

03063



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"REDES NEURONALES Y ALGORITMOS GENÉTICOS:  
ANÁLISIS DE EQUIVALENCIA FUNCIONAL APLICADO A  
SERIES DE TIEMPO EN INVERSIONES"

**T E S I S**

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS  
(COMPUTACIÓN)

**P R E S E N T A :**

**LIC. JAVIER ROSAS HERNÁNDEZ**

DIRECTOR DE TESIS: DR. ÁNGEL FERNANDO KURI MORALES

MÉXICO, D.F.

JUNIO 2005

m345143



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

## **Gracias ...**

A mi mamá, María de Lourdes, por su amor, paciencia y ejemplo de lucha constante.

A mis hermanos, Jaime, Verónica, Celia y Alejandra, por todos los momentos que hemos compartido.

A la memoria de mi padre Isidro.

A mis sobrinos, particularmente Ernesto, Leslie Vanessa, Omar, Lizeth, Valeria Michel y Montserrat, mejor ejemplo de una nueva generación.

A mis tíos y primos, porque indirectamente colaboraron en la elaboración de este trabajo.

A aquellos familiares dondequiera que se encuentren.

A CC.

Al Dr. Ángel Kuri, por compartir sus conocimientos para la elaboración de esta investigación.

A los miembros del jurado por sus aportaciones para mejorar este trabajo, el Dr. Fernando Arámbula, Dra. Katya Rodríguez, Dr. Yu Tang y M. Nicolás Kemper.

A mis alumnos, profesores y compañeros de trabajo por los momentos de aprendizaje compartidos.

A la UNAM, particularmente el IIMAS, la FESA y FESC. Así como el apoyo del CONACYT.

**... a Dios**

# ÍNDICE

Resumen.....	i
<b>Tema I. Series de Tiempo en Análisis de Inversiones .....</b>	<b>1</b>
I.1 Sistema Financiero Mexicano.....	1
I.2 Casas de Bolsa.....	3
I.3 Sociedades de Inversión .....	5
I.4 Series de tiempo en inversiones.....	8
<b>Tema II. Redes Neuronales y Algoritmos Genéticos.....</b>	<b>16</b>
II.1 Redes Neuronales.....	17
II.1.1 Definiciones .....	17
II.1.2 Arquitectura .....	20
II.1.3 Algoritmo de retropropagación .....	26
II.2 Algoritmos Genéticos.....	33
II.2.1 Características.....	35
II.2.2 Operadores.....	37
II.2.3 Algoritmo elitista .....	40
<b>Tema III. Equivalencia entre métodos de Redes Neuronales y Algoritmos Genéticos .....</b>	<b>45</b>
III.1 Análisis paramétrico de Redes Neuronales .....	47
III.2 Análisis multivariado en Algoritmos Genéticos.....	53
III.3 Análisis funcional de equivalencia.....	60
<b>Tema IV. Pronóstico de Sociedades de Inversión.....</b>	<b>67</b>
IV.1 Características del caso de estudio .....	68
IV.2 Pronóstico con Redes Neuronales.....	72
IV.3 Pronóstico con Algoritmos Genéticos .....	84
IV.4 Análisis estadístico de resultados .....	89
<b>Conclusiones .....</b>	<b>94</b>
<b>Glosario .....</b>	<b>95</b>
<b>Referencias .....</b>	<b>99</b>

## RESUMEN

En los últimos años se ha tenido un gran avance en la investigación sobre Inteligencia Artificial, la nueva tecnología ha permitido que se desarrollen técnicas que faciliten el entorno con la computadora de manera más sencilla con la finalidad de imitar el comportamiento y razonamiento humano.

Los algoritmos genéticos y las redes neuronales son dos técnicas muy utilizadas en la actualidad para resolver diferentes tipos de problemas. Algunos estudios integran ambas técnicas con la finalidad de encontrar mejores soluciones disminuyendo así, la complejidad computacional.

Algunos investigadores han considerado estas dos técnicas como algoritmos matemáticos que son implementados en una computadora y, por ende, no son considerados parte de la Inteligencia Artificial. Para solventar los problemas filosóficos que esto conlleva, los investigadores integraron estas técnicas en una nueva rama de la computación denominada Inteligencia Computacional.

La Inteligencia Computacional pretende integrar técnicas y herramientas que aplican modelos matemáticos a la solución de problemas de toda índole. La principal diferencia con otras ramas de la computación es que aquí, los modelos están basados en estudios sobre la inteligencia humana y de cómo el ser humano utiliza su intelecto para resolver problemas. En este sentido, los algoritmos genéticos están basados en el concepto de la evolución natural neodarwiniana y las redes neuronales en la forma en que se procesa la información a través de neuronas.

En el presente trabajo, se presenta el análisis de algoritmos genéticos y redes neuronales aplicados a series de tiempo para pronóstico en el sector financiero. El sector financiero es una de las principales áreas en las que se han aplicado técnicas de Inteligencia Artificial, debido a las implicaciones que tiene para la economía de una nación.

La principal hipótesis que se plantea en esta investigación es mostrar que existe un mecanismo o modelo matemático del comportamiento de las series de tiempo para inversiones en el sector financiero que equipara las técnicas de Redes Neuronales y

Algoritmos Genéticos aplicando los modelos estudiados a través de pronósticos de Sociedades de Inversión.

La principal premisa que sustenta esta hipótesis es que por medio de ambas técnicas es posible realizar un pronóstico en series de tiempo. Esto quiere decir que al hacer el pronóstico en ambas técnicas y obtener los mismos resultados, es posible encontrar un método alternativo que muestre la equivalencia entre ambas técnicas de Inteligencia Artificial, aún cuando en la presente investigación el análisis formal presentado es básico. Por otro lado, hasta el día de hoy, no se han encontrado investigaciones que muestren la equivalencia con fundamentos matemáticos de ambas técnicas.

La presente investigación se desarrolla en los siguientes capítulos:

En el capítulo I, se presenta el análisis de inversiones desde el punto de vista del entorno Financiero Mexicano. El análisis se centra particularmente en el análisis de Sociedades de Inversión en Casas de Bolsa. El mercado financiero se desarrolla a través del tiempo y en este sentido el modelo matemático adecuado es el estudio de series de tiempo.

En el capítulo II, se presentan los fundamentos para ambas técnicas, Redes Neuronales y Algoritmos Genéticos, y también se analizan las características y algoritmos principales.

El capítulo III presenta el análisis polinomial de equivalencia entre ambas técnicas a través de un análisis multivariado en Algoritmos Genéticos y análisis paramétrico en Redes Neuronales.

El capítulo IV valida el modelo presentado en el capítulo III a través de casos prácticos en pronósticos de Sociedades de Inversión que cotizan en la Bolsa Mexicana de Valores.

Finalmente, este trabajo no representa una investigación exhaustiva sobre los algoritmos desarrollados para modelar Series de Tiempo con Algoritmos Genéticos y/o Redes Neuronales, debido a que no es el objetivo de la investigación. Sin embargo, se consideran algunos algoritmos más comúnmente utilizados en el desarrollo de la investigación.

# CAPÍTULO I. ANÁLISIS DE INVERSIONES

*A veces sentimos que lo que hacemos es tan solo una gota en el mar,  
pero el mar sería menos si le faltara una gota.  
Madre Teresa de Calcuta*

El análisis de inversiones en el entorno Financiero Mexicano presenta una oportunidad interesante para el desarrollo de nuevas técnicas debido a la dinámica que se presenta durante un día de operaciones.

El pronóstico de sociedades de inversión es de particular complejidad debido a que éstas se conforman por diferentes instrumentos financieros y que son en algunos casos variables.

La actividad bursátil mexicana ha presentado grandes cambios a lo largo de la historia. En la actualidad un cliente puede estimar el valor del instrumento y comprar libremente acciones en la Bolsa Mexicana de Valores, lo que representa un gran desafío para las instituciones que ofrecen los servicios de intermediarios.

## I.1 Sistema Financiero Mexicano

El Sistema Financiero Mexicano está constituido por tres subsistemas, el Bancario, Bursátil e Instituciones auxiliares de Crédito, los cuales encauzan el flujo de recursos monetarios dinámicamente. Éstos subsistemas son regulados por un Régimen Jurídico Moderno que se interrelaciona dentro del contexto Económico Nacional e Internacional.

Las instituciones que conforman el Sistema Financiero Mexicano captan, administran y canalizan a la inversión, el ahorro tanto de nacionales como extranjeros. El Sistema está integrado principalmente por Grupos Financieros, Banca Comercial, Banca de Desarrollo,

Casas de Bolsa, Aseguradoras, Arrendadoras Financieras, Afianzadoras y Casas de Cambio.

El objetivo central de estas instituciones es captar los recursos económicos de algunas personas para ponerlo a disposición de otras empresas e instituciones gubernamentales que lo requieren para invertirlo. Éstas últimas harán negocios y devolverán el dinero que obtuvieron, además de una cantidad extra (rendimiento), lo cual genera una dinámica en la que el capital es el motor principal del movimiento dentro del sistema.

Las bolsas de valores son mercados organizados que contribuyen a que la canalización de financiamiento se realice de manera libre, eficiente, competitiva, equitativa y transparente, atendiendo a ciertas reglas acordadas previamente por todos los participantes del mercado.

Los organismos que integran el mercado de valores de México, en respuesta a la importancia de este sector dentro de la economía nacional, han experimentado un intenso proceso de modernización e incorporación de tecnología avanzada, situando la actividad bursátil en un nivel de alta eficiencia y competitividad.

Los integrantes del sistema reciben, de manera directa o indirecta, un beneficio económico por el desempeño de su actividad:

- En el caso de una institución privada con fines lucrativos (bancos, casas de bolsa, etc.), mediante una comisión o interés.
- En el de una institución privada no lucrativa (asociaciones, academias, etc.), a través de las cuotas de sus agremiados que si obtienen ganancias económicas.
- En los organismos gubernamentales (Secretaría de Hacienda y Crédito Público, Comisión Nacional Bancaria y de Valores, etc.), vía impuestos sobre las actividades económicas que se generan dentro del sistema o a raíz del mismo y que recauda el gobierno en su conjunto.

El máximo organismo administrativo para el Sistema Financiero Mexicano es la Secretaría de Hacienda y Crédito Público (SHCP). De acuerdo con la Ley Orgánica de la Administración Pública, la SHCP tiene a su cargo: Planear, coordinar, evaluar y vigilar el sistema bancario del país, que comprende al Banco Central, a la Banca Nacional de

Desarrollo y las demás instituciones encargadas de prestar el servicio de Banca y Crédito. La SHCP también es la encargada de ejercer las atribuciones que le señalen las leyes en materia de seguros, fianzas, valores y de organizaciones y actividades auxiliares de crédito.

## 1.2 Casas de Bolsa

En 1975, se promulgó la Ley del Mercado de Valores, con la cual la Bolsa Mexicana de Valores (BMV) dejó de ser una institución de crédito. Las bolsas de valores tienen por objeto facilitar las transacciones con valores y procurar el desarrollo del mercado respectivo. Las bolsas de valores deberán constituirse como sociedades anónimas de capital variable.

La finalidad del mercado bursátil es que los inversionistas adquieran o vendan valores inscritos en la Bolsa, de acuerdo con los precios resultantes del libre juego de la oferta y la demanda. Estas transacciones se realizan en el piso de remates (actualmente es electrónico) de la Bolsa por los intermediarios autorizados.

Las Casas de Bolsa son intermediarios (corredurías) entre oferentes y demandantes de recursos, que además realizan operaciones por cuenta propia con el fin de facilitar la colocación de valores o que coadyuvan a dar mayor estabilidad a los precios de éstos. Las casas de bolsa también prestan servicios de asesoría financiera y bursátil, reciben fondos por concepto de operaciones con valores y están constituidas como sociedades anónimas de capital variable.

En México, la actividad bursátil se encuentra regulada por la Ley del Mercado de Valores y las Leyes Mercantiles que le sean aplicables.

Las casas de bolsa proporcionan los servicios que demandan los inversionistas y empresas emisoras. Las empresas emisoras son aquellas sociedades anónimas que, cubriendo las disposiciones normativas correspondientes, ofrecen al mercado de valores, los títulos representativos del capital social, acciones o valores que amparan un crédito colectivo a su cargo (obligaciones).

Los valores, atendiendo a su naturaleza y rentabilidad se clasifican en renta variable o renta fija. Esta clasificación resulta de la mayor importancia respecto a las estrategias de inversión, ya que marca diferentes situaciones de riesgo.

En el mercado de valores, los recursos son captados por el emisor a través de las casas de bolsa. Para éstos, los títulos de deuda o capital se fraccionan y se colocan para su cotización en la bolsa de valores, mediante oferta pública, de tal manera que cada inversionista adquiera una parte de la deuda o capital representada por un título o valor. Las operaciones de financiamiento se realizan ya sea en la forma de una emisión de capital fraccionada (cuando se trata de acciones), o bien, en forma de una emisión de crédito fraccionado (cuando se trata de Papel Comercial, Obligaciones, Cetes u otros títulos de deuda).

El mercado bursátil se divide en dos tipos: a) El primario donde se emite la oferta pública (colocación) de valores de los organismos o empresas; y b) En el mercado secundario, una vez colocados los valores entre los inversionistas, éstos pueden ser comprados y vendidos en la BMV a través de una casa de bolsa.

Estos mercados son supervisados por cinco instituciones reguladoras, SHCP, Banco de México (BM), Comisión Nacional Bancaria y de Valores (CNBV), Comisión Nacional de Seguros y Fianzas (CNSF) y Comisión Nacional de Sistema de Ahorro para el Retiro (CON SAR).

La amplia gama de opciones de inversión que ofrece el mercado de capitales y el mercado de dinero permite al inversionista establecer diversas estrategias de acuerdo a su liquidez, expectativas y nivel de riesgo que se desea asumir. Toda estrategia de inversión debe conjugar tres variables básicas: riesgo, rentabilidad y liquidez.

- **Riesgo.** Es la probabilidad de que el rendimiento esperado de una inversión no se cumpla, por el contrario, genere pérdidas. El incremento del riesgo está relacionado en proporción directa con la posibilidad de lograr mayores beneficios. Es así como los títulos que implican alto riesgo suelen otorgar una mejor tasa como premio a quien acepta el riesgo.

- Rentabilidad. Expresa la ganancia que es capaz de producir una inversión. Esta ganancia puede ser generada por dividendos, intereses, descuentos y diferencia entre los precios de compra-venta.
- Liquidez. Facilidad con que una inversión puede convertirse en dinero. La liquidez es condición deseable cuando los fondos invertidos podrían destinarse a otros fines en un plazo relativamente incierto, o bien, si se mantiene una estrategia de inversión dinámica de aprovechamiento de oportunidades de mayor rentabilidad.

Con la finalidad de obtener liquidez a través de operaciones bursátiles, una Casa de Bolsa tiene participación en los diferentes mercados financieros del país. Entiéndase a este conjunto como Mercado de Valores.

El término bursátil es concerniente con a las operaciones de la bolsa y a los valores cotizables. Cuando se usa para calificar un título o valor, se pretende significar su bursatilidad, es decir, la relativa facilidad con la que puede comprarse o venderse y proporcionar liquidez.

### 1.3 Sociedades de Inversión

Es más fácil hablar de los valores de acuerdo a sus tipos, pues cada uno tiene diferentes características, pero lo común sigue siendo que valen por lo que representan. Los tipos básicos de valores son:

**Acciones.** Estos valores representan una parte de una empresa, es decir, al poseer una acción se convierte en socio de ella y por ello participa de las ganancias y pérdidas que tenga la compañía. El valor de la acción sube por las expectativas buenas de la compañía y baja por las malas noticias económicas o la disminución de ganancias. Las ganancias (pérdidas) se obtienen al vender o cuando la empresa hace pagos de dividendos (pagos parciales por disminución de capital).

Deuda. Estos valores representan parte de una deuda que gira (emite) la empresa, es decir, cuando la contraparte adquiere deuda, ésta se convierte en uno de los acreedores a los que la empresa debe pagar capital o intereses. Si la contraparte que adquiere la deuda al emitirse se espera a recuperar su dinero invertido al vencimiento del plazo, entonces no es importante la variación del precio del valor; sin embargo, si desea vender antes del vencimiento, para cambiar el valor por dinero, entonces las variaciones en el mercado pueden afectar el dinero que se recupera. Los factores que pueden mermar su dinero son, entre otros: variación a la alza de las tasas, las perspectivas de la empresa y la economía en su conjunto.

A los valores comúnmente se les llama instrumentos y dependiendo del tipo son instrumentos del mercado de capitales (acciones) o instrumentos del mercado de dinero (deuda). A estos mercados se adicionan el de Sociedades de Inversión, el de metales, el sistema internacional de cotizaciones cada uno con sus instrumentos, el mercado de productos derivados y el mercado de cambios o monedas.

Las sociedades de inversión son parte del mercado de valores y representan un mecanismo para invertir en los diversos mercados a través de una cartera manejada por administradores profesionales, de modo que muchas de sus características se pueden asociar con el mercado o los mercados en que invierten.

El decidir operar a través de Sociedades de Inversión, resta la flexibilidad de las inversiones directas en cada instrumento. Las sociedades de inversión tienen las siguientes características:

Los instrumentos propios de este mercado son las acciones de los siguientes tipos de sociedades de inversión:

- Renta variable
- Instrumentos de deuda
- Capitales

Están dirigidas a todo tipo de inversionistas, por tanto en una sociedad de inversión participa más de un inversionista, de modo que la sociedad de inversión busca satisfacer las necesidades de varias personas al mismo tiempo, así que requiere poner reglas para

evitar condiciones desordenadas en su operación y perfil de inversión. Toda esta información viene detallada en el prospecto de información que el inversionista firmará antes de comprar las acciones de dicha sociedad de inversión, reconociendo que ha leído las reglas y que está de acuerdo con ellas.

Para las operaciones en cada sociedad de inversión, su prospecto de información debe señalar lo siguiente:

- Días y horarios de operación.
- Los límites de recompra de acciones que será el máximo que podrá retirar de la sociedad de inversión en un periodo determinado.
- Los límites de tenencia de acciones de la sociedad de inversión por inversionista. La Ley de Sociedades de Inversión permite un máximo de 10% del total de las acciones, sin embargo, el perfil de la sociedad puede imponer límites inferiores.

Las Sociedades de Inversión a veces también son llamadas Fondos de Inversión, pero debido a que existen fondos de inversión y fideicomisos privados que pueden causar confusión por no ser instituciones financieras es conveniente verificar en el Registro de Prestadores de Servicios Financieros de la CONDUSEF<sup>1</sup> aquellas sociedades de inversión autorizadas.

Para invertir en Sociedades de Inversión se puede acudir a Distribuidoras de Sociedades de Inversión, cuya función principal es la distribución (venta y recompra) de las acciones de las sociedades de inversión que la contratan. La distribución de las sociedades se puede realizar a través de diferentes instituciones financieras:

- Distribuidoras de sociedades de inversión independientes o afiliadas a grupos financieros.
- Administradoras de activos de sociedades de inversión.
- Bancos.
- Casas de bolsa.
- Instituciones de seguros.
- Organizaciones auxiliares de crédito.

---

<sup>1</sup> CONDUSEF Comisión Nacional para la Protección y Defensa de los Usuarios de Servicios Financieros, [www.conducef.gob.mx](http://www.conducef.gob.mx)

- Casas de cambio.
- Sociedades financieras de objeto limitado (SOFOLSES).

Los precios e índices diarios de las sociedades de inversión los difunde la Bolsa Mexicana de Valores, S.A. de C.V. a través de:

- Boletín bursátil en la sección mercado de capitales.
- Página de la BMV, <http://www.bmv.com.mx/bmv/opsocinv.html>
- Página <http://www.mapafinanciero.com.mx/JSP/v21/mayorescambios.jsp>, que corresponde a la página del mapa financiero.
- Página de Banamex, <http://www.banamex.com> y particularmente en el sitio de Accitrade <http://www.accitrade.com> perteneciente a la Casa de Bolsa Accival del Grupo Financiero Banamex Citigroup.

Al invertir en Sociedades de Inversión, se tiene un costo por cobro de comisiones por concepto de administración y custodia, y otro por operación que se realice. El primero es similar a la comisión que cobra un banco por concepto de anualidad y es independiente a la comisión que se cobra por una operación. Es importante considerar que tanto las comisiones por compra-venta y las administrativas, así como por otros conceptos, disminuyen la rentabilidad que se obtiene de la inversión.

#### I.4 Series de tiempo en inversiones

Realizar un pronóstico a través de series de tiempo ha sido exitosamente utilizado para ayudar en la toma de decisiones en diversas aplicaciones correspondientes a un gran número de áreas.

Se han desarrollado un gran número de técnicas para modelar y predecir series de tiempo, entre las que destacan la metodología de Box-Jenkins (1994) y las Redes Neuronales Artificiales. Box-Jenkins ha sido ampliamente utilizado debido a la formalidad de la técnica; sin embargo, esta técnica solo es capaz de construir modelos lineales. En contraste, las Redes Neuronales Artificiales permiten modelar funciones no lineales, la

principal dificultad en estas técnicas se refieren a elegir la arquitectura apropiada para la red que ayude a construir el modelo de series de tiempo a pronosticar.

Diversos problemas del mundo real son observados durante intervalos de tiempo regulares y los datos resultantes pueden ser usados para diferentes propósitos, por ejemplo, para registrar el consumo mensual de electricidad en una casa e identificar algunos patrones de uso, o para registrar continuamente la temperatura de un proceso químico con la finalidad de inferir información de utilidad acerca del proceso.

De acuerdo a Box, G. E., una serie de tiempo es un conjunto de observaciones generadas secuencialmente en el tiempo. El análisis de series de tiempo es un proceso inductivo que es adecuado para inferir a partir de observaciones, información acerca del fenómeno del cuál se generaron los datos.

Uno de los principales objetivos del análisis de series de tiempo es la construcción de modelos de predicción o pronósticos. Estos modelos pueden ser utilizados para ayudar a la toma de decisiones en una gran diversidad de dominios, disminuyendo los riesgos asociados a la decisión de un experto. Entre las aplicaciones que más utilizan el análisis de series de tiempo para predicción, sobresalen la planeación de líneas de producción, finanzas y el control de procesos industriales.

El diseño de modelos de predicción es un proceso que involucra el conocimiento del modelo de series de tiempo así como del dominio de la aplicación. En consecuencia, diferentes modelos arrojan resultados diferentes.

El modelo de Box y Jenkins es lineal con pocos parámetros. En consecuencia, es más fácil y rápido diseñar un modelo de este tipo que un modelo no lineal. Sin embargo, los modelos lineales no describen adecuadamente todos los fenómenos. Un modelo alternativo que implementa modelos no lineales es utilizar Redes Neuronales Artificiales, Drossu y Obradovic (1995) puntualizan dos características principales de los modelos con redes neuronales para predicción en series de tiempo: la capacidad para aproximar funciones y la relación directa con modelos estadísticos clásicos, como el Box-Jenkins.

En términos formales, una serie de tiempo es una secuencia de vectores, dependiente del tiempo  $t$

$$\vec{x}(t), \quad t = 0, 1, \dots \quad (1.1)$$

Los componentes de los vectores pueden ser alguna variable observada, por ejemplo:

- la temperatura del aire en un edificio
- el precio de un artículo en particular
- el número de pájaros en una ciudad
- la cantidad de agua consumida en una comunidad
- el precio de una sociedad de inversión en cada día

Teóricamente,  $\vec{x}$  puede ser considerado como una función continua de la variable  $t$  en el tiempo. Sin embargo, para efectos prácticos, el tiempo es usualmente visto en términos de periodos de tiempo discretos, adelantando una instancia de  $\vec{x}$  en cada punto final de un intervalo de tiempo, usualmente un tamaño fijo. El tamaño del intervalo de tiempo, generalmente depende del problema de estudio y puede ser desde milisegundos, horas, días o inclusive años. Esta es la razón por la que se dice una secuencia o series de tiempo.

En muchas situaciones, las observaciones están disponibles sólo en pasos de tiempo discretos. En otros casos, los valores son acumulados o promediados sobre el intervalo de tiempo para obtener las series. En dominios donde el tiempo es continuo, se deberá medir la variable en puntos particulares en el intervalo del tiempo. Elegir la frecuencia de esos puntos, es un parámetro crucial debido a que diferentes frecuencias pueden esencialmente cambiar las características principales de los resultados en la serie de tiempo.

Si el vector  $\vec{x}$  contiene solo un componente, el cual es el caso en la mayoría de las aplicaciones, se dice que se trata de una serie de tiempo univariada, en otro caso, se conoce como multivariada.

Dependiendo del objetivo del análisis de la serie de tiempo, los tipos de aplicaciones típicas para series de tiempo se distinguen como sigue:

1. Pronóstico de desarrollos futuros en series de tiempo

2. Clasificación de series de tiempo en una de varias clases
3. Descripción de series de tiempo en términos de los parámetros de un modelo
4. Mapeo de una serie de tiempo en otra

Las aplicaciones del tipo 1 son las más utilizadas. Desde economía hasta en planeación de energía se pueden encontrar un sin fin de problemas en los que se desea predecir los valores futuros para el vector  $\vec{x}$ . Por ejemplo, para decidir una estrategia de compra en un mercado financiero.

Formalmente, el problema de series de tiempo se describe como sigue: Encontrar una función  $F: R^{k+1} \rightarrow R^k$ , donde  $k$  es la dimensión de  $\vec{x}$ , tal que se obtenga un estimado  $\hat{\vec{x}}(t+d)$  del vector  $\vec{x}$  en el tiempo  $t+d$ , dados los valores de  $\vec{x}$  hasta el tiempo  $t$ , más un número de variables independientes del tiempo (características exógenas)  $\pi_i$

$$\hat{\vec{x}}(t+d) = F(\vec{x}(t), \vec{x}(t-1), \dots, \pi_1, \dots, \pi_l) \quad (I.ii)$$

$d$  es llamada el retraso para la predicción. Típicamente,  $d = 1$ , lo cual significa que el vector subsecuente será estimado, sin embargo  $d$  puede tomar cualquier valor mayor o igual a 1. Por otro lado, las variables exógenas  $\pi_i$  pueden ser decisivas en algunas aplicaciones, por ejemplo, el tamaño del cuarto donde se está midiendo la temperatura.

En este sentido, un pronóstico llega a ser un problema de aproximación de una función, donde el método elegido es aplicado para aproximar el valor continuo de la función  $F$  tan exacto como sea posible. Visto de esta forma, pronosticar puede ser comparado con la aproximación de funciones o problemas de regresión que incluyen vectores de datos estáticos.

Generalmente, la evaluación del rendimiento del pronóstico realizado es generada al computar una medida de error  $E$  sobre un número de elementos en la serie de tiempo, tal como una validación o un conjunto de prueba:

$$E = \sum_{i=0}^N e(\hat{\vec{x}}(t-i), \vec{x}(t-i)) \quad (I.iii)$$

$e$  es una función que mide un simple error entre el estimado (pronóstico) y el elemento actual en la secuencia. Típicamente, se utiliza una medida de distancia (Euclidiana,

Mahalanobis, Hamming, etc.), pero dependiendo del problema se puede inclusive utilizar una función que calcule el costo de realizar el pronóstico de  $\hat{x}(t+d)$  incorrectamente.

En muchos problemas de pronóstico, el valor exacto de  $\hat{x}(t+d)$  es no requerido, solo una indicación de si  $\hat{x}(t+d)$  es mayor (incrementa) o menor (disminuye) que  $\hat{x}$  o aproximadamente es el mismo. Si es este el caso, el problema de pronóstico cambia a un problema de clasificación, mapeando la secuencia de la serie de tiempo en clases de aumento, incremento o, inclusive, constante.

En términos más generales, la clasificación en series de tiempo (aplicación tipo 2), se puede expresar como el problema de contar una función  $F_c: R^{k+1} \rightarrow B^k$  asignando una clase a la serie de tiempo.

$$F_c: (\bar{x}(t), \bar{x}(t-1), \dots, \pi_1, \dots, \pi_j) \rightarrow c_i \in C \quad (1.4)$$

donde  $C$  es el conjunto de etiquetas para las clases disponibles. Formalmente, no hay una diferencia esencial para el problema de aproximación de funciones, es decir, la clasificación puede ser considerada como un caso especial de aproximación de funciones, donde la función que será aproximada mapea a vectores continuos con valores binarios. Típicamente, una función de error para este tipo de aplicación tiene la forma

$$E = 1 - \frac{1}{N} \sum_{i=1}^N \delta_{\hat{c}_i, c_i} \quad (1.5)$$

Expresando los porcentajes de entradas que no han sido correctamente asignadas a la clase deseada.  $\delta_{i,j}$  es el símbolo delta de Kronecker, es decir,  $\delta_{i,j} = 1$  si y sólo si  $i = j$ , y 0 en otro caso  $c_i$  es la etiqueta para la clase conocida de la entrada  $i$ .

La aplicación del tipo 3 (modelado de series de tiempo) está implícitamente contenida en los dos tipos anteriores, pronóstico y clasificación. La función  $F$  puede ser considerada como un modelo de la serie de tiempo la cuál es capaz de generar las series, a través de sustituir sucesivamente las entradas por los valores estimados. Un modelo útil debería tener menos parámetros que elementos en la serie de tiempo, sus parámetros deberían ser vistos como una característica sobresaliente en el modelo.

El último tipo (aplicación 4), se debe considerar que el modelado es una forma de mapear una serie de tiempo en si misma, es decir, encontrar los parámetros para el modelo de series de tiempo con la finalidad de reproducir las series, por consiguiente, el mapeo de una serie de tiempo a otra diferente, es posible. Por ejemplo, el pronóstico del valor de series de tiempo para el precio del IPC (Índice de Precios al Consumidor) en la BMV da los valores para otra, las tasas de interés. El mapeo entre series de tiempo es un caso especial de series de tiempo multivariada.

Debido a la medición de errores y la influencia de factores desconocidos o incontrolables, casi siempre se tiene que asumir que aún el modelo óptimo acarreará un error residual el cuál no podrá ser eliminado. En la mayoría de los casos, este error es asumido como el resultado de un proceso de ruido, es decir, producido aleatoriamente por una fuente desconocida. Entonces, el modelo de pronóstico se extiende como sigue

$$\hat{\bar{x}}(t+d) = F(\bar{x}(t), \bar{x}(t-1), \dots, \pi_1, \dots, \pi_l) + \tilde{e}(t) \quad (I.vi)$$

Este ruido  $\tilde{e}(t)$  no puede ser incluido explícitamente en el modelo, sin embargo, muchos modelos asumen una cierta característica del ruido a través de parámetros que pueden ser incluidos en el modelo.

Solamente en muy pocos casos es apropiado usar las medidas observadas inmediatamente para procesamiento. Sin embargo, en la mayoría de los casos, es necesario preanalizar y preprocesar las series de tiempo para asegurar una salida óptima del procesamiento. Por ejemplo, en la mayoría de los métodos es conveniente identificar la tendencia y periodicidad en los datos.

Gráficamente, una serie de tiempo se representa en una forma bidimensional en donde el eje x representa el periodo y en el eje y se presenta el valor de la variable en esa unidad de tiempo. Por ejemplo, la siguiente figura muestra la serie de tiempo para una sociedad de inversión; en el eje x se presentan los días hábiles del periodo, mientras que en el eje y se asignan los precios para esas sociedades.

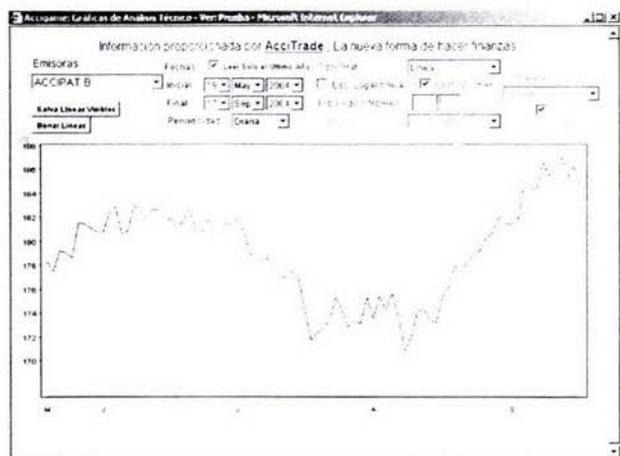


Fig. I.1 Ejemplo de la gráfica para una serie de tiempo

Además, la unidad de tiempo para una serie de tiempo varía dependiendo del problema. En la siguiente gráfica se considera una unidad de medida mensual.

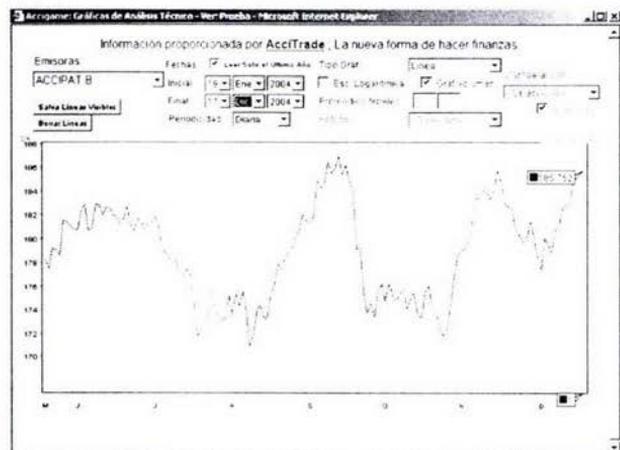


Fig. I.2 Serie de tiempo mensual

Por otro lado, el análisis de series de tiempo resulta en ocasiones intuitivo y para ayudar a la toma de decisiones es posible utilizar inclusive modelos sencillos como análisis comparativo entre emisoras y cálculo de promedios móviles, como se muestra en las siguientes figuras.



Fig. I.3 Análisis gráfico comparativo de emisoras

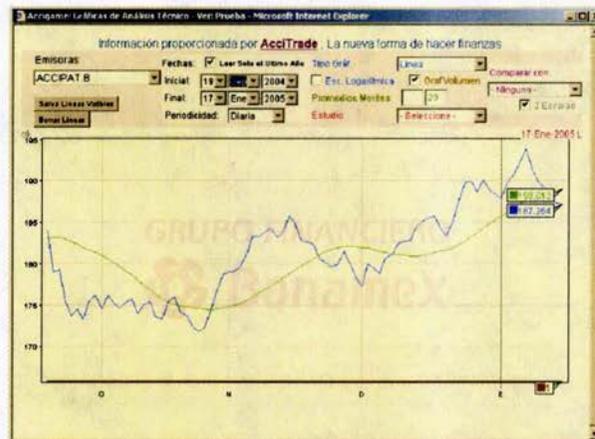


Fig. I.4 Análisis gráfico con promedios móviles

# CAPÍTULO II. ALGORITMOS GENÉTICOS Y REDES NEURONALES

*A veces podemos pasarnos años sin vivir en absoluto  
y de pronto toda nuestra vida se concentra en un solo instante.*

*Oscar Wilde*

El objetivo de la Inteligencia Artificial (IA) es desarrollar paradigmas o algoritmos aplicados en máquinas para realizar tareas cognitivas en las que el ser humano es actualmente mejor (Sage, 1990). En este sentido, un sistema IA deberá ser capaz de hacer tres cosas: 1) almacenar el conocimiento; 2) aplicar el conocimiento almacenado para resolver problemas; y 3) adquirir nuevo conocimiento a través de la experiencia.

En los últimos años, las redes neuronales y algoritmos genéticos han demostrado una gran capacidad para resolver diversos problemas, almacenar el conocimiento, utilizarlo e inclusive generar el conocimiento. Considerando estas dos técnicas como parte de la Inteligencia Artificial, en general, están basadas en principios simples de la naturaleza aplicados a modelos matemáticos que con la ayuda de la computadora representan una gran ventaja: la iteración no lineal.

Las redes neuronales a través del aprendizaje de retropropagación han mostrado buenos resultados en diversos tipos de problemas. Sin embargo, la elección de los parámetros básicos, la topología de la red, tasa de aprendizaje y pesos iniciales, frecuentemente determina el proceso de entrenamiento. La selección de estos parámetros en ocasiones recae en el uso de reglas prácticas, sin embargo los valores que ellos toman es lo más importante para resolver exitosamente el problema.

Los algoritmos genéticos son métodos de búsqueda global que están basados en los principios de selección natural, cruzamiento y mutación. Los algoritmos genéticos son utilizados para resolver problemas de optimización. Considerado que se han desarrollado

varios tipos de algoritmos, la característica común en todos es que utilizan una función de rendimiento o *fitness* la cual depende de una o más variables y de parámetros desconocidos.

## II. 1 Redes Neuronales

El desarrollo de las redes neuronales artificiales, o simplemente "red neuronal", fue motivado por el análisis de cómo el cerebro humano computa, que es sustancialmente diferente al procesamiento digital de una computadora.

Como se verá a través de este capítulo, una red neuronal deriva su poder de cómputo principalmente por dos cosas: a) su estructura masiva de distribución paralela y b) la habilidad para aprender y generalizar<sup>1</sup>. Estas dos capacidades de procesamiento de información hacen posible que una red neuronal resuelva problemas complejos, actualmente intratables con otras técnicas.

En la práctica, las redes neuronales no generan una solución aislada, sino que necesitan integrarse a metodologías de sistemas de ingeniería, por lo que las redes neuronales son asignadas a un subconjunto de tareas de un problema descompuesto en diversas tareas.

En la literatura, las redes neuronales también se refieren a neurocomputadoras, redes conexionistas o procesadores distribuidos paralelos.

### II.1.1 Definiciones

Las redes neuronales se inspiraron, en un principio, en la analogía con un cerebro humano. El cerebro humano es un sistema de proceso de información que realiza cálculos muy complejos, no lineales y paralelos. Para ello, tiene la capacidad de organizar las neuronas con la finalidad de realizar ciertos cálculos, por ejemplo, reconocimiento de patrones, percepción y control motor. Un cerebro tiene una estructura compleja y la

---

<sup>1</sup> Generalización se refiere a que la red neuronal produce salidas *razonables* de acuerdo a las entradas recibidas y que no han sido encontradas durante el proceso de entrenamiento (aprendizaje).

habilidad para construir sus propias reglas es conocida como "experiencia". La experiencia es construida a través del tiempo.

La red es implementada utilizando componentes electrónicos o es simulada con software en una computadora digital. Para un mejor rendimiento, las redes neuronales emplean una interconexión masiva de celdas de cómputo simples, conocidas como "neuronas" o "unidades de procesamiento".

Alexander Morton (1990), define una red neuronal vista como una máquina adaptable: "Una red neuronal es un procesador distribuido en paralelo masivamente compuesto de unidades simples de procesamiento, el cual es propenso naturalmente para almacenar conocimiento experimental y hacerlo disponible para utilizarlo. Esto es semejante al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red desde su medio ambiente a través de un proceso de aprendizaje<sup>2</sup>.
2. Los pesos de conexión entre las neuronas, conocido como los pesos sinápticos, son utilizados para almacenar el conocimiento adquirido".

El uso de redes neuronales ofrece las siguientes propiedades y capacidades:

1. No linealidad. Una neurona artificial puede ser lineal o no. Una red neuronal se compone de la interconexión de neuronas no lineales.
2. Mapeo Entrada-Salida. La red aprende de los ejemplos a través de construir un mapeo de entrada-salida para el problema, se trata del paradigma más popular llamado aprendizaje con maestro o aprendizaje supervisado.
3. Adaptabilidad. Las redes neuronales tienen la capacidad de adaptar sus pesos sinápticos para producir los cambios en el medio ambiente.
4. Respuesta evidente. Utilizada principalmente en clasificación de patrones que ayuda a decidir acerca de patrones ambiguos.
5. Información contextual. El conocimiento es representado por la estructura de la red neuronal y el estado de activación. **Cada neurona en la red es potencialmente afectada por la actividad global de todas las neuronas en la red.**
6. Tolerancia a fallas. Debido a la naturaleza distribuida de la información almacenada en una red, a medida que se acerca a la respuesta el rendimiento de

---

<sup>2</sup> El procedimiento utilizado para realizar el proceso de aprendizaje es llamado algoritmo de aprendizaje.

la red se degrada. Sin embargo, para evitar ésto es necesario hacer medidas correctivas en el diseño del algoritmo utilizado para entrenar la red.

7. Implementación VLSI. La tecnología VLSI, very-large-scale-integrated es utilizada debido a la naturaleza paralela masiva de una red neuronal haciendo más rápido el cálculo en ciertas tareas.
8. Uniformidad de análisis y diseño. Básicamente, las redes neuronales se acoplan universalmente como procesadoras de información en el sentido de que se cuenta con neuronas, se comparte teorías y algoritmos de aprendizaje y la integración de redes modulares.
9. Analogía neurobiológica. El diseño de una red neuronal es motivado por la analogía con el cerebro incluyendo que el procesamiento paralelo de tolerancia a fallas no es sólo posible físicamente sino rápido y poderoso. Los ingenieros buscan en la neurobiología nuevas ideas para resolver problemas más complejos.

La característica principal en una red neuronal es la capacidad de la red para aprender del medio ambiente y mejorar su rendimiento a través del aprendizaje. Idealmente, la red obtiene más conocimiento acerca de su medio ambiente después de cada iteración del proceso de aprendizaje.

Haykin (1999) define el aprendizaje en el contexto de una red neuronal como "un proceso mediante el cual los parámetros libres de una red neuronal son adaptados a un proceso de estimulación por el medio ambiente en el que se encuentra la red. El tipo de aprendizaje es determinado por la manera en la que se realizan los cambios de los parámetros".

De acuerdo a la definición anterior, el proceso de aprendizaje implica la siguiente secuencia de eventos.

1. La red neuronal es estimulada por un medio ambiente.
2. La red neuronal **cambia experimentalmente** en los parámetros libres como **resultado de la estimulación.**
3. La red neuronal **responde de una forma nueva** al medio ambiente debido a los cambios ocurridos en la estructura interna.

Un algoritmo de aprendizaje es un conjunto de reglas bien definidas para la solución del problema de aprendizaje. Básicamente, los algoritmos difieren en la forma en la que

ajustan los pesos sinápticos de una neurona. Las principales reglas de aprendizaje para el diseño de redes neuronales son:

- Aprendizaje de corrección de error, trata de encontrar un filtro óptimo.
- Aprendizaje basado en memoria, opera memorizando el entrenamiento de los datos explícitamente.
- Aprendizaje Hebbiano, inspirado en consideraciones neurobiológicas con reglas de aprendizaje autoorganizados no supervisado.
- Aprendizaje competitivo, también inspirado en neurobiología.
- Aprendizaje de Boltzmann, basado en mecanismos estadísticos.

## II.1.2 Arquitectura

Una neurona es una unidad de procesamiento de información, la cual es fundamental para la operación de una red neuronal. En la Figura II.1 se muestra el modelo de un perceptrón, el cual representa la base para el diseño de redes neuronales artificiales.

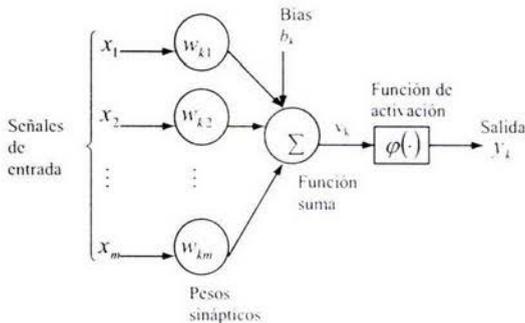


Figura II.1 Modelo no lineal de una neurona

En la Figura II.1, se observan tres elementos básicos del modelo neuronal:

- Un conjunto de sinapsis o ligas conectadas, cada una de las cuales es caracterizada por un peso o una fuerza. Particularmente, una señal  $x_j$  en la entrada de la sinapsis  $j$  conectada a la neurona  $k$  es multiplicada por el peso sináptico  $w_{kj}$ . El primer subíndice se refiere a la neurona y el segundo a la entrada final de la sinapsis para los cuales se refiere el peso. A diferencia de la sinapsis en el

cerebro, el peso sináptico de una red artificial puede ser números negativos y positivos.

2. Un sumador para adicionar las señales de entrada, pesado por las respectivas sinapsis de la neurona, las operaciones constituyen una combinación lineal.
3. Una función de activación que limita la amplitud de la salida de una neurona a un rango de valores finitos. Generalmente, el rango normalizado se encuentra en el intervalo  $[0, 1]$  o alternativamente  $[-1, 1]$ .

El modelo neuronal incluye una medida de tendencia umbral (*bias*), denotada por  $b_k$ . El bias tiene el efecto de incrementar o disminuir la entrada a la red de la función de activación, dependiendo de si el valor es positivo o negativo, respectivamente.

En términos matemáticos, una neurona se describe a través de las siguientes ecuaciones:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (\text{II.i})$$

$$y_k = \varphi(u_k + b_k) \quad (\text{II.ii})$$

donde  $x_1, x_2, \dots, x_m$  son las entradas;  $w_{k1}, w_{k2}, \dots, w_{km}$  los pesos de la neurona  $k$ ;  $u_k$  es la salida de la combinación lineal de las entradas;  $b_k$  es la bias;  $\varphi(\cdot)$  es la función de activación; y  $y_k$  es la salida de la neurona.

Considerando que la bias  $b_k$  es un parámetro externo de una neurona artificial  $k$  y que tiene el efecto de aplicar una afinación a la salida  $u_k$ , sea

$$v_k = u_k + b_k \quad (\text{II.iii})$$

Equivalentemente, se combinan las ecuaciones ( II.i ) y ( II.iii ) como sigue

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (\text{II.iv})$$

$$y_k = \varphi(v_k) \quad (\text{II.v})$$

En la ecuación ( II.iv ) se agrega una nueva sinapsis cuya entrada es  $x_0 = +1$  y su peso es  $w_{k0} = b_k$ . En la Figura II.2 se reformula el modelo para una neurona  $k$ . En esta figura, el efecto de la bias es contabilizado de dos formas: 1) agregando una nueva señal de

entrada con el valor de +1, y 2) agregando un nuevo peso sináptico igual a la bias  $b_k$ . Aunque el modelo de las Figuras II.1 y II.2 son aparentemente diferentes, matemáticamente son equivalentes.

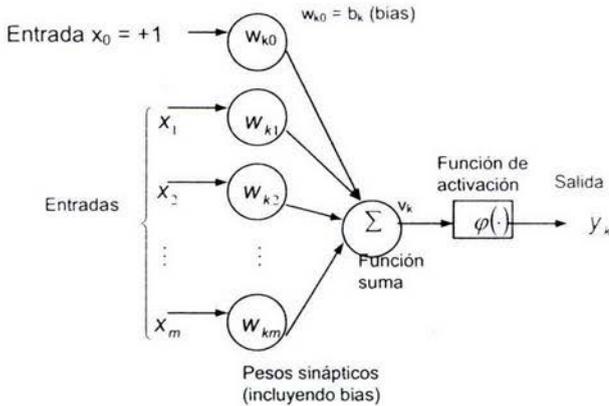


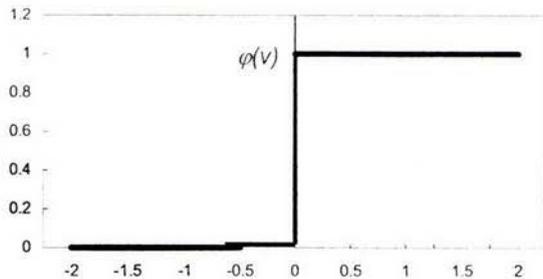
Figura II.2 Otro modelo no lineal de una neurona

### Funciones de activación

La función de activación, denotada por  $\varphi(v)$ , define la salida de una neurona en términos del campo local  $v$  inducido. Básicamente, se utilizan tres tipos de funciones de activación.

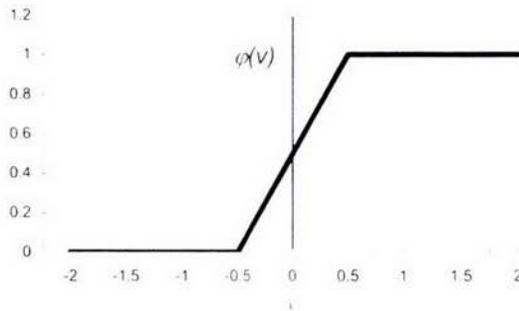
#### 1. Función Escalón

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



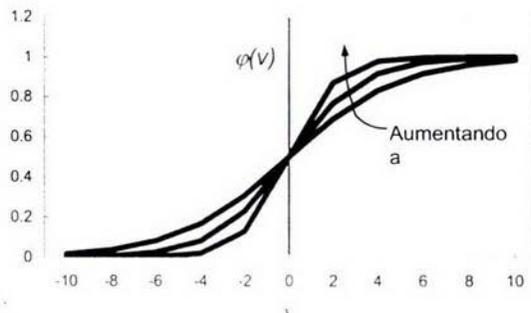
2. Función Lineal

$$\varphi(v) = \begin{cases} 1 & v \geq +\frac{1}{2} \\ v & +\frac{1}{2} > v > -\frac{1}{2} \\ 0 & v \leq -\frac{1}{2} \end{cases}$$



3. Función Sigmoide

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$



La función sigmoide, definida por la función logística con forma de s aplastada, es la función de activación más utilizada en redes neuronales debido a que es una función que incrementa exhibiendo un balance entre un ambiente lineal y no lineal.

Una función escalón asume valores de 0 o 1, la función sigmoide asume un rango continuo entre 0 y 1. Además, la función sigmoide es diferenciable mientras que la función escalón no.

Es de notar que las tres funciones anteriores comprenden un rango entre 0 y 1. Sin embargo, algunas ocasiones es necesario que la función de activación tenga un rango de -1 a 1. En este caso la función escalón se define como

$$\varphi(v) = \begin{cases} 1 & \text{si } v > 0 \\ 0 & \text{si } v = 0 \\ -1 & \text{si } v < 0 \end{cases} \quad (\text{II.vi})$$

y corresponde a la función signo. Por otra parte, la función correspondiente a la función sigmoide se utiliza la función tangente hiperbólica, definida como

$$\varphi(v) = \tanh(v) \quad (\text{II.vii})$$

### Arquitecturas de red

La manera en la cual las neuronas de una red neuronal están estructuradas está ligada con el algoritmo de aprendizaje utilizado para entrenar la red. Sin embargo, en ocasiones se habla de algoritmos de aprendizaje (reglas) utilizadas en diseño de redes neuronales (estructuras). En general, hay tres clases de arquitecturas de red:

#### 1. Redes progresivas de una capa.

Es la forma más simple de una red neuronal compuesta de capas, se tiene una capa de entrada de nodos fuentes que proyecta sobre una capa de salida de neuronas (nodos calculados), pero no viceversa, es decir, es una red acíclica o hacia adelante. La capa de entrada no se cuenta debido a que en ella no se realizan cálculos. Esta red es llamada red simple o red de una capa y se muestra en la Figura II.3.

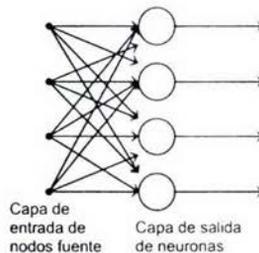


Figura II.3 Red progresiva con una capa de neuronas

#### 2. Redes progresivas multicapa.

En esta clase de redes se tiene la presencia de una o más capas escondidas cuyos nodos calculados son llamados neuronas escondidas o unidades escondidas. Debido a la(s) capa(s) ocultas, la red adquiere una perspectiva global a pesar de la conectividad local y

las interacciones extras entre neuronas. Además, agregando al menos una capa, la red es capaz de extraer estadísticas de orden mayor.

Típicamente, las neuronas en cada capa de la red tienen sus entradas de las salidas de la capa precedente. La gráfica de la Figura II.4, muestra una red multicapa con una capa oculta.

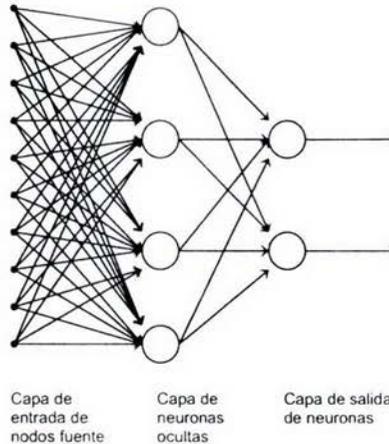


Figura II.4 Red progresiva completamente conectada con una capa oculta y una de salida.

Para abreviar, la red de la Figura II.4 es referida como una red 10-4-2 porque tiene 10 nodos fuente, 4 neuronas ocultas y 2 neuronas de salida. Además, se dice que la red es fuertemente conectada en el sentido de que cada nodo en cada capa está conectado con cada nodo de la capa hacia delante.

### 3. Redes recurrentes.

Una red neuronal recurrente es aquella donde existe al menos un ciclo hacia atrás. Por ejemplo, una red que consiste de una capa de neuronas con una neurona de salida que regresa las entradas de las otras neuronas, es decir, la salida de una neurona representa la entrada para la propia neurona. El uso de ciclos involucra el uso de ramas particulares compuestas de elementos de retraso de unidad (denotados por  $z^{-1}$ ), lo que origina un ambiente dinámico no lineal. En la Figura II.5 se muestra una red neuronal con una capa oculta.

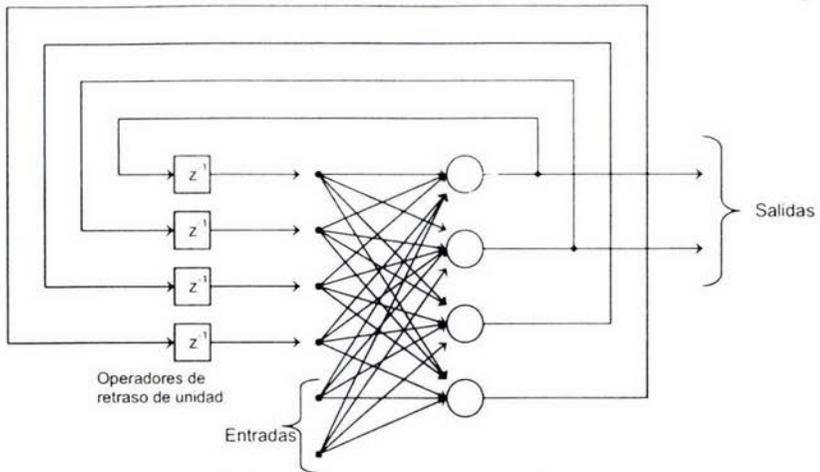


Figura II.5 Red recurrente con neuronas ocultas.

### II.1.3 Algoritmo de retropropagación

En el periodo en que se crearon las redes neuronales (1943-1958) varios investigadores tuvieron contribuciones importantes:

- McCulloch y Pitts (1943) introdujeron la idea de las redes neuronales como máquinas calculadoras.
- Hebb (1949) propuso la primer regla para aprendizaje autoorganizado.
- Rosenblatt (1958) propuso el perceptrón como el primer modelo para aprendizaje con maestro, es decir, aprendizaje supervisado.

El perceptrón es la forma más simple de una red neuronal utilizada para la clasificación de patrones que se dice son linealmente separables, es decir, patrones que se encuentran en lados opuestos de un hiperplano. Básicamente, consiste de una simple neurona con un peso sináptico ajustable y el umbral. El algoritmo utilizado para ajustar los parámetros libres de esta red neuronal apareció primero en un procedimiento de aprendizaje desarrollado por Rosenblatt para su modelo de cerebro perceptrón (1958,1962)<sup>3</sup>. El proceso de aprendizaje en el perceptrón es realizado por un número finito de iteraciones.

<sup>3</sup> La organización de la red de la versión original del perceptrón, planteado por Rosenblatt (1962) tiene tres tipos de unidades: unidades sensoriales, unidades de asociación y unidades de respuesta. Las conexiones de las unidades sensoriales a las de asociación tienen pesos fijos, mientras que las conexiones de las unidades de asociación a las de respuesta tienen pesos variables. Las unidades de asociación actúan como

La primer crítica al perceptrón de Rosenblatt fue presentada por Minsky y Selfridge (1961), quienes argumentaron que el perceptrón no generalizaba en la noción de paridad binaria. Posteriormente, en términos matemáticos Minsky y Papert (1969,1988) escribieron en su libro *Perceptrons* las limitaciones computacionales del perceptrón de Rosenblatt para hacer generalizaciones en los ejemplos aprendidos localmente. También argumentaron que esas limitaciones se extendían a las variantes del perceptrón, más específicamente, a las redes neuronales multicapa.

A mediados de los 80's los investigadores concluyeron que las conjeturas de Minsky y Papert no eran justificadas debido a que había varias formas avanzadas de redes neuronales computacionalmente más poderosas que el perceptrón de Rosenblatt, por ejemplo perceptrones multicapa entrenados con el algoritmo de retropropagación.

Las redes neuronales multicapa consisten de un conjunto de unidades sensoriales (nodos fuente) que constituyen la capa de entrada, una o más capas ocultas de nodos computados y una capa de salida de nodos computados. La señal de entrada es propagada a través de la red en dirección hacia delante, sobre la base de capa por capa. Estas redes neuronales son comúnmente llamadas Perceptrones Multicapa (MLPs), los cuales representan una generalización del perceptrón de una capa.

Los perceptrones multicapa han sido aplicados satisfactoriamente para resolver problemas diversos y difíciles entrenándolos de una manera supervisada con el más popular de los algoritmos conocido como algoritmo de retropropagación del error. Este algoritmo está basado en la regla de aprendizaje de corrección del error, visto como la generalización del algoritmo de filtro de mínimos cuadrados, least-mean-square (LMS). Aunque, no se ha dicho que el algoritmo de retropropagación genere una solución óptima para todos los problemas solucionables, se ha superado el pesimismo sobre las máquina multicapa de aprendizaje inferido por Minsky y Papert.

Básicamente, el aprendizaje de retropropagación del error consiste de dos fases a través de las diferentes capas de la red: un paso hacia delante y un paso hacia atrás. En el paso

---

preprocesadores diseñados para extraer un patrón del medio ambiente de entrada. En tanto que los pesos variables son afectados, la operación del perceptrón original de Rosenblatt es esencialmente el mismo para el caso de una simple unidad de respuesta, es decir, una neurona

hacia delante, un patrón de actividades o vector de entrada es aplicado a los nodos sensoriales de la red y su efecto se propaga a través de la red capa por capa. Finalmente, un conjunto de salidas es generado como la respuesta actual de la red. Durante el paso hacia delante los pesos sinápticos son calculados. En el paso hacia atrás, los pesos sinápticos son ajustados de acuerdo con una regla de corrección del error. Específicamente, la respuesta actual de la red es sustraída de la respuesta deseada para producir una señal de error. Esta señal de error es propagada hacia atrás a través de la red. Los pesos sinápticos son ajustados para generar la respuesta actual de la red más cercana a la respuesta deseada en un sentido estadístico.

El algoritmo de retropropagación del error es referido en la literatura como el algoritmo de retropropagación o simplemente back-propagation. En tanto, el proceso de aprendizaje que se realiza con el algoritmo es llamado aprendizaje de retropropagación.

Un perceptrón multicapa tiene tres características principales:

1. El modelo de cada neurona en la red incluye una función de activación no lineal. La función más utilizada es la sigmoide no lineal representada por la función logística:

$$y_j = \frac{1}{1 + \exp(-v_j)} \quad (\text{II.viii})$$

donde  $v_j$  es el campo local inducido de una neurona  $j$ , es decir, la suma de todas las entradas sinápticas incluyendo la bias, y  $y_j$  es la salida de la neurona.

2. La red contiene una o más capas de neuronas ocultas que no son parte de las entradas o salidas de la red. El objetivo de estas neuronas ocultas hace que la red aprenda tareas complejas extrayendo progresivamente las características principales de los patrones o vectores de entrada.
3. La red presenta un grado alto de conectividad determinado por las sinapsis de la red. Un cambio en la conectividad implica un cambio en las conexiones sinápticas o sus pesos.

En la Figura II.6 se muestra una porción de un perceptrón multicapa, el cual presenta dos tipos de señales:

1. Señales de función. Una señal de función es una señal de entrada (estímulo) que llega al final de la red, se propaga hacia delante (de neurona a neurona) a través

de la red y emerge a la salida de la red como una señal de salida. Además, se presupone realizar una función útil a la salida de la red y en cada neurona por la que pasa la señal de función, la señal es calculada como una función de las entradas y asocia los pesos a la neurona. La señal de función es también referida como una señal de entrada.

2. Señales de error. Una señal de error se origina a la salida de la neurona y es propagada hacia atrás (capa a capa) a través de la red. Se refiere a una señal de error debido a que su cálculo por cada neurona involucra una función dependiente del error en una u otra manera.

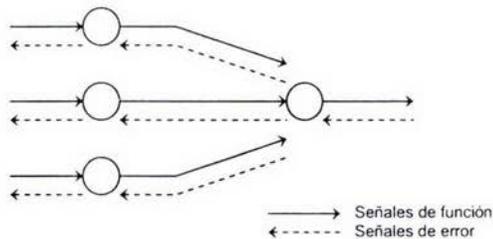


Figura II.6 Flujos en un perceptrón multicapa

### Algoritmo de retropropagación

#### Notación

- Los índices  $i$ ,  $j$  y  $k$  se refieren a neuronas diferentes, las señales se propagan a través de la red de izquierda a derecha, la neurona  $j$  se encuentra en la capa de la derecha de la neurona  $i$  y la neurona  $k$  en la capa a la derecha de la neurona  $j$  cuando la neurona  $j$  está en una unidad oculta.
- En la iteración (época)  $n$ , el  $n$ -ésimo patrón de entrenamiento (ejemplo) es presentado a la red.
- El símbolo  $\varepsilon(n)$  se refiere a la suma instantánea del cuadrado de los errores o energía del error en la iteración  $n$ . El promedio de  $\varepsilon(n)$  sobre todos los valores de  $n$ , es decir, todo el conjunto de entrenamiento, es denominado la energía del error promedio  $\varepsilon_{av}$ .

- El símbolo  $e_j(n)$  se refiere a la señal de error a la salida de la neurona  $j$  en la iteración  $n$ .
- El símbolo  $d_j(n)$  se refiere a la respuesta deseada para la neurona  $j$  y es utilizada para calcular  $e_j(n)$ .
- El símbolo  $y_j(n)$  se refiere a la señal de función a la salida de la neurona  $j$  en la iteración  $n$ .
- El símbolo  $w_{ij}(n)$  denota el peso sináptico que conecta la salida de la neurona  $i$  a la entrada de la neurona  $j$  en la iteración  $n$ . La corrección realizada a este peso en la iteración  $n$  es denotada por  $\Delta w_{ij}(n)$ .
- El campo local inducido, es decir, la suma de todos los pesos de las entradas incluyendo la bias, de una neurona  $j$  en la iteración  $n$  es denotada por  $v_j(n)$ ; éste representa la señal para la función de activación asociada con la neurona  $j$ .
- La función de activación que describe la relación funcional entrada-salida de no linealidad asociada con la neurona  $j$  es denotada por  $\phi_j(\cdot)$ .
- La bias aplicada a la neurona  $j$  es denotada por  $b_j$ ; su efecto es representado por la sinapsis de peso  $w_{j0} = b_j$  conectada a un entrada igual a +1.
- El  $i$ -ésimo elemento del vector de entrada (patrón) es denotado por  $x_i(n)$ .
- El  $k$ -ésimo del vector de salida (patrón) es denotado por  $o_k(n)$ .
- El parámetro de tasa de aprendizaje es denotado por  $\eta$ .
- El símbolo  $m_l$  denota el tamaño, es decir, el número de nodos, en la capa  $l$  del perceptrón multicapa:  $l = 0, 1, \dots, L$ , donde  $L$  es la profundidad de la red. Entonces,  $m_0$  denota el tamaño de la capa de entrada,  $m_1$  el tamaño de la primer capa oculta y  $m_L$  el tamaño de la capa de salida. También se utiliza  $m_L = M$ .

La señal de error de la salida de una neurona  $j$  en la iteración  $n$ , es decir, la presentación del  $n$ -ésimo entrenamiento, está definido por

$$e_j(n) = d_j(n) - y_j(n) \text{ donde la neurona } j \text{ está en un nodo salida} \quad (\text{II.ix})$$

Se define el valor instantáneo de la energía del error para la neurona  $j$  como  $\frac{1}{2}e_j^2(n)$ .

Correspondientemente, el valor instantáneo  $\varepsilon(n)$  de la energía del error total por la suma del valor instantáneo de todas las neuronas en la capa de salida, debido a que en esta capa se encuentran las neuronas que son calculadas directamente. Es decir,

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (\text{II.x})$$

donde  $C$  es el conjunto que incluye las neuronas de la capa de salida. Ahora, sea  $N$  el conjunto de todas muestras utilizadas en el entrenamiento. La energía del error cuadrado promedio se obtiene por sumar  $\varepsilon(n)$  sobre todas las  $n$  y normalizar con respecto al tamaño  $N$ , como se muestra a continuación

$$\varepsilon_{av} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad (\text{II.xi})$$

Para un conjunto de entrenamiento,  $\varepsilon_{av}$  representa la función de costo como una medida de rendimiento del aprendizaje. El objetivo del proceso de aprendizaje es ajustar los parámetros libres de la red para minimizar  $\varepsilon_{av}$ . El ajuste a los pesos son hechos de acuerdo a los errores respectivos calculados para cada patrón representado en la red.

El campo de acción inducido  $v_j(n)$  producido en la entrada de la función de activación asociado con la neurona  $j$  es

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n) \quad (\text{II.xii})$$

donde  $m$  es el número total de entradas (excepto las bias) aplicado a la neurona  $j$ . El algoritmo de retropropagación aplica una corrección  $\Delta w_{ji}(n)$  a los pesos sinápticos  $w_{ji}(n)$ , el cual es proporcional a la derivada parcial  $\partial \varepsilon(n) / \partial w_{ji}(n)$ <sup>4</sup>.

La corrección  $\Delta w_{ji}(n)$  aplicado a los pesos sinápticos que conectan la neurona  $i$  a la neurona  $j$  es definida por la regla delta:

$$\begin{pmatrix} \text{Corrección} \\ \text{del peso} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{Parámetro} \\ \text{tasa de aprendizaje} \\ \eta \end{pmatrix} \cdot \begin{pmatrix} \text{Gradiente} \\ \text{Local} \\ \delta_j(n) \end{pmatrix} \cdot \begin{pmatrix} \text{Señal de entrada} \\ \text{de la neurona } j \\ y_i(n) \end{pmatrix} \quad (\text{II.xiii})$$

Además, el gradiente local  $\delta_j(n)$  depende de si la neurona  $j$  es un nodo de salida o un nodo oculto:

1. Si la neurona  $j$  es un nodo de salida,  $\delta_j(n)$  es igual al producto de la derivada  $\phi_j'(v_j(n))$  y la señal de error  $e_j(n)$  asociados con la neurona  $j$ .

<sup>4</sup> De acuerdo a la regla de la cadena del cálculo, se expresa este gradiente como  $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$ . Para un detalle de la demostración, consultar Haykin (1999), pág 163.

2. Si la neurona  $j$  es un nodo oculto,  $\delta_j(n)$  es igual al producto de la derivada  $\phi'_j(v_j(n))$  y la suma de los pesos de los  $\delta$ s calculados para las neuronas en la siguiente capa oculta o la de salida que se conecta a la neurona  $j$ .

$$\delta_j(n) = \phi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (\text{II. xiv})$$

En resumen, el algoritmo de retropropagación<sup>5</sup> se cicla para cada entrenamiento de acuerdo a los siguientes cinco pasos:

- 1) Inicialización. Asumiendo que no se cuenta con la suficiente información, se estiman los pesos sinápticos y los umbrales desde una distribución uniforme con media cero y varianza elegida de tal manera que la desviación estándar induzca al campo local de las neuronas, teniendo así la transacción entre la función de activación lineal y las partes saturadas de la función sigmoideal.
- 2) Presentación de las muestras de entrenamiento. Presentar a la red con una época de muestras de entrenamiento. Para cada muestra del conjunto, ordenados de alguna manera, realizar la secuencia de cálculos hacia adelante y hacia atrás descrita en los pasos 3 y 4, respectivamente.
- 3) Cálculo hacia adelante. Sea una muestra en la época denotado como  $(x(n), d(n))$ , con el vector de entrada  $x(n)$  aplicado a la capa de entrada de los nodos sensoriales y el vector de salida deseado  $d(n)$  presentado a la capa de salida de los nodos calculados. Calcular el campo local inducido y las señales de función de la red hacia adelante, capa por capa. El campo local inducido  $v_j^{(l)}(n)$  para la neurona  $j$  en la capa  $l$  es calculado como

$$v_j^{(l)}(n) = \sum_{i=1}^{m_{l-1}} w_{ji}^{(l)}(n) y_i^{l-1}(n) \quad (\text{II. xv})$$

donde  $y_i^{l-1}(n)$  es la señal de salida (función) de la neurona  $i$  en la capa anterior  $l-1$  en la iteración  $n$  y  $w_{ji}^{(l)}(n)$  es el peso sináptico de la neurona  $j$  en la capa  $l$  a partir de la neurona  $i$  en la capa  $l-1$ . Para  $i=0$ , se tiene  $y_0^{l-1}(n) = +1$  y  $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$ , es decir, la bias aplicada a la neurona  $j$  en la capa  $l$ . Suponiendo que se utiliza la función sigmoideal, la señal de salida de la neurona  $j$  en la capa  $l$  es  $y_j^{(l)} = \phi(v_j^{(l)}(n))$ .

<sup>5</sup> Los fundamentos matemáticos de las redes neuronales y particularmente del algoritmo de retropropagación quedan fuera del alcance de esta investigación, para un mayor detalle consultar la bibliografía propuesta.

Si la neurona  $j$  está en la primer capa oculta, es decir,  $l=1$ , hacer  $y_j^{(l)} = x_j(n)$  donde  $x_j(n)$  es el  $j$ -ésimo elemento del vector de entrada  $x(n)$ . Si la neurona  $j$  está en la capa de salida, es decir  $l = L$  ( $L$  es llamado la profundidad de la red), hacer  $y_j^{(L)} = o_j(n)$ .

Calcular la señal de error como  $e_j(n) = d_j(n) - o_j(n)$ , donde  $d_j(n)$  es el  $j$ -ésimo elemento del vector de respuesta deseado  $d(n)$ .

- 4) Cálculo hacia atrás. Consiste en calcular los  $\delta$ s, es decir, los gradientes locales de la red, definido como

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\varphi_j'(v_j^{(L)}(n)) & \text{para la neurona } j \text{ en la capa de salida } L \\ \varphi_j'(v_j^{(l)}(n))\sum_k \delta_k^{(l-1)}(n)w_{kj}^{(l-1)}(n) & \text{para la neurona } j \text{ en la capa oculta } l \end{cases} \quad (\text{II.xvi})$$

donde el apóstrofe en  $\varphi_j'(\cdot)$  denota la diferenciación con respecto al argumento.

Se ajustan los pesos sinápticos de la red en la capa  $l$  de acuerdo a la regla delta generalizada:

$$w_{\mu j}^{(l)}(n+1) = w_{\mu j}^{(l)}(n) + \alpha [w_{\mu j}^{(l)}(n-1)] + \eta \delta_j^{(l)}(n) y_{\mu}^{(l-1)}(n) \quad (\text{II.xvii})$$

donde  $\eta$  es la tasa de aprendizaje y  $\alpha$  es la constante momentum.

- 5) Iteración. Iterar los cálculos hacia delante y hacia atrás de acuerdo a nuevas épocas de muestras de entrenamiento de la red hasta que se llegue a un criterio de paro, por ejemplo que el error promedio sea menor a 0.00001. Además, el orden de la presentación de las muestras del entrenamiento será aleatorio de época en época. En general, los parámetros tasa de aprendizaje y momentum serán ajustados (usualmente a la baja) a medida que incrementa el número de iteraciones de entrenamiento.

## II. 2 Algoritmos Genéticos

La metáfora biológica para los algoritmos genéticos se deriva en la evolución de las especies por la sobrevivencia descrita por Charles Darwin. En una población de animales o plantas, un nuevo individuo es generado por la cruce de la información genética de dos padres.

Existen una gran cantidad de problemas interesantes para los que no se han desarrollado algoritmos razonablemente rápidos. Muchos de estos problemas son problemas de optimización que son utilizados en aplicaciones diversas. Debido a la complejidad de la optimización, en general se utiliza un algoritmo que da una solución aproximada. Para este tipo de problemas también se utilizan algoritmos probabilísticos; sin embargo, aunque éstos algoritmos no garantizan el valor óptimo de las muestras seleccionadas aleatoriamente, la probabilidad del error disminuye tanto como se desee.

En general, una tarea abstracta es resolver un problema en el que se realiza una búsqueda a través de un espacio de soluciones potenciales. Una vez que se obtiene la mejor solución, se considera la tarea como un proceso de optimización. Para espacios de búsqueda pequeños, los métodos clásicos exhaustivos usualmente son suficientes. Sin embargo, para espacios grandes se emplean técnicas de inteligencia artificial especiales.

Los algoritmos genéticos son una de esas técnicas. Se trata de algoritmos estocásticos cuyos métodos de búsqueda modelan algunos fenómenos naturales. Los algoritmos genéticos han sido aplicados satisfactoriamente a problemas de optimización como trayectorias, calendarización, control adaptado, juegos, modelos cognitivos, problemas de transporte, problemas del agente viajero, problemas de control óptimo, optimización de consultas en base de datos, etc.

Durante la última década, la importancia de optimizar ha aumentado y algunos problemas combinatorios y de ingeniería solo son resueltos con las computadoras actuales a través de aproximaciones. El objetivo de los algoritmos genéticos son estos problemas complejos.

Los algoritmos genéticos pertenecen a la clase de algoritmos probabilísticos, sin embargo, se diferencian de los algoritmos aleatorios debido a que combinan las búsquedas dirigida y estocástica. Debido a esto, los algoritmos genéticos son mucho más robustos que los métodos de búsqueda dirigida debido a que mantienen una población de soluciones potenciales mientras que la mayoría de los otros métodos procesa un solo punto del espacio de búsqueda.

## II.2.1 Características

La información genética para la construcción de los individuos está almacenada en el ADN (Ácido Desoxirribonucleico). El genoma ADN humano consiste de 46 cromosomas, los cuales son cadenas de cuatro bases diferentes, abreviadas A, T, G y C. Un trio de bases es traducido en uno de 20 aminoácidos o en una señal de inicio o paro de la construcción de proteínas. En total, hay cerca de tres billones de nucleótidos. Éstos son estructurados en genes, los cuales llevan una o más piezas de información acerca de la construcción del individuo. Sin embargo, se estima que solo 3% de los genes contienen información significativa, la gran mayoría de los genes no son utilizados.

La información genética en sí, el genoma, es llamado el genotipo del individuo. El resultado, el individuo, es llamado fenotipo. El mismo fenotipo puede resultar de diferentes genotipos, por ejemplo, los gemelos.

Un algoritmo genético trata de simular el proceso de evolución natural. Su propósito es optimizar el conjunto de parámetros. En la idea original, propuesta por John Holland (1975), la información genética es codificada en una cadena de bits de longitud variable, llamada la cadena de parámetros o individuo. Un valor posible de un bit es llamado un alelo e inclusive se cuentan con diferentes técnicas de codificación.

Cada parámetro de la cadena representa una posible solución al problema de estudio. La calidad de la solución es almacenada en el valor de aptitud o *fitness*. El *fitness* es calculado a través de una función que representa los resultados de aplicar los operadores básicos de un algoritmo genético cruza, selección y muta.

Un algoritmo genético, como cualquier programa evolutivo, se utiliza para resolver un problema en particular y tiene los siguientes cinco componentes:

- Una representación genética para las soluciones potenciales al problema.
- Una forma de crear una población inicial de soluciones potenciales.
- Una función de evaluación que juega el papel del medio ambiente, la cual ordena las soluciones en términos de su aptitud (*fitness*).
- Operadores genéticos que modifican la composición de los hijos.

- Valores para varios parámetros que utiliza el algoritmo genético, por ejemplo, tamaño de la población, probabilidades para aplicar los operadores genéticos, etc.

Los algoritmos genéticos pertenecen a un término común, Programas Evolutivos (EP – Evolution Programs), para todos los sistemas basados en la evolución. La estructura de un programa evolutivo se muestra a continuación:

```

Procedimiento programa_evolutivo
Inicio
  t ← 0
  inicializar P(t)
  evaluar P(t)
  mientras (no se cumpla la condición de terminación) hacer
  inicio
    t ← t + 1
    seleccionar P(t) de P(t-1)
    modificar P(t)
    evaluar P(t)
  fin
fin

```

Figura II.7 Estructura de un programa evolutivo

El programa evolutivo es un algoritmo probabilístico el cual mantiene una población de individuos,  $P(t) = \{x_1^t, \dots, x_n^t\}$  para la iteración  $t$ . Cada individuo representa una solución potencial al problema de estudio y, en algunos programas evolutivos, es implementado como una estructura de datos  $S$  (posiblemente compleja). Cada solución  $x_i^t$  es evaluada para dar alguna medida de su aptitud. Posteriormente, una nueva población (iteración  $t+1$ ) está formada por seleccionar los mejores individuos (paso de selección). Algunos miembros de la nueva población sufren transformaciones (paso de modificación) al aplicar operadores genéticos para formar nuevas soluciones.

Existen transformaciones unitarias  $m_i$  (tipo mutación), la cual crea un nuevo individuo a través de aplicar un pequeño cambio en un individuo ( $m_i : S \rightarrow S'$ ), y transformaciones de orden superior  $c_i$  (tipo cruzamiento), la cual crea nuevos individuos por combinar partes de

varios (dos o más) individuos ( $c_i : S_1 \times \dots \times S_n \rightarrow S'$ ). Después de algún número de generaciones, el programa converge, esperando que el mejor individuo represente una solución cercana a la óptima (razonablemente).

### II.2.2 Operadores

Como se mencionó en la sección anterior, los operadores básicos de un algoritmo genético son selección, cruzamiento y mutación. En la Figura II.8 se muestra la estructura principal de un algoritmo genético. Se observa que se inicia con una generación aleatoria de un conjunto de individuos iniciales, llamados la población inicial.



Figura II.8 Estructura principal de un algoritmo genético

Los individuos son evaluados y clasificados (ranking) considerando que el número de individuos en cada población es constante, es decir, para cada nuevo individuo uno más tendrá que ser eliminado, en general será el que tenga el peor valor de fitness.

Se consideran dos operadores básicos para generar nuevos individuos: mutación y cruzamiento.

### Mutación

La mutación es la más sencilla de implementar. Durante la mutación, los bits de la cadena paramétrica son intercambiados aleatoriamente. La mutación también se aplica a los descendientes producidos por cruzamiento o como un operador independiente para generar el nuevo individuo al aplicarla a un individuo seleccionado aleatoriamente en la población.

### Cruzamiento

El cruzamiento simula la generación sexual de un hijo o descendiente de dos padres. Para realizar esto se toma parte de la cadena de bits de uno de los padres y la otra parte del otro padre, combinándose en el hijo. Existen tres tipos básicos para realizar el cruzamiento: en un punto, en dos puntos y uniforme.

En la crucea en un punto, ambos padres son cortados en el mismo punto, así el hijo es generado tomando una parte de cada padre, manteniendo el mismo lugar de los padres. Es importante notar, que es seleccionado al azar un punto de corte que es independiente del significado actual de los bits. Un parámetro de la cadena de bits puede ser codificado en más de un bit y su codificación cortada en dos partes durante el cruzamiento, esto resulta en un nuevo valor, completamente diferente del padre. Un ejemplo de cruzamiento en un punto es el siguiente:

Padre 1:	<u>001010011</u>	0101001010101110
Padre 2:	010101110	<u>1010101101110101</u>
Hijo:	001010011	1010101101110101

La crucea en dos puntos difiere de la versión anterior en que se realizan dos cortes aleatorios, en consecuencia, se obtienen tres partes que se unen en el mismo orden para generar el descendiente. Por ejemplo:

Padre 1:	<u>001010011</u>	01010010	<u>10101110</u>
Padre 2:	010101110	<u>10101011</u>	01110101
Hijo:	001010011	10101011	10101110

Estas dos operaciones de cruce son las originales. La cruce uniforme fue sugerida por Syswerda en 1989. Con esta opción, para cada bit, se decide aleatoriamente si es copiado por el padre uno o por el segundo.

Padre 1: 000000000000000000000000  
 Padre 2: 111111111111111111111111  
 Hijo: 1001001101110100100111001

Durante una generación se realiza un número fijo de cruzamientos y mutaciones.

**Selección**

La selección de individuos para cruce y muta está basada en los mejores individuos. En la ruleta clásica basada en ajustes (fitness), la oportunidad de que un individuo sea seleccionado está basada en su ajuste relativo en la población, como se muestra en el ejemplo de la Figura II.9. La selección por ajustes trabaja bien al inicio, pero a medida que se acerca al final, los valores del ajuste de los individuos cambian ligeramente. Whitley (1993) propuso una alternativa: la selección no se realiza en el ajuste relativo, sino en el ordenamiento del individuo en la población.

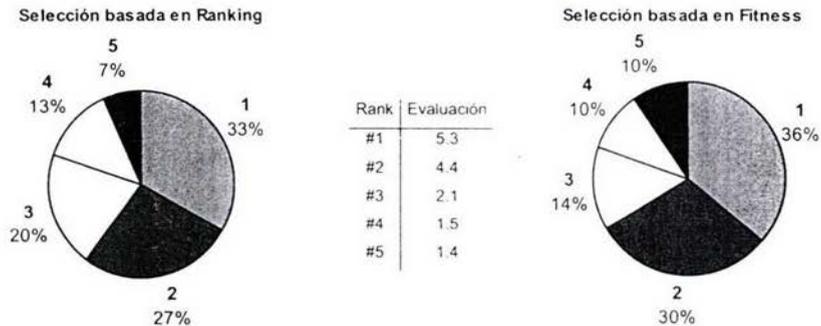


Figura II.9 Selección basada en Ranking vs. Basada en Fitness

Por ejemplo, en una población de 5 individuos, la oportunidad de que el primer individuo calificado sea seleccionado es 5 a 15, mientras que para el quinto individuo es 1 a 15,

esto es debido a que  $\sum_{n=1}^s n = 15$ . En general, ambos métodos funcionan y se elige la técnica para la selección de acuerdo a la experiencia o las características del problema.

### II.2.3 Algoritmo Elitista

Como explica Kuri y Galaviz en su libro Algoritmos Genéticos (2002), Golberg popularizó el Algoritmo Genético Simple (AGS) propuesto por Holland considerando que los códigos genéticos utilizados son binarios. Explicado a detalle, el proceso de un AGS es:

1. Decidir cómo codificar el dominio del problema.
2. Generar un conjunto aleatoria (población inicial) de N posibles soluciones codificadas al problema. Llamada población actual.
3. Calificar cada individuo de la población actual.
4. Seleccionar dos individuos de la población actual con una probabilidad proporcional a su calificación.
5. Lanzar una moneda al aire (con probabilidad  $p_c$  de caer cara).
6. Si cayó cara mezclar los códigos de los dos individuos seleccionados para formar dos híbridos, a éstos se les llaman nuevos individuos.
7. Si cayó cruz, los individuos seleccionados son los nuevos individuos.
8. Por cada bit de cada nuevo individuo lanzar otra moneda (con probabilidad  $p_m$  cae cara).
9. Si cae cara cambiar el bit en turno por su complemento.
10. Si cae cruz el bit permanece inalterado.
11. Incluir a los dos nuevos individuos en una nueva población.
12. Si la nueva población tiene ya N individuos, llamarla población actual y regresar al paso 3, a menos que se cumpla alguna condición de terminación. Esta condición es determinada de muchas maneras, por ejemplo, un número máximo de generaciones que se pretende ejecutar el algoritmo.
13. Si no, regresar al paso 4.

En general, se hace referencia a un AGS como aquel caracterizado por tener<sup>6</sup>:

<sup>6</sup> El teorema del esquema permite determinar la manera en que un esquema dado se propaga en una población, bajo los operadores de selección proporcional, cruzamiento en un punto y mutación uniforme. Los

1. Tamaño de población fijo en todas las generaciones.
2. Selección proporcional (de ruleta).
3. Cruzamiento de un punto. La probabilidad de cruza se mantiene fija para todas las generaciones y todas las parejas.
4. Mutación uniforme, todas las posiciones de las cadenas genéticas tienen la misma probabilidad de ser cambiadas. La probabilidad de mutación permanece fija para todas las generaciones y todas las posiciones de los individuos.
5. Selección no elitista, es decir, no se copian individuos de una generación a otras sin pasar por el proceso de selección aleatoria.

Se considera que la convergencia global no es una propiedad inherente de un AGS, es decir, el AGS original no es considerado un algoritmo de optimización porque es demostrable que no convergerá a ningún subconjunto de estados que contenga cuando menos una solución global, aún en un tiempo infinito.

Con el simple mecanismo de mantener el mejor individuo observado se garantiza la convergencia global, por esta razón los algoritmos genéticos son una alternativa robusta y general a los problemas de optimización. Existen modelos llamados no convencionales de los algoritmos genéticos que presentan una forma de hacer las técnicas más eficientes.

En un AGS no existe ningún medio para asegurar que una buena aproximación que ya se encontró se preserve o que sea seleccionada para dejar descendientes en la siguiente generación. Es cierto que el mejor individuo de la población tiene la más alta probabilidad de ser seleccionado por la ruleta, pero hay una probabilidad distinta de cero de que esto no ocurra. También puede ser que sea elegido y se pierda al cruzarse con algún otro seleccionado distinto de él mismo, o bien que logre sobrevivir a la cruza pero la mutación lo altere. No se provee de un mecanismo que asegure que el mejor individuo de la generación  $i$  pase intacto a la generación  $i+1$ .

Ésto trae como consecuencia que la mejor calificación de cada generación del AGS fluctúe. Sin embargo, sería deseable que el comportamiento de la calificación máxima de la población fuera monótono, es decir, no decreciente al asegurar que el mejor individuo

---

fundamentos matemáticos de los algoritmos genéticos queda fuera del alcance de esta investigación, para un mayor detalle consultar la bibliografía anexa.

de la generación  $i+1$  tendrá una calificación al menos tan alta como el mejor de la generación  $i$ .

Para lograr lo anterior, se propone el elitismo. El elitismo es un mecanismo que consiste en conservar, en la generación  $i$  de un algoritmo genético, a los  $k$  mejores individuos de las  $r$  mejores generaciones. La manera más sencilla del elitismo consiste de conservar, copiar sin alteraciones, al mejor individuo de la generación inmediata anterior.

El AGS en combinación con elitismo es una variante sencilla del AGS. Sin embargo, existen variantes como elitismo parcial y elitismo total. Por elitismo parcial se entiende que en una población de tamaño  $n$  se mantiene una copia de los mejores  $\tau < n$  individuos hasta la generación  $k$ . Por elitismo total se quiere decir que se mantiene una copia de los mejores  $n$  individuos hasta la generación  $k$ . Es decir, debido a que se han probado  $nk$  individuos hasta la generación  $k$ , la población consistirá de los mejores  $n$  hasta ese punto.

En la Figura II.10 se muestra el caso de elitismo total. En la generación  $k$  se han elegido los mejores  $n$  individuos posibles del total de los  $nk$  individuos considerados hasta ese punto. Al hacer esto, artificialmente se fuerza a la población a orientarse hacia un conjunto focalizado de hiperplanos. En este contexto, un hiperplano es el espacio de búsqueda de las posibles soluciones al problema. El riesgo obvio es que se restrinja el espacio de búsqueda de manera que sesgue el algoritmo excesivamente.

Generación

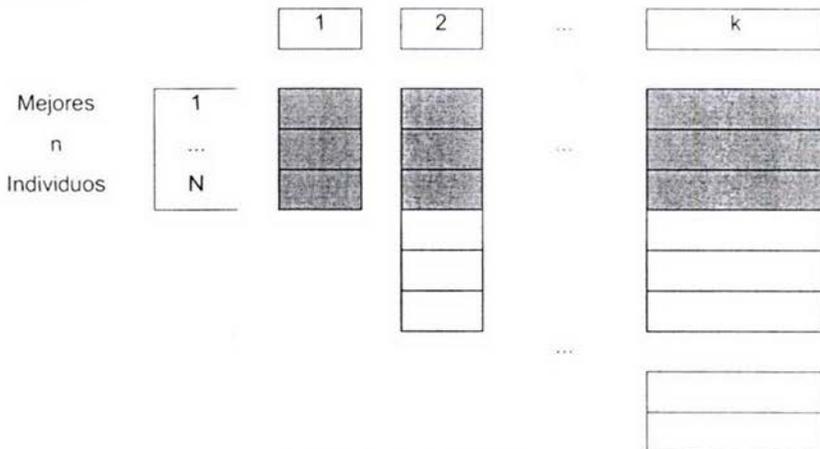


Figura II.10 Elitismo total

Para evitar el riesgo de sesgo excesivo, se altera la estrategia de selección proporcional como se describe. En el AGS se considera la selección proporcional, sin embargo, en el algoritmo elitista se hace la selección determinística. En este tipo de selección no se considera el ajuste del individuo para determinar los descendientes más deseables. En su lugar, se propone enfatizar la variedad genética imponiendo una estrategia que refuerce el cruzamiento de individuos predefinidos. En este sentido, existen dos puntos de vista contrastantes, en uno se favorece que los genes del mejor individuo se crucen entre ellos mismos; en el otro se favorece que los mejores individuos se crucen con los peores.

La primera estrategia se denomina el modelo de Nietzsche (MN), en donde los mejores elementos de la población se entremezclan en un esfuerzo por preservar los mejores genes. La otra estrategia se denomina el modelo de Vasconcelos (MV), en donde los mejores individuos se mezclan con los peores en un esfuerzo por explorar el más amplio espectro del espacio de soluciones.

El modelo de Nietzsche adopta la estrategia de seleccionar determinísticamente el individuo  $i$  para cruzarlo con el individuo  $i+1$  para  $i = 1, 3, 5, \dots$ . Se supone que se ha aplicado elitismo total; por ello, se fuerza que los genes de los individuos 1 y 2 se crucen; que los genes de los individuos 3 y 4 se crucen, es decir, que los genes de los individuos  $n-1$  y  $n$  se crucen.

De acuerdo a Kuri y Galaviz, un análisis superficial sugiere que un mecanismo en el cual se refuerzan las características benéficas de los individuos llevará a un mejor algoritmo. Sin embargo, la simulación no sustenta esta visión debido a que los posibles esquemas de esta forma conducen a lo que se denomina convergencia prematura, es decir, la selección determinística de los mejores individuos de esta forma no permite que haya una exploración lo suficientemente vasta del espacio de soluciones. En cierto sentido, se restringe la fase de exploración en tanto se maximizan las características positivas de los individuos.

En el modelo de Vasconcelos, se adopta la estrategia de cruzar al individuo  $i$  con el individuo  $n-i+1$  determinísticamente. Como en MN se supone elitismo total, así se adopta una estrategia que, superficialmente, destruye las características deseables de los

mejores individuos cruzándolos deliberadamente con los peores. Sin embargo, cuando esto se hace en conjunción con elitismo total, esta estrategia conduce al análisis implícito de una más amplia variedad de esquemas, es decir, maximiza la exploración del espacio de soluciones.

El MV permite al algoritmo atrapar los mejores esquemas sin restringir el espacio de búsqueda en una forma sensible. Además, no se fija un valor en un número grande de las cadenas de la población debido a que determinísticamente se afecta las posibles fijaciones problemáticas con una cruce mejor-peor. Debido a que se trabaja con elitismo, se preservan los esquemas deseables y se evita la correlación espuria al cruzar individuos disímiles.

Finalmente, el elitismo total garantiza que, aún en el caso del MV, el peor individuo de la población  $k$  está en el estrato que corresponde al  $1/k$  porcentual. Por ejemplo, si  $n = 50$  y se observa la población  $k=20$ , los individuos en ésta están dentro del mejor  $1/k=5\%$  del total de los individuos analizados. Esta característica se incrementa a medida que el algoritmo genético avanza.

# CAPÍTULO III. EQUIVALENCIA ENTRE MÉTODOS DE REDES NEURONALES Y ALGORITMOS GENÉTICOS

*Enseñar no es una función vital, porque no tienen el fin en sí misma,  
la función vital es aprender  
Aristóteles*

Los fundamentos para la hipótesis de la presente investigación se describirán en este capítulo. La inquietud para este análisis surge de dos puntos principales:

1. La importancia que en la actualidad han tenido los Algoritmos Genéticos y las Redes Neuronales como parte de nuevas técnicas de Inteligencia Artificial aplicados a problemas prácticos con muy buenos resultados.
2. La comparación de ambas técnicas de Inteligencia Artificial para resolver el mismo problema con enfoques distintos, es decir, el análisis de equivalencia entre ambas técnicas analizado desde puntos de vista diferentes.

Cuando se habla de Inteligencia Artificial existen diversas técnicas que son aplicadas a problemas específicos y en algunos casos se aplican los modelos para un cierto tipo de problemas. Sin embargo, a medida que se realizan investigaciones más detalladas se crean nuevas técnicas o se amplía el campo de acción de una técnica para problemas más difíciles o complejos.

Como se describió en el Capítulo I, en análisis de inversiones y particularmente para la estimación de pronósticos es común utilizar técnicas como Redes Neuronales. Sin embargo, no ha sucedido lo mismo con la aplicación de los Algoritmos Genéticos para este tipo de problemas. En general, la mayoría de las investigaciones se han enfocado a integrar ambas técnicas para cubrir el mismo objetivo, es decir, utilizar los Algoritmos

Genéticos como una técnica para optimización de las Redes Neuronales y así obtener mejores resultados.

Se han realizado investigaciones en Bäck (1992), Billings (1999) y De Jong (1995) para optimizar los parámetros de una Red Neuronal, como la tasa de aprendizaje, las capas ocultas, número de neuronas ocultas, etc. a través de un Algoritmo Genético. Por otro lado, poco se ha encontrado acerca del análisis que muestren una equivalencia entre ambas técnicas para resolver el mismo problema, particularmente, problemas de aproximación a pronósticos en inversiones.

Como se verá a lo largo de este capítulo y del siguiente, se presenta la investigación del análisis fundamental de equivalencia entre ambas técnicas para problemas de aproximación, sin perder ninguna de ellas su esencia en la forma en que resuelven los problemas. Las interrogantes principales para este estudio son ¿es posible resolver un mismo problema utilizando AG o RN, indistintamente? ¿cuáles son los resultados que se obtienen al aplicar ambas técnicas al mismo problema? ¿cuál es el rendimiento esperado de utilizar una u otra técnica? ¿qué ventajas / desventajas se tienen al utilizar una técnica en comparación con la otra? ¿es posible aplicar ambas técnicas para pronósticos de manera independiente? Básicamente, éstos cuestionamientos serán desarrollados a través del capítulo y del siguiente.

El análisis paramétrico en Redes Neuronales está basado en series de tiempo en el sentido de la función de señal que se utiliza, generalmente la función sigmoideal. Las Redes Neuronales están caracterizadas porque ayudan a resolver un problema de manera eficiente. Sin embargo, no es sencillo comprender cómo funciona o por qué es que funciona esta técnica. En general, se dice que se resuelve el problema pero no se explica a detalle cómo se obtiene la respuesta.

El análisis multivariado en Algoritmos Genéticos está caracterizado en la optimización de una serie de tiempo para realizar el pronóstico. Los Algoritmos Genéticos ayudan a resolver el problema apoyándose en otras técnicas, es decir, como complemento a otras técnicas, por ejemplo, Redes Neuronales o Estadísticas. Con los AG, como método de optimización, es posible obtener el detalle de cada operación realizada para obtener el resultado deseado, es decir, esta técnica explica cómo se resuelve el problema.

En este sentido, se pretende explicar cómo es que ambas técnicas son aplicadas a problemas de pronóstico en análisis de inversiones para la toma de decisiones. Las interrogantes planteadas en esta sección abren la pauta para analizar nuevas técnicas para el análisis de inversiones y formas más eficientes para calcular pronósticos, sin descartar las técnicas comúnmente utilizadas como Series de Tiempo.

### III.1 Análisis paramétrico de Redes Neuronales

Las Redes Neuronales son una técnica ampliamente utilizada para el análisis de predicciones, sin embargo, no permiten un análisis detallado de los resultados obtenidos por lo que es común que investigadores complementen el análisis con Algoritmos Genéticos para optimización de parámetros y la estructura de la red.

El diseño de redes neuronales para predicción de series de tiempo consiste de tres pasos básicos: Identificación, Estimación y Evaluación.

- a) Identificación. El primer paso consiste en elegir la arquitectura de la red neuronal artificial para modelar la serie de tiempo. Entre los parámetros principales de la arquitectura de la red que se necesitan determinar son: la topología de la red, la longitud de la ventana de tiempo, el número de capas ocultas<sup>1</sup>, el número de nodos ocultos, los parámetros de aprendizaje y las funciones de activación.

Para cada problema, los valores de cada uno de estos parámetros son determinados por diferentes estrategias. Para definir la arquitectura completa, se integran diferentes estrategias, cada una de las cuales define un parámetro. El problema es que algunas veces, las estrategias seleccionadas no consideran la interdependencia entre los parámetros y no necesariamente encuentran un buen punto en el espacio completo de los parámetros.

Recientemente, se han utilizado los Algoritmos Genéticos para determinar la arquitectura de las redes neuronales en lugar de estimar los parámetros por

---

<sup>1</sup> En la práctica, solo se utiliza una capa oculta debido a que la mayoría de los problemas se resuelven con este tipo de arquitectura.

separado. Los algoritmos genéticos son capaces de definir, al mismo tiempo, un gran número de parámetros realizando la optimización global en el espacio de búsqueda de los parámetros.

- b) **Estimación.** Este paso consiste de la aplicación de un algoritmo de aprendizaje para estimar los pesos de la red; en la práctica el algoritmo más común utilizado es el de retropropatación (Backpropagation Algorithm), descrito en el capítulo 2. Este algoritmo está basado en la técnica del gradiente descendiente. Existen otros algoritmos basados en la segunda derivadas que son en muchas situaciones más rápidos que el Backpropagation y son recomendados para aplicaciones de gran escala.
- c) **Evaluación.** El paso de evaluación consiste del análisis de los errores de predicción. Un modelo es generalmente evaluado por la suma o el promedio de los errores cuadráticos generados por el modelo, los cuales serán lo mas pequeños posibles. Además, los errores generados tendrán un comportamiento aleatorio a través del tiempo, de lo contrario, será necesario hacer ajustes al modelo.

El enfoque de la presente investigación no se dirige a la aplicación de los algoritmos genéticos para ajustar los parámetros de una red neuronal y así modelar el problema de series de tiempo integrando las dos técnicas. Por el contrario, el propósito es equiparar ambas técnicas, con el mismo nivel de formalidad, para resolver el problema de pronóstico con cualquiera de ambas técnicas.

Los parámetros básicos que componen la red neuronal son:

- a) **La estructura de la neurona.** Las neuronas o unidades de procesamiento son elementos simples que se componen de cuatro componentes. Una función de conexión, una función de entrada, una función de activación o transferencia y una función de salida.
- b) **La topología de la red.** Una vez seleccionadas las neuronas que serán distribuidas en las capas, la red será fuertemente conectada de tal manera que otro parámetro básico son el número de capas y las neuronas que se encuentran en cada nodo. Respecto al número de capas, por lo menos serán dos (la capa de entrada y la de salida) y el número de capas ocultas (que solo procesan resultados).

- c) La regla de aprendizaje o de adaptación. Una vez seleccionada la red neuronal, la red tendrá la tarea de aprender a través de una fase de entrenamiento, es decir, se tiene una regla de aprendizaje o algoritmos relacionados a interpretar el conocimiento implícito en cada neurona conectada a la red de acuerdo a los pesos específicos, que generalmente son iniciados con valores aleatorios y que son adaptados de acuerdo a un límite de error. Los procesos de aprendizaje, generalmente, son divididos en tres: supervisado, no supervisado (auto-organizados) y estrategias estocásticas.

Como se describió en el capítulo II, sección II.1.3, con el algoritmo de retropropagación de Rosenblat, el algoritmo no explica cómo se obtienen los resultados debido a que la aplicación de dicho algoritmo deriva básicamente en la asignación de valores para cada uno de los parámetros que componen la arquitectura de la red neuronal.

De acuerdo al algoritmo básico de retropropagación, existen otros parámetros adicionales que permiten una mejor aproximación del algoritmo, los principales son:

- **Número de capas.** El tamaño de la red depende del número de capas ocultas y el número de neuronas con que cuente cada una. Una red de tres capas es recomendable para resolver problemas cuando la función consiste de un número finito de puntos, cuando la función es continua en un conjunto compacto (conjunto con límite claros) y, en algunos casos para funciones distribuidas normalmente. Para todas las demás funciones con una red de cuatro capas es suficiente. En la mayoría de las aplicaciones prácticas una red de tres capas es suficiente.
- **Tasa de aprendizaje.** Un parámetro importante al aplicar el algoritmo de retropropagación es seleccionar la tasa de aprendizaje. Esta tasa depende de la naturaleza del problema y de la arquitectura o diseño de la red. Una tasa de aprendizaje alta implica que la red aprenda rápido y que se hagan mayores ajustes a los pesos para una convergencia más rápida. Sin embargo, cuando se acerca a la solución, continuar aplicando grandes ajustes a los pesos ya no es una ventaja, en este caso una tasa de aprendizaje pequeña sería más deseable. Un método para resolver este problema es no utilizar una tasa constante y a medida que disminuye el error, se reduce también la tasa de aprendizaje. Para asegurar la

convergencia en el proceso de aprendizaje, se recomienda que la tasa siga la condición  $0 < \text{Tasa de aprendizaje} < 2/\text{No. de neuronas en la capa precedente}$ . Cuando se utilizan redes para funciones de aproximación, se recomienda una tasa de aprendizaje entre 0.01 y 0.0001 para las neuronas lineales (salida) y entre 0.05 y 0.001 para las neuronas ocultas (no lineales).

- **Tamaño de la red.** Considerando que una red mayor aprende mejor y que una red pequeña produce mejores generalizaciones, el tamaño de la red es seleccionado heurísticamente. Así, el número de neuronas disponibles se identifica con el número de pesos sinápticos y se relaciona directamente con el número de muestras para el entrenamiento. Como no existe una regla universal, se recomienda que el número de ejemplos se encuentre entre dos y cuatro veces el número de pesos en la red neuronal. Otra alternativa consiste en adaptar los métodos durante el proceso de aprendizaje utilizando dos estrategias básicas: i) incrementando el número de neuronas en la capa oculta, y ii) reduciendo el número de conexiones entre las neuronas, suponiendo que se inicia con una red grande (aplicar "podado").
  
- **Estrategia de aprendizaje y orden de presentación.** Se refiere a la estrategia de aprendizaje y al orden de presentación de los ejemplos de entrenamiento. Otro factor que afecta la velocidad de la convergencia, se refiere al momento en que se ajustan los pesos que básicamente hay dos posibilidades: la adaptación después de un paso simple (estrategia delta) y la adaptación después de una época (retropropagación acumulada o delta acumulada). La segunda estrategia no es recomendada para épocas grandes debido al gran número de cálculos que son requeridos.
  
- **Podado (Pruning).** En el caso de perceptrones multicapa, una interrogante básica es ¿cuántas neuronas deberían utilizarse en las capas ocultas?. Si se seleccionan pocas neuronas existe el riesgo de que el problema no se resuelva adecuadamente. Por otro lado, muchas neuronas tienen la desventaja de entorpecer la generalización de la red neuronal. Los procedimientos de truncado están basados en iniciar con una red grande que mantiene todas las conexiones entre las capas individuales, es decir, fuertemente conectada, y las conexiones

menos importantes o significativas son eliminadas al quitar su peso sináptico correspondiente. Los pesos menos importantes son aquellos que aportan la menor cantidad a la suma total de los pesos de las neuronas, es decir, aquellos que tienen pesos cercanos a cero por lo que se establece un umbral para el pruning.

- **Neurona de umbral (bias).** Una neurona natural es excitada si la suma de los valores de entrada excede un umbral. En los diseños más recientes de redes se tiene una capa de red interna que consiste de una neurona cuya entrada es uno. Esta neurona de umbral es conectada a todas las neuronas de la red y actúa como un umbral ajustable para la excitación de las neuronas.
- **Momentum.** El objetivo del proceso de aprendizaje es encontrar el error mínimo global, sin embargo, un factor importante para ello es la tasa de aprendizaje que a medida que es más pequeña implica cálculos más extensos originando que se pueda llegar a un mínimo local. Para evitar llegar a un mínimo local se introduce el concepto de momentum, al modificar la ecuación II.xvi para adaptar los pesos se introduce  $\alpha$  como el momentum que modifica la derivada parcial correspondiente (paso 4 del Algoritmo de Retropropagación, pág. 33, Cap. II). Dependiendo de la estrategia de aprendizaje se recomienda el rango del momentum entre 0.7 y 0.9 para la estrategia delta y  $< 0.3$  para la estrategia delta acumulada.
- **Disminución de peso (Weight-Decay).** Al trabajar con pesos pequeños en redes neuronales se tiene la cualidad de generalizar mejor, por lo que para utilizar pesos pequeños se aplica una descomposición de los mismos. Para llevar a cabo esto se aplica una modificación a la función de costo, sumando los valores de los pesos a los costos provocando que sean reducidos por el gradiente descendiente, es decir, no solo se minimiza el error sino que también los valores de los pesos. Aplicar un factor de descomposición en los pesos conlleva a dos efectos importantes: prevenir pesos largos que originen un aprendizaje no favorable y la facilidad para la poda (pruning), cuando los pesos se acercan a cero tienen una menor relevancia. La tasa de descomposición es útil con la estrategia delta.
- **Inicialización de pesos.** La inicialización de los valores de los pesos que conectan las neuronas tienen una influencia sustancial en la convergencia del

proceso de convergencia. En general, un rango inicial adecuado está entre 0.1 y 0.9, siempre que se utilice la función de transferencia sigmoideal.

- **Procedimiento de entrenamiento.** Un entrenamiento satisfactorio de una red neuronal se obtiene cuando la red es capaz de reproducir correctamente los datos conocidos con un error mínimo así como clasificar datos desconocidos, es decir, datos que no se utilizaron en el proceso de entrenamiento. En general, se espera que el error de prueba sea ligeramente mayor que el error de entrenamiento. De lo contrario implica que la red aprendió los detalles de cada ejemplo de aprendizaje y no de las relaciones existentes. Un sobreentrenamiento implica que el error de prueba empieza a incrementar por lo que se deberá detener el entrenamiento en el error mínimo por lo que se recomienda guardar la red después de cada fase de prueba.
  
- **Pre-procesamiento de datos.** Debido a que las redes neuronales se aplican a diversas aplicaciones con resultados satisfactorios y que solo se requiere un conocimiento preliminar para entrenar una red neuronal, los datos presentados a la red no tienen un procesamiento previo. Sin embargo, preparar los datos es un factor decisivo en la velocidad de aprendizaje de la red y la calidad de la aproximación obtenida. En este sentido, las siguientes interrogantes surgen antes de iniciar el proceso: ¿se cuenta con los datos suficientes y la información que representan es correcta? ¿los datos cubren el rango de variables? ¿están cubiertos los casos de los límites con los datos disponibles? ¿los datos contienen información relevante? ¿existen transformaciones o combinaciones de variables que describen el problema más efectivamente que las variables individuales? Una vez analizadas las preguntas anteriores, se recomienda normalizar los datos. Para ello existen diversos métodos, uno de los más comunes implica generar un subconjunto de los datos transformados al rango deseado por un factor de escala. Un error frecuente se refiere a normalizar los datos de entrenamiento, prueba y carga por separado de acuerdo a sus respectivos máximos y mínimos; sin embargo, el mismo proceso de normalización debería ser aplicado a todas las fases con la finalidad de asegurar que todos los datos se encuentran en el mismo rango. El método más comúnmente utilizado para normalizar los datos implica

mapear los datos linealmente sobre un rango específico, donde cada valor de la variable  $x$  se transforma como sigue:

$$x' = \frac{x'_{\max} - x'_{\min}}{x_{\max} - x_{\min}} \cdot (x - x_{\min}) + x'_{\min} = \frac{x'_{\max} - x'_{\min}}{x_{\max} - x_{\min}} \cdot x - x'_{\min} \cdot \frac{x'_{\max} - x'_{\min}}{x_{\max} - x_{\min}} + x'_{\min} = \text{Factor} \cdot x - \text{Offset}$$

donde  $x_{\max}$  y  $x_{\min}$  son los valores mínimos y máximos esperados de la variable concerniente y  $x'_{\max}$  y  $x'_{\min}$  especifica los valores de los rangos para la variable transformada. Un rango de aproximadamente 0.1 y 0.9 es apropiado para la transformación de una variable sobre el rango de la función de transferencia sigmoideal.

En la práctica, existe software para redes neuronales que manejan valores predeterminados para los parámetros requeridos e inclusive hay aplicaciones que no consideran todos los parámetros. A medida que se tiene un dominio más profundo de los parámetros y se da una mejor interpretación, se observan mejores resultados al aproximar la red a los resultados deseados.

### III.2 Análisis multivariado en Algoritmos Genéticos

El Algoritmo Genético, como se describió en el capítulo anterior (Sección II.2), es un método de optimización el cual a través de iterar un procedimiento evalúa una función que se aproxima a la solución del problema.

La aplicación de algoritmos genéticos como técnicas de aprendizaje hacen más complicada la solución y corresponden a lo que en Inteligencia Artificial está clasificado como aprendizaje no supervisado. En este caso, el algoritmo genético se aplica a la configuración de las soluciones en vez de a las mismas soluciones, denominándolos en este sentido, algoritmos genéticos de orden<sup>2</sup>.

Existen casos prácticos en los que se desea encontrar una expresión matemática que sintetice las relaciones entre un conjunto de variables. Debido a la facilidad que representa la manipulación numérica, los datos son más fáciles de analizar aunque

<sup>2</sup> Kuri, A. y Galaviz, J. Algoritmos Genéticos, IPN, UNAM, CFE, pp. 135

también presenta algunas limitantes al encontrar las soluciones utilizando estas técnicas. Por ejemplo, que los datos tienen que estar en una forma predefinida, se tiene un número limitado de variables para procesar y datos y principalmente que no siempre se puede garantizar una estabilidad numérica.

En general, cuando se plantea un problema de estas características se tiene que se encuentran sobredeterminados, es decir, que existen más datos que parámetros. En consecuencia, se observa que los procesos de aproximación arrojan ajustes parciales por lo que es necesario medir la curva que representa la relación entre las variables independientes y la variable dependiente. Los algoritmos genéticos de orden libre no solo evolucionan la solución del problema, sino que también evolucionan la forma de la solución. En términos generales, el método consiste de:

- a) Se supone que los datos referentes al sistema de estudio se encuentran en forma de tabla. En ésta existe una columna por cada uno de los parámetros,  $p$ , que caracterizan al sistema.
- b) Se desea expresar una de las variables en función de las otras que corresponden al número de parámetros,  $p$ .
- c) Se elige, a priori, la forma del aproximante. La forma primordial es polinomial

$$\text{completa del tipo } f(v_1, \dots, v_p) = \sum_{i_1=0}^{g_1} \dots \sum_{i_p=0}^{g_p} C_{i_1 \dots i_p} v_1^{i_1} \dots v_p^{i_p} \quad (\text{III.i})$$

en donde  $p$  es el número de parámetros,  $g_i$  es el grado máximo que puede tomar la  $i$ -ésima variable independiente.

- d) Se escoge, a priori, una métrica que permita definir la distancia entre el aproximante y los valores de la tabla especificada en a).
- e) En este caso, la norma recomendada es la norma  $L_\infty$  o minimax, es decir, se desean encontrar los coeficientes  $C_{0 \dots 0}, C_{0 \dots 1}, \dots, C_{g_1 \dots g_p}$  tales que minimice

$$\varepsilon_i = (f_{1 \dots p}) - \left( \sum_{i_1=0}^{g_1} \dots \sum_{i_p=0}^{g_p} C_{i_1 \dots i_p} v_1^{i_1} \dots v_p^{i_p} \right)_i \text{ con } i = 1, \dots, m \quad (\text{III.ii})$$

Para la solución al conjunto  $M$ , se elige arbitrariamente  $m$  vectores de la tabla, en donde

$$m = (g_1 + 1) \times (g_2 + 1) \times \dots \times (g_p + 1) + 1 = 1 + \prod_{i=1}^p (g_i + 1) \quad (\text{III.iii})$$

Considerando en ( III.ii )  $(\sum C\vec{v})_i + \varepsilon_i = f_i$  ( III.iv )

Donde  $f = f_{1 \dots p}$ ,  $C = C_{i1 \dots ip}$ ,  $\vec{v} = v_1^{g_1} \dots v_p^{g_p}$  y

$$(\sum C\vec{v}) = (\sum_{i_1=0}^{g_1} \dots \sum_{i_p=0}^{g_p} C_{i_1 \dots i_p} v_1^{i_1} \dots v_p^{i_p})$$
 ( III.v )

Lo que se desea es minimizar  $\varepsilon_\theta$  considerando que

$$\varepsilon_\theta = \max\{|\varepsilon_1|, \dots, |\varepsilon_m|\} \text{ y } \varepsilon_i = \eta_i \varepsilon_\theta \text{ con } \eta_i \leq 1$$
 ( III.vi )

sustituyendo en ( III.iv ) se tiene  $(\sum C\vec{v})_i + \eta_i \varepsilon_\theta = f_i$  ( III.viii )

En términos matriciales, el sistema ( III.vii ) se expresa como

$$\begin{bmatrix} (v_1^0 \dots v_p^0)_1 & \dots & (v_1^{g_1} \dots v_p^{g_p})_1 & \eta_1 \\ \dots & \dots & \dots & \dots \\ (v_1^0 \dots v_p^0)_m & \dots & (v_1^{g_1} \dots v_p^{g_p})_m & \eta_m \end{bmatrix} \begin{bmatrix} C_{0 \dots 0} \\ \dots \\ C_{g_1 \dots g_p} \\ \varepsilon_\theta \end{bmatrix} = \begin{bmatrix} f_1 \\ \dots \\ f_m \end{bmatrix}$$
 ( III.viii )

escribiendo ( III.viii ) como  $A \bullet C = f$

Aplicando la regla de Cramer<sup>3</sup> ( III.viii ), se escribe la matriz A como

$$A = \begin{bmatrix} (v_1^0 \dots v_p^0)_1 & \dots & (v_1^{g_1} \dots v_p^{g_p})_1 & \sigma_1 \\ \dots & \dots & \dots & \dots \\ (v_1^0 \dots v_p^0)_m & \dots & (v_1^{g_1} \dots v_p^{g_p})_m & \sigma_m \end{bmatrix}$$
 ( III.ix )

y los coeficientes resultantes de  $C = f A^{-1}$  son los coeficientes minimax que se buscan para el conjunto de  $m$  vectores. Donde  $\varepsilon_\theta$  se minimiza si

$$\varepsilon_\theta = \frac{\Delta_i}{\sigma_1 \Delta_1 + \dots + \sigma_m \Delta_m}$$
 ( III.x )

es decir, el ajuste de minimax para el conjunto implica que las  $\varepsilon_i$ 's tienen el mismo valor absoluto y que los signos de tales  $\varepsilon_i$ 's son todos iguales (o todos diferentes) a los signos de las  $\Delta_i$ 's.

Lo que se busca es el conjunto de coeficientes C que minimicen  $\varepsilon_\theta$  para los  $n$  elementos del conjunto original. En la explicación anterior, se supone que los  $m$  vectores elegidos son arbitrarios. Para encontrar la  $\varepsilon_\theta$  global, es decir, aquella que resuelva el ajuste de

<sup>3</sup> Para el detalle de la aplicación de la regla de Cramer, revisar en Algoritmos Genéticos, Kuri y Galaviz, pp. 167-170

minimax para los  $n$  vectores originales, Kuri y Galaviz proponen el algoritmo iterativo de ascenso como sigue:

1. Hacer  $i \leftarrow 1$
2. Elegir un conjunto de vectores arbitrario  $M_i$  de tamaño  $m$ . Este conjunto es llamado el conjunto interno, a los vectores o incluidos en este conjunto, se agrupan en otro conjunto  $E_i$ , llamado conjunto externo y que es de tamaño  $n - m$ .
3. Encontrar la matriz  $A_i$  de acuerdo con ( III.ix ).
4. Encontrar los coeficientes minimax  $C_i$  y el máximo error de ajuste  $[\varepsilon_{\theta}]_i$  para el conjunto interno.
5. Calcular el máximo error de ajuste  $[\varepsilon_{\phi}]_i$  para el conjunto externo, en donde

$$[\varepsilon_{\phi}]_i = \max \left\{ (f_{i \dots \rho})_j - \left( \sum_{l_1=0}^{g_{l_1}} \dots \sum_{l_r=0}^{g_{l_r}} C_{l_1, l_2, \dots, l_r} v_1^{l_1} \dots v_r^{l_r} \right)_j \right\} \text{ con } \vec{v}_j \in E_i \quad (\text{ III.xi })$$

6. Si  $[\varepsilon_{\theta}]_i \geq [\varepsilon_{\phi}]_i$ , fin del algoritmo.
7. Si  $[\varepsilon_{\theta}]_i < [\varepsilon_{\phi}]_i$ , intercambiar un vector en  $M_i$  por un vector en  $E_i$  tal que  $||[\varepsilon_{\theta}]_{i+1}|| > ||[\varepsilon_{\theta}]_i||$
8. Hacer  $i \leftarrow i + 1$
9. Ir al paso 3.

El algoritmo de ascenso converge, debido a que por definición  $i < \tau \Rightarrow ||[\varepsilon_{\theta}]_i|| < ||[\varepsilon_{\theta}]_{i+1}||$  en donde  $\tau$  denota el índice del conjunto objetivo  $M_{\tau}$ .

El análisis para algoritmos genéticos está basado en la generación de un polinomio de  $n$  variables (multivariado) que corresponda a los resultados esperados, por lo que los parámetros necesarios son mínimos en comparación con las redes neuronales, sin embargo, el nivel de abstracción es mayor con respecto a las mismas. En contraste al algoritmo utilizado en esta investigación para las Redes Neuronales, en el Algoritmo Genético Elitista, descrito en el capítulo II, sección II.2.3, el polinomio que se obtiene describe la forma en que se va obteniendo los resultados, es decir, a medida que se van evaluando las generaciones en el algoritmo la solución se aproxima a la ecuación polinomial de grado  $n$  que se desea obtener.

La idea del algoritmo es generar un método general para obtener el valor de una variable dependiente como una función de dos variables independientes con las características principales:

- a) Los datos no tienen que conformar un patrón predeterminado, esto quiere decir que una colección aleatoria de los valores  $(x_i, y_i, z_i)$  mantienen una relación de la forma  $x = f(y, z)$ . De hecho, también se obtienen las relaciones  $y = f(x, z)$  y  $z = f(x, y)$ .
- b) El algoritmo siempre llega a un valor mínimo global bajo una norma minimax.
- c) La estabilidad numérica se lleva a cabo al seleccionar el valor apropiado del factor de perturbación aleatorio, por lo que es factible que cuando dos o más datos (de los tres o  $n$ ) son idénticos, el algoritmo converge.
- d) La forma para la aproximación, que se asume es polinomial, se modifica fácilmente para ajustar los datos para los casos donde el cambio sea sugerido por el conjunto de los datos.

Cuando se logra encontrar una expresión funcional como la descrita en ( III.i ) lo que se hace es sintetizar de manera muy compacta el conocimiento del sistema. En este sentido, el método "aprende" de la experiencia pasada (datos históricos). En un aproximante de este tipo, el número de coeficientes crece exponencialmente con el número de variables independientes y los grados máximos estipulados. Por ejemplo, para 3 variables y  $g_1=g_2=g_3=4$  el número de coeficientes es  $5^3=125$ . Una solución cuya caracterización implique un gran número de datos no es útil, incluyendo la complicación de estabilidad numérica que existe en sistemas de ecuaciones de gran tamaño.

Para evitar el problema anterior, se reemplaza el sistema como

$$f(v_1, \dots, v_p) = \sum_{i_1=0}^{g_1} \dots \sum_{i_p=0}^{g_p} \mu_{i_1 \dots i_p} C_{i_1 \dots i_p} v_1^{i_1} \dots v_p^{i_p} \quad ( III.xii )$$

en donde las variables  $\mu_{i_1 \dots i_p}$  solamente toman valores 0 o 1 y su propósito es excluir/incluir el coeficiente del cual son factor. Incluyendo un nuevo parámetro al problema, el cual indica el número máximo de coeficientes deseados ( $\gamma$ ). Es decir, ahora el problema consiste en encontrar los  $\gamma$  coeficientes más convenientes para minimizar  $\epsilon_\theta$  y encontrar los valores de los coeficientes elegidos.

Con estos ajustes, se buscan dos vectores de solución: el vector  $\mu$  y el vector C. El vector  $\mu$  tiene  $1 + \prod_{i=1}^p (g_i + 1)$  elementos y el vector C tiene  $\gamma$  elementos. Ambos son problemas altamente no lineales y NP completos, al utilizar algoritmos genéticos para encontrar la solución se genera el algoritmo genético de orden.

Algoritmo genético de orden

En primera instancia, para que se pueda resolver el problema planteado, el genoma tiene las siguientes características adicionales :

- a) Tiene una longitud m igual al número de coeficientes de la expresión ( III.xii )
- b) Cada uno de los bits del genoma corresponde con un coeficiente de ( III.xii ). Los coeficientes se indexan de manera canónica de izquierda a derecha, de manera que los índices corren desde 0 hasta m.
- c) Existen  $\gamma$  posiciones en las cuales el genoma toma el valor de 1; y m -  $\gamma$  posiciones en las cuales el genoma toma el valor de 0. Al número de 1's del genoma se le denomina el orden del genoma y se representa por  $\omega$ .
- d) Un 1 en el genoma significa que el coeficiente correspondiente existe ( $\mu=1$ ); por el contrario, un 0 en el genoma significa que el coeficiente no existe. En la siguiente tabla, la primera fila corresponde a las potencias asociadas a los coeficientes y el orden presentado representa el índice de los coeficientes. La segunda fila corresponde al genoma

000	001	002	010	011	012	020	021	022	100	101	102	110	111	112	120	121	122
1	1	0	0	0	0	1	0	1	0	1	0	0	1	1	0	0	1

El aproximante correspondiente al genoma anterior tiene la forma

$$f(v_1, v_2, v_3) = C_{000} + C_{001}v_3 + C_{020}v_2^2 + C_{022}v_2^2v_3^2 + C_{101}v_1v_3 + C_{111}v_1v_2v_3 + C_{112}v_1v_2v_3^2 + C_{122}v_1v_2^2v_3^2$$

Sea  $\Gamma = (g_1, g_2, \dots, g_p)$  un conjunto de grados máximos de cardinalidad p y  $f(v_1, \dots, v_p)$  el aproximante minimax de ( III.xii ). La longitud del genoma está dada por  $\prod_{i=1}^p (g_i + 1)$ , dado el índice de los coeficientes del genoma es posible encontrar las potencias asociadas a los coeficientes de la siguiente relación:

$$\tau_i = \frac{(N-1) - \sum_{j=1}^{i-1} \tau_j \prod_{k=j+1}^p (g_k + 1)}{\prod_{j=i-1}^p (g_j + 1)} \quad (\text{III. xiii})$$

en donde  $a > b \Rightarrow \sum_{x=a}^b x \equiv 0$  y  $\prod_{x=a}^b x \equiv 1$ . Los términos de las potencias  $(\tau_p, \tau_{p-1}, \dots, \tau_1)$  corresponden a las posiciones  $p, p-1, \dots, 1$  de la potencia.

De manera análoga, dados los términos  $\tau_p, \dots, \tau_1$  es posible encontrar el índice correspondiente de la expresión

$$N = 1 + \sum_{i=1}^p \tau_i \prod_{j=i+1}^p (g_j + 1) \quad (\text{III. xiv})$$

Estos genomas son susceptibles de ser tratados genéticamente de manera que la función de ajuste sea el valor del error  $\varepsilon_0$  y se trata de un proceso de minimización de dicho error. En cada generación se derivan dos componentes para cada individuo:

- a) El máximo error de ajuste ( $\varepsilon_0$ ).
- b) El vector de coeficientes C para el algoritmo de ascenso.

De esta manera, la población evoluciona hacia la combinación de coeficientes que minimicen  $\varepsilon_0$  bajo la condición de que el número de 1's en el genoma sea igual a  $\gamma$ . El algoritmo genético, en este caso, tiene una función objetivo en la cual, además de minimizar el error  $\varepsilon_0$ , debe mantener un genoma cuyo número de 1's sea siempre igual a  $\gamma$  ( $\omega=\gamma$ ). Es decir, los operadores genéticos de cruzamiento y mutación están limitados en el sentido de que éstos no dan origen a genomas válidos en todos los casos. Consecuentemente, se adopta una de las siguientes estrategias:

- a) Establecer una medida de castigo para aquellos genomas, resultantes de los operadores genéticos. Esta estrategia es más general, aunque menos eficiente.
- b) Modificarlos para que el orden del genoma se mantenga constante, aplicando un algoritmo de reparación de manera que los genomas que violen la restricción  $\omega=\gamma$  sean reemplazados por otros que no la violen.

El ajuste paramétrico de orden libre no busca la solución del problema (el vector C) de manera directa. La búsqueda de la forma de la solución, el mejor individuo de orden  $\gamma$ ,

conduce a la solución del problema de fondo, que es el conjunto de coeficientes que mejor ajusten los datos. Con la metodología descrita existe la posibilidad de interpolar y, evidentemente, una vez obtenido C, el polinomio resultante es susceptible de aceptar cualquier argumento, aún aquellos que se encuentran fuera de los límites del conjunto de datos original (extrapolación).

En resumen, el algoritmo genético comprende los siguientes pasos:

- a) Una población inicial de tamaño  $\Psi$  de cadenas binarias de longitud 
$$\lambda = \prod_{i=1}^n \text{deg}(V_i) + 1.$$
 Donde  $\text{deg}(V_i)$  es el grado más alto de  $V_i$  y cada bit en los individuos de la población hace referencia a uno de los  $\mu_i$  de la expresión ( III.xii ). Si el  $i$ -ésimo bit es 0, entonces  $\mu_i=0$  y en restricción solo se permite que  $\gamma$  bits sean igual al 1.
- b) Se obtiene el aproximante de acuerdo a la expresión III.x.
- c) Se obtiene el valor de rendimiento o fitness  $\Phi_i$  para cada uno de los individuos de la población.
- d) Se realiza una prueba y el proceso se detiene si  $\Phi_i$  es más pequeño que una constante  $\phi_k$  o la generación ha llegado a una constante  $\psi_k$ , en caso contrario, el algoritmo procede con el paso (e).
- e) Se realiza la reproducción dando un más alto porcentaje de supervivencia a los individuos para los que  $\Phi_i$  es más grande.
- f) Se realiza el cruzamiento
- g) Se realiza la mutación con probabilidad  $p_m$ .
- h) Se continúa con el paso (b).

### III.3 Análisis funcional de equivalencia

Ambas técnicas, Algoritmos Genéticos y Redes Neuronales, son adecuadas para aplicarlas a series de tiempo o aproximaciones de manera independiente, sin embargo, otros estudios integran ambas técnicas con la finalidad de obtener resultados más eficientes y robustos. En dichos estudios, en general, una vez que se ha definido la arquitectura de una red neuronal para realizar el pronóstico se aplica algún algoritmo

genético para mejorar el ajuste de parámetros y converger de manera más rápida a la solución<sup>4</sup>.

Como se describió a lo largo de este capítulo, y con apoyo de los fundamentos explicados en el capítulo II, ambas técnicas convergen a soluciones aproximadas para un mismo problema, particularmente series de tiempo. Una de las interrogantes más inquietantes para cualquier persona es ¿se puede predecir el futuro? Precisamente la presente investigación ayuda a responder la pregunta de acuerdo al enfoque del que se aplique el concepto de predicción de un evento. Los principales puntos que sustentan la afirmación anterior son los siguientes:

### Algoritmos Genéticos

Debido a las relaciones altamente no lineales entre los términos del polinomio aproximante, es difícil que los argumentos fuera del dominio de datos arrojen resultados muy significativos. Para subsanar esta limitación, es posible expresar una de las variables en función de sí misma, desplazada con respecto a las entradas de la tabla. Como ejemplo, supóngase la tabla de valores siguientes:

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$f(v_1, v_2, v_3, v_4, v_5)$
1	18.31200	30.96213	34.37500	61.37500	10.18800	10.31200
2	18.50000	30.96213	34.25000	62.37500	10.31200	11.25000
3	19.56200	30.96213	34.50000	62.25000	11.25000	11.56200

En la tabla anterior, los datos correspondientes a  $v_1, \dots, v_5$  constituyen un registro histórico en donde cada fila de la tabla corresponde a un momento del tiempo y  $f(v_1, \dots, v_5)$  corresponde al valor de  $v_5$  al tiempo  $t+1$ . Es decir,  $f(v_1, \dots, v_5) = v_5(t+1) = f(v_1(t), \dots, v_5(t))$

En general,  $f_i(v_1, \dots, v_\eta) = v_i(t + \Delta t) = f(v_1(t), \dots, v_\eta(t))$  con  $i = 1, \dots, \eta$  (III.xv)

en donde  $\Delta t$  determina qué tantos intervalos de tiempo se desean dejar pasar entre el momento de observación del fenómeno en cuestión y el momento de predicción.

<sup>4</sup> El enfoque de esta investigación queda fuera del alcance de mostrar la funcionalidad que se obtiene al integrar ambas técnicas, sin embargo, el lector podrá consultar la bibliografía para un análisis más detallado.

Otra alternativa es poner la variable dependiente o variable de proyección en función de otras variables y de sí misma en distintos instantes de tiempo

$$f_k(v_1, \dots, v_{j,\tau}) = v_k(t + \Delta t) = f(v_1(t), \dots, v_k(t), v_k(t+1), \dots, v_j(t)) \quad \text{con } \tau = 0, \dots, n \quad (\text{III.xvi})$$

en donde  $\tau$  especifica cuántas instancias de la variable independiente  $-1$  aparecen en la tabla de datos.

Considerado así el algoritmo genético, junto con el algoritmo de intercambio y el método de condicionamiento de datos, permiten encontrar una expresión matemática de la relación que existe entre un número indeterminado de variables y una variable de proyección. En general, este proceso es llevado a cabo de forma más genérica encontrando no una variable de proyección sino un conjunto de variables de proyección como se indica en III.xv. Así, el número de posibilidades y variantes es prácticamente ilimitado.

### Redes Neuronales

Las redes neuronales se aplican frecuentemente a modelos de predicción y debido a la facilidad en el manejo del software desarrollado, se pierde en cierto sentido el interés por conocer los fundamentos matemáticos que implica un modelo de esta naturaleza. En este sentido, resulta interesante el conocer el funcionamiento de una red neuronal. Como ya se explicó en el apartado anterior, es posible obtener una proyección a través de un algoritmo genético cuando el problema se interpreta en una expresión polinomial. Asimismo, las redes neuronales también yacen en un fundamento polinomial.

La red neuronal que utiliza el algoritmo de retropropagación funcionan gracias a una afinación de los parámetros de la red y un parámetro fundamental se refiere a la función de transferencia o activación. Se trata de enfocar el análisis en la estrategia de retropropagación. De acuerdo a Simon Haykin el entrenamiento de un perceptrón multicapa con el algoritmo de retropropagación es considerado como un método práctico para realizar un mapeo no lineal de entrada/salida de un problema natural.

Sea  $m_0$  el número de nodos de entrada (origen) de un perceptrón multicapa y sea  $M=m_L$  el número de neuronas en la capa de salida de una red. La relación entrada/salida de la red define un mapeo de un espacio de entrada Euclidiana  $m_0$ -dimensional a un espacio de

salida Euclidiana M-dimensional, el cuál es diferenciable en un espacio continuo infinito cuando la función de activación así lo permite y en este sentido surge la interrogante: ¿Cuál es el mínimo número de capas ocultas en un perceptrón multicapa con un mapeo entrada/salida para realizar una aproximación de algún mapeo continuo?

La respuesta a la interrogante anterior está planteada en el Teorema de Aproximación Universal<sup>5</sup> para un mapeo no lineal de entrada/salida, que dice:

Sea  $\varphi(\bullet)$  una función continua monótona creciente, limitada y no constante. Sea  $I_m$  el hipercubo de unidad  $m_0$ -dimensional  $[0,1]^{m_0}$ . El espacio de funciones continuas sobre  $I_m$  es denotado por  $C(I_m)$ . Entonces, dada una función  $f \in C(I_m)$  y  $\varepsilon > 0$ , existe un entero  $m_1$  y un conjunto de constantes reales  $\alpha_i$ ,  $b_i$  y  $w_{ij}$ , donde  $i=1, \dots, m_1$  y  $j=1, \dots, m_0$  tal que se define

$$F(x_1, \dots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \varphi \left( \sum_{j=1}^{m_0} w_{ij} x_j + b_i \right) \quad (\text{III.xvii})$$

como el aproximante de realización de la función  $f(\bullet)$ , esto es

$$|F(x_1, \dots, x_{m_0}) - f(x_1, \dots, x_{m_0})| < \varepsilon \quad (\text{III.xviii})$$

para todas las  $x_1, x_2, \dots, x_{m_0}$  que se encuentran dentro del espacio de entrada.

El Teorema de Aproximación Universal se aplica directamente a los perceptrones multicapa. En primer lugar, la función logística  $1/[1 + \exp(-v)]$  utilizada para la linealidad en un modelo neuronal para la construcción de un perceptrón multicapa es una función monótona incremental, limitada y no constante; en consecuencia, satisface las condiciones impuestas sobre la función  $\varphi(\bullet)$ . En segundo lugar, la ecuación ( III.xvii ) representa la salida de un perceptrón multicapa como sigue:

1. La red tiene  $m_0$  nodos de entrada y una capa oculta que consiste de  $m_1$  neuronas, las entradas son denotadas por  $x_1, x_2, \dots, x_{m_0}$ .
2. La neurona oculta  $i$  tiene los pesos sinápticos  $w_{i1}, \dots, w_{im_0}$  y bias  $b_i$ .
3. La red de salida es una combinación lineal de las salidas de las neuronas ocultas, con  $\alpha_1, \dots, \alpha_{m_1}$  definiendo los pesos sinápticos de la capa de salida.

<sup>5</sup> El Teorema de Aproximación Universal es considerado una extensión natural del Teorema de Weierstrass (1885) que indica que "cualquier función continua expresada en un intervalo real cerrado sobre el eje x, puede ser expresado en ese intervalo como una absoluta y uniforme serie de polinomios que convergen".

El teorema de aproximación universal es un teorema existencial en el sentido de que provee una justificación matemática para la aproximación de una función continua arbitraria contrario a la representación exacta. La ecuación ( III.xvii ), la cual es la base del teorema, generaliza aproximaciones por series finitas de Fourier. Así, el teorema indica que una sola capa oculta es suficiente para un perceptrón multicapa para calcular una aproximación  $\epsilon$  uniforme a partir de un conjunto de entrenamiento dado representado por el conjunto de entradas  $x_1, \dots, x_{m_0}$  y una salida deseada  $f(x_1, \dots, x_{m_0})$ . Sin embargo, el teorema no dice que una sola capa oculta sea óptima en el sentido del tiempo de aprendizaje, fácil de implementar o, más aún, la generalización<sup>6</sup>.

Un perceptrón multicapa entrenado con el algoritmo de retropropagación es en sí mismo un esquema sigmoideal anidado, escrito de la siguiente forma compacta para el caso de una salida como

$$F(x, w) = \varphi \left( \sum_k w_{ok} \varphi \left( \sum_l w_{kl} \varphi \left( \dots \varphi \left( \sum_i w_{li} x_i \right) \right) \right) \right) \quad ( \text{III.xix} )$$

donde  $\varphi(\bullet)$  es una función de activación sigmoideal común,  $w_{ok}$  es el peso sináptico de la neurona  $k$  en la última capa oculta a la neurona de salida simple  $o$ , y así sucesivamente, para los otros pesos sinápticos, y  $x_i$  es el  $i$ -ésimo del vector de entrada  $\mathbf{x}$ . El vector de pesos  $\mathbf{w}$  denota el conjunto ordenado de pesos sinápticos por capa, después las neuronas en una capa y después las sinápsis en una neurona. El esquema de funciones no lineales anidadas descritas en ( III.xvii ) es inusual en teoría de aproximación clásica, sin embargo, se trata del aproximador universal descrito en el Teorema de Aproximación Universal<sup>7</sup>.

Finalmente, la complejidad computacional de un algoritmo es usualmente medido en términos del número de multiplicaciones, sumas y almacenamiento necesarios para su implementación. Un algoritmo de aprendizaje se dice que es computacionalmente

---

6 Barron, en 1993, definió las propiedades de la aproximación a través de un perceptrón multicapa, asumiendo que la red tiene una capa de neuronas oculta utilizando la función sigmoideal y una neurona de salida lineal. Además, la red es entrenada utilizando el algoritmo de retropropagación y que se prueba con datos nuevos. Durante el entrenamiento, la red aprende los puntos específicos de la función de entrada  $f$  de acuerdo con los datos de entrenamiento y, en consecuencia, produce una función de aproximación  $F$ , como fue definida en (xv). Cuando la red es expuesta a datos de prueba que no se habían utilizado antes, la red de la función  $F$  actúa como un "estimador" de nuevos puntos de la función de entrada, es decir,  $F = \hat{f}$ .

7 Otra propiedad probada por Hornik y otros, en 1990, indica que los perceptrones multicapa pueden aproximar funciones que no son diferenciables en un sentido clásico, pero posee una derivada generalizada como el caso de la función escalón.

eficiente cuando su complejidad computacional es polinomial en cuanto al número de parámetros ajustables que son actualizados de una iteración a otra. En este sentido, es considerado que el algoritmo de retropropagación es computacionalmente eficiente; específicamente, al utilizar el perceptrón multicapa al realizar el entrenamiento ya que contiene un total de  $W$  pesos sinápticos (incluyendo la neurona bias), su complejidad es lineal en  $W$ , es decir, es  $O(W)$ .

#### Equivalencia de técnicas

La importancia de mostrar la equivalencia entre ambas técnicas deriva de que hay investigadores que trabajan sobre alguna técnica en especial (Algoritmos Genéticos o Redes Neuronales) y en otros casos la conjunción de ambos algoritmos permite fortalecer la investigación, sin embargo, en la práctica algunas aplicaciones están enfocadas a una herramienta. Al diversificar las técnicas de Inteligencia Artificial para resolver un problema de Series de Tiempo en Inversiones, se observa que con cualquiera de las técnicas analizadas, RN y AG, obteniendo resultados similares. Esto permite desarrollar aplicaciones más flexibles y confiables al incluir modelos alternativos para comparar los resultados apoyando directamente la toma de decisiones en casos donde la incertidumbre se incrementa pero que aún así se desea obtener siempre la mejor decisión, como es el caso de series de tiempo en inversiones.

La equivalencia de funciones entre algoritmos genéticos y redes neuronales se sigue de los siguientes hechos:

- En las redes neuronales se utilizan los pesos que corresponden a valores  $w_1, w_2, \dots, w_m$  que son conocidos y que se modifican o transfieren en cada época a través de una función de transferencia de tipo sigmoideal.
- La función sigmoideal corresponde a una exponencial que expresada en términos de polinomios, específicamente, la aproximación de  $e^x$  por medio del polinomio de Taylor en el punto 0 es:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots \quad (\text{III.xx})$$

- La aproximación de  $e^x$  es mejor en la medida en que el grado del polinomio de Taylor es mayor.

- En algoritmos genéticos se obtienen coeficientes,  $C_1, C_2, \dots, C_n$  que corresponde a los monomios de un polinomio de grado  $n=3$ .

- En términos generales, existe una relación funcional del tipo

$$w_1=f_1(C_1, C_2, \dots, C_n) \qquad C_1=F_1(w_1, w_2, \dots, w_m)$$

$$w_2=f_2(C_1, C_2, \dots, C_n) \qquad C_2=F_2(w_1, w_2, \dots, w_m)$$

$$w_3=f_3(C_1, C_2, \dots, C_n) \qquad C_3=F_3(w_1, w_2, \dots, w_m)$$

$$w_m=f_m(C_1, C_2, \dots, C_n) \qquad C_n=F_n(w_1, w_2, \dots, w_m)$$

Donde  $w_1, w_2, \dots, w_m$  corresponde a los pesos a través de una función  $f_1, f_2, \dots, f_m$  que hace referencia a los coeficientes de los polinomios encontrados con el algoritmo genético. Asimismo,  $C_1, C_2, \dots, C_n$  hacen referencia a una función  $F_1, F_2, \dots, F_n$  evaluada sobre los pesos de una red neuronal.

- Esta relación funcional permite encontrar las aproximaciones necesarias para cada estimación y mostrar la equivalencia al determinar los valores del caso de estudio.
- Sin perder generalidad, lo que se indica es que existe una relación de equivalencia entre los pesos de una red neuronal y los coeficientes que forman cada uno de los monomios de un polinomio de grado  $n$ .
- Finalmente, la equivalencia no está definida en términos de los resultados de aplicar ambas técnicas, RN y AG, aun mismo caso de estudio, debido a que si fuera este el caso muchas técnicas serían equivalentes. Sin embargo, a través de expresiones de polinomios se puede generar la correspondencia deseada sobre estas técnicas específicas de Inteligencia Artificial.

## CAPÍTULO IV. PRONÓSTICO DE SOCIEDADES DE INVERSIÓN

*Digamos que existen dos tipos de mentes poéticas:  
una apta para inventar fábulas y otra dispuesta a creerlas.*

*Galileo Galilei*

En el presente capítulo se presenta un caso de estudio donde se aplican los conceptos explicados en los capítulos precedentes y el punto importante es sustentar de forma pragmática la investigación realizada. Aunque en ningún sentido se pretende generalizar las técnicas mostradas al caso de series de tiempo, si facilita el inicio para la continuidad de la investigación en términos más generales.

A lo largo de este capítulo se realiza la estimación de pronósticos de tres sociedades de inversión de acuerdo al análisis técnico que comprende la inversión en un mercado de valores, particularmente la Bolsa Mexicana de Valores. Para ello se utilizan datos de la Casa de Bolsa Acciones y Valores Banamex integrante del Grupo Financiero Banamex Citigroup. El objetivo particular es aplicar las técnicas estudiadas al mismo caso de estudio bajo las mismas condiciones y al analizar los resultados concluir sobre la presente investigación.

La importancia del análisis de inversiones radica en la gran cantidad de dinero que se maneja y la incertidumbre que se presenta al desconocer todos los factores que intervienen en una decisión con estas dos características básicas. El objetivo es estimar un valor para una sociedad de inversión en particular, puede ser aplicado a acciones que coticen en la BMV, a partir de los datos que se han obtenido en diferentes instantes del tiempo. En este sentido, se considera que cuando los valores estimados se encuentran fuera del rango de observación, entonces se está prediciendo el comportamiento del sistema en estudio.

## IV.1 Características del caso de estudio

En ambas técnicas, Redes Neuronales y Algoritmos Genéticos, aplicadas a pronósticos en Sociedades de Inversión se utilizaron los mismos datos compuestos por seis meses de precios de las sociedades seleccionadas para realizar el pronóstico. Estos valores representan cada día de operaciones en la BMV, aproximadamente 152 datos (del 1 de julio 2004 al 31 de enero 2005), los cuales se muestran en la Tabla IV.1 para cada una de las Sociedades estimadas.

FECHA	HZPLAZO	ACCIPAT	ACCIVAL	FECHA	HZPLAZO	ACCIPAT	ACCIVAL
01-Jul-04	1.478150	180.924220	124.606756	01-Nov-04	1.506942	179.115497	133.385061
02-Jul-04	1.483473	178.910594	124.235706	02-Nov-04	1.507080	179.792589	133.189267
05-Jul-04	1.483492	178.894612	124.125482	03-Nov-04	1.507211	181.199736	134.394189
06-Jul-04	1.483605	178.276106	123.393519	04-Nov-04	1.507685	183.424976	135.002144
07-Jul-04	1.483865	178.628611	122.608331	05-Nov-04	1.507831	183.343716	134.402876
08-Jul-04	1.483580	177.888121	121.833056	08-Nov-04	1.506970	183.859071	134.156593
09-Jul-04	1.485224	177.156294	122.339234	09-Nov-04	1.505916	184.303385	134.507414
12-Jul-04	1.486631	176.938256	122.659703	10-Nov-04	1.506269	183.234426	134.518288
13-Jul-04	1.488595	177.580305	122.098265	11-Nov-04	1.506597	184.059321	136.101750
14-Jul-04	1.489169	176.868420	120.447439	12-Nov-04	1.508329	185.739563	136.032872
15-Jul-04	1.488132	174.320934	120.524734	15-Nov-04	1.509283	184.726574	136.040719
16-Jul-04	1.492104	171.663201	120.952716	16-Nov-04	1.508894	182.788041	135.905939
19-Jul-04	1.492325	172.360357	121.028851	17-Nov-04	1.509928	182.721179	136.248833
20-Jul-04	1.489283	172.690809	121.780875	18-Nov-04	1.509784	182.441274	136.280741
21-Jul-04	1.486258	173.512948	121.405333	19-Nov-04	1.509371	180.716929	134.658877
22-Jul-04	1.486612	175.359451	121.690888	22-Nov-04	1.509697	180.245346	134.974171
23-Jul-04	1.485803	174.108205	121.086078	23-Nov-04	1.509513	179.697474	134.927701
26-Jul-04	1.484384	172.916356	121.189910	24-Nov-04	1.510250	179.982478	135.073071
27-Jul-04	1.481654	173.270053	122.568854	25-Nov-04	1.510687	181.500088	135.735175
28-Jul-04	1.483045	173.191846	122.155165	26-Nov-04	1.511640	180.146418	136.349941
29-Jul-04	1.484271	175.379319	121.978592	29-Nov-04	1.511927	178.428115	137.279220
30-Jul-04	1.485852	173.488888	122.505007	30-Nov-04	1.512129	177.272966	136.620128
02-Ago-04	1.485626	175.464811	123.021727	01-Dic-04	1.513155	180.055392	137.692028
03-Ago-04	1.485309	174.251613	123.225528	02-Dic-04	1.513104	179.522318	136.743195
04-Ago-04	1.484821	175.699498	122.150070	03-Dic-04	1.515343	178.893735	136.627724
05-Ago-04	1.484672	173.779569	122.080998	06-Dic-04	1.517400	180.761967	137.458164
06-Ago-04	1.488129	170.858605	120.499232	07-Dic-04	1.517702	181.168379	136.965930
09-Ago-04	1.488846	171.900817	120.225585	08-Dic-04	1.517206	182.509860	136.913374
10-Ago-04	1.488063	174.088441	121.181147	09-Dic-04	1.516872	182.785043	137.093421
11-Ago-04	1.488045	174.400291	120.737927	10-Dic-04	1.517575	182.868690	138.033123
12-Ago-04	1.488952	173.378231	120.390270	13-Dic-04	1.518127	184.521282	138.177383
13-Ago-04	1.491383	173.246891	120.247288	14-Dic-04	1.518621	185.171248	139.445586
16-Ago-04	1.491611	175.392406	121.014838	15-Dic-04	1.519620	185.555475	139.813350

17-Ago-04	1.493516	176.193878	121.660560		16-Dic-04	1.519390	185.751565	139.966422
18-Ago-04	1.492532	177.931738	122.427159		17-Dic-04	1.520147	184.334288	140.072924
19-Ago-04	1.491288	177.433661	122.585719		20-Dic-04	1.521064	183.441111	140.059892
20-Ago-04	1.491620	178.353869	122.913428		21-Dic-04	1.522004	185.198429	141.229360
23-Ago-04	1.490042	178.859648	123.197647		22-Dic-04	1.522254	187.781741	141.763404
24-Ago-04	1.490590	179.191378	123.481273		23-Dic-04	1.522970	189.849627	142.072518
25-Ago-04	1.491332	180.224604	123.523504		24-Dic-04	1.523698	189.942259	142.483142
26-Ago-04	1.492188	180.701784	123.183122		27-Dic-04	1.523377	188.710819	142.623193
27-Ago-04	1.493285	181.997997	123.356859		28-Dic-04	1.522812	190.136484	143.407240
30-Ago-04	1.492897	181.461794	123.254405		29-Dic-04	1.523386	188.916710	144.433294
31-Ago-04	1.494444	181.408794	123.780859		30-Dic-04	1.523581	188.447373	143.870508
01-Sep-04	1.494647	181.727533	124.309274		31-Dic-04	1.524307	187.846132	143.483996
02-Sep-04	1.494594	184.900686	125.386423		03-Ene-05	1.524560	189.884717	144.366316
03-Sep-04	1.494556	184.303218	124.683920		04-Ene-05	1.524231	190.868698	142.706880
06-Sep-04	1.494741	184.420078	124.995341		05-Ene-05	1.524615	192.399435	141.033505
07-Sep-04	1.494334	186.536179	125.947546		06-Ene-05	1.525532	193.882087	141.885712
08-Sep-04	1.495058	185.482558	126.249711		07-Ene-05	1.526059	191.454085	139.763569
09-Sep-04	1.495622	185.903717	126.462164		10-Ene-05	1.526184	189.916516	139.666224
10-Sep-04	1.496344	186.983387	127.293589		11-Ene-05	1.526681	189.199056	137.844332
13-Sep-04	1.497543	185.377952	126.879484		12-Ene-05	1.527202	187.785742	138.894448
14-Sep-04	1.497307	186.170401	127.573513		13-Ene-05	1.527680	187.341288	139.890990
15-Sep-04	1.498349	184.894981	127.096144		14-Ene-05	1.528534	187.264232	141.746990
17-Sep-04	1.499663	184.052231	128.011389		17-Ene-05	1.528724	187.356540	142.723637
20-Sep-04	1.499450	179.082867	127.858019		18-Ene-05	1.528826	190.942778	144.397140
21-Sep-04	1.500328	179.338352	128.134147		19-Ene-05	1.529534	190.083665	144.558052
22-Sep-04	1.501252	175.451536	127.336281		20-Ene-05	1.529324	189.322168	142.426168
23-Sep-04	1.501148	173.722023	127.645888		21-Ene-05	1.530185	190.221091	141.597683
24-Sep-04	1.500678	174.562430	127.874813		24-Ene-05	1.530617	190.791324	142.036777
27-Sep-04	1.501165	173.317501	126.337885		25-Ene-05	1.530550	191.877936	143.048053
28-Sep-04	1.501509	175.536314	127.483539		26-Ene-05	1.530389	191.486786	144.438551
29-Sep-04	1.500546	176.244379	128.501579		27-Ene-05	1.530132	192.105722	143.828326
30-Sep-04	1.500177	174.624964	128.132291		28-Ene-05	1.530048	191.468366	144.113295
01-Oct-04	1.501012	176.234348	129.076589		31-Ene-05	1.529960	192.610504	144.664513
04-Oct-04	1.501401	175.315453	129.926488		01-Feb-05	1.530106	193.972628	146.465010
05-Oct-04	1.502076	174.660608	129.499870		02-Feb-05	1.530359	194.062980	146.382695
06-Oct-04	1.502140	175.419604	129.132449		03-Feb-05	1.530218	193.029012	147.309194
07-Oct-04	1.502285	175.795517	129.104617		04-Feb-05	1.531128	195.126937	147.377046
08-Oct-04	1.504487	174.008182	127.706225		07-Feb-05	1.531221	195.438526	147.411613
11-Oct-04	1.505936	175.087424	128.089149		08-Feb-05	1.531340	195.995715	147.877665
12-Oct-04	1.506466	175.462464	128.558721		09-Feb-05	1.531997	194.013259	149.263176
13-Oct-04	1.508212	173.599601	128.095713		10-Feb-05	1.532950	193.509733	149.435209
14-Oct-04	1.507494	173.357669	127.645678		11-Feb-05	1.534449	194.764065	149.559114
15-Oct-04	1.507405	175.501919	128.612908		14-Feb-05	1.534408	193.969352	148.580303
18-Oct-04	1.506199	176.077703	129.155889		15-Feb-05	1.533835	194.790497	149.539946
19-Oct-04	1.506774	174.188629	128.828851		16-Feb-05	1.532170	194.201288	149.035075
20-Oct-04	1.506858	173.875092	129.290523		17-Feb-05	1.532170	194.201288	149.035075
21-Oct-04	1.507704	172.521228	129.975744		18-Feb-05			

22-Oct-04	1.507180	171.820649	130.166697		21-Feb-05			
25-Oct-04	1.506539	172.327380	129.691734		22-Feb-05			
26-Oct-04	1.506339	174.199619	130.870464		23-Feb-05			
27-Oct-04	1.505849	176.745270	132.264175		24-Feb-05			
28-Oct-04	1.506174	178.739871	131.853953		25-Feb-05			
29-Oct-04	1.506955	179.051895	132.936815		28-Feb-05			

Tabla IV.1 Precios de Sociedades de Inversión

Es importante mencionar que la sociedad HZPLAZO es una sociedad de Instrumentos de Deuda, es decir, la inversión se realiza principalmente en papeles gubernamentales y sociedades de media y baja bursatilidad. Por otra parte, las sociedades ACCIPAT y ACCIVAL corresponden a Renta Variable cuyo portafolio también incluye papel gubernamental pero también incluyen un capital importante de acciones de alta bursatilidad. A priori, en las sociedades de instrumento de deuda se espera una ganancia menor en comparación con las de renta variable pero también el riesgo es menor por lo que depende en muchos sentidos del tipo de inversionista para decidir la manera de construir el portafolio. En términos generales, los especialistas recomiendan diversificar el capital para mantener un equilibrio entre ganancia - riesgo inherente al tipo de inversión. Por ejemplo, un inversionista moderado puede invertir el capital en 60% de instrumentos de deuda y 40% de renta variable.

Las gráficas originales de los datos se presentan en las gráficas IV.1, IV.2 y IV.3. Se obtiene la gráfica original de los datos del 1 de julio 2004 al 31 de enero 2005 (152 datos).

### HZPLAZO

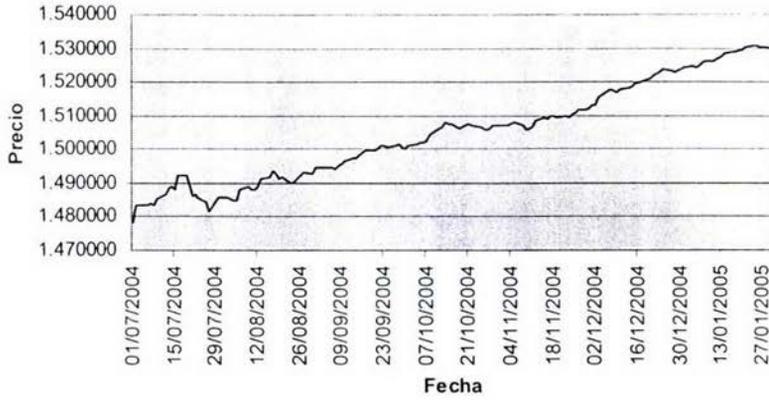


Figura IV.1 Gráfica original de HZPLAZO

### ACCIPAT

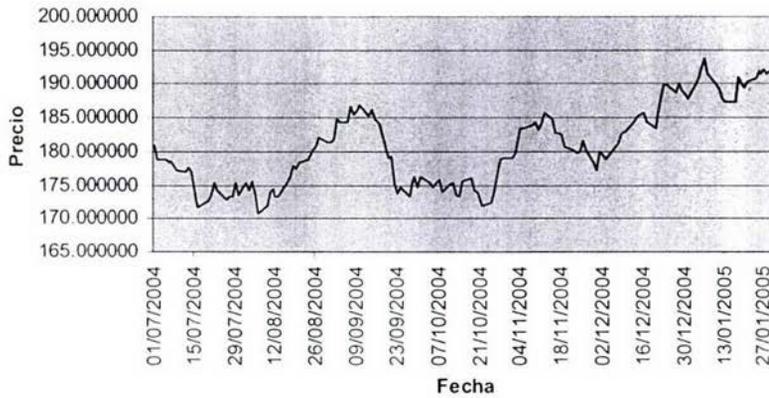


Figura IV.2 Gráfica original de ACCIPAT

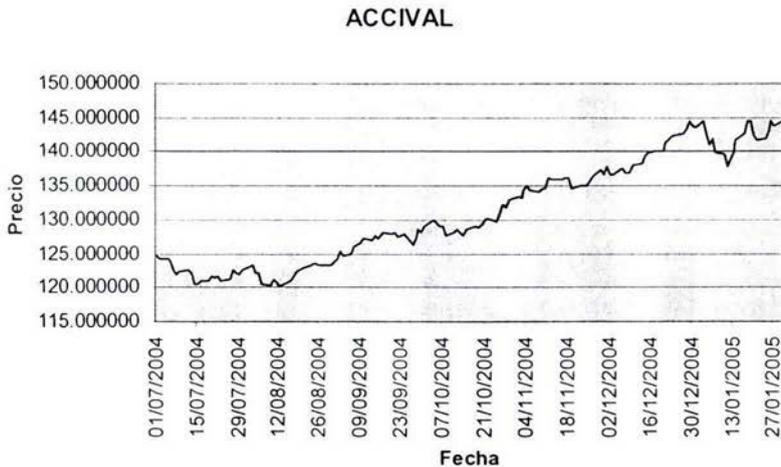


Figura IV.2 Gráfica original de ACCIVAL

#### IV.2 Pronósticos con Redes Neuronales

Para realizar el pronóstico con redes neuronales se utilizó el DataEngine versión 4.0, el cual utiliza el algoritmo de retropropagación multicapa de Rosenblatt, descrito en los capítulos II y III de esta investigación. El procedimiento para estimar los valores de cada sociedad de inversión se describe a continuación.

#### HZPLAZO

- a) Se obtiene la gráfica original de los datos del 1 de julio 2004 al 31 de enero 2005 (152 datos). Figura IV.1.
- b) Se obtiene el promedio móvil a 5 días para **cada evento** observado con la finalidad de afinar la aproximación. Posteriormente se **obtiene** la diferencia entre el valor original y el promedio móvil con el objetivo de eliminar la tendencia. Con éstos datos actualizados se realiza el proceso.

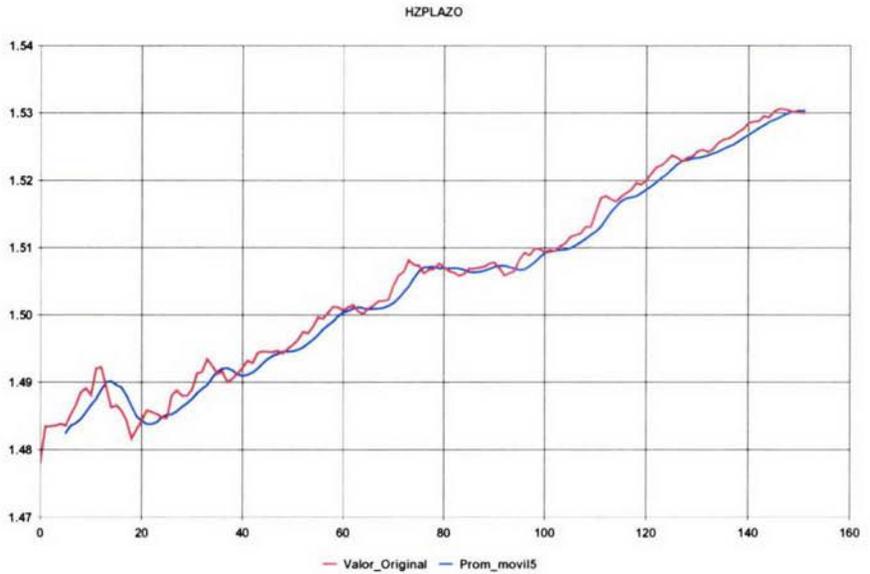


Figura IV.4 Gráfica original de HZPLAZO comparada con promedio móvil a 5 días.

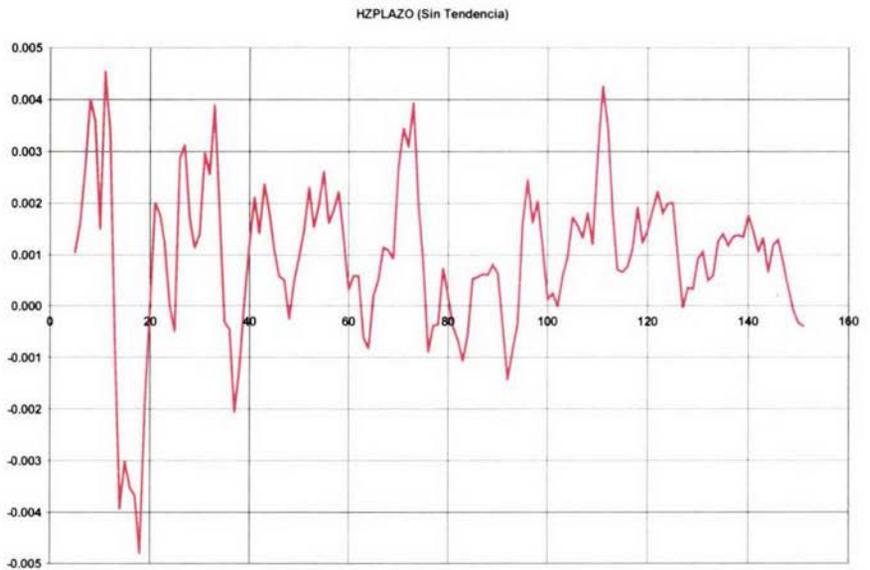


Figura IV.5 Gráfica del promedio móvil de HZPLAZO sin tendencia.

- c) Se normalizan los datos en una escala entre [0.1, 0.9] para garantizar que todos los datos se encuentran en el mismo rango. Este preprocesamiento de datos reduce los valores para la estimación a 142 de los cuales 115 datos (81%) fueron utilizados para entrenar la red y los 27 restantes (19%) en la etapa de pruebas.
- d) Se definió y entrenó una red neuronal para la sociedad de inversión de acuerdo a los resultados de errores mínimos observados al entrenar la red.

Parámetro	Valor
Número de capas	3
Número de conexiones	6
Algoritmo de aprendizaje	Retropropagación
Estrategia de aprendizaje	Delta Simple
Orden de presentación	Aleatorio
Podado	0.02
Épocas	20000
Inicialización de pesos	[-0.1, 0.1]

Tabla IV.2 Parámetros para la Red Neuronal

Parámetro	Capa de Entrada	Capa Oculta	Capa de Salida
Neuronas	5	1	1
Función de transferencia	Lineal	Sigmoidal	Lineal
Tasa aprendizaje	0.3	N/A	0.001
Momentum	0.3	N/A	0.3
Disminución de pesos	0.999999	N/A	0.999999

Tabla IV.3 Parámetros por capa para la red neuronal

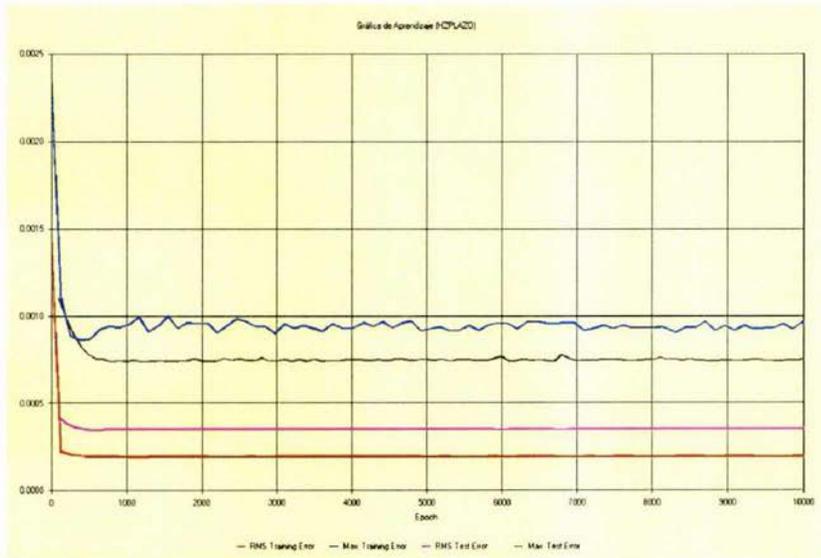


Figura IV.6 Gráfica de errores durante el proceso de entrenamiento para HZPLAZO.

- e) Se generó una carta en DataEngine. Una carta es una secuencia de bloques con la finalidad de automatizar los procesos. En este caso se toman los valores originales y con los datos preprocesados como archivos de entrada, se entrena la red con el perceptrón multicapa (NetHzplazo.mlp) definido y se generan las gráficas de los datos resultantes.

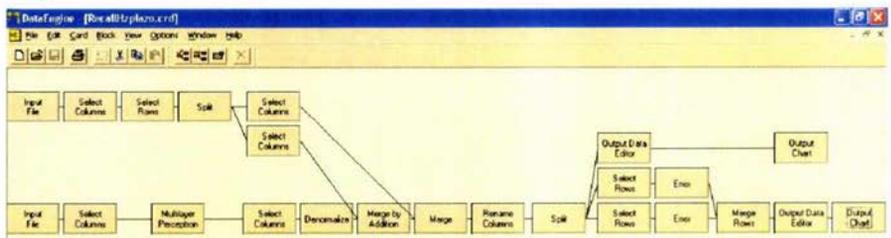


Figura IV.7 Carta automatizada para pronosticar

- f) Una vez entrenada la red, se generó el pronóstico y se comparó con el valor original.

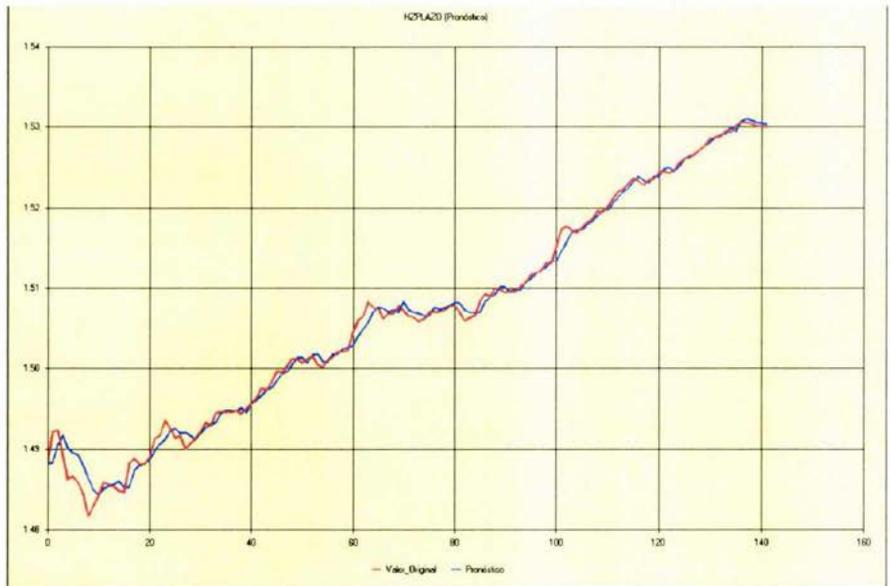


Figura IV.8 Gráfica de comparación de datos originales y pronóstico para HZPLAZO.

#### ACCIPAT

- a) Se obtiene la gráfica original de los datos del 1 de julio 2004 al 31 de enero 2005 (152 datos). Figura IV.2.
- b) Se obtiene el promedio móvil a 5 días para cada evento observado con la finalidad de afinar la aproximación. Posteriormente se obtiene la diferencia entre el valor original y el promedio móvil con el objetivo de eliminar la tendencia. Con éstos datos actualizados se realiza el proceso.

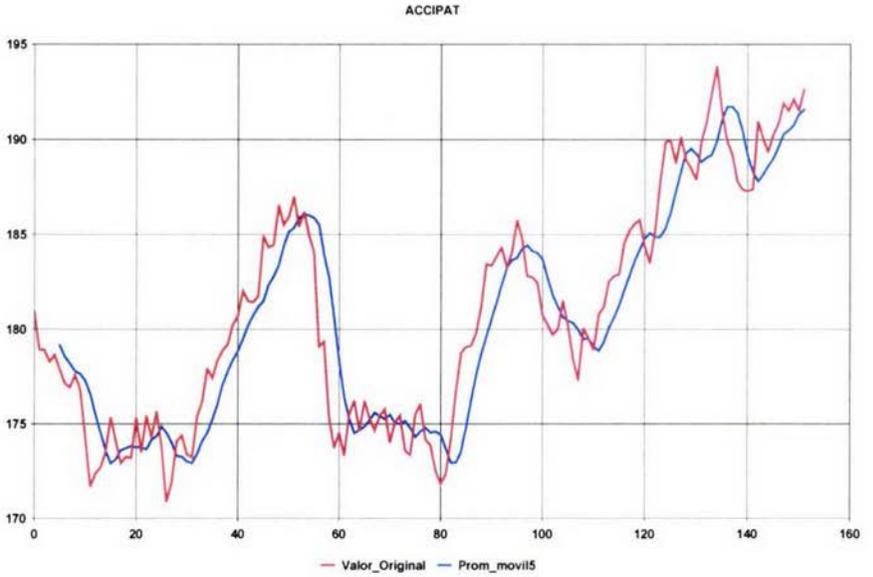


Figura IV.9 Gráfica original de ACCIPAT comparada con promedio móvil a 5 días.

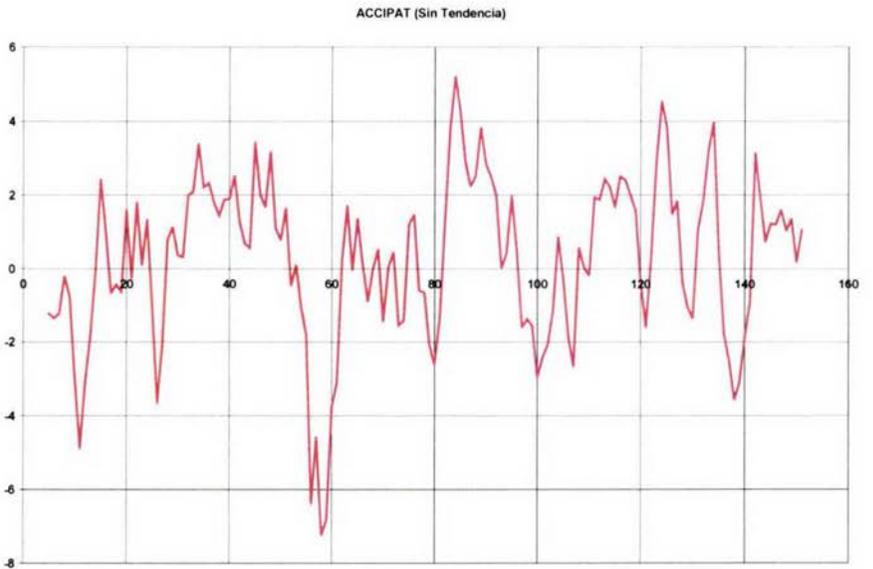


Figura IV.10 Gráfica del promedio móvil de ACCIPAT sin tendencia.

- c) Se normalizan los datos en una escala entre [0.1, 0.9] para garantizar que todos los datos se encuentran en el mismo rango. Este preprocesamiento de datos reduce los valores para la estimación a 142 de los cuales 115 datos (81%) fueron utilizados para entrenar la red y los 27 restantes (19%) en la etapa de pruebas.
- d) Se definió y entrenó una red neuronal para la sociedad de inversión de acuerdo a los resultados de errores mínimos observados al entrenar la red.

Parámetro	Valor
Número de capas	3
Número de conexiones	6
Algoritmo de aprendizaje	Retropropagación
Estrategia de aprendizaje	Delta Acumulada
Orden de presentación	Aleatorio
Podado	0.02
Épocas	2400
Inicialización de pesos	[-0.1, 0.1]

Tabla IV.4 Parámetros para la Red Neuronal

Parámetro	Capa de Entrada	Capa Oculta	Capa de Salida
Neuronas	5	1	1
Función de transferencia	Lineal	Sigmoidal	Lineal
Tasa aprendizaje	0.45	N/A	0.002
Momentum	0.2	N/A	0.3
Disminución de pesos	0.9999	N/A	0.99999

Tabla IV.5 Parámetros por capa para la red neuronal

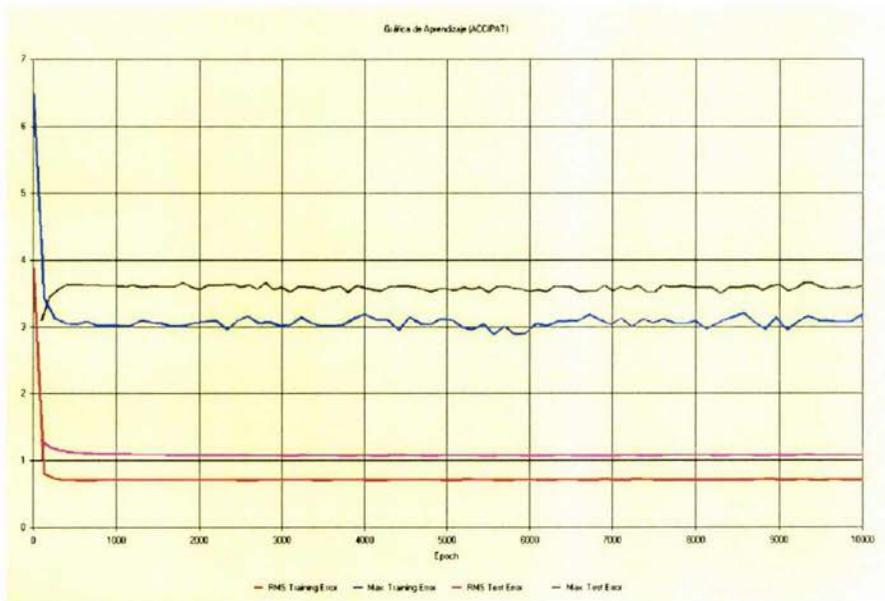


Figura IV.11 Gráfica de errores durante el proceso de entrenamiento para ACCIPAT.

- e) Se generó una carta en DataEngine. Una carta es una secuencia de bloques con la finalidad de automatizar los procesos. En este caso se toman los valores originales y con los datos preprocesados como archivos de entrada, se entrena la red con el perceptrón multicapa (NetAccipat.mlp) definido y se generan las gráficas de los datos resultantes (Figura IV.7)
- f) Una vez entrenada la red, se generó el pronóstico y se comparó con el valor original.

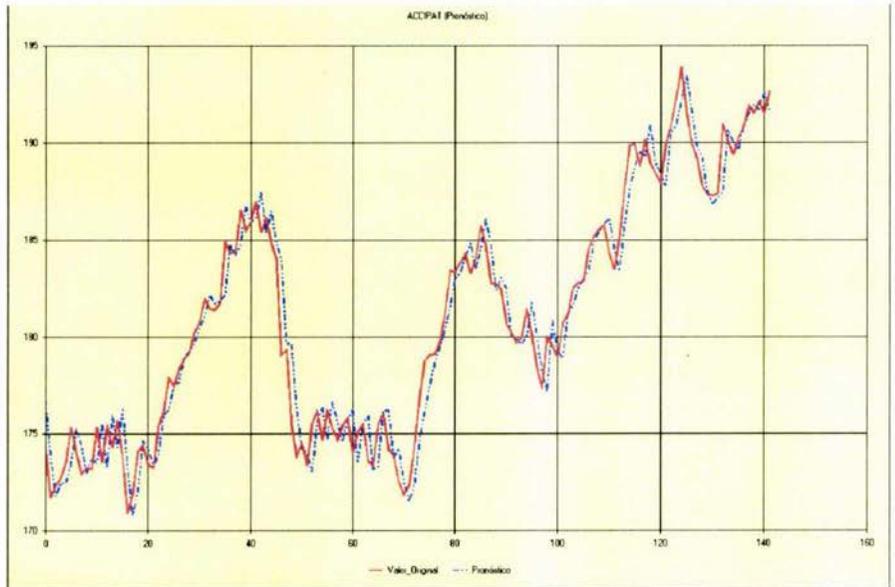


Figura IV.12 Gráfica de comparación de datos originales y pronóstico para ACCIPAT.

#### ACCIVAL

- a) Se obtiene la gráfica original de los datos del 1 de julio 2004 al 31 de enero 2005 (152 datos). Figura IV.3.
- b) Se obtiene el promedio móvil a 5 días para cada evento observado con la finalidad de afinar la aproximación. Posteriormente se obtiene la diferencia entre el valor original y el promedio móvil con el objetivo de eliminar la tendencia. Con éstos datos actualizados se realiza el proceso.

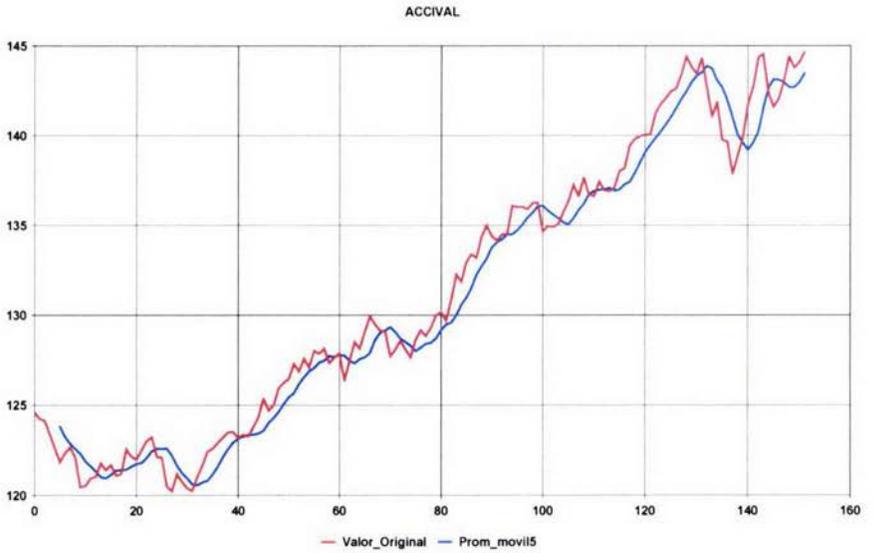


Figura IV.13 Gráfica original de ACCIVAL comparada con promedio móvil a 5 días.

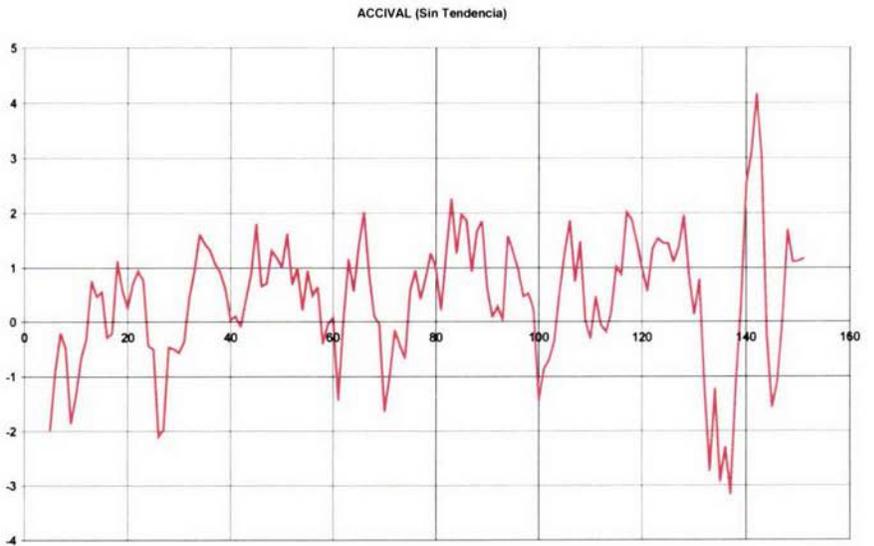


Figura IV.14 Gráfica del promedio móvil de ACCIVAL sin tendencia.

- c) Se normalizan los datos en una escala entre [0.1, 0.9] para garantizar que todos los datos se encuentran en el mismo rango. Este preprocesamiento de datos reduce los valores para la estimación a 142 de los cuales 115 datos (81%) fueron utilizados para entrenar la red y los 27 restantes (19%) en la etapa de pruebas.
- d) Se definió y entrenó una red neuronal para la sociedad de inversión de acuerdo a los resultados de errores mínimos observados al entrenar la red.

Parámetro	Valor
Número de capas	3
Número de conexiones	12
Algoritmo de aprendizaje	Retropropagación
Estrategia de aprendizaje	Delta Simple
Orden de presentación	Aleatorio
Podado	0.02
Épocas	39100
Inicialización de pesos	[-0.1, 0.1]

Tabla IV.6 Parámetros para la Red Neuronal

Parámetro	Capa de Entrada	Capa Oculta	Capa de Salida
Neuronas	5	2	1
Función de transferencia	Lineal	Sigmoidal	Lineal
Tasa aprendizaje	0.3	N/A	0.01
Momentum	0.3	N/A	0.2
Disminución de pesos	0.999999	N/A	0.999999

Tabla IV.7 Parámetros por capa para la red neuronal

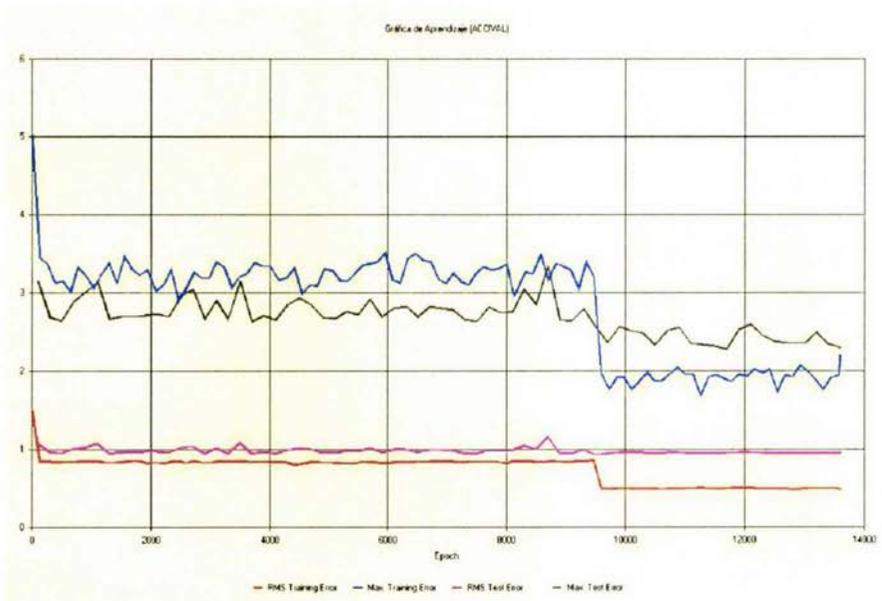


Figura IV.15 Gráfica de errores durante el proceso de entrenamiento para ACCIVAL.

- e) Se generó una carta en DataEngine. Una carta es una secuencia de bloques con la finalidad de automatizar los procesos. En este caso se toman los valores originales y con los datos preprocesados como archivos de entrada, se entrena la red con el perceptrón multicapa (NetAccival.mlp) definido y se generan las gráficas de los datos resultantes (Figura IV.7)
- f) Una vez entrenada la red, se generó el pronóstico y se comparó con el valor original.

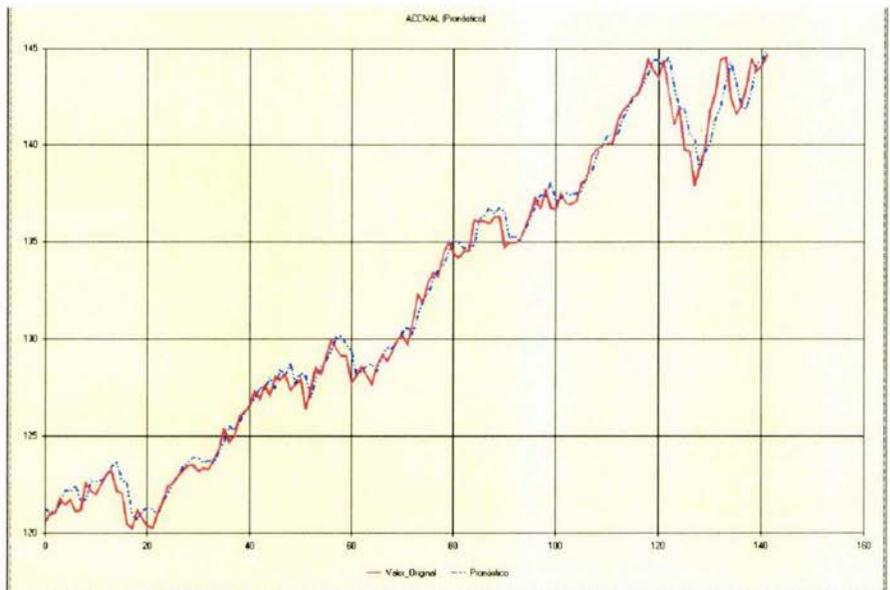


Figura IV.16 Gráfica de comparación de datos originales y pronóstico para ACCIVAL.

### IV.3 Pronósticos con Algoritmos Genéticos

La técnica descrita para el pronóstico con Algoritmos Genéticos es diferente en el sentido de que no se necesita entrenar el polinomio para que aprenda los datos, es decir, el interés no recae en el origen de los datos. Similar al análisis realizado para Redes Neuronales, en la sección anterior IV.2, se toma como base los datos de la Tabla IV.1, para realizar la estimación del promedio móvil a 5 días se desea expresar una variable en función de otras 5 variables independientes y el grado máximo que cada variable independiente tendrá en el polinomio será de 3. El algoritmo utilizado fue desarrollado en MatLab 5.2.

Con este planteamiento, el polinomio necesario para esta solución comprendería

$$\prod_{i=1}^5 \{\text{deg}(V_i) + 1\} = 4^5 = 1024$$
 coeficientes, en términos prácticos se trata de un polinomio

difícilmente manejable debido a la cantidad de almacenamiento necesaria y tiempo de procesamiento de cómputo, sin embargo, para el problema la complejidad del método utilizado no debería ser un factor determinante en cuanto a la solución que se busca.

Lo que se necesita es un algoritmo que encuentre no solo la solución para un subconjunto del aproximante

$$f(v_1, \dots, v_p) = \sum_{i_1=0}^{g_1} \dots \sum_{i_p=0}^{g_p} \mu_{i_1, \dots, i_p} C_{i_1, \dots, i_p} v_1^{i_1} \dots v_p^{i_p} \quad (IV.i)$$

sino que también encuentre los mejores  $\mu_i$  de acuerdo a un determinado número de términos,  $\eta$ . En este sentido, la variable  $\mu_i$  toma los valores de 0 o 1 y solamente indica la presencia o ausencia del i-ésimo término del polinomio. Por ejemplo, si tenemos 2 variables independientes,  $n=2$  de grado 2, la ecuación (i) corresponde a

$$P(V_1, V_2) = \mu_{00}C_{00} + \mu_{01}C_{01}V_2 + \mu_{02}C_{02}V_2^2 + \dots \\ \mu_{10}C_{10}V_1 + \mu_{11}C_{11}V_1V_2 + \mu_{12}C_{12}V_1V_2^2 + \dots \\ \mu_{20}C_{20}V_1^2 + \mu_{21}C_{21}V_1^2V_2 + \mu_{22}C_{22}V_1^2V_2^2 \quad (IV.ii)$$

Para el caso de estudio,  $n=5$  y el grado máximo es de 3 para cada variable independiente, se consideraron  $\eta=20$  términos significativos por lo que el espacio de búsqueda de la

$$\text{solución corresponde a } \binom{1024}{20} = \frac{1024!}{20! 1004!} = \frac{1024 \times \dots \times 1005}{20!} > \frac{1.33 \cdot 10^{60}}{2.43 \cdot 10^{18}} > 5.48 \cdot 10^{41}.$$

Se asume que la serie de observaciones del proceso toma la forma de un conjunto de pares,  $[(\bar{x}_1, y_1), \dots, (\bar{x}_n, y_n)]$ , donde  $\bar{x}_i = (v_1, \dots, v_p)_i$  y  $y_i = (v_{p+1})_i$ , entonces, los datos son colocados en una tabla donde cada columna corresponde a uno de los parámetros del sistema. El número de columnas es denotado por  $p+1$  donde  $p$  es el número de variables independientes. El número de registros es denotado por  $n$  mientras que el conjunto de datos se denota por  $N$ . En la Tabla IV.8 se ejemplifica esta idea para los datos de la sociedad HZPLAZO de acuerdo a los datos presentados en la Tabla IV.1

$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6=(v_1+v_2+v_3+v_4+v_5)/5$
1.478150	1.483473	1.483492	1.483605	1.483865	1.482517
1.483473	1.483492	1.483605	1.483865	1.483580	1.483603
1.483492	1.483605	1.483865	1.483580	1.485224	1.483953
1.483605	1.483865	1.483580	1.485224	1.486631	1.484581

1.483865	1.483580	1.485224	1.486631	1.488595	1.485579
1.483580	1.485224	1.486631	1.488595	1.489169	1.486640
1.528826	1.529534	1.529324	1.530185	1.530617	1.529697
1.529534	1.529324	1.530185	1.530617	1.530550	1.530042
1.529324	1.530185	1.530617	1.530550	1.530389	1.530213
1.530185	1.530617	1.530550	1.530389	1.530132	1.530375
1.530617	1.530550	1.530389	1.530132	1.530048	1.530347
1.530550	1.530389	1.530132	1.530048	1.529960	1.530216
1.528826	1.529534	1.529324	1.530185	1.530617	1.529697

Tabla IV.8 Fragmento de los datos en el tiempo para HZPLAZO

$$v_6 = v_5(t+1) = f(v_1(t), v_2(t), v_3(t), v_4(t), v_5(t)) = (v_1 + v_2 + v_3 + v_4 + v_5) / 5$$

Los parámetros adicionales para el Algoritmo Genético se presentan en la Tabla IV.9

Parámetro	Valor
Número de individuos	50
Número de generaciones	50
Número de vectores de datos	148
Número de variables independientes	5
Máximo grado de las variables	3
Tamaño del genoma, $\Gamma$	1024
Número de términos	20
Probabilidad de cruce, $P_c$	0.9
Probabilidad de mutación, $P_m$	0.005
Elitismo total, individuos	50

Tabla IV.9 Parámetros para el Algoritmo Genético

Debido a que en la propuesta se incluye el número de términos que serán considerados al encontrar el polinomio que se ajuste a los datos ya que si no se hiciera así se estaría trabajando con los 1024 coeficientes lo cual sería impráctico en tiempo de respuesta e interpretación, por este motivo se introduce también un operador de preservación de orden. El algoritmo genético deberá mantener un genoma cuyo número de 1's ( $\omega$ ) siempre es igual a  $\gamma=20$ . Sea  $\Gamma$  la longitud del genoma y  $\Gamma(i)$  su  $i$ -ésimo gen, la preservación de  $\gamma$  se obtiene al aplicar el siguiente criterio para cada individuo de la población:

- Obtener el orden del genoma ( $\omega$ ).
- Si  $\omega = \gamma$ , terminar.
- Si  $\omega > \gamma \Rightarrow s \leftarrow 1; t \leftarrow 0$                       Si  $\omega < \gamma \Rightarrow s \leftarrow 0; t \leftarrow 1$
- Generar un número aleatorio  $k$ , donde  $1 \leq k \leq \Gamma$ .
- Hacer  $i \leftarrow k$ , buscar en el genoma hasta  $\Gamma(i) = s$ . Si se llega al final de la cadena, hacer  $i \leftarrow i - 1$  y continuar con la búsqueda. Posteriormente, hacer  $\Gamma(i) \leftarrow t$  y pasar al paso (d).
- Repetir los pasos (d) y (e) hasta que  $\omega = \gamma$

Cada bit en el genoma corresponde a uno de los  $\mu_i$  en (i), si  $\mu_i = 0$  es por que el  $i$ -ésimo bit es 0. En otras palabras, cada bit corresponde a un coeficiente. Los coeficientes son indexados de izquierda a derecha y van desde 1 a  $\Gamma=1024$ .

Potencias asociadas a los coeficientes

00001	00002	00003	...	33331	33332	33333
Índices de los coeficientes						
1	2	3	...	1022	1023	1024

Tabla. IV.10 Relación entre potencias e índices

Sea  $\Gamma=(g_1, g_2, \dots, g_p)$  un conjunto de grados máximos de cardinalidad  $p$ , para el caso de estudio  $g_1=g_2= \dots=g_p=3$ , y  $f(v_1, \dots, v_p)$  el aproximante minimax de (i). También se sabe que

la longitud del genoma está dada por  $\prod_{i=1}^p (g_i + 1)$ . Entonces, dado el índice  $N$  de los coeficientes del genoma, es posible encontrar las potencias asociadas a los coeficientes de la siguiente relación:

$$\tau_i = \left[ \frac{(N-1) - \sum_{j=1}^{i-1} \tau_j \prod_{k=j+1}^p (g_k + 1)}{\prod_{j=i+1}^p (g_j + 1)} \right] \quad (IV.iii)$$

en donde si  $a > b \Rightarrow \sum_{i=a}^b x \equiv 0$  y  $\prod_{i=a}^b x \equiv 1$ . Los términos de las potencias ( $\tau_1, \tau_2, \dots, \tau_p$ ) corresponden a las posiciones 1, 2, ...,  $p$  de la potencia. Por ejemplo, los términos  $\tau_1=3, \tau_2=3, \tau_3=3, \tau_4=3$  y  $\tau_5=1$  en el genoma de la Tabla IV.10 corresponden al índice 1022.

De manera análoga, con los términos ( $\tau_1, \tau_2, \dots, \tau_p$ ) se puede encontrar el índice correspondiente de la expresión

$$N = 1 + \sum_{i=1}^p \tau_i \prod_{j=1}^p (g_j + 1) \quad (\text{IV.iv})$$

en donde, como antes,  $a > b \Rightarrow \prod_a^b x \equiv 1$ .

De acuerdo a la metodología expuesta para algoritmos genéticos se obtiene la tabla de términos de coeficientes para cada una de las sociedades de inversión. Asimismo, se obtiene el valor del grado del monomio para el término correspondiente de cada una de las 5 variables asociadas el coeficiente.

<b>Término</b>	<b>Grado monomio</b>	<b>Coefficientes</b>	<b>Término</b>	<b>Grado monomio</b>	<b>Coefficientes</b>
1	22311	-1.5070E+06	11	30302	7.5562E-09
2	22323	2.3453E-01	12	30303	-1.0304E-15
3	23203	-4.7014E-10	13	31131	1.2863E-14
4	23210	-3.5812E-06	14	31232	7.9135E-18
5	23220	3.1057E-09	15	32002	-5.8916E+07
6	23231	6.8138E-15	16	32012	-3.7589E+12
7	23232	2.5365E+07	17	32013	6.7896E+08
8	23233	-6.1245E+08	18	32200	9.7641E+06
9	23300	4.2295E+12	19	32210	-3.7564E+12
10	23310	-7.6671E-03	20	32220	-7.6340E+22

Tabla IV.11 Coeficientes y exponentes asociados al polinomio para HZPLAZO

<b>Término</b>	<b>Grado monomio</b>	<b>Coefficientes</b>	<b>Término</b>	<b>Grado monomio</b>	<b>Coefficientes</b>
1	01200	-4.3088E+04	11	01311	7.3769E-09
2	01201	4.6890E-02	12	03221	-7.0447E-17
3	01202	-8.5269E-11	13	12132	1.3306E-15
4	01203	-5.8186E-07	14	12133	5.9671E-18
5	01220	3.0131E-09	15	12133	-5.8578E+07
6	01221	4.0134E-15	16	12200	-1.2732E+12
7	01222	2.5298E+07	17	12221	2.5638E+08
8	01223	-4.0806E+08	18	32021	9.0815E+06
9	01230	1.2256E+12	19	32330	-1.1228E+12
10	01310	-7.5377E-03	20	32331	-1.1710E+22

Tabla IV.12 Coeficientes y exponentes asociados al polinomio para ACCIPAT

<b>Término</b>	<b>Grado monomio</b>	<b>Coefficientes</b>	<b>Término</b>	<b>Grado monomio</b>	<b>Coefficientes</b>
1.	11023	-3.2446E+05	11	12301	8.7506E-10
2	11101	2.3616E-02	12	13102	-6.6721E-17
3	11102	-1.9946E-12	13	13123	1.4944E-15

4	11112	-3.9470E-08	14	13211	8.6187E-19
5	11120	2.6992E-10	15	13330	-4.9393E+06
6	11121	6.2112E-17	16	21001	-1.0071E+12
7	12110	1.7773E+06	17	21003	3.2051E+07
8	12112	-5.5956E+06	18	21020	3.2100E+05
9	12120	8.3244E+10	19	21031	-7.0305E+11
10	12203	-7.7319E-04	20	21332	-1.0881E+22

Tabla IV.13 Coeficientes y exponentes asociados al polinomio para ACCIVAL

## IV.4 Análisis estadístico de resultados

El objetivo es realizar el pronóstico de una semana<sup>1</sup> para las sociedades seleccionadas. En el caso de redes neuronales se utiliza la funcionalidad de Recall en DataEngine que indica utilizar los datos en memoria una vez que se ha definido la red, entrenado y probado con la finalidad de generar el pronóstico a 5 días. Para los 5 días estimados se calculó el Error Cuadrático Medio definido como

$$ECM = \frac{1}{N} \sum_{i=1}^N (ValorActual_i - ValorEstimado_i)^2$$

donde N es el número de eventos

observados.

HZPLAZO				
Día	Pronóstico	Real	Error Cuadrático	Error Cuadrático Medio
01/Febrero/2005	1.530254	1.530106	2.1904E-08	
02/Febrero/2005	1.530616	1.530359	6.6049E-08	
03/Febrero/2005	1.531092	1.530218	7.6388E-07	
04/Febrero/2005	1.531634	1.531128	2.5604E-07	
07/Febrero/2005	1.532096	1.531221	7.6562E-07	
		Suma	1.8735E-06	3.747E-07

Tabla IV.14 Pronóstico a 5 días para HZPLAZO con RN

Para el caso de la sociedad HZPLAZO de los valores estimados se obtiene un coeficiente de correlación de 0.914 con respecto al valor real.

ACCIPAT				
Día	Pronóstico	Real	Error Cuadrático	Error Cuadrático Medio

<sup>1</sup> En términos generales, los especialistas recomiendan realizar estimaciones frecuentes considerando espacios de proyección mínimos debido a que existen factores que no son modelados como eventos sociales, políticos, internacionales, etc. El pronóstico a 5 días considera el posible comportamiento para una sociedad durante una semana hábil, así se evita que los eventos de fin de semana sean considerados como variable clave en el modelo de pronóstico.

01/Febrero/2005	193.15446	193.97263	0.66939347	
02/Febrero/2005	193.4097	194.06298	0.42677277	
03/Febrero/2005	193.76402	193.02901	0.54024345	
04/Febrero/2005	194.07029	195.12694	1.11649894	
07/Febrero/2005	194.57436	195.43853	0.74678337	
			3.49969201	0.6999384

Tabla IV.15 Pronóstico a 5 días para ACCIPAT con RN

Para el caso de la sociedad ACCIPAT de los valores estimados se obtiene un coeficiente de correlación de 0.674 con respecto al valor real.

## ACCIVAL

Día	Pronóstico	Real	Error Cuadrático	Error Cuadrático Medio
01/Febrero/2005	145.11534	146.46501	1.82160921	
02/Febrero/2005	145.51197	146.3827	0.75815342	
03/Febrero/2005	145.74902	147.30919	2.43415291	
04/Febrero/2005	146.12789	147.37705	1.5603853	
07/Febrero/2005	146.53607	147.41161	0.76657101	
			7.34087185	1.46817437

Tabla IV.16 Pronóstico a 5 días para ACCIVAL con RN

Para el caso de la sociedad ACCIVAL de los valores estimados se obtiene un coeficiente de correlación de 0.84 con respecto al valor real.

Para el caso de Algoritmos Genéticos, la estimación se presenta en las Tablas IV.17, IV.18 y IV.19.

## HZPLAZO

Día	Pronóstico	Real	Error Cuadrático	Error Cuadrático Medio
01/Febrero/2005	1.531021	1.530106	8.3722E-07	
02/Febrero/2005	1.532552	1.530359	4.8092E-06	
03/Febrero/2005	1.533122	1.530218	8.4332E-06	
04/Febrero/2005	1.533312	1.531128	4.7699E-06	
07/Febrero/2005	1.534220	1.531221	8.994E-06	
		Suma	2.7844E-05	5.5687E-06

Tabla IV.17 Pronóstico a 5 días para HZPLAZO con AG

## ACCIPAT

Día	Pronóstico	Real	Error Cuadrático	Error Cuadrático Medio
01/Febrero/2005	193.5854	193.97263	0.1499432	

02/Febrero/2005	193.32075	194.06298	0.5509093	
03/Febrero/2005	193.47354	193.02901	0.1976079	
04/Febrero/2005	193.14557	195.12694	3.9258259	
07/Febrero/2005	193.99021	195.43853	2.0976319	
			6.9219181	1.3843836

Tabla IV.18 Pronóstico a 5 días para ACCIPAT con AG

ACCIVAL				
Día	Pronóstico	Real	Error Cuadrático	Error Cuadrático Medio
01/Febrero/2005	144.62217	146.46501	3.3960771	
02/Febrero/2005	145.1196	146.3827	1.5953991	
03/Febrero/2005	145.76554	147.30919	2.3828624	
04/Febrero/2005	145.16109	147.37705	4.9104687	
07/Febrero/2005	146.69366	147.41161	0.5154631	
			12.80027	2.5600541

Tabla IV.19 Pronóstico a 5 días para ACCIVAL con AG

Los datos presentados en las Tablas IV.14 – IV.19 corresponden a la información obtenida de generar el proceso descrito en el presente capítulo con una muestra de 50 eventos para cada dato, esto se realizó con la finalidad de disminuir la generación de información que no sea representativa para la solución del problema. En este sentido, una vez que se obtenían las 50 muestras se calculó el promedio estadístico que corresponde al dato presentado en las tablas antes descritas.

#### Análisis de Varianza

Con la finalidad de validar estadísticamente el modelo presentado se presenta un análisis de varianza para el modelo expuesto. Una hipótesis estadística es una afirmación con respecto a una característica que se desconoce de una población de interés.

Para el caso de estudio se asume que se han estimado los resultados de 5 días para una serie de tiempo de una sociedad de inversión a través de dos métodos: Redes Neuronales y Algoritmos Genéticos. Se decidió utilizar la estadística F para comprobar que los resultados obtenidos por cada técnica corresponde a la misma secuencia de datos. En otras palabras, se utilizó la técnica del análisis de varianza para un experimento con sólo un factor completamente aleatorizado.

La hipótesis nula a probar indica validar si existe alguna razón para creer que los pronósticos obtenidos para 5 días en una Sociedad de Inversión son diferentes para las dos técnicas, es decir, se desea probar la hipótesis nula de que

$$H_0 : \mu_1 = \mu_2$$

Donde  $\mu_1$  representa la media de la población para RN y  $\mu_2$  la media para la técnica de AG. Para éste análisis de varianza, las fuentes de variación, grados de libertad, sumas de cuadrados, cuadrados medios y el cociente F están expresados en la tabla ANOVA<sup>2</sup> Tabla IV.20

Fuente de Variación	gl	Suma de Cuadrados	Cuadrados Medios	Estadística F
Tratamientos	k-1	$\sum \sum (\bar{Y}_{.j} - \bar{Y}_{..})^2$	$\sum \sum (\bar{Y}_{.j} - \bar{Y}_{..})^2 / (k-1)$	$F = \frac{\sum \sum (Y_{.j} - \bar{Y}_{..})^2 / (k-1)}{\sum \sum (Y_{ij} - \bar{Y}_{.j})^2 / (N-k)}$
Error	N-k	$\sum \sum (Y_{ij} - \bar{Y}_{.j})^2$	$\sum \sum (Y_{ij} - \bar{Y}_{.j})^2 / (N-k)$	
Total	N-1	$\sum \sum (Y_{ij} - \bar{Y}_{..})^2$		

Tabla IV.20 Tabla de análisis de varianza para la estadística F

Las sumas de los cuadrados pueden calcularse mediante fórmulas algebraicamente equivalentes

$$STC = \sum_{j=1}^k \sum_{i=1}^{n_j} (Y_{ij} - \bar{Y}_{..})^2 = \sum_{j=1}^k \sum_{i=1}^{n_j} Y_{ij}^2 - \frac{T^2}{N} \quad (\text{IV.v})$$

$$SCTR = \sum_{j=1}^k \sum_{i=1}^{n_j} (Y_{.j} - \bar{Y}_{..})^2 = \sum_{j=1}^k \frac{T_j^2}{n_j} - \frac{T^2}{N} \quad (\text{IV.vi})$$

$$SCE = STC - SCTR \quad (\text{IV.vii})$$

Para la Sociedad de Inversión ACCIPAT, de acuerdo a las Tablas IV.15 y IV.18 se indica que el tamaño de las muestras es  $n_1 = n_2 = 5$ , así  $N = 5 + 5 = 10$ .  $k = 2$  debido a que son dos técnicas o tratamientos los que se analizan.

<sup>2</sup> Tabla de análisis de varianza para la estadística F, Canavos 1994, Sec. 12.4

La suma de los tratamientos, es decir, la suma de los pronósticos para RN y AG, respectivamente, son  $T_1=968.97283$  y  $T_2=967.51547$ . La suma total es  $T=1936.4883$ . La suma de los cuadrados son los siguientes:

$$STC= 193.15446^2 + 193.4097^2 + \dots + 193.99021^2 - 1936.4883^2/10 = 1.8613$$

$$SCTR= 968.97283^2/5 + 967.51547^2/5 - 1936.4883^2/10 = 0.2124$$

$$SCE = 1.8613 - 0.2124 = 1.6489$$

La información se ha agrupado en la tabla ANOVA IV.21

Fuente de Variación	gl	Suma de Cuadrados	Cuadrados Medios	Estadística F
Tratamientos	1	0.2124	0.2124	F=1.0306
Error	8	1.6489	0.2061	
Total	9	1.8613		$F_{0.99,1,8} = 11.26$

Tabla IV.20 Tabla de análisis de varianza para la estadística F (ACCIPAT)

Debido a que  $F = 1.0306 < F_{0.99,1,8} = 11.26$  se acepta la hipótesis  $H_0$  de que ambas medias son equivalentes, es decir, existe una razón para considerar que ambas técnicas RN y AG generan los mismos resultados al pronosticar la serie de tiempo ACCIPAT con un nivel de confianza  $\alpha = 0.99$ .

Asimismo, en las Tablas IV.21 y IV.22 se expresa el análisis de varianza para las sociedades de inversión HZPLAZO y ACCIVAL.

Fuente de Variación	gl	Suma de Cuadrados	Cuadrados Medios	Estadística F
Tratamientos	1	0.0000073	0.0000073	F=8.11
Error	8	0.0000078	0.0000009	
Total	9	0.000028		$F_{0.99,1,8} = 11.26$

Tabla IV.21 Tabla de análisis de varianza para la estadística F (HZPLAZO)

Fuente de Variación	gl	Suma de Cuadrados	Cuadrados Medios	Estadística F
Tratamientos	1	0.2816	0.2816	F=0.6048
Error	8	3.7250	0.4656	
Total	9	4.0066		$F_{0.99,1,8} = 11.26$

Tabla IV.22 Tabla de análisis de varianza para la estadística F (ACCIVAL)

# CONCLUSIONES

En el presente trabajo se presentó la investigación realizada para aplicar técnicas de Inteligencia Artificial a un caso de estudio del Sector Financiero sobre pronósticos de Sociedades de Inversión. La primera motivación deriva de la inquietud por conocer nuevos métodos para la toma de decisiones orientado a resultados y utilizando los beneficios de la nueva tecnología.

El análisis de varianza realizado permiten confirmar que ambas técnicas, Redes Neuronales y Algoritmos Genéticos, coadyuvan en la toma de decisiones cuando se trata de invertir grandes cantidades de dinero, independientemente de la metodología utilizada o la herramienta de cómputo que facilita la obtención de los resultados, la aproximación en ambos casos fue superior a un nivel de confianza del 99%. Es decir, con cualquiera de las dos técnicas estudiadas es posible realizar un pronóstico para estimar el valor de una Sociedad de Inversión en el Mercado de Valores de México.

En términos matemáticos, se encuentra una equivalencia de interpretación funcional por un lado de las redes neuronales y, por otro, a mostrar una mayor aplicación de los algoritmos genéticos. El objetivo de considerar que ambas técnicas son aplicadas al mismo nivel para un problema en particular permite ayudar al tomador de decisiones a utilizar indistintamente una u otra técnica de acuerdo a las posibilidades del investigador.

El trabajo presentado es un enfoque y, no está limitado a futuras implementaciones, sin considerar que el caso de estudio fue aplicado a Sociedades específicas, de alguna manera se abre la pauta para avanzar en la investigación de nuevas metodologías y técnicas de Inteligencia Computacional y procesamiento de soluciones integrales en la toma de decisiones.

Finalmente, los conocimientos adquiridos en la maestría me permitieron aplicar y comprender los enfoques que ambas técnicas, Redes Neuronales y Algoritmos Genéticos, presentan y proponer la aplicación en casos de dichos modelos matemáticos y computacionales a situaciones tan prácticas, dinámicas y complejas como las que se dan en el mundo financiero.

# GLOSARIO

**Acción:** Título que representan una parte proporcional del capital social de una empresa, al poseer una acción se convierte en socio de ella y por ello participa de las ganancias y pérdidas que tenga la compañía.

**ADN:** Ácido DesoxirriboNucleico

**BANXICO:** Banco de México, Institución con personalidad de derecho público, con carácter autónomo, encargada de las operaciones de banca central del país y regulación del sistema financiero de México.

**BMV:** Bolsa Mexicana de Valores.

**Bonos (Bond):** Títulos de deuda emitidos por una empresa o por el Estado. En ellos se especifica el monto a reembolsar en un determinado plazo, los amortizaciones totales o parciales, los intereses periódicos y otras obligaciones del emisor.

**Bursatilidad: (Marketability):** Facilidad de comprar o vender la acción de una emisora en particular.

**Cartera de inversión (Investment portfolio):** Conjunto de inversiones financieras que cumplen con los objetivos de liquidez, rendimiento, plazo y riesgo fijados por el inversionista.

**Casas de bolsa (Broking house):** Intermediarios autorizados para realizar intermediación en el mercado bursátil. Se ocupan de las siguientes funciones: realizar operaciones de compra-venta de valores; brindar asesoría a las empresas en la colocación de valores y a los inversionistas en la constitución de sus carteras; recibir fondos por concepto de operaciones con valores y realizar transacciones con valores a través de los sistemas electrónicos de la BMV, por medio de sus operadores.

**Casa de corretaje:** Institución que se encarga de realizar operaciones de compra o venta de una acción o títulos que circula en un mercado financiero.

**CNBV:** Comisión Nacional Bancaria y de Valores.

**CONDUSEF:** Comisión Nacional para la Protección y Defensa de los Usuarios de Servicios Financieros.

**Diversificación (Diversification):** Operación de estructurar una cartera con diferentes clases de valores, con objeto de disminuir el riesgo.

**Emisión:** Generación de deuda a través de títulos.

**Emisora:** Son las personas morales ( Empresas – Bancos y Gobierno) que colocan sus valores (deuda o capital) y que forman el mercado primario al venderlos a los inversionistas .

**Indicador financiero:** Son valores y tasas publicados por los organismos oficiales con la finalidad de regular el comportamiento de los mercados financieros.

**Indice de Precios y Cotizaciones (IPC/Quotations and price index):** Es el principal indicador de la BMV, expresa el rendimiento del mercado accionario en función de las variaciones de precios de una muestra balanceada, ponderada y representativa del conjunto de acciones cotizadas en la BMV. Este indicador, aplicado en su actual estructura desde 1978, expresa en forma fidedigna la situación del mercado bursátil y su dinamismo operativo.

**INCP:** Indice Nacional de Precios al Consumidor.

**Intereses (Interest):** Porcentaje fijo que sobre el monto de un capital y su uso paga periódicamente al dueño del mismo la persona física o moral que toma en préstamo dicho capital.

**Intermediación bursátil:** Es la actividad que realizan las Casas de Bolsa al contactar a compradores y vendedores de valores.

**Instrumento:** A los valores comúnmente se les llama instrumentos y dependiendo del tipo son instrumentos del mercado de capitales (Acciones) o instrumentos del mercado de dinero (deuda); además de los instrumentos de sociedades de inversión, el de metales y el sistema internacional de cotizaciones y del mercado de Productos Derivados y el mercado de cambios de monedas.

**Inversionistas:** Personas físicas o morales, nacionales o extranjeras, que, a través de una Casa de bolsa colocan sus recursos a cambio de valores, para obtener rendimientos. En los mercados Bursátiles del mundo destaca la participación del grupo de los llamados "inversionistas institucionales", representado por sociedades de inversión, fondos de pensiones y otras entidades con alta capacidad de inversión y amplio conocimiento del mercado y de sus implicaciones.

**Liquidación:** Intercambio de dinero que realizan los participantes en el mercado como consecuencia de la compra o venta de valores. Se entenderá por liquidación el complemento de una transacción, es decir, el mecanismo para que los valores sean traspasados a la (s) cuenta(s) de valores del nuevo propietario de los mismos y el efectivo acreditado a la(s) cuenta(s) de la contraparte correspondiente.

**Liquidez:** Calidad del activo de un banco que puede fácilmente transformarse en dinero efectivo.

**Mercado de capitales:** Aquel en el que comercian valores de largo plazo (más de un año). Principalmente acciones y obligaciones, su objetivo es el financiamiento de activos fijos.

**Mercado de dinero:** Aquel en el que se comercian valores a corto plazo (menos de un año), sus principales instrumentos son los CETES, PAGAFES, Aceptaciones bancarias, Papel comercial y metales preciosos. Su objetivo es el financiamiento del capital de trabajo para emisores privados, así como el financiamiento del gasto corriente y regulación del circulante monetario por parte del gobierno federal.

**Mercado de valores:** Mercado organizado para la compra-venta de valores (inversiones financieras). Esta compuesto por varios mercados: un mercado de capitales (para inversión a largo plazo), un mercado de dinero (para inversión a corto plazo), un mercado primario (para la emisión de valores) y un Mercado secundario (para la compra-venta de valores ya emitidos) financiamiento corporativo de las empresas.

**Mercado Primario:** Cuando el título es negociado por primera vez directamente del emisor al inversionista, representado esto un movimiento de efectivo para el primero, es el mercado donde se colocan las nuevas emisiones de valores con una oferta pública inicial, una oferta primaria.

**Mercado Secundario:** Mercado donde se operan instrumentos de inversión después de su colocación inicial en el mercado primario, es cuando el título y los recursos se negocian e intercambian entre el gran público inversionista.

**Minusvalía:** Resultado en la valuación de títulos que representa una diferencia negativa entre el valor del título en posición y el valor del mercado, el valor del mercado es mas alto.

**Operación en Directo:** Compra – venta de títulos en tenencia propia, los títulos salen de la posición.

**Posición Financiera:** Son las operaciones activas o pasivas con títulos o efectivo que una institución mantiene vigentes al cierre de un día de operación.

**Precio :** Valor en que se estima algo.

**Precio de mercado:** Valor de un título determinado a partir de las cotizaciones de éste en el mercado.

**Rendimientos:** Ganancia de capital. Beneficio que produce una inversión. El rendimiento anualizado y expresado porcentualmente respecto a la inversión se denomina tasa de rendimiento.

**Sociedades de Inversión:** Sociedades anónimas constituidas con el objeto de adquirir valores y documentos seleccionados de acuerdo al criterio de diversificación de riesgos, con recursos provenientes de la colocación de las acciones representativas de su capital social entre el público inversionista.

**Tasa de rendimiento:** Proporción o porcentaje sobre algún principal activo o pasivo.

**Timing:** Elección del momento oportuno de comprar o vender una categoría de inversión o una inversión específica.

**Título:** Es un papel con valor monetario y puede representar un activo o un pasivo.

**Valor Nominal:** Valor monetario de un título.

**Vector:** Conjunto de precios de mercado.

**Vía de liquidación:** Medio por el que se hace la liquidación o cobro de operaciones.

**Volatilidad:** Grupo de fluctuación que manifiesta el precio del subyacente a través del tiempo.

**Valor presente:** Valor calculado como la diferencia entre la suma de flujos esperados de una inversión descontados por una tasa de rendimiento

# REFERENCIAS

- ✓ Blanco H., Gonzalo et al., El sistema financiero en México, Captus Prees, 1996.
- ✓ Box, G.E., G. M. Jenkins, G. C. Reinsel, Time Series Analysis: Forecasting and Control, third edition, Prentice Hall, 1994.
- ✓ Brealey, Richard, et al., Introducción a las finanzas corporativas, Mc Graw-Hill 5a. Ed. España 1998.
- ✓ Canavos, George C. Probabilidad y Estadística. Aplicaciones y Métodos, Mc Graw-Hill, México, 1994.
- ✓ Cheney, E. W., Introduction to Approximation Theory, McGraw Hill Book Company, p. 39, 1966.
- ✓ Fogel, David B., Evolutionary Computation. Toward a New Philosophy of Machine Intelligence, The Institute of Electrical and Electronic Engineers, New York, 1995.
- ✓ Galaviz Casas, José y Kuri Morales, Ángel F., Algoritmos Genéticos, IPN, UNAM, FCE, México 2002.
- ✓ Goldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, pp. 40-41, 1989.
- ✓ Goldberg, David E. & Deb, Kalyanmoy, A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, in Gregory J. E. Rawlins (Editor), Foundations of Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California, pp. 69-93, 1991.
- ✓ Haykin, Simon, Neural Networks a comprehensive foundation, Prentice Hall 2a. Ed. E.U. 1999
- ✓ Hebb, D. O. The Organization of Behavior, New York, John Wiley, 1949.
- ✓ Hertz, J., Krogh, A. y Palmer, R. G., Introduction to the Theory of Neural Computation, Addison-Wesley, 1991.
- ✓ Holland, John H., Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, Massachusetts, Second Edition, 1992.
- ✓ Holland, John H., Hidden Order, Addison-Wisley, 1995.
- ✓ Koza, John R., Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, Massachusetts, 1992.

- ✓ Menchaca Trejo, Mauricio, El mercado de dinero en México, Ed. Trillas, México 1998
- ✓ Mendenhall, W., Wackerly, D. y Scheaffer, R. Estadística Matemática con Aplicaciones, 2a. Edición, Grupo Editorial Iberoamerica, México, 1994.
- ✓ Michalewicz, Zbigniew, Genetic Algorithms + Data Structures = Evolution Programs, Springer 3a. Ed., E.U. 1999.
- ✓ Michalewicz, Zbigniew y Fogel, David B., How to Solve It: Modern Heuristics, Springer, Berlin, 2000.
- ✓ Minsky, M. Y Papert, S. Perceptrons, MIT Press, Cambridge, 1969.
- ✓ Morton, Alexander, H., An Introduction to Neural Computing, Chapman & Hall – 1990.
- ✓ Rich, Elaine y Knight, Kevin, Artificial Intelligence, Mc Graw-Hill 2a. ed. International Edition 1991
- ✓ Rosenblatt, F. Principles of Neurodynamics, Spartan, New York, 1962.
- ✓ Ross, Stephen, et al., Finanzas Corporativas, Mc Graw-Hill 5ª. Ed., México, D.F. 2001
- ✓ Trippi, Robert y Turban, Efraim, Neural Networks in Finance and Investing, Probus Publishing Company, USA, 1993.
- ✓ Whitley, Darrel, A Genetic Algorithm Tutorial, Statistics and Computing Colorado State University, 1993.

#### Sitios

Banco de México	<a href="http://www.banxico.org.mx">www.banxico.org.mx</a>
Comisión Nacional Bancaria y de Valores	<a href="http://www.cnbv.gob.mx">www.cnbv.gob.mx</a>
Bolsa Mexicana de Valores	<a href="http://www.bmv.com.mx">www.bmv.com.mx</a>
Portal Banamex	<a href="http://www.banamex.com">www.banamex.com</a>
Accitrade, Casa de Bolsa Accival	<a href="http://www.accitrade.com">www.accitrade.com</a>

#### Artículos

- ✓ Bäck, Thomas, Self-Adaptation in Genetic Algorithms, En F. Varela & P. Bourgine (Editores), Proceedings of the First European Conference on Artificial Life, Paris, France, MIT Press, pp. 263-271, 1992.
- ✓ Barron, A. R., Universal approximation bounds for superpositions of a sigmoidal function, IEEE Transactionson Information Theory 39(3):930-945, 1993.

- ✓ Bastos, Ricardo y Ludermir, Teresa, Design of Neural Networks for Time Series Prediction using case-inicialized Genetic Algorithm, Brasil.
- ✓ Beasley, David; Bull, D. R. y Martin, R. R., An Overview of Genetic Algorithms : Part 1, Fundamentals, University Computing, 15(2), pp. 58-69, 1993.
- ✓ Billings, Stephen A., Liu, G. P., Kadirkamanathan, V., Variable neural networks for adaptive control of nonlinear systems, IEEE Transactions on Systems, Man and Cybernetics, Part C 29(1): 34-43, 1999.
- ✓ Coello Coello, Carlos A. Introducción a los Algoritmos Genéticos, Soluciones Avanzadas. Tecnologías de Información y Estrategias de Negocios, Año 3, Número 17, pp. 5-11, Enero de 1995.
- ✓ Coello Coello, Carlos A., La importancia de la representación en los algoritmos Genéticos (Parte I), Soluciones Avanzadas. Tecnologías de Información y Estrategias de Negocios, Año 7, Número 69, pp. 50-56, 15 de mayo de 1999.
- ✓ Coello Coello, Carlos A., La importancia de la representación en los algoritmos Genéticos (Parte II), Soluciones Avanzadas. Tecnologías de Información y Estrategias de Negocios, Año 7, Número 70, pp. 44-48, 15 de junio de 1999.
- ✓ De Jong, K. Y Spears, W., An Análisis of the Interacting Roles of Population size and crossover in Genetic Algorithms, 1995.
- ✓ Dorffner, Georg, Neural Networks for Time Series Processing, IEEE, Austria.
- ✓ Drossu, R. y Obradovic, Z., Efficient Design of Neural Networks for Time Series Prediction, <http://citeseer.nj.nec.com/183217.html>, 1995.
- ✓ Fogel, David B. An Introduction to Simulated Evolutionary Programming, IEEE Transactions on Neural Networks, Vol. 5, No. 1, pp. 3-14, January 1994.
- ✓ Fonseca, C. M., Fleming, P. J., Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part I: A Unified Formulation, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 1998.
- ✓ Fonseca, C. M., Mendes, E. M., Fleming, P. J., Billings, S. A., Non-linear Model Term Selection with Genetic Algorithms, in IEE/IEEE Workshop on Natural Algorithms in Signal Processing, vol. 2, pages 27/1–27/8, 1993.
- ✓ Galaviz, J. y Kuri, M., A. A self-adaptive Genetic Algorithm for Function Optimization, Proceedings of ISAI/IFIPS 1996, pág.156
- ✓ Goldberg, D. E.; Korb, B. y Deb, K. Messy genetic algorithms: Motivation, analysis, and first results, Complex Systems, Vol. 3, pp. 493-530, 1989.

- ✓ Goldberg, David E., Krjazew, D. Solving permutation problems with the ordering messy genetic algorithm, Source Natural Computing Series archive Advances in evolutionary computing: theory and applications archive, 321 – 350, Springer-Verlag New York, Inc. New York, NY, USA, 2003.
- ✓ Hinterding, Robert; Michalewicz, Z. y Eiben, A. E., Adaptation in Evolutionary Computation: A Survey, Proceedings of the 4th IEEE Conference on Evolutionary Computation, Indianapolis, USA, pp. 65-69. April 1997.
- ✓ Hitoshi, Iba y Nikolaev, Nikolay Y., Polynomial harmonic GMDH learning networks for time series modeling, Neural Networks Volume 16, Issue 10, December 2003, Pages 1527-1540, 2003.
- ✓ Hornik, K., Stinchcombe, M. and White, H. Multilayer feedforward networks are universal approximators, Neural Networks 2: 359-366, 1989.
- ✓ Jain, A. K., y Jianchang, M., Artificial Neural Networks: A Tutorial, Computer, Vol. 29, No. 3, pp. 31-44, , marzo, 1996.
- ✓ Kargupta, H. SEARCH, polynomial complexity, and the fast messy genetic algorithm (IlligAL Report No. 95008). Urbana, IL: University of Illinois at Urbana-Champaign, 1995.
- ✓ Kargupta, H., Deb, K., y Goldberg, D. E., Ordering genetic algorithms and deception, Parallel Problem Solving from Nature - PPSN II, 47-56, 1992.
- ✓ Koehn, Philipp, Combining Genetic Algorithms and Neural Networks. The Encoding Problem, Thesis of Master of Science Dec. 1994
- ✓ Kuri M., Ángel, Multivariate Approximation via a Genetic Algorithm, IPN 1996
- ✓ Kuri, A., Heuristic Techniques for the generalization and acceleration of an algorithm for multivariate minimax approximation, Agrociencia 1993.
- ✓ Kuri, A., An alternative model of Genetic Algorithms as Learning Machines, Fourth World Congress on Expert Systems, March, 1998.
- ✓ Leung, Frank et al., Tuning of the structure and parameters of a Neural Network using an improved Genetic Algorithm, Hong Kong July 2002.
- ✓ McCulloch, W. S. y Pitts, W., A logical calculus of the ideas immanent in nervous activity, Bulletin of mathematical biophysics, 5, 115-133, 1943.
- ✓ McCulloch, W. S., Pitts, W., Landahl, H. D., A Statistical Consequence of the Logical Calculus of Nervous Nets, Bulletin of Mathematical Biophysics. 1943.
- ✓ Minsky, M., Neural nets and the brain model problem, doctoral dissertation, Princeton University, Princeton, N.J, 1954.

- ✓ Minsky, M. y Selfridge, O. G., Learning in random nets, Proc. Fourth London Symposium. on Information Theory, Butterworth, Ltd., London, 1961.
- ✓ Nissenen, A. S. y Koivisto, H., Identification of multivariate volterra series using genetic algorithms. In J.T. Alander, editor, Second Nordic Workshop on Genetic Algorithms and Their Applications, pages 151-161, University of Vaasa, Finland, 1996.
- ✓ Rudolph, Günter, Convergence Analysis of Canonical Genetic Algorithms, IEEE Transactions on Neural Networks, 5:96-101, January 1994.
- ✓ Sage, Andrew P. y Post, Stephen, An Overview of Automated Reasoning. IEEE Transactions on Systems, 1990, Man and Cybernetics, 20 (1): 202 -224.
- ✓ Syswerda, G., Uniform crossover in genetic algorithms, Third International Conference on Genetic Algorithms, pages 2-8. Morgan Kaufman, 1989.
- ✓ Timothy, Johnson, Husbands,P., System Identification Using Genetic Algorithms, PPSN 1990: 85-89
- ✓ Vose, M., Modeling Simple Genetic Algorithms, Foundations of Genetic Algorithms -2-, D. Whitley, ed., Morgan Kaufmann. pp: 63-73, 1993.
- ✓ Whitley, Darrell, A Genetic Algorithm Tutorial, Technical Report CS-93-103, Computer Science Department, Colorado State University, March 10, 1993. Publicado como: Whitley, Darrell, A Genetic Algorithm Tutorial, Statistics and Computing, Vol. 4, pp. 65-85, 1994.
- ✓ Whitley, L. D., Fundamental principles of deception in genetic algorithms, G. Rawlins, editor, Foundations of Genetic Algorithms 1, pages 221-- 241, 1990.
- ✓ Whitley, L. D., Heckendorn, R. B., Stevens, S., Hyperplane ranking, nonlinearity and the simple genetic algorithm, Lecture Notes in Computer Science, Inf. Sci. 156(3-4): 123-145, 2003.
- ✓ Whitley, D., y Yoo, N.-W., Modeling simple genetic algorithms for permutation problems, Foundations of Genetic Algorithms, 3, 163-184, 1994.
- ✓ Wise, B. M., B. R. Holt, N. B. Gallagher and S. Lee, A Comparison of Neural Networks, Non-linear Biased Regression and a Genetic Algorithm for Dynamic Model Identification, Chemo. and Intell. Lab. Sys. 30: 81-89, 1995.
- ✓ Yao, Xin, Evolving Artificial Neural Networks, proceedings of the IEEE UK Sep 1999.