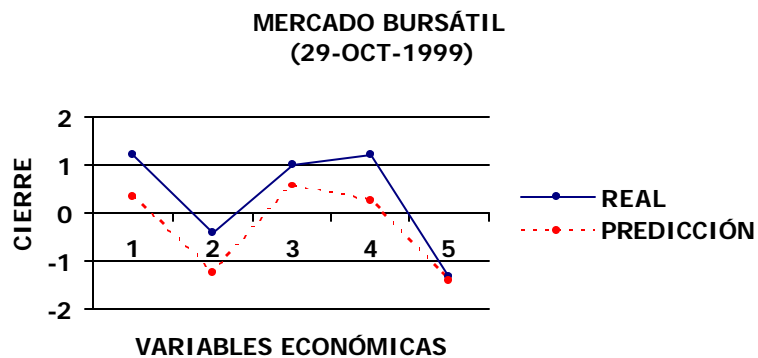
IV.3.1 EJERCICIO I

En la siguiente tabla se muestran los valores "reales" y las gráficas correspondientes a los cuatro días pronosticados.

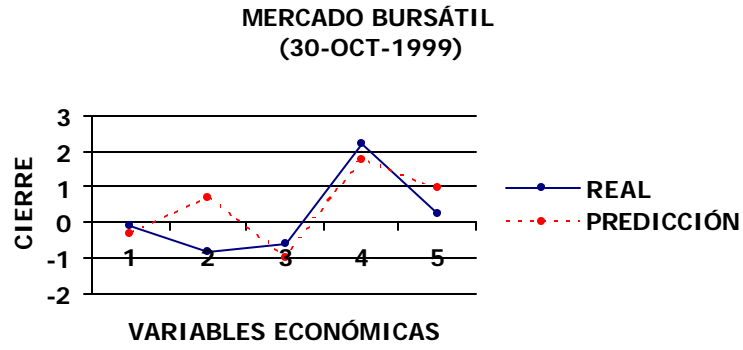
	MES	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
VALOR REAL	29-Oct	1.21	-0.422	0.99	1.21	-1.342
PREDICCIÓN	29-Oct	0.311	-1.242	0.565	0.212	-1.412
VALOR REAL	30-Oct	-0.136	-0.832	-0.62	2.196	0.242
PREDICCIÓN	30-Oct	-0.348	0.694	-1.003	1.785	0.939
VALOR REAL	31-Oct	-0.606	1.294	-0.834	0.494	-0.707
PREDICCIÓN	31-Oct	-0.493	0.852	-0.187	0.062	0.075
VALOR REAL	01-Nov	1.268	0.603	-0.045	-0.243	0.758
PREDICCIÓN	01-Nov	0.505	1.095	-0.671	0.075	0.717

GRÁFICAMENTE:

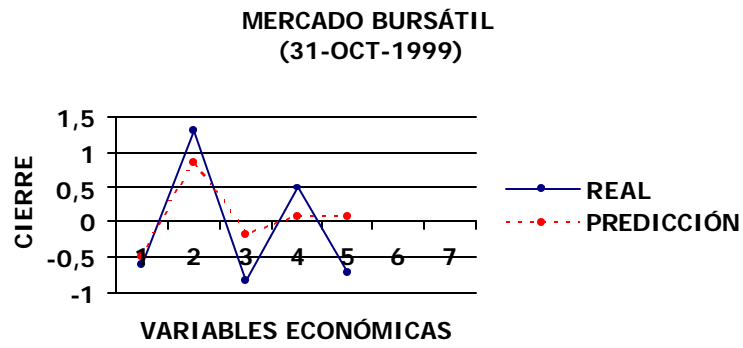
1.



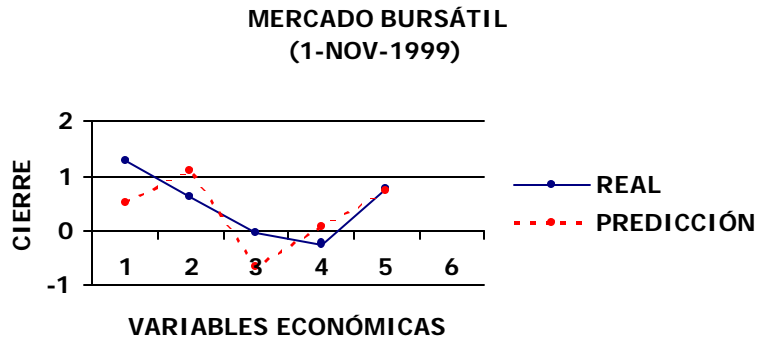
2.



3.

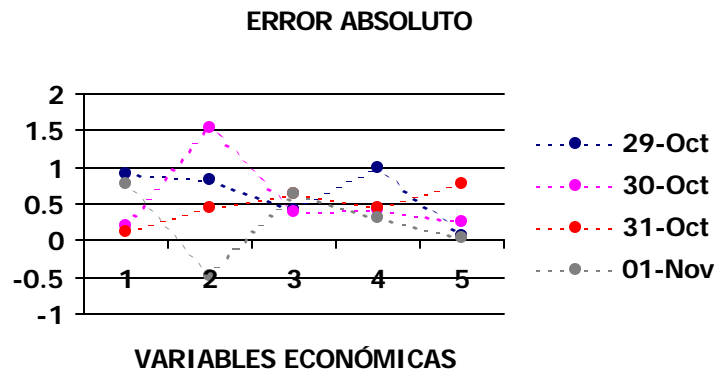


4.



En la siguiente tabla y gráfica se muestra el **error absoluto** entre el valor real y el valor pronosticado de las cinco variables económicas.

ERROR	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
29-Oct	0.899	0.82	0.425	0.998	0.07
30-Oct	0.212	1.526	0.383	0.411	0.242
31-Oct	0.113	0.442	0.647	0.432	0.782
01-Nov	0.763	-0.492	0.626	0.318	0.041



IV.3.2 EJERCICIO II.

En este ejercicio, los valores elegidos fueron muy distintos a los anteriores, estos valores no pertenecen al archivo general IO (entrenamiento y verificación). La única restricción fue que estuvieran dentro del rango de normalización.

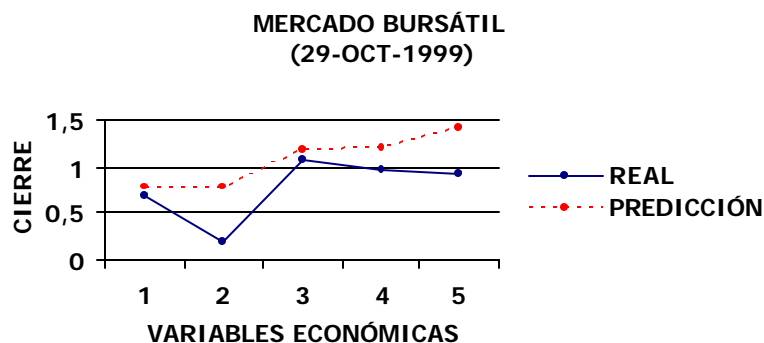
A continuación se muestran los resultados obtenidos.

RESULTADOS:

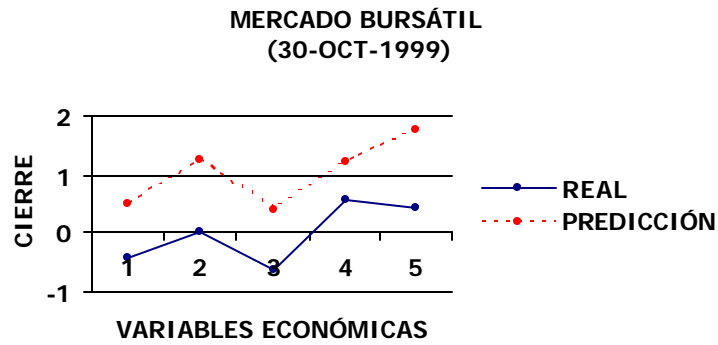
	MES	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
VALOR REAL	29-Oct	0.686	0.181	1.069	0.982	0.94
PREDICCIÓN	29-Oct	0.785	0.785	1.175	1.194	1.421
VALOR REAL	30-Oct	-0.429	0.022	-0.643	0.564	0.409
PREDICCIÓN	30-Oct	0.469	1.251	0.397	1.232	1.778
VALOR REAL	31-Oct	0.717	-0.597	-0.036	0.98	0.932
PREDICCIÓN	31-Oct	0.237	0.852	-0.885	0.863	1.213
VALOR REAL	01-Nov	0.079	-0.924	0.279	0.456	-0.279
PREDICCIÓN	01-Nov	0.785	0.95	0.863	0.808	1.386

GRÁFICAMENTE:

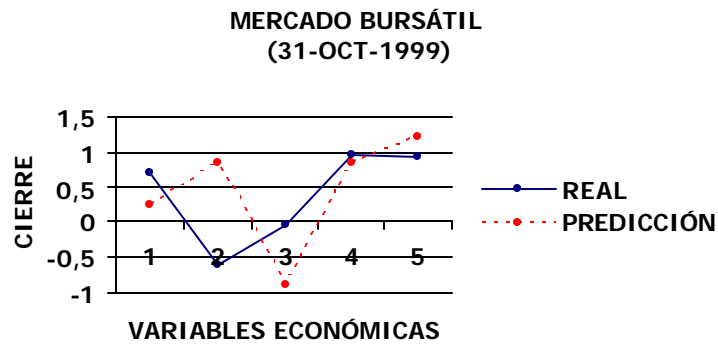
1.



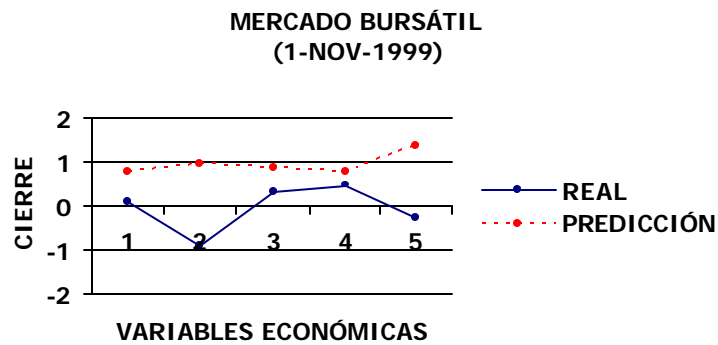
2.



3.



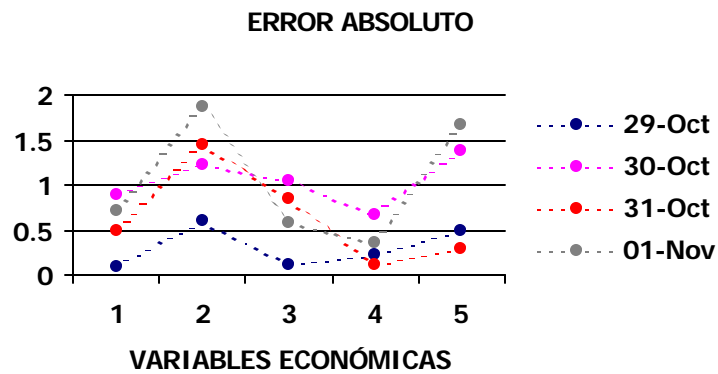
4.



A continuación se muestra el **error absoluto** de las cinco variables económicas dentro de los cuatro días.

ERROR	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
29-Oct	0.099	0.604	0.106	0.212	0.481
30-Oct	0.898	1.229	1.04	0.668	1.369
31-Oct	0.48	1.449	0.849	0.117	0.281
01-Nov	0.706	1.874	0.584	0.352	1.665

GRÁFICAMENTE:



V.3.3 OBSERVACIONES

Observando a primer instancia los resultados de ambos ejercicios; lo que se puede afirmar es de que la red obtiene buena pronosticación cuando los datos numéricos pertenecen ó forman parte de la información original (ejercicio 1); ya que en el segundo caso, la información que se utilizó para efectuar las predicciones al no pertenecer al campo de los datos originales, la red no pronostico bien; sin embargo y afirmando una vez más, estas observaciones son relativamente engañosas, hay que realizar el proceso inverso de la normalización de estos datos; para de esta forma tener un criterio más amplio y determinar la eficiencia de la red neuronal.

IV.4 RESULTADOS FINALES.

Una vez que se han obtenido los resultados que presenta la red neuronal hay que efectuar el proceso inverso de normalización, recordando que para estructurar la base de datos, fue necesario emplear el proceso de **Auto Scaling**, un método de normalización; que proporciona DynaMind.

La fórmula que se empleó para la normalización de cada uno de los valores de las variables económicas es:

$$V_{nor} = \frac{V_{real} - V_{prom}}{Sx}$$

donde:

V_{nor}: Valor Normalizado

V_{prom}: Valor Promedio

V_{real}: Valor Real

S_x: Desviación Standar

Entonces para conocer los valores finales, se emplea la siguiente fórmula:

$$V_{\text{real}} = (V_{\text{nor}})(S_x) + V_{\text{prom}}$$

A continuación se muestran los valores finales que la red pronosticó en los dos ejercicios anteriores.

Nota: La desviación estándar y el promedio son distintos en cada una de las cinco variables económicas. Tal como se muestra a continuación:

	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
Promedio	87.58	155.133	9512.1	10396.92	14.96
Desviación Estándar	0.659	1.516	162.666	177.85	0.417

Ejercicio 1:

Por ejemplo; para conocer el valor verdadero del día 29 de octubre tanto del real como el pronosticado, se tiene:

$$V_{\text{prom}}=87.58;$$

$$S_x=0.659;$$

$$V_{\text{nor}}=1.21;$$

$$V_{\text{real}} = (v_{\text{nor}} * s_x) + v_{\text{prom}}$$

$$V_{\text{real}} =$$

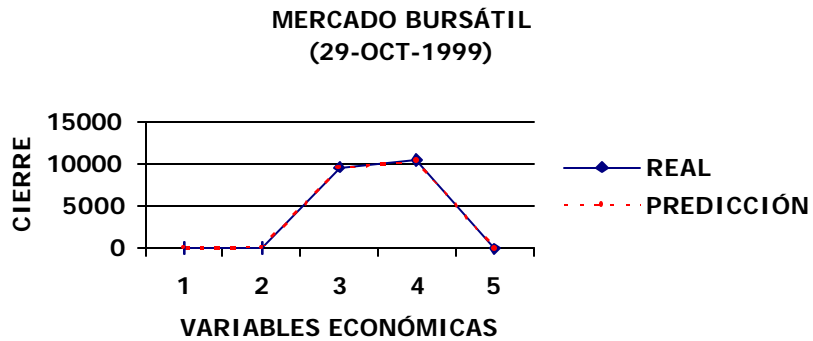
$$88.3774$$

Por lo que los resultados finales son:

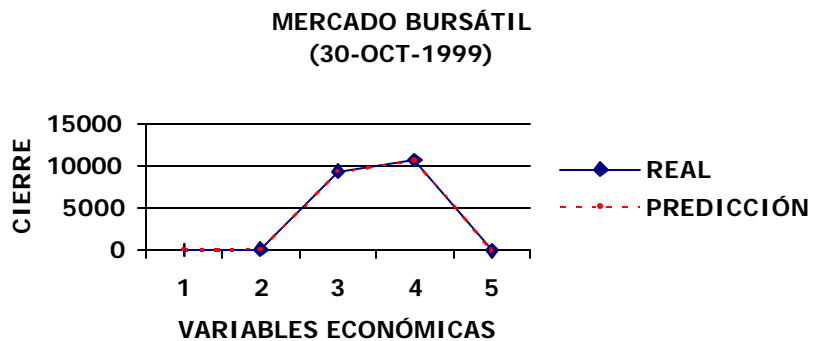
	MES	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
VALOR REAL	29-Oct	88.37	154.49	9673.1	10612	14.4
PREDICCIÓN	29-Oct	87.78	153.25	9604	10435	14.37
VALOR REAL	30-Oct	87.49	153.87	9411.2	10787	15.06
PREDICCIÓN	30-Oct	87.35	156.18	9348.9	10714	15.35
VALOR REAL	31-Oct	87.18	157.09	9376.4	10485	14.66
PREDICCIÓN	31-Oct	87.25	156.42	9481.7	10408	14.99
VALOR REAL	01-Nov	88.41	156.04	9504.8	10354	15.27
PREDICCIÓN	01-Nov	87.91	156.79	9403	10410	15.25

Gráficamente:

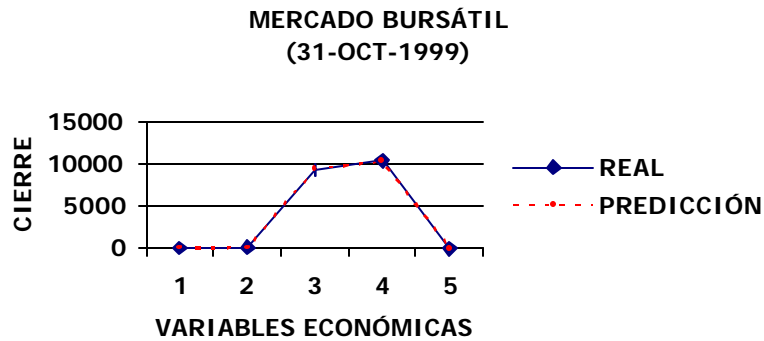
1.



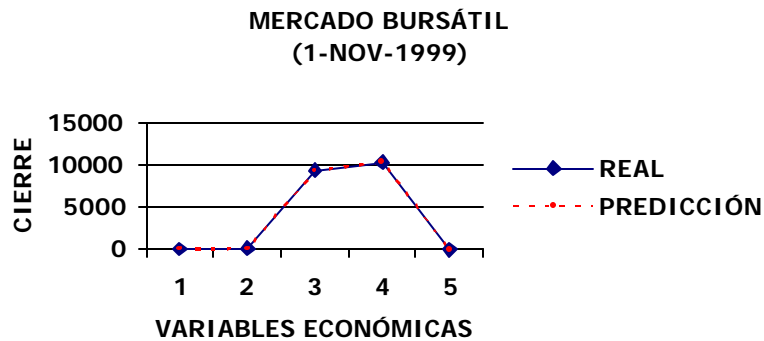
2.



3.



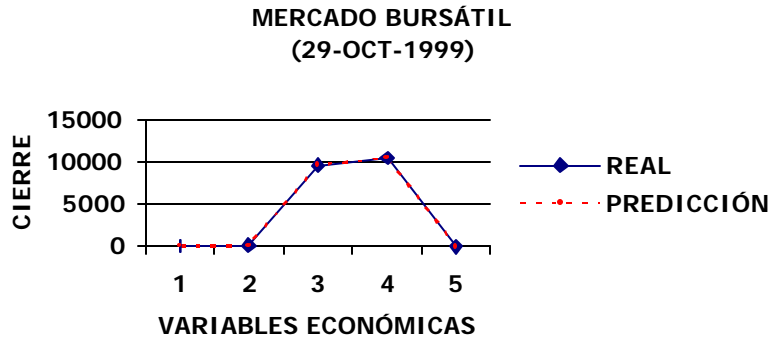
4.

*Resultados:**Ejercicio II.*

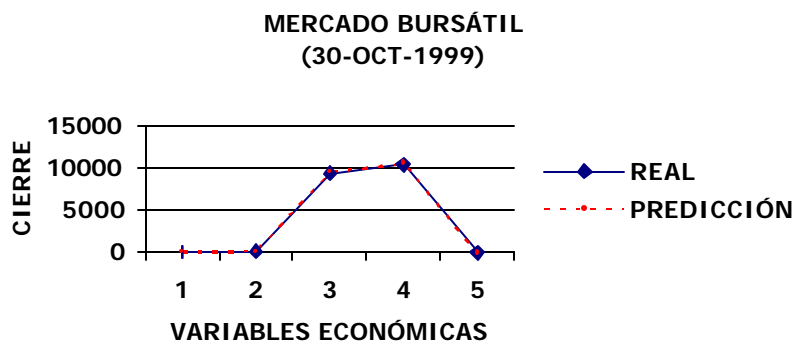
	MES	BONO	DIVISA	INDBOL1	INDBOL2	VALOR1
VALOR REAL	29-Oct	88.03	155.4	9686	10572	15.35
PREDICCIÓN	29-Oct	88.09	156.32	9703.2	10609	15.55
VALOR REAL	30-Oct	87.29	155.16	9407.5	10497	15.13
PREDICCIÓN	30-Oct	87.88	157.02	9576.7	10616	15.7
VALOR REAL	31-Oct	88.05	154.22	9506.2	10571	15.34
PREDICCIÓN	31-Oct	87.73	156.42	9368.1	10550	15.46
VALOR REAL	01-Nov	87.63	153.73	9557.5	10478	14.84
PREDICCIÓN	01-Nov	88.09	156.57	9652.5	10541	15.53

Gráficamente:

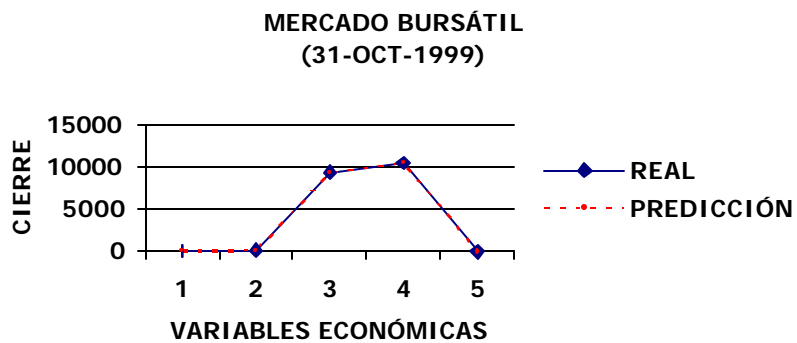
1.



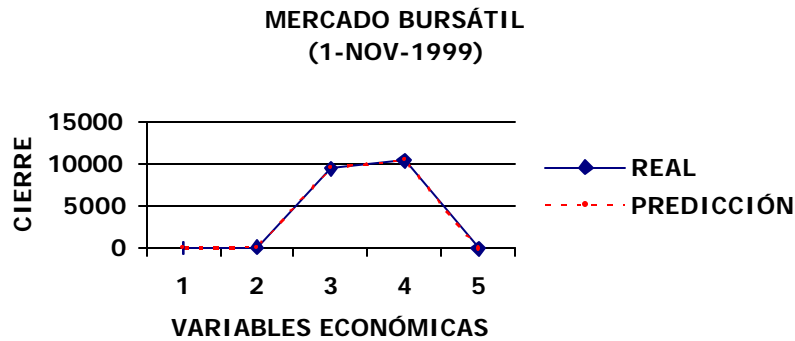
2.



3.



4.



IV.5 CONCLUSIONES.

Con estos nuevos resultados se tiene:

1. La red neuronal obtuvo una excelente capacidad de pronóstico; los valores reales son muy cercanos a los pronosticados; gráficamente en ambos ejercicios las gráficas tienen un comportamiento "predefinido"; permitiendo con esto a pensar que este es el comportamiento bursátil de estas cinco variables económicas del mes de octubre y de noviembre (29 de octubre al 1 de noviembre 1999), tomando como referencia información perteneciente al mes de octubre del año pasado.
2. El proceso de "desnormalización", cambio radicalmente la visión ó perspectiva de los resultados obtenidos inicialmente; afirmando que la red es eficaz.

¿Por qué eficaz?; por el hecho de que en el primer ejercicio, los datos que sirvieron para evaluar su capacidad de pronosticación fueron aleatorios pero pertenecientes a la información con la que aprendió la red neuronal; mostrando buenos resultados; quizás uno podría pensar que era obvio esperar tales resultados; por lo que se realizó el segundo ejercicio; donde la información **no** pertenecía al archivo de entrenamiento ni al de verificación y además de ser aleatorios los datos una vez más; dando como resultados que la red presentará buenos resultados .

“Cada pensamiento que tenemos está
creando nuestro futuro”
“anónimo”

CAPÍTULO V

“ CONCLUSIONES ”

A continuación se presentan las principales conclusiones y contribuciones de este trabajo de investigación que abarcó desde el conocimiento básico y elemental de las redes neuronales biológicas y artificiales hasta adquirir y lograr el entendimiento adecuado y correcto del sistema de desarrollo **Dynamind**, empleado para realizar uno de los principales objetivos de este trabajo; “Explorar la posibilidad de redes neuronales artificiales de predecir eventos futuros en una serie”. Pero, ¿qué es una serie temporal ? o ¿en qué consiste una serie de eventos temporales?. Lo que se realizó en este trabajo fue la predicción de las series temporales constituidas por:

1. Una **señal senoidal**.
2. **Números aleatorios de una base de datos de números aleatorios**.
3. El comportamiento **de cinco variables económicas del mercado bursátil** a lo largo de un tiempo (un mes).

Formalmente, se puede definir a una serie temporal como una secuencia de vectores que son función del tiempo y que pueden ser resultado de una función continua, como las medidas de la temperatura en una habitación, ó son producto de funciones discontinuas, por ejemplo, una secuencia de números aleatorios u otros fenómenos como algunos de los estudiados en esta tesis.

Dependiendo del objetivo del problema que se esta analizando , el procesamiento de series temporales por medio de redes neuronales se puede dividir en cuatro tipos, estos son:

- Predicción de valores futuros de la serie.
- Clasificación de los eventos de la serie temporal o parte de ella en una o varias clases.
- Descripción de la serie mediante un conjunto de parámetros que definan un modelo.
- Transformación de una serie en otra.

En nuestro caso, el problema de predicción en series temporales puede ser contemplado como la búsqueda de una función que la describa; siempre y cuando se encuentre una relación existente entre las variables que definen al fenómeno de estudio.

Por otra parte, no es necesario conocer el valor exacto de la predicción, sino solo alguna característica como es: si los valores de la serie van a crecer o, a disminuir; o por lo menos lograr una aproximación del comportamiento del fenómeno.

Para el caso de la señal senoidal, el interés de su análisis se caracteriza por ser una señal continua porque su comportamiento a través del tiempo es equidistante y se puede hablar de la presencia de tendencia cuando la señal se presenta en más de dos ciclos cierta tendencia.

En el capítulo III (Resultados), se tiene que el número de neuronas en la red entrenada con la señal senoidal no afectó el desempeño de la red para caso en particular; los resultados arrojados por cada una de las redes neuronales (seno2.net y seno8.net).

Se observa que de 40 puntos que se tomaron para verificar el aprendizaje de la red, el 30% de ellos fue muy cercano a los valores reales, pues varían una centésima del punto original; tal como se observa en el tiempo 6, a este punto le

corresponde el **0.389**, la red de dos neuronas pronosticó **0.388** mientras que la de ocho neuronas pronosticó un punto de **0.377**.

Por otro lado, al analizar los errores presentados después del entrenamiento, se consideran que estos son aceptables.

Se puede afirmar entonces, que las redes neuronales **SENO2.NET** y **SENO8.NET** obtuvieron una buena **capacidad de pronosticar** una señal senoidal.

Cabe aclarar que el número de neuronas necesarias en la red depende del tipo del problema a resolver, siendo la única variable afectada por éste número el **tiempo de aprendizaje**, ya que se puede considerar que es proporcional al número de neuronas.

En la aplicación I los quince puntos que pronosticó son muy parecidos a los reales, esto se debe a que el archivo IO (Archivo de entradas y salidas) utilizado para ese ejercicio, pertenecía al archivo de entrenamiento de la red neuronal. Comprobando una vez mas que la red neuronal obtuvo un excelente aprendizaje. En la segunda aplicación, donde se le introdujo a la red una señal con frecuencia distinta, para conocer su comportamiento, se observa que a esta nueva señal no le afectó la pronosticación ya que los resultados presentados también son cercanos a los reales; demostrando con esto que la red neuronal **seno2.net** pronosticó muy bien.

Para el segundo fenómeno estudiado: **Predicción del siguiente número aleatorio de una base de datos de números aleatorios**. A diferencia de la señal senoidal, los eventos para este caso no presentan un comportamiento continuo sino que presentan un comportamiento de tipo discreto. Implicando la no existencia de alguna tendencia o comportamiento dentro de un patrón fijo que expliqué el fenómeno.

El reto en este caso para la redes neuronales artificiales, fue el de encontrar a través del tiempo el valor de esa variable aleatoria, por lo que la pregunta a

contestar en su momento fue ¿Esta nueva red neuronal tendrá la misma capacidad de pronosticación que tuvo la red senoidal, cuando la información es aleatoria?

Para contestar tal pregunta se vio la necesidad de generar tres redes neuronales artificiales que fuesen capaces de pronosticar la serie de números aleatorios.

Esa base de datos esta conformada por 50 números aleatorios los cuales fueron presentándose también aleatoriamente. En cada una de las tres redes, como se justifica y observa en el capítulo III, se modificaron algunos de sus parámetros.

Observando detalladamente los resultados, en realidad no se llegó a obtener una **consistencia** en las predicciones de alguna de esas redes neuronales, ya que los pronósticos en algunos eventos de las tres redes a veces están muy alejados de lo esperado y en otras ocasiones los pronósticos son muy cercanos .

De hecho, ninguna de las tres redes tuvo la capacidad de pronosticar el valor mínimo de la base de datos, siendo este el 1424; ninguna de las redes se acercó a tal valor quizás se debido a la poca frecuencia repetitiva de estos números de dimensión pequeña en la lista del archivo de aprendizaje.

Por lo tanto, el criterio que se estableció para definir cuál de las tres redes era la más “eficiente” fue el de calcular el error promedio existente entre el valor real menos el valor pronosticado. Otro criterio para comparar la eficiencia fue el de considerar cuál de las tres redes empleó menos tiempo en su aprendizaje (siendo esta **RAND1**, la cual empleó 1:58 [min]), estableciéndose así que esta era la idónea, además el error promedio presentado por esta red fue de 846.9.

Un punto a destacar es la inconsistencia de las tres redes neuronales en el pronostico de los eventos aleatorios ya que proviene del mismo comportamiento del fenómeno, al no contar éste con una tendencia predefinida. Además, hay que considerar que al generar y trabajar con redes neuronales tiene su grado de complejidad como cualquier sistema que estudia o trabaja con fenómenos no lineales .

Analizando la aplicación II (los 300 números de la lotería nacional), en este ejercicio en particular no se logró crear una red más eficiente que las tres primeras redes; parece que al incrementar la base de datos no se logró la idea que se tenía en mente planteada en un inicio que, al incrementar la base de datos, el aprendizaje de la red sería más efectiva; sin embargo lo que se logró fue un comportamiento muy similar al de las tres redes anteriores, es decir, un comportamiento no definido. Pero aún así considero que las pronosticaciones efectuadas por esta red no están del todo mal.

Para el caso del fenómeno del **mercado bursátil**, este presenta un comportamiento con cierta tendencia y alguna aleatoriedad al mismo tiempo.

Analizando la aplicación de las redes del **Mercado Bursátil**, se tiene que los días entre el 18 y el 25 de octubre la red pronosticó excelentemente, difiriendo sus predicciones en una décima del valor real.

En este ejemplo, la red pronosticó muy bien considerando que la base de datos de su aprendizaje fue muy pobre (14 datos); pocos en comparación con la base de datos de la senoidal (76 datos) y de los números aleatorios (300 datos).

En este caso; la experiencia adquirida anteriormente, permitió estructurar el archivo de entradas y salidas (IO) adecuadamente.

Otro punto que ayudó es el hecho de que ésta base de datos no fue totalmente aleatoria, como se mencionó anteriormente; por ejemplo la variable económica "divisa", en los primeros ocho días se mantuvo constante, permitiendo con esto que la red tuviese un reforzamiento en su aprendizaje, es decir, esa variable económica al repetirse por lo menos ocho veces, permitió que la red guardará en sus pesos el valor de esa variable.

Como una observación, se tiene que al interpretar los resultados obtenidos para pronosticar los días del 29 de octubre al 1 de noviembre de 1999, aparentemente

la red no había pronosticado adecuadamente, sin embargo, la perspectiva cambió cuando se realizó el proceso de "desnormalización" de esos resultados obtenidos; entonces éstos se aproximaron a los reales.

En general, evaluando el funcionamiento de cada una de las redes generadas para las tres distintas series temporales, se tiene que en la predicción **SENOIDAL** y **MERCADO BURSÁTIL**, la red pronosticó puntos ó eventos muy cercanos a los reales:

1. Cuando la nueva información IO que se utiliza como referencia una vez que se ha evaluado su aprendizaje, pertenece al archivo de entrenamiento y verificación.
2. Cuando la información IO es repetible en ciertos intervalos de tiempo, la red le permite establecer un mejor y mayor aprendizaje.

En otras palabras estamos hablando de la presencia de tendencia o comportamiento continuo.

En el caso contrario cuando a la red se le presentan series temporales aleatorias (**números aleatorios**), la red neuronal tiene un funcionamiento menos eficiente, la red no encuentra una tendencia dentro del comportamiento del fenómeno quizás se deba también a que la información presentada fue escasa para que la red llegase a encontrar cierta tendencia si es que existiese esta en el desarrollo del fenómeno.

Por otro lado se tiene que al trabajar con esta metodología computacional de las redes neuronales artificiales y hablando primeramente del sistema de desarrollo **Dynamind** se tiene que:

1. El proceso empieza desde el hecho de obtener la base de datos idónea para que la red neuronal aprenda. Esta información debe ser totalmente apegada

al fenómeno a estudiar, es decir, ser real y confiable porque de esta forma nuestra red será real y confiable al presentar los resultados.

Una vez que se tiene una buena base de datos, el siguiente punto a tomar en cuenta es el tipo con el que está constituida la información, es decir, si es de tipo numérico, alfanumérico o totalmente compuesto por caracteres.

Las tres series temporales a pronosticar fueron totalmente de tipo numérico, sin embargo se tuvo la necesidad, en el segundo y tercer caso (Números Aleatorios / Mercado Bursátil) de aplicar un método de normalización; los números originales al ser de tipo flotante no son aceptados por el sistema de desarrollo; ahora bien si la información fuese de tipo carácter, entonces se tendría que emplear un método donde se convierta esos caracteres a tipo numérico.

2. El siguiente punto crucial es, una vez que se ha normalizado la información, estructurar y delinear adecuadamente la base de datos para generar el archivo de entradas y salidas (IO), ya que a partir de éste se obtendrán los archivos de entrenamiento y verificación para la red neuronal. Hay que recordar que al crear el archivo general de IO, se tiene que establecer correctamente el número de vectores de (IO) que conformarán cada secuencia ya que, por experiencia, es importante establecerlos adecuadamente pues esto determina o influye en el aprendizaje de la red neuronal.
3. Otro punto importante es determinar el número adecuado de neuronas que conforman la red. Generalmente este tipo de variable depende del fenómeno que se desea analizar o estudiar.

Finalmente, otros parámetros que determinan y forman parte del éxito o fracaso del aprendizaje de la red neuronal son: el número de épocas de aprendizaje, la

razón de aprendizaje (learning rate) y la función de transferencia (Máximo, Mínimo y Ganancia), el número de épocas, etc.

Como se observa son varios los parámetros ó variables que hay que considerar para controlar y obtener una excelente red neuronal para efectuar predicciones dentro del paradigma de las redes neuronales artificiales.

Recapitulando se tiene que al inicio de este trabajo el primer fenómeno es estudio (Señal Senoidal) consiste en una señal continua ; posteriormente, en el segundo fenómeno su comportamiento no presenta con un patrón fijo y por ende una tendencia; finalmente, en el último fenómeno este presenta cierta tendencia y al mismo tiempo un comportamiento variable con respecto al tiempo. Fenómenos cuyo comportamiento difiere uno del otro, por lo que la pregunta ahora a contestar con respecto al sistema de desarrollo Dynamind es ¿en realidad éste sistema realiza pronosticaciones con su algoritmo de aprendizaje denominado True Time o sencillamente lo que está efectuando son sólo **interpolaciones?**. Aquí hay que recordar que interpolar consiste en construir una función que pase por valores conocidos.

Por otro lado no todo es y son las redes neuronales artificiales sino que existen realidades, prejuicios, ventajas y desventajas en el uso de las mismas, por ejemplo:

- Las redes neuronales artificiales son una herramienta más en el mercado para hacer un análisis ó construcción de modelos de predicción.
- Una de las desventajas como se ha mencionado anteriormente, es de que requiere la definición de muchos parámetros antes de aplicar la metodología, por ejemplo, hay que decidir la arquitectura más apropiada, el número de capas ocultas, la función de transferencia, etc.

- Otra desventaja es que no ofrecen una interpretación fácil, aunque este punto débil puede convertirse relativamente en un factor positivo, es decir, cuando mas difícil es de entender la lógica del sistema, mas difícil es su manipulación.
- Una afirmación errónea es que las redes neuronales no son estadísticas y, por ello, más fáciles de usar. La metodología es, sin duda, matemática y, bajo ciertas condiciones, puede generar resultados idénticos a otras técnicas estadísticas, sencillamente se puede afirmar que sirven para situaciones en las que existen muestras complejas de datos, pero no se conoce su comportamiento. Es aquí de donde se desprende una de las ventajas de las redes neuronales artificiales, que permiten integrar en un mismo modelo **información cuantitativa** y **cualitativa**, que se relacione lineal o no linealmente con la variable a predecir; lo cual no es permisible para otro tipo de modelos o metodologías computacionales.

Esto se debe hasta cierto punto, al tipo de la topología que se utiliza especialmente en el campo de predicciones, estamos hablando de las **redes neuronales recurrentes**, además de ser uno de los modelos posibles para el tratamiento de secuencias temporales. Hay que recordar que su principal característica o ventaja consiste en la posibilidad de almacenar una representación de la historia reciente de la secuencia, por lo que en las redes no recurrentes, no existe esta "memoria".

La diferencia principal entre las redes no recurrentes y las redes recurrentes, es la conexión cíclica (recurrente) entre las neuronas que la configuran. Por sus propiedades, las redes recurrentes presentan una gran capacidad de pronosticación (capacidad de producir salidas correctas para entradas no vistas durante el entrenamiento).

Estas propiedades son:

- **No linealidad.** Una red neuronal puede ser lineal o no lineal.
- **Adaptabilidad.** Las redes neuronales son capaces de reajustar sus pesos para adaptarse a los cambios en el entorno.
- **Tolerancia ante fallos.** Una red neuronal es tolerante ante fallos en el sentido de que los posibles fallos operacionales en partes de la red solo afectan débilmente al rendimiento de ésta.

Por todas estas propiedades y recapitulando, se tiene que la aplicación de éstas redes en especial en el procesamiento de secuencias de acuerdo a Forcada y Gori son:

- Predicción en series temporales.
- Procesamiento del lenguaje humano.
- Ecuación de canales digitales.
- Codificación del habla.
- Reconocimiento del habla.
- Control de sistemas.

Fenómenos que de una u otra forma tienen un comportamiento no lineal y aleatorio dentro del tiempo.

Para perfeccionar el aprendizaje en este tipo de algoritmos, Christian W.Omlin y C. Lee Giles consideran que es de gran importancia conocer a fondo las "reglas" que controlan las redes neuronales recurrentes, ya que al trabajar estas en el tiempo discreto, pueden ser entrenadas desde la forma de un **estado autónoma finito determinístico** (DFAs) aplicados a los grupos de algoritmos en la salida de las redes de estado recurrente.

O bien la propuesta elaborada por Williams y Zipser los cuales proponen dos algoritmos de aprendizaje para las redes recurrentes. El primero de ellos consiste en el algoritmo del gradiente para problemas donde los datos consisten en épocas, y el segundo método denominado **aprendizaje recurrente en tiempo real** (RTRC) donde se usa para describir temporalmente las cadenas de entradas y

salidas, las cuales son marcadas por las épocas, la idea principal de este método consiste en el seguimiento del gradiente del algoritmo de aprendizaje.

Las redes RTRC consisten en dos capas, una capa de entrada con m neuronas y una capa de salida con n neuronas.

Como se ha mencionado anteriormente, el número de neuronas depende de la complejidad del problema.

La problemática de los algoritmos RTRC es que mientras que en el aprendizaje para una gran cantidad de datos, el cambio de los pesos entre el inicio del aprendizaje ($t = 0$) y el final de este ($t = \quad$), generalmente no es tan pequeño. Esto se debe a la técnica del seguimiento del gradiente (gradient- following).

Con esto no se afirma que con solo éste clase de topologías (redes recurrentes), se pueden realizar predicciones de series temporales, sino también existen por ejemplo las redes neuronales de tipo ART y las redes neuronales que emplean el método de aprendizaje aplicado en capa por capa.

Éstas dos opciones consisten por ejemplo, en que las redes de tipo ART (Teoría de la Resonancia Adaptiva), tiene como características, que emplean un aprendizaje rápido y además de ser un sistema con tolerancia al ruido siendo su principal característica o peculiaridad de estas redes es la de mantener la "plasticidad" necesaria para aprender nuevos patrones, y a la vez previenen la modificación de otros que se han aprendido previamente, este tipo de paradigmas fueron implementados por Carpenter y Grossberg.

De hecho a este tipo de redes se le considera que están dentro de la clasificación de redes auto-organizativos, como son las redes de Hammin y Maynet, las redes de retro-propagation y los mapas de Kohonen, siendo estos tipos de algoritmos los que más se utilizan en la actualidad.

Por el lado de las redes que emplean la metodología FIR, siendo un nuevo modelo para la predicción de series temporales pues su característica es de igual forma la utilización del gradiente descendente.

Este tipo de algoritmo de aprendizaje también conocido como optimización de las capas, ha sido generalmente propuesto para redes con *perceptron multicapa* y han demostrado ser excelente en el aprendizaje en la fase de entrenamiento.

¿Por qué FIR?; simplemente porque se basa en los filtros empleados en el procesamiento de señales digitales o discretos, permitiendo con esto que la red sea capaz de aproximarse a la función original, que sería por ejemplo en reconocimiento de voz.

Con todas estas referencias sobre las redes neuronales que no fueron estudiadas ni tratadas sobre el desarrollo de este trabajo, simplemente la idea es dar a conocer la existencia de otros tipos de topologías de las redes así como también las diversas metodologías que existen en el mercado para trabajar con redes neuronales o la existencia de otros tipos de paradigmas, como son por ejemplo:

- La lógica difusa, metodología introducida por Zadeh, la ventaja que se tiene es que con una misma sintaxis y semántica se pueden utilizar para expresar cierto conocimiento sobre el problema a resolver.
- Los algoritmos genéticos, metodología que proporciona un mecanismo de búsqueda global en un espacio de posibles soluciones.
- El cálculo probabilístico, mecanismo para evaluar el resultado de aquellos sistemas afectados por datos aleatorios o datos que son afectados por cualquier tipo de incertidumbre probabilística.

Todas estas son posibles soluciones para los posibles problemas a resolver, que dependen de la propia naturaleza del fenómeno en estudio.

También se presenta el caso de la combinación de una metodología con otra, como la presentada por Goodman (*et.al.*) donde presenta un sistema híbrido, es decir, una red neuronal **FUZZY ARTMAP**, como una alternativa a los algoritmos de predicción en el análisis de bases de datos masivos.

Y no olvidar los **Neurochips** que contienen mil o dos mil redes neuronales artificiales o células del cerebro de una rata, como por ejemplo, en el laboratorio de Neuroingeniería de Georgia, en donde han creado un robot híbrido, que contiene una parte mecánica y la otra parte biológica, en la que ésta última parte, opera básicamente como si fuese la actividad neuronal del cerebro de la rata, para ello, las señales neuronales son analizadas en la computadora y son observados los patrones emitidos por las células del cerebro para posteriormente ser trasladados tales patrones dentro del movimiento y comportamiento del robot.

Aplicaciones de gran alcance como la que recientemente se realizó en la última guerra entre Irak y los Estados Unidos de Norteamérica, donde los meteorólogos, oficiales militares e ingenieros, desarrollaron un modelo computacional donde el objetivo principal era predecir cuándo se presentarían tormentas de arena en el desierto de Irak.

Tecnologías, metodologías sobre las redes neuronales donde no se puede establecer cuál es mejor o peor, sino simplemente depende del área o áreas de aplicación y de los resultados que se deseen obtener.

CONTRIBUCIÓN DE ESTA TESIS.

La principal aportación sobre este trabajo de investigación fue el de implementar uno de los sistemas de desarrollo de redes neuronales con los que cuenta el laboratorio de Cibernética "Alejandro Medina" de la Facultad de Ciencias de la Universidad Nacional Autónoma de México, denominado **Dynamind**.

Este sistema de desarrollo permite trabajar en el área de predicción, clasificación y reconocimiento de patrones que para las personas interesadas en introducirse en este mundo maravilloso e interesante de las redes neuronales artificiales realice sus propias aplicaciones apoyándose en el manual proporcionado dentro del apéndice del trabajo de investigación. O simplemente como una opción más existente en el mercado de las redes neuronales para trabajar paralelamente con otros sistemas de desarrollo como son por ejemplo, Matlab, SprinN 2.6, etc.

Aplicaciones que quizás no sean tan relevantes como los descritos anteriormente por los escasos recursos humanos, tecnológicos y económicos que se llegan a presentar dentro de todo trabajo de investigación sin embargo si se pueden realizar aplicaciones interesantes e importantes por ejemplo en sistemas de control, o el simple hecho de conocer la existencia de éste tipo de tecnología que es relativamente joven y además no es muy conocida en el ambiente ingenieril.

Por lo que propongo que se establezca en las carreras de Ingeniero Eléctrico-Electrónico, Ingeniero Eléctrico-mecánico, una asignatura optativa para aquellos que se interesen en el campo de las redes neuronales artificiales.

APÉNDICE

“SISTEMA DE DESARROLLO DYNAMIND”



Objetivo: Conocer y explicar las características principales del sistema de desarrollo DynaMind para generar redes neuronales artificiales.

DynaMind es un sistema de simulación y emulación para elaborar redes neuronales artificiales.

Al ser posible construir redes neuronales; con DynaMind se tiene la posibilidad de encontrar soluciones para diversos problemas y efectuar diversas aplicaciones, por ejemplo:

- Procesamiento de señales: diagnóstico de señales acústicas, detección de imperfecciones, reconocimiento de voz, identificación de señales biológicas, etc.
- Procesamiento de imagen: identificación de huellas digitales, inspección industrial, procesamiento de imagen médico.
- Clasificación: reconocimiento óptico de caracteres, análisis espectral, identificación de células.

ARCHIVOS QUE MANEJA DYNAMIND.

DynaMind maneja diversos tipos de archivos para cada problema. Los archivos son nombrados dentro del mismo problema con sus propias extensiones; estos son:

- * . **net**: Este archivo contiene toda la información estructural de la red neuronal, como es el algoritmo de aprendizaje utilizado, en número de neuronas de entrada, de salida y ocultas, etc.
- * . **IO**: Es el archivo de entrada/salida del proceso de entrenamiento y verificación para el problema en particular.
- * . **set**: Este archivo agrupa a todos los archivos de la red neuronal, es decir, el *.net y el *. IO.
- * . **sig**: Este archivo solo indica la función de transferencia utilizada para la red neuronal.
- * . **iox**: Es el archivo de salidas (resultados) y targets para el problema en particular, pero en un formato IOX.
- * . **out**: Es el archivo de salidas, targets y entradas del problema en particular.

ELABORACIÓN DE LA RED NEURONAL

Son básicamente cinco los pasos para generar una red neuronal con DynaMind, estos son:

1. Preparación de dato o información.
2. Creación de la red neuronal.
3. Entrenamiento de la red neuronal.
4. Verificación de la red neuronal.
5. Utilización de la red neuronal.

I. ELABORACIÓN Y/O PREPARACIÓN DEL DATO.

La base de datos es la información numérica con que se generara la red neuronal, es decir, es la documentación recabada durante un experimento, una encuesta, un estudio de campo, un comportamiento físico, biológico, etc.

1.1 PREPARACIÓN DEL DATO:

La preparación del dato para Dynamind consiste en tomar en cuenta, las siguientes 4 características:

1. Determine que dato va utilizar para el entrenamiento y verificación de la red. Para un aprendizaje más efectivo, el dato debe ser relevante, significativo y completo.
2. Dividir el dato para el entrenamiento y la verificación. El archivo de verificación sirve para validar el entrenamiento de la red, este archivo debe de estar constituido por una tercera parte del dato "general" de IO; ya que las dos terceras partes constituyen el archivo de entrenamiento.

3. Mapear el dato a una cadena de números (simbólicamente). El dato debe estar conformado por números para ser transformados internamente en entradas y salidas numéricas.

3.1 Para ordenar los datos, ó convertir la información cuando ésta es escrita a forma numérica; existen tres métodos, estos son:

- Binary coding
- One - of- N
- Max/ Min scaling

Dependiendo del tipo de problema, es posible elegir alguno de estos métodos o la combinación de ellos.

3.1.1 Binary coding:

El **código binario**, es un método simple de input / output que utiliza los campos **"on"** y **"off"**, donde **on** es asignado a valores de +1 y **off** para valores de -1.0

Este método puede ser utilizado por ejemplo, donde se encuentre el campo, de sexo, es decir, se puede utilizar el código +1.0 para masculino y -1.0 para femenino

3.1.2 Código ONE - OF- N.

Cuando se cuenta con un campo que esta dividido por varias categorías, es aquí donde es necesario mapear el dato mediante el código OF - N; donde una entrada es usada para representar cada categoría.

Por ejemplo un estudio médico sobre una persona puede contener las siguientes categorías:

- 1.Estatura
- 2.Peso
- 3.Edad
- 4.Sexo

La categoría o el vector que es "estudio medico", esta conformado por cuatro campos que son: estatura, peso, edad y sexo.

Entonces, el vector o categoría "estudio medico" puede estar mapeado de la siguiente forma:

+1.0 -1.0 -1.0 -1.0

Donde una de las entradas es encendida con ON, mientras que las otras entradas están asignadas a OFF (asignada a la mínima función de transferencia)

En realidad este método es el mismo que el Binary coding.

3.1.3 Código de escala MAX / MIN:

En algunos casos el dato numérico esta constituido por un rango continuo y pueden ser diversos y/ o diferentes. Cuando se cuenta con este tipo de datos es recomendable utilizar el **código de escala máxima y mínima**.

Esto se debe a que cada entrada análoga debe estar a una escala apropiada dentro del rango seleccionado para la función de transferencia.

Para ello se aplica la siguiente función matemática para cada uno de los puntos que constituyen al dato:

$$X_s = [(X_d - d_{min} / d_{max} - d_{min}) (T_{fMax} - T_{fMin})] + T_{fMin}$$

Donde:

Xs: Es el punto o el dato nuevo que se encontrará dentro de la escala Max / Min.

Xd: Es el dato no escalado (el punto que se va a escalar)

dMin : El valor mínimo del dato a escalar

dMax: El valor máximo del dato a escalar

TFMax: El valor máximo de la función de transferencia.

TFMin: El valor mínimo de la función de transferencia.

Una vez generada la base de datos adecuada se procede a salvarla en la ruta **c:/ndx/examples**, con su nombre y la extensión **.dat (*. dat)**.

El siguiente paso, es crear el archivo de entradas y salidas (*** .io**); para lo cual, desde Ms-Dos, y en ruta, **c:/ndx/examples**, se introduce la siguiente instrucción: **iobuild [nombre del archivo de datos] [nombre de archivo IO]**; inmediatamente DynaMind despliega la siguiente pantalla donde se despliega una serie de preguntas que hay que contestar y estas dependen de la red neuronal que se vaya a generar.

```

                                IOBUILDer Version 2.0
                                [C] 1992 NeuroDynamax,Inc.
Please specify the format for your ASCII file
By choosing a number of the list below.
FEEDFORWARD NETWORKS:
    [1] Inputs followed by outputs
    [2] Inputs and outputs locations in a record to be specified
    [3] Inputs only
RECURRENT NETWORKS (True Time):
    [4] Single or multiple data streams

>>>
IOBUILDer Method 2 ( Feedforward networks)
Input a one line description of the problem:
>>>
Specify the length of a record >>>
Specify which fields in a record are inputs,e.g 1, 8 , 3-6 [1 is a the first field];
>>>
Specify which fields in a record are outputs,e.g 2,5,6-8 [1 is a the first field];
>>>
Input the IO pair label
>>>
Records being parsed:
IOBUILDer finished.

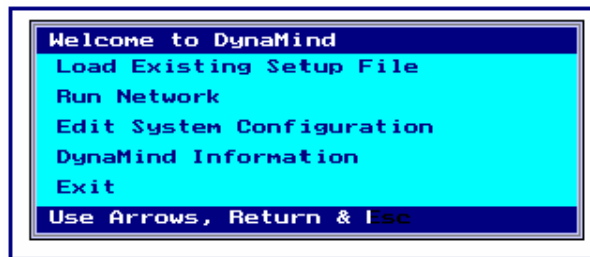
```

Una vez generado el archivo de entradas y salidas, se procede a crear la red neuronal.

II. ELABORACIÓN DE LA RED NEURONAL

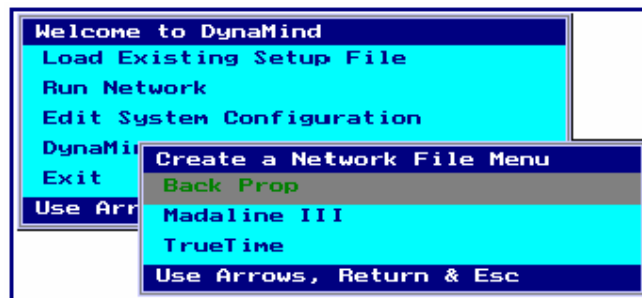
Para la generación de la red neuronal, desde Ms-Dos se teclea `c:/ndx/Dyna` para entrar al ambiente de desarrollo.

En esta imagen se observa la bienvenida al ambiente de desarrollo. Para empezar a generar la red, se selecciona "Edit System Configuration / Network file/Create a Network"

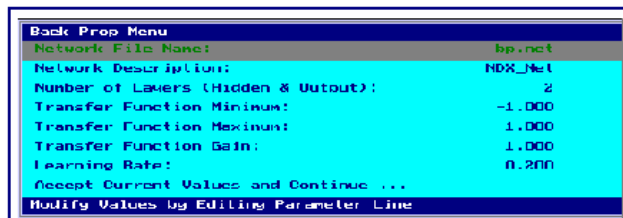


Mostrándose los tres algoritmos de aprendizaje:

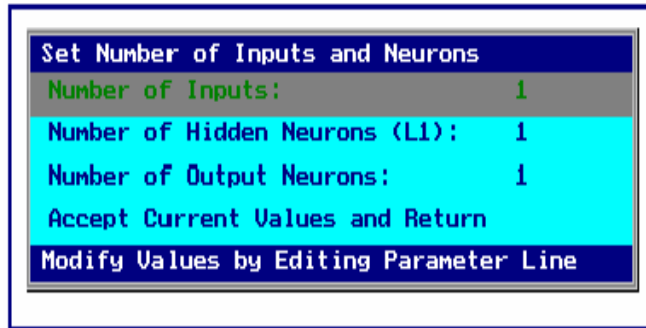
- Back Propagation
- Madaline III
- True Time



Al elegir el algoritmo de aprendizaje Back Propagation, es menester establecer los parámetros correspondientes, como se muestran en esta imagen.



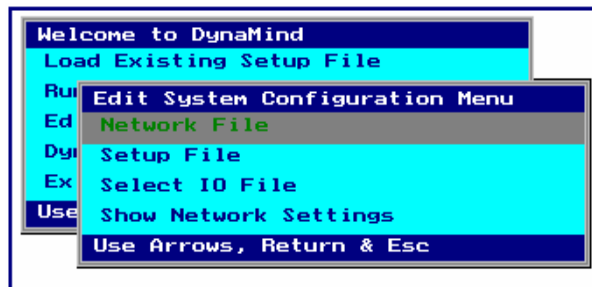
En la siguiente pantalla, se indica la estructura de la red neuronal, es decir, el número de neuronas intermedias (hidden), el número de neuronas tanto de entrada como de salida.



Finalmente se procede a guardar la red neuronal asignándole el nombre con su extensión **.net**

Seleccionar:

**"Edit System Configuration /
Network File / Save"**



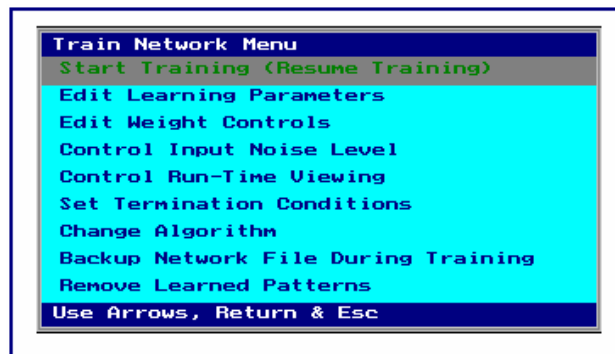
III. ENTRENAMIENTO DE LA RED NEURONAL

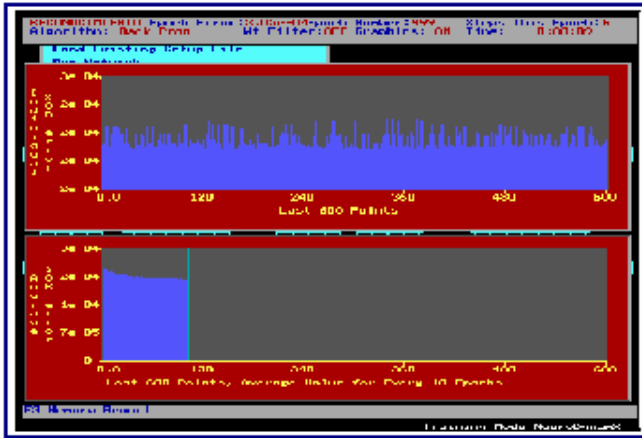
Para efectuar el proceso de entrenamiento del menú principal seleccionar:

" Run Network"

posteriormente;

**"Train Network /Start Training"
(Resume Training)**



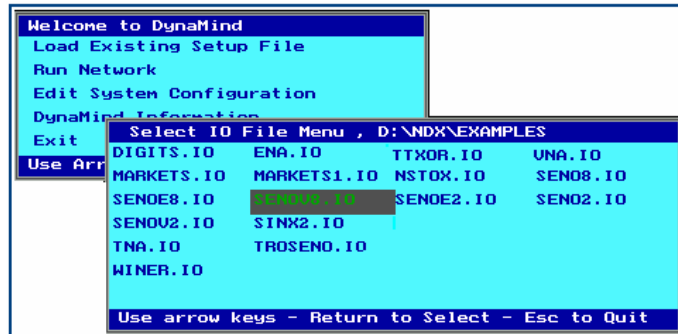


Iniciándose así el proceso de entrenamiento que solo dura unos minutos, dependiendo del tipo de microprocesador con que cuenta la PC, el número de épocas asignadas para el aprendizaje, etc.

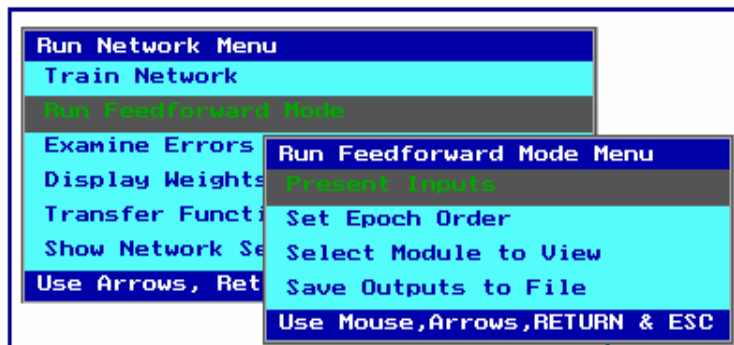
IV. VERIFICACIÓN DE LA RED NEURONAL

Una vez entrenada la red el siguiente paso es verificar su aprendizaje, por lo que es necesario cargar el archivo de entradas y salidas (el de verificación). Hay que recordar que este archivo surge de una partición.

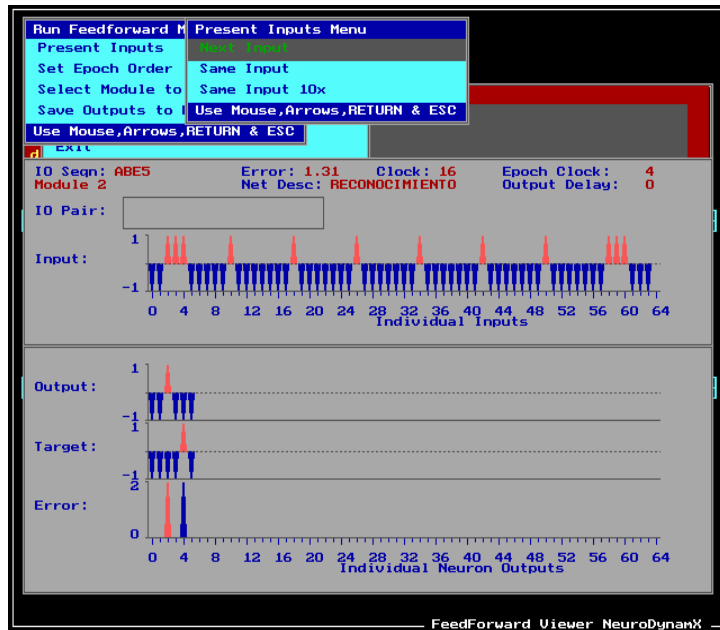
En el menú principal, seleccionar " Edit System Configuration / Select IO File " Para seleccionar así, el archivo de verificación y cargarlo al sistema.



Una vez cargado el archivo, seleccionar: "Run Network /Run Feedforward Mode/ Present Inputs" Para visualizar cada una de las entradas y salidas de este archivo.

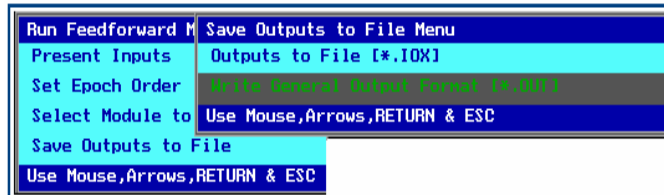


En la parte superior de la gráfica se muestran las entradas; donde los triángulos rojos indican entradas positivas (on) mientras que los azules son negativas (off). En gráfica inferior se muestra las salidas, las targets y el error existente entre entradas y salidas.



Para obtener y observar el archivo de resultados, del menú principal seleccionar:

“Run Network/Run Feedforward/Save Outputs to File/Write General Output Format [*.out]



En la siguiente imagen, se muestra un archivo de resultados donde se exhiben los diversos parámetros y por ende las entradas y salidas de la efectuadas por la red neuronal.

NeuroDynamX, Inc. (c) 1992							
OUT File generated by DynaMind 3.00 X							
IO information:							
File: 2SENSAM.IO							
Number of Inputs: 1							
Number of Outputs: 2							
Number of Patterns: 23							
Setup File: RNS2.SET							
Network information:							
File: RNS2.NET							
Sigmoid: Min: -1.500 Max: 1.500 Gain: 0.250							
Learning Parameters: Alg: TrueTime Learning rate: 0.250							
Perturbation: N/A Output Delay: 1							
Format for each line: IO Seq Label, IO Pair Label, Sum Squared Error							
Outputs (OUT module:neuron), Targets (TRG module:neuron), Inputs							
SeqLab	PairLab	SSError	OUT1:1	OUT1:2	TRG1:1	TRG1:2	IN1
SEQ1	PART1	0.365411	-0.052	0.007	0.552		0
SEQ1	PART2	0.414469	-0.015	0.238	0.629		0.333
SEQ1	PART3	0.241856	0.36	0.388	0.852		0.629
SEQ2	PART1	0.106693	0.651	0.127	0.978		0.333
SEQ2	PART2	0.255992	0.346	0.395	0.852		0.629
SEQ2	PART3	0.098923	0.663	0.491	0.978		0.852

V. UTILIZACIÓN DE LA RED NEURONAL.

Finalmente si son satisfactorios los resultados que presenta la red una vez efectuado el proceso de verificación, el último paso que resta es utilizarla para la aplicación para la cual fue generada.

BIBLIOGRAFÍA

1. DYNAMIND Developer User's Guide
2. DEL RIO MARTÍN, SANZ MOLINA A. *Redes Neuronales y Sistemas Borrosos*, Ed. Ra-Ma.
3. IBARRA, LUZ MARÍA. *Aprender Mejor con Gimnasia Cerebral*, México, D.F. Ed. Garnik Ediciones, 2000 .
4. TAPÍA, RICARDO. *Las células de la mente*, Ed. Fondo de Cultura Económica, ed. 1996 .
5. LEVITON, RICHARD. *Construya su cerebro*, Ed. Diana, ed. 2001 .

- Artículos:

1. HEE-YEAL YU & SUNG – YANG BANG, "An Improved Time Series Prediction by Applying the Layer – by – Layer Learning Method to FIR Neural Networks". *Neural Networks*, Vol. 10. No. 9, pp. 1717.
2. GAIL A. CARPENTER, "Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks". *Neural Networks*, Vol. 10, No. 8, pp. 1473.
3. CHRISTIAN W. OMLIN & C. LEE GILES, "Extraction of Rules From Discrete-time Recurrent Neural Networks". *Neural Networks*, Vol. 9, No. 1, pp. 41.
4. FAOUZI BOUSLAMA & AKIRA ICHIKAWA; "Application of Neural Networks to Fuzzy Control". *Neural Networks*, Vol. 6, pp. 791
5. ANNE EISENBERG; "Wired to the Brain of a Rat, a Robot Takes on the Word" *The New York Times.com*, <http://www.starbucks.com/default.asp?ci>
6. ANDREW C. REVKIN; "A Formidable Enemy: Dust" *The New York Times*, <http://www.nytimes.com/2003/04/01/science/earth/01dust.html?>

- Direcciones Electrónicas:
 1. [http:// www.monografias.com / trabajos10 / expers.shtml](http://www.monografias.com/trabajos10/expers.shtml)
 2. [http:// www.monografias.com / trabajos / redesneuro / redesneuro.shtml](http://www.monografias.com/trabajos/redesneuro/redesneuro.shtml)
 3. [http:// \[www.modeloingenieria.edu.ar\]\(http://www.modeloingenieria.edu.ar\) / utnfrro / orientación / monografías / matich – redes neuronales.pdf](http://www.modeloingenieria.edu.ar/utnfrro/orientacion/monografias/matich-redes%20neuronaes.pdf)
 4. [http:// \[www.geocities.com\]\(http://www.geocities.com\) / txmetsb / sistemas-de tiempo-real.htm](http://www.geocities.com/txmetsb/sistemas-de-tiempo-real.htm)
 5. [http://www.iiia.csic.es/ ~mario/tutorial/RNA_tipos.htm](http://www.iiia.csic.es/~mario/tutorial/RNA_tipos.htm)
 6. <http://www.aernsoft.com/cas/sprinn/index.htm>