

Entrenamiento de una red neuronal multicapa para la tasa de cambio euro - dólar (EUR/USD)

Training a multilayer neural network for the Euro-dollar (EUR/USD) exchange rate

Jaime Alberto Villamil Torres¹ y Jesús Alberto Delgado Rivera²

RESUMEN

Tanto para los inversionistas como para las autoridades económicas es necesario que se desarrolle una herramienta matemática que logre dar cuenta de la dirección de una variable como el tipo de cambio (el precio relativo entre dos monedas). Muchos de los mecanismos usados actualmente están basados en el uso de técnicas estadísticas, en particular series de tiempo lineales. Las redes neuronales artificiales (RNA) son modelos matemáticos que pretenden emular el funcionamiento del cerebro humano, su aplicación en economía e ingeniería surge a finales de los años ochenta con buenos resultados. Las RNA se presentan como una alternativa para simular el comportamiento de variables financieras que, por lo general, tienden a parecerse a un paseo aleatorio. En este trabajo se muestran los resultados del entrenamiento de una red neuronal para negociación de la tasa de cambio EUR/USD y las bondades del algoritmo de entrenamiento chemotaxis, que permite entrenar redes que maximicen una función objetivo que relacione aciertos en la predicción con las ganancias de un trader.

Palabras clave: chemotaxis, estrategias de negociación, Forex, redes neuronales artificiales, redes multicapa.
JEL: F310, C450

ABSTRACT

A mathematical tool or model for predicting how an economic variable like the exchange rate (relative price between two currencies) will respond is a very important need for investors and policy-makers. Most current techniques are based on statistics, particularly linear time series theory. Artificial neural networks (ANNs) are mathematical models which try to emulate biological neural networks' parallelism and nonlinearity; these models have been successfully applied in Economics and Engineering since the 1980s. ANNs appear to be an alternative for modelling the behaviour of financial variables which resemble (as first approximation) a random walk. This paper reports the results of using ANNs for Euro/USD exchange rate trading and the usefulness of the algorithm for chemotaxis leading to training networks thereby maximising an objective function re predicting a trader's profits. JEL: F310, C450.

Keywords: artificial neural network, chemotaxis, FOREX, trading strategy.

Recibido: agosto 25 de 2007

Aceptado: noviembre 1 de 2007

Introducción

La teoría económica subraya la dificultad para predecir los precios de los activos financieros y pone en duda la capacidad de que los inversionistas deriven beneficios indefinidamente a través de la especulación. Esta afirmación se conoce como *hipótesis de los mercados eficientes* y fue expuesta por Eugene Fama a finales de los años sesenta. El presente planteamiento señala que el conocimiento del comportamiento del precio en el pasado no aporta información para dilucidar su nivel futuro. No obstante existen

enfoques diferentes que intentan explicar el rumbo de los precios de los activos financieros, uno, con base en los indicadores de la salud económica de un país (el análisis fundamental) y, otro, por medio de indicadores y figuras construidas a partir de la cotización histórica (el análisis técnico). En este artículo se propone el uso de redes neuronales multicapa, alimentadas de cualquiera de estos indicadores, que proporcionen señales de negociación en el mercado de la tasa de cambio euro-dólar (EUR/USD).

¹ Economista y M. Sc. en Matemática Aplicada, Universidad Nacional de Colombia. Trabajador, Bancolombia. jaime_villamil@yahoo.com

² Ingeniero electrónico. M.Sc. en Ingeniería Eléctrica, Ph.D. en Cibernética. Profesor Titular, Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Colombia, Bogotá. jadelgador@unal.edu.co, adelgado@ieee.org

Marco teórico

Las redes neuronales son sistemas de procesamiento de información que están inspirados en la estructura y el funcionamiento de los sistemas nerviosos biológicos. Su importancia reside en la capacidad que tienen para aproximar funciones desconocidas con base en el conocimiento de un patrón de datos de entrada y salida. La primera representación de una neurona biológica; fue ofrecida en 1911 por Santiago Ramón y Cajal. En la Figura 1 se puede observar un dibujo de una neurona biológica, esta es en general una célula nerviosa provista de un *núcleo* que genera un impulso eléctrico dependiendo de un grado de excitación (por ejemplo, la dificultad de la tarea), y unas prolongaciones que se conocen como *axón* y, finalmente, las *dendritas*.

Es por medio de las ramificaciones más pequeñas derivadas del axón, las dendritas, que unas neuronas se comunican con otras. Esta conexión no es física, las dendritas de una neurona están cerca de las dendritas de otra, pero ambas terminaciones se encuentran inmersas en una sustancia neurotransmisora que, dependiendo de la intensidad del impulso eléctrico que viene de una neurona, le es comunicado a las neuronas contiguas, a este espacio entre dendritas se le conoce como *sinapsis*. La intensidad del impulso eléctrico que permite la activación de las neuronas vecinas está determinada por un umbral, si el impulso es mayor a este, entonces se transmite a las demás, de lo contrario no. En 1943 el esquema de funcionamiento neurofisiológico de la neurona fue emulado matemáticamente por McCulloch y Pitts mediante una representación como la de la Figura 2, a la que se le conoce como *neurona artificial* o *perceptron*.

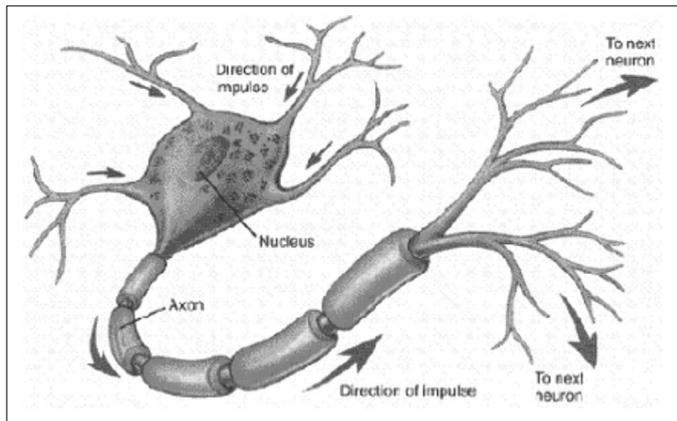


Figura 1. Neurona Biológica. Fuente: <http://psych.wisc.edu/faculty/pages/croberts/topic3.html>

En la Figura 2 existe un vector x_r con r variables de entrada, un vector w_{ix} que consta de ponderadores (que son ajustables) para cada variable de entrada y, además, se involucra un parámetro escalar b llamado *umbral*. La activación (el impulso eléctrico representado por la letra a de la neurona

se expresa por medio de la *función de activación*³ f cuyo argumento es el producto interno de los vectores w y x más el parámetro b (también ajustable si se desea).

En breve, una *red neuronal* se define como una interconexión en paralelo de neuronas, donde las salidas de una neurona sirven de entrada a otras. La red neuronal es capaz de *aprender* cuál es la asociación entre unas variables explicativas y unas explicadas. El modo en que estas conexiones se presentan se llama *arquitectura de la red*. Y al proceso gradual de ajuste de los pesos y umbrales a valores que se adecúan a un correcto aprendizaje de la tarea presentada se le llama *entrenamiento*.

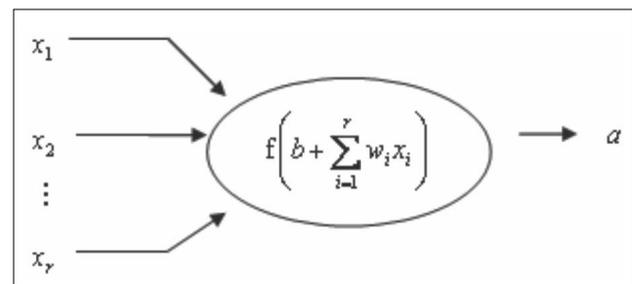


Figura 2. Neurona artificial

La investigación en redes neuronales, después del hallazgo de Minsky y Papert (1969), entró en un estado de abandono hasta finales de los ochenta. El resultado de estos autores señaló que, sin importar la función no lineal usada como función de activación, un perceptron con solamente la capa de salida, tiene éxito en tareas de clasificación de patrones solo si estos son linealmente separables. El ejemplo clásico para ilustrar esta idea ha sido el aprendizaje de la función lógica o *exclusiva* (XOR). Esta operación lógica entre valores de verdad de dos proposiciones se representa mediante la siguiente tabla:

Tabla 1. Operación XOR

X1	X2	Y
1	0	1
0	1	1
0	0	0
1	1	0

El valor 0 indica falsedad y el valor 1 lo contrario. De esta manera la operación lógica o *exclusiva* entre dos proposiciones es *verdadera*, sólo si alguna de las dos proposiciones es verdadera, pero no ambas. En cálculo de proposiciones la operación XOR se puede expresar en términos de operaciones más conocidas como AND, OR y la negación (\neg)⁴. XOR es equivalente a la siguiente proposición $(\neg x_1 \text{ AND } x_2) \text{ OR } (x_1 \text{ AND } \neg x_2)$

³ La función de activación se usa para acotar el rango de su respuesta o salida (a) entre dos valores deseables. Las funciones de activación más usadas son la logística, que garantiza valores en el intervalo $(0, 1)$, y la tangente hiperbólica, que garantiza valores en el intervalo $(-1, 1)$.

⁴ En teoría de conjuntos estas operaciones son semejantes a la intersección, la conjunción y el complemento, respectivamente.

La dificultad para resolver el problema reside en el hecho de que los patrones no son linealmente separables. Es decir, que no existe una línea recta que permita separar el espacio de patrones⁵. En la Figura 3 se dibujan los valores de la Tabla 1, en las abscisas van los valores de x_1 y en las coordenadas los de x_2 . Con blanco se representa un valor verdadero (1) de la proposición $x_1 \text{ XOR } x_2$ y con negro un valor falso (0). Se observa en dicha Ilustración que no es posible trazar una línea recta que divida las respuestas falsas de las verdaderas.

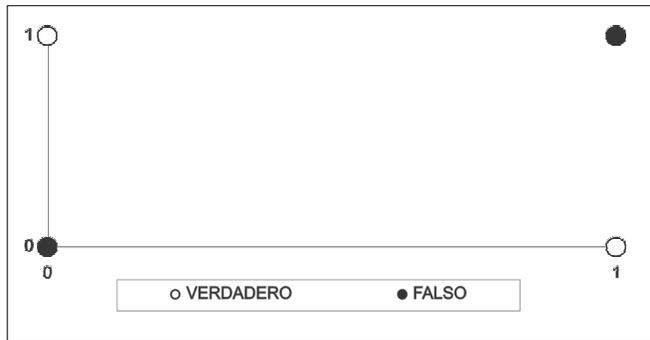


Figura 3. No separabilidad lineal

La forma en la que se pueden hacer separables las respuestas de la Figura 3 es mediante una o varias capas de funciones no lineales entre las entradas y las salidas de un perceptron, a esta configuración se le llama *perceptron multicapa*. Este, según la demostración Funahashi (1989)⁶, es capaz de aproximar cualquier asociación *no lineal* entre variables.

Un *perceptron multicapa* consiste en un arreglo de neuronas por capas: la capa de neuronas para las variables de entrada se llama *capa de entrada*, una o varias capas intermedias se llaman *capas ocultas*, y una capa de neuronas para las variables de salida se llama *capa de salida*, las interconexiones que son permitidas entre neuronas sólo van de una neurona de una capa a las neuronas de la capa siguiente. Esta es la principal característica de las redes multicapa o estáticas, no existe retroalimentación, esto es, no se permite que la salida de una capa vaya como entrada a una capa anterior. Cuando se permite se habla de redes *recurrentes* o *dinámicas*. La notación abreviada para la arquitectura de una red neuronal multicapa es (n_i, n_j, n_k) . Donde n_i y n_k son el número de variables de entrada y de salida respectivamente, en tanto que n_j es el número de neuronas en la capa oculta.

En la Figura 4 se ilustra una red neuronal multicapa con arquitectura (4, 3, 3, 2). Las capas se representan con la letra mayúscula S y el subíndice da cuenta del orden en que se distribuyen las capas; la ilustración muestra una red con cuatro variables de entrada en la capa S_1 , dos de salida en la capa S_4 , y dos capas ocultas (S_2 y S_3) cada una con tres neuronas como las neuronas, artificial de la Figura 2.

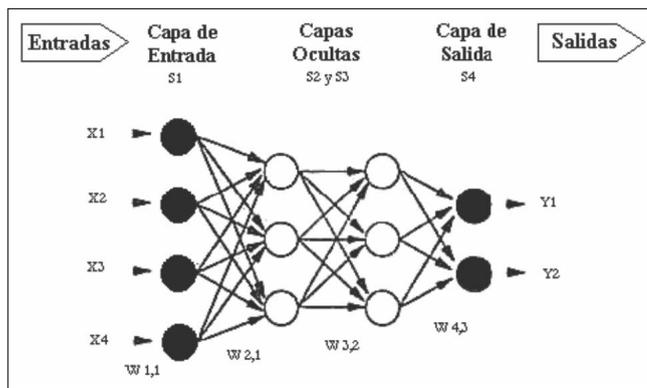


Figura 4. Red neuronal multicapa

Toda *red multicapa* se encuentra definida en términos de su arquitectura, sus funciones de activación, los umbrales y los pesos, siendo estas dos últimas las variables de ajuste en el momento de utilizar un *algoritmo de entrenamiento* para que la red aprenda. En el entrenamiento, además de ajustar los pesos y los umbrales, es preciso optimizar el número de neuronas que la conforman porque de esto depende la velocidad que adquiere la red para aprender.

El *entrenamiento* de una red multicapa es de dos tipos: *incremental* si el aprendizaje se da al presentarle a la red cada *entrada* (variables explicativas) con su respectiva *salida* (variables explicadas), o *batch*, cuando el aprendizaje de la red ocurre sólo después de haberle presentado todo el conjunto de datos de entrada y salida. El objetivo del *entrenamiento* es minimizar una función de costo, la que se usa con frecuencia es el error cuadrático medio.

El algoritmo de entrenamiento más usado es el de *propagación inversa* (en inglés *back propagation*) que parte de unos valores iniciales de todos los pesos y umbrales de la red generados aleatoriamente y los actualiza con base en un término de corrección. El proceso iterativo se realiza desde la última capa de neuronas hasta la primera en la medida en que cada patrón es presentado a la red. En virtud de que el aprendizaje se hace con base en la repetición, no solo es necesario presentarle a la red una lista de datos (patrones) disponibles en la muestra de entrenamiento, sino que es preciso presentárselos varias veces hasta que la red aprenda. A cada listado de patrones se le llama *época*. El número de iteraciones del algoritmo de propagación inversa es igual al número de épocas multiplicado por el número de patrones disponibles en la muestra de entrenamiento. En la n -ésima iteración la corrección $\Delta w(n)$ es igual a una constante de proporcionalidad η (llamada *tasa de aprendizaje*), que multiplica el cambio de dirección de la suma de errores cuadrados $E(n)$ con respectos a los pesos $w(n)$, es decir:

$$\Delta w(n) = -\eta \frac{\partial E(n)}{\partial w(n)} \tag{1}$$

⁵ Cuando se tienen variables de entrada la no separabilidad lineal consiste en que no es posible que un hiperplano separe dos hipervolumenes.

⁶ Citado por Delgado, (1998).

La regla (1) es necesario examinarla por separado en la capa de salida y en la capa oculta, debido a que en esta última no existen unos valores deseables que puedan ser observados y con los cuales medir los errores de salida de las neuronas en esa capa. La deducción del mecanismo de actualización de los pesos como lo plantea el algoritmo de propagación inversa se encuentra expuesto en Haykin (1994). En una red neuronal con sólo una capa oculta, las ecuaciones (2) y (3) buscan valores óptimos de los pesos que minimicen el valor del error cuadrado medio. La optimización de los pesos en la capa de salida se consigue mediante la siguiente regla:

$$w_{kj}(n+1) = w_{kj}(n) + \eta [e_k(n) f'_k(s_k(n)) y_j(n)] \quad (2)$$

Por su lado, los parámetros de la capa oculta son actualizados por medio de la regla:

$$w_{ji}(n+1) = w_{ji}(n) + \eta \left[y_i(n) f'_j(s_j(n)) \sum_{k \in K} e_k(n) f'_k(s_k(n)) w_{kj}(n) \right] \quad (3)$$

Donde:

$w_{ji}(n)$ es el j -ésimo peso en la capa oculta para la i -ésima variable de entrada.

$w_{kj}(n)$ es el k -ésimo peso en la capa de salida para la j -ésima variable de entrada.

$f_j(\cdot)$ es la función de activación de la neurona j que genera la salida y_j

$y_j(n)$ es la salida de la neurona j en la n -ésima iteración. $y_j = f_j(s_j(n))$

$s_j(n)$ es el nivel de actividad interna de la neurona j igual a $\sum_{i \in I} w_{ji}(n) y_i(n)$

$e_k(n)$ es el error de aproximación entre la salida simulada y_i y la deseada y_i^*

N es el número de patrones de entrenamiento

$E(n)$ es la suma de errores al cuadrado $E(n) = \frac{1}{2} \sum_{k \in K} e_k^2(n)$

ECM es el error cuadrado medio es

$$ECM = \frac{1}{N} \sum_{n=1}^N E(n)$$

El algoritmo de propagación inversa tiene deficiencias: una tasa de aprendizaje muy pequeña hace que el entre-

namiento sea demasiado lento; por el contrario, una tasa muy alta hace que el algoritmo sea inestable. La solución depende notablemente de los pesos con los que se inicializa el entrenamiento, puesto que dependiendo de ello se puede llegar a un mínimo local y no a uno global. En consecuencia, el entrenamiento mediante propagación inversa es un ejercicio de ensayo y error que resulta algo engorroso, por esta razón se han propuesto algunas modificaciones del algoritmo.

Una propuesta de entrenamiento que se mantiene en la idea de emular los mecanismos biológicos es la de Bremermann y Anderson (1991). Ellos propusieron entrenar redes neuronales artificiales imitando el mecanismo de *chemotaxis*, que es el movimiento aleatorio de organismos unicelulares, por ejemplo de las bacterias, cuando responden al estímulo de las sustancias químicas que les sirven de nutrientes⁷. Las ventajas que tiene la *chemotaxis* en comparación con la propagación inversa son, primero, que este algoritmo no usa el gradiente del error, ofreciendo así libertad para usar funciones objetivo mucho más complejas (para las cuales el cálculo del gradiente sería difícil o imposible de calcular) y, segundo, que la tasa de aprendizaje es variable y no fija, como en el algoritmo de retropropagación.

El algoritmo de *chemotaxis* obedece a la siguiente secuencia:

- i-). Se inicia con una matriz arbitraria de pesos⁸ W_0
- ii-). Los pesos iniciales de i-) son perturbados por un término ΔW que sigue una distribución normal estándar. $W_t = W_0 + h * \Delta W$ donde h es una constante similar a la tasa de aprendizaje en el algoritmo de propagación inversa que toma valores entre cero y uno.
- iii-). La función objetivo de interés a ser optimizada se evalúa con los nuevos pesos $f(W_t)$. Si los nuevos pesos disminuyen (en el caso que se desee minimizar) o aumentan (en el caso que se desee maximizar) en la función objetivo con relación al resultado que se obtiene con los pesos iniciales $f(W_0)$, se toman estos nuevos pesos como referencia y se vuelve al paso ii-), de lo contrario se regresa a i-).
- iv-). Durante el ciclo de iteraciones el valor de h es ajustado de la siguiente forma: después de s pasos exitosos el nuevo valor es $h + \delta_s$, es decir, se hace más rápido el aprendizaje. Después de f pasos fallidos el nuevo valor es $h - \delta_f$, esto es, se hace más lento el aprendizaje.

⁷ "Bacteria are too small to be able to measure spatial concentration gradients of chemoattractants. When swimming in a medium with varying concentrations they generate random directions instead and keep going as long as concentration increases. If attractant concentration does not or no longer increases, then they stop, tumble, then emerge in a new direction at random angles to the old direction. In this way they move towards larger and larger concentration values of the attractant. In other words, they optimize the function that describes the concentration of the chemoattractant in the medium", Bremermann y Anderson (1991, pp. 127).

⁸ En el caso de los perceptrones multicapa se parte de dos matrices, una en la capa oculta y otra en la de salida.

El proceso puede detenerse definiendo un número máximo de iteraciones, o sino un valor mínimo del error cuadrado medio que resulte satisfactorio. En Delgado (2000) se encuentra el diagrama de flujo del algoritmo *chemotaxis*.

El objetivo de cualquier algoritmo de entrenamiento es hacer mínimo el error cuadrado medio (o cualquier otra función de pérdida), pero la experiencia ha demostrado que las redes tienden a sobreajustar los datos. Por esta razón es usual destinar el 70% de la muestra para el entrenamiento y el 30% para la prueba del modelo. La selección de la mejor red se realiza con base en la arquitectura que obtenga el menor valor del error cuadrado medio en el conjunto de prueba. En la Figura 5 se muestra cómo es el comportamiento del error cuadrado medio a medida que el algoritmo avanza en cada iteración, se ve que en el conjunto de entrenamiento este valor siempre decae con el número de iteraciones, pero para seleccionar la red óptima sólo hay un número de iteraciones que hace mínimo el error cuadrado medio en el conjunto de prueba. De ese número de iteraciones en adelante se representan arquitecturas que ajustan muy bien en el entrenamiento pero muy mal con los valores de prueba.

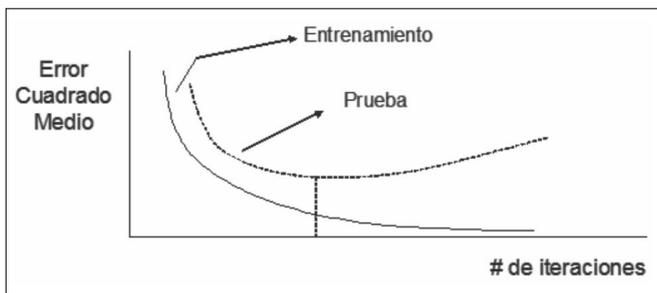


Figura 5. Selección de la Red Óptima.

En finanzas no es importante examinar si un modelo pronóstica adecuadamente el precio en un momento del tiempo, sino si es capaz de predecir los incrementos o descensos de estos con relación a un periodo anterior; esta información es más relevante para tomar decisiones económicas (Pesaran y Timmermann, 1992). Un indicador sencillo para medir este aspecto es el estadístico de *cambio de dirección* (CD), que presenta el porcentaje de veces que el modelo predice acertadamente un aumento o caída de la variable.

En un contexto de modelos de negociación las medidas anteriores no dan cuenta de la rentabilidad que produce una estrategia determinada. Autores como Kuan y Liu (1995) afirman que un tema de investigación en finanzas neuronales debe explorar la posibilidad de que los algoritmos de entrenamiento usen como funciones de pérdida la rentabilidad anualizada de una estrategia de negociación en lugar del error cuadrado medio⁹.

El cálculo de la rentabilidad anualizada (RA) de una estrategia está ligado a la regla de negociación que se siga. Si se supone un modelo que predice el precio del activo en la hora siguiente y_t^* y el valor que tomará dentro de seis horas y_{t+6} , u n a regla de negociación sencilla es:

Si $y_{t+6}^* > y_t^*$ entonces en t se compra el activo al precio de ese momento

O Si $y_{t+6}^* < y_t^*$ entonces en t se vende el activo al precio de ese momento

El capital final K_T , que tendrá un inversionista que se introduce en el negocio con un capital inicial K_0 durante un periodo $[0, T]$, sin tener en cuenta los costos de transacción, no liquidando la posición antes de la generación de una señal de negociación, asumiendo que el precio de compra es el mismo de venta y reinvertiendo la totalidad del capital acumulado en cualquier momento,¹⁰ viene dado por:

$$K_T = K_0 \prod_{t=0}^T \left(1 + c_t \left| \frac{y_{t+6}}{y_t} - 1 \right| \right) \quad (4)$$

Donde $c_t = \text{signo} \left(\frac{y_{t+6}}{y_t} - 1 \right) * \text{signo} \left(\frac{y_{t+6}^*}{y_t^*} - 1 \right)$.

Si en un periodo de seis horas el modelo de pronóstico que respalda la decisión de comprar o vender arroja una señal y acierta, entonces $c_t = 1$ y el incremento de los precios con respecto a seis horas atrás es sumado al capital que se tiene hasta ahora. Si no se acierta, entonces $c_t = -1$ y se pierde ese aumento. En el caso en que el modelo no detecte un cambio de dirección o que el precio observado se mantenga constante, el inversionista ni pierde ni gana.

La rentabilidad anualizada de la estrategia de negociación descrita arriba va a estar dada por la siguiente expresión:

$$RA = \left[\frac{K_T}{K_0} \right]^{M/T} - 1 \quad (5)$$

Donde M es el número de días de negociación en un año, algunos autores escogen 252; otros, 360 días. Y T es el tiempo que duró la inversión expresado en días.

Los inversionistas prefieren que el crecimiento de su capital sea positivo y constante, las grandes variaciones en las rentabilidades son peligrosas, por esta razón es preferible ajustar la rentabilidad por una medida del riesgo. La *razón de Sharpe* es un indicador que da cuenta de esto y permite comparar dos estrategias de negociación, este se calcula dividiendo la rentabilidad anualizada por la desviación estándar de los retornos (también anualizada).

⁹ "As our estimation methods are based on MSE minimization, which is not a loss function for sign predictions, it would be very interesting to construct estimation methods based on a suitable loss function and compare the resulting sign prediction results" [Kuan y Liu, 1995, pp. 359].

¹⁰ Estos supuestos implican que no existen reglas para administrar el riesgo en el manejo del capital (tales como el take profit o el stop loss) ni reglas de entrada y salida en el mercado.

Antecedentes

Hay trabajos anteriores en los que se explora el poder predictivo de las redes neuronales en los mercados de divisas. Kuan y Liu (1995) trabajaron con las tasas CHF/USD, DEM/USD, GBP/USD, USD/JPY y el dólar canadiense; la muestra usada por ellos tomaba datos diarios de los precios a los que el mercado abre en la Bolsa de New York desde el primero de marzo de 1980 hasta el 28 de enero de 1985, en total 1.245 observaciones. No usaron como variable de salida el nivel de la tasa de cambio sino su nivel logarítmico, y como variables de entrada consideraron únicamente entre uno y seis rezagos de la misma serie. Usaron dos tipos de redes; las estáticas o multicapa y las dinámicas o recurrentes, con funciones de activación logística (entre dos y seis neuronas) en la capa oculta, y una función de identidad en la capa de salida. El desempeño de los modelos fue comparado con el de modelos Arima, encontrando que no existe una diferencia sorprendente entre el pronóstico del cambio de dirección de los dos tipos de modelos.

Yao y Tan (2000) trabajaron con las mismas divisas que Kuan y Liu, excluyendo el dólar canadiense y agregando el dólar australiano. En este trabajo se examinó el desempeño de una estrategia de negociación con base en las simulaciones de redes neuronales entrenadas con ayuda del algoritmo de propagación inversa. Los autores, teniendo en cuenta el efecto de los costos de transacción, consideraron un modelo para realizar operaciones cada viernes, ellos construyeron una muestra de 510 observaciones con los precios de cierre desde el 18 de mayo de 1984 hasta el 7 de julio de 1995. El periodo de entrenamiento tomaba cifras hasta el 12 de julio de 1991. A diferencia de Kuan y Liu, trabajaron con el nivel de la tasa de cambio, y de igual manera que ellos, compararon sus modelos contra un Arima, pero aquí prestaron especial énfasis a indicadores de desempeño financiero sobre la estrategia de negociación simulada. Por ejemplo, compararon el resultado financiero de negociar con los pronósticos de la red con la rentabilidad de comprar al inicio del periodo y vender hasta el final (a esta estrategia se le llama *comprar y mantener*). Las redes que entrenaron las usaron como variables de entrada, rezagos de la serie y promedios móviles de las mismas. Yao & Tan concluyeron que la rentabilidad de la estrategia simulada con las redes neuronales para todas las divisas, excepto para el USD/JPY, son superiores al retorno reportado por las estrategias de referencia, y recomendaron reentrenar con frecuencia los modelos neuronales, por lo menos cada 20 semanas. También sugieren que los trabajos futuros se concentren en realizar el entrenamiento de las redes teniendo como objetivo la maximización de la rentabilidad de una estrategia de negociación.

Dunis y Willians (2002) trabajaron la tasa de cambio EUR/USD en frecuencia diaria, usaron una muestra de precios de cierre del 17 de octubre de 1997 hasta el 18 de mayo de 2000 para el entrenamiento, y del 19 de mayo de 2000 hasta el 3 de julio de 2001 para la

validación. La variable de salida fue la tasa de crecimiento de la divisa en función de rezagos de ella misma y de los retornos de los índices de precios de bolsas importantes, como el DAX 30 y el STOXX 50. Emplearon una red neuronal con funciones logísticas y las entrenaron por propagación inversa. Probaron el desempeño financiero de la estrategia en la que se compran euros si el pronóstico de la red es positivo y se venden euros si sucede lo contrario, y compararon los resultados con otros tipos de modelos como el ARMA, la regresión logística y la estrategia MACD. Ellos concluyeron que en términos del error cuadrado medio no hay mucha diferencia entre los ARMA y las redes, pero la rentabilidad de la red es seis veces superior al ARMA, es el doble del modelo logístico y es sólo un poco superior a la estrategia MACD.

El presente trabajo se diferencia de los anteriores porque considera una frecuencia menor. Se propone un modelo de negociación para intervalos de seis horas. Se complementan los anteriores estudios al proponer un mecanismo de entrenamiento que permite ajustar los pesos de la red neuronal usando como función objetivo la rentabilidad o indicadores de esta. Se emplea una función de activación poco usada como lo es la doble exponencial o Gumbel. Y en la selección de los modelos se usa la técnica multivariante del *análisis de componentes principales* (ACP) para seleccionar el mejor modelo dentro de varios entrenados, atendiendo a múltiples criterios.

Metodología

Se eligió una muestra de 623 observaciones con frecuencia en horas de Londres. Se dividió la muestra en un periodo de entrenamiento (70% de los datos) y otro de prueba (el 30% restante), como lo recomiendan las referencias consultadas –en particular Gibb (1996) y Kröse y Van der Smagt (1996)– para evitar el problema de sobre-entrenamiento. El 70% de la muestra (436 datos) para el entrenamiento de la red comprende registros del 7 de diciembre de 2004 a las 13 horas hasta el 31 de diciembre del 2004 a las 11 horas, día en que la cotización EUR/USD registró su valor máximo histórico. El 30% restante de la muestra (187 datos) va hasta el 12 de enero de 2005 a las 6 horas de Londres. En la Figura 6 se encuentra el conjunto de patrones de entrenamiento y de prueba.

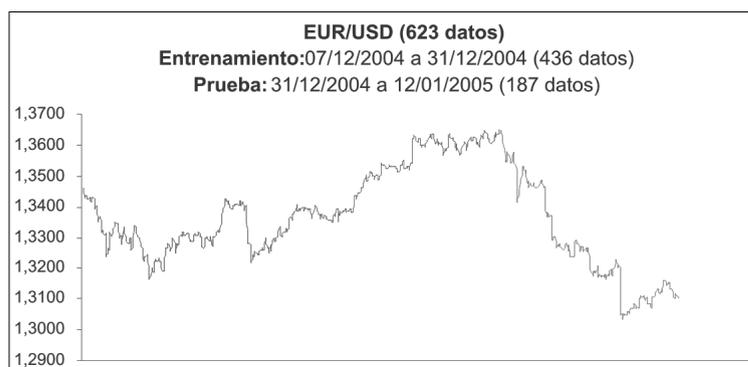


Figura 6. Período de entrenamiento y prueba para el EUR/USD.

Fueron entrenadas 780 redes neuronales, 630 usando variables fundamentales y 150 con indicadores técnicos. A las primeras se les dio el nombre de modelos tipo I y a las segundas de modelos tipo II. En todos los modelos se consideraron dos variables de salida, el pronóstico de la cotización del EUR/USD en la hora siguiente y dentro de seis horas. Se usó el algoritmo de propagación inversa con tasa de aprendizaje 0,4. El número de épocas fue 400. Se implementó el entrenamiento en Matlab haciendo que el número semilla del generador de números aleatorios de la máquina fuera el mismo en todos los entrenamientos, esto con el propósito de hacer comparaciones entre ellos. A continuación se describen ambos tipos de modelos, y en la Tabla 4 se resumen sus características:

a) *Modelos que Incluyen como entradas las variables fundamentales (modelos tipo I):* en principio se usaron como variables de entrada los precios de cierre de EUR/USD rezagados en el tiempo. Ya que para la periodicidad que se maneja (una hora) no es posible construir una serie con variables económicas relevantes como la base monetaria o la cuenta de capitales de corto plazo, se tomó en su lugar, y en la misma periodicidad, la serie de la tasa de cambio USD/JPY y EUR/JPY debido a que el Japón se ha caracterizado por tener gran importancia en la inversión de corto plazo en Estados Unidos y en la balanza comercial de la Comunidad Europea. Todos los valores de las variables de entrada y salida se reescalaron entre cero y uno.

b) *Modelos que incluyen como entradas indicadores del análisis técnico (modelos tipo II):* se usaron los indicadores TRIX y el oscilador estocástico (%K)¹¹ construidos para el EUR/USD y para el EUR/JPY. El índice TRIX se calculó con un valor de *lambda* igual a 0,9 para la primera tasa de cambio y para la última con 0,6. Estos valores de *lambda* son los que generaron mayor rentabilidad en el periodo de entrenamiento cuando se empleó la estrategia TRIX. De igual forma los patrones de entrenamiento se llevaron a un intervalo entre cero y uno.

En la Tabla 2 se resumen los 780 modelos entrenados, discriminados por sus variables de entradas y por las funciones de activación usadas en las capas oculta y de salida. Las dos últimas columnas de la tabla muestran los rezagos usados para las variables de entrada y el número de neuronas en capa oculta que se utilizó.

Se consideraron dos funciones de activación: la logística y Gumbel (o valor extremo). Para cada arquitectura se usaron tres combinaciones de estas funciones:

- 1) Logística en capa oculta y de salida (se llamó Log).
- 2) Gumbel en capa oculta y de salida (se llamó Gumbel).
- 3) Logística en capa oculta y Gumbel en capa de salida (se llamó Loggumbel).

En los modelos entrenados la regla de negociación que se evaluó fue:

Si $y_{t+6}^* > y_t^*$ Entonces en t se compran euros al precio EUR/USD_t

O Si $y_{t+6}^* < y_t^*$ Entonces en t se venden euros al precio EUR/USD_t

Tabla 2. Modelos Entrenados.

MODELOS TIPO I (1a Etapa)					
NOMBRE	V/BLES DE ENTRADA	ACTIVACIÓN	No. de REDES	REZAGOS	NCO
A	EUR/USD	LOG	60		
		GUMBEL	60		
		LOG-GUMBEL	60	1 y 24	
B	EUR/USD & USD/JPY	LOG	60	1 a 2 y 24	2, 4, 6, 8
		GUMBEL	60	1 a 3 y 24	10, 12, 14
		LOG-GUMBEL	60	1 a 4 y 24	16, 18 y 20
C	EUR/USD & EUR/JPY	LOG	60	1 a 5 y 24	
		GUMBEL	60	1 a 6 y 24	
		LOG-GUMBEL	60		

MODELOS TIPO I (2a Etapa)					
NOMBRE	V/BLES DE ENTRADA	ACTIVACIÓN	No. de REDES	REZAGOS	NCO
C	EUR/USD & EUR/JPY	LOG	30	1 a 4	3,7,9,11,17
		GUMBEL	30	1 a 8	19, 21, 23
		LOG-GUMBEL	30	1 a 12	27 y 31

MODELOS TIPO II					
NOMBRE	V/BLES DE ENTRADA	ACTIVACIÓN	No. de REDES	REZAGOS	NCO
A	TRIX (EUR/USD) & %K (EUR/JPY)	LOG-GUMBEL	50	1 a 2	2, 4, 6, 8
				1 a 3	10, 12, 14
B	TRIX (EUR/USD) & %K (EUR/USD)	LOG-GUMBEL	50	1 a 4	16, 18 y 20
				1 a 5	
C	TRIX (EUR/USD) & TRIX (EUR/JPY)	LOG-GUMBEL	50	1 a 6	

A cada ajuste se le calculan diez indicadores:

- La raíz del error cuadrado medio (RECM) con los datos de entrenamiento. A menor valor es mejor el ajuste.
- RECM con los datos del conjunto de prueba.
- RECM total que es la suma de las dos cifras anteriores. Este indicador selecciona modelos que sean buenos tanto en el entrenamiento como en la prueba.
- El índice Theil con datos del conjunto de prueba. Este indicador, a medida que se acerca a cero, señala un mayor poder predictivo, se construye con base en la siguiente expresión:

$$Theil = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - y_i^*)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i)^2} + \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i^*)^2}} \in [0,1]$$

¹¹ El TRIX es el nombre para la estrategia "triple promedio móvil exponencial", que consiste en realizar un suavizamiento sobre una serie usando un factor de decaimiento lambda. Al promedio se le aplica nuevamente el promedio móvil una y otra vez. La señal de negociación consiste en comparar el valor suavizado con un periodo atrás, si es más alto se compra, y se vende si sucede lo contrario. El "oscilador estocástico" construye una serie histórica a partir del precio más alto, más bajo y de cierre, en una jornada de negociación; se divide la diferencia del cierre con el precio más bajo entre la diferencia del precio más alto con el más bajo, esta razón ofrece una idea de si la tendencia del precio en una secuencia de rondas de negociación es creciente o decreciente; dependiendo de esto, se compra o vende.

- El criterio de información de Aikake (AIC) con datos del conjunto de prueba. Este indicador, además de ayudar a seleccionar el modelo con el menor error cuadrado medio, busca el modelo más *parsimonioso*, o sea, aquel que con menos parámetros (pesos) tenga un buen desempeño predictivo. Entre menor sea el valor del indicador es más preferible el modelo; se calcula de acuerdo con la siguiente fórmula:

$$AIC = \ln \left[\frac{1}{N} \sum_{i=1}^N (v_i - y_i^*)^2 \right] + \frac{2p}{N}$$

Donde p es el número de pesos de la red que el algoritmo de entrenamiento debe ajustar y N es el número total de datos en el conjunto de prueba.

- El criterio de información de Schwarz (SIC) con datos del conjunto de prueba. Tiene la misma interpretación que el criterio AIC, pero este castiga mucho más a los modelos sobreparametrizados. Se calcula así:

$$SIC = \ln \left[\frac{1}{N} \sum_{i=1}^N (v_i - y_i^*)^2 \right] + \frac{p \cdot \ln(N)}{N}$$

- El cambio de dirección (CD) con datos del conjunto de prueba. Es un porcentaje que, entre más grande sea, más favorable es la predicción del modelo.
- El coeficiente de asociación C con datos del conjunto de prueba. Es una medida de asociación construida a partir del estadístico de prueba del *test* de asociación de *Pearson*, en el que se organizan los datos en una tabla de contingencia 2×2 y se asocia lo que ocurre en la realidad (el precio sube o baja) con el pronóstico de la red, si hay poca asociación este indicador toma valores pequeños. Sobre esta prueba, que también se conoce como *ji-cuadrado*, se puede consultar en Peña (1998).
- La rentabilidad anualizada (RA) se expresa en porcentaje, y habla bien de la estrategia de negociación entre más alta sea.
- Y la *razón de Sharpe* con datos del conjunto de prueba, que en la medida en que sea más alta indica que la estrategia es más rentable y menos riesgosa.

En la mayoría de trabajos consultados, por ejemplo Dunis & Willians (2002), la mejor red se elige realizando un ordenamiento por la función de pérdida; la red con menor valor del ECM es el mejor modelo; no obstante, también existe interés en que otros indicadores tengan simultáneamente un buen desempeño. En este trabajo se empleó el análisis de componentes principales (ACP) para seleccionar la mejor red neuronal dentro de varias que ya han sido entrenadas.

El ACP permite la reducción de la dimensionalidad de un conjunto multivariado de datos, llevando las observaciones a un nuevo sistema de ejes que conserva la misma variabilidad, estos nuevos ejes son combinaciones lineales de las variables originales. El procedimiento parte de la obtención de los valores propios de la matriz de varianzas y covarianzas de los datos. El vector propio asociado al valor propio más grande es el que resume la mayor parte de la variabilidad global de los datos; este porcentaje de varianza explicada es igual al valor propio dividido por la suma de todos los valores propios obtenidos. Entre más dependencia exista entre las variables, mayor será esta proporción.

En este caso es indicado usar ACP porque hay bastante dependencia entre las variables que miden el ajuste de los datos en el modelo neuronal en términos del error, del pronóstico del cambio de dirección y de la rentabilidad de la estrategia. Al aplicar la combinación lineal sugerida por el vector propio de la componente principal se obtiene un puntaje con el cual se pueden organizar los modelos.

Resultados

En el ACP se consideraron únicamente las variables: rentabilidad anualizada (RA), razón de Sharpe, pronóstico en el cambio de dirección (CD) y el criterio de información de Schwarz (SIC). En la Tabla 3 se muestran los valores propios asociados a los nuevos ejes o componentes, el primero de ellos retiene el 58,64% de la inercia (o varianza) del total de datos.

Tabla 3. ACP Modelos tipos I y II.

Componente	V/rs Propios	% de Varianza Exp.	% Acumulado
1	2,3455	58,64	58,64
2	1,0227	25,57	84,21
3	0,4777	11,94	96,15
4	0,1540	3,85	100,00

En la Tabla 4 se señala el vector propio asociado a la primera componente. Se observa que en la construcción del nuevo eje los indicadores RA y razón de Sharpe reciben más importancia que el índice CD. El criterio SIC tiene menor peso en la conformación de la componente principal.

Tabla 4. Primera Componente modelos tipos I y II.

Variable	1a Componente
RA	0,9040
SHARPE	0,9072
CD	0,7912
SIC	0,2817

En el Anexo se exhiben los 20 primeros modelos seleccionados de acuerdo al puntaje que ofrece la primera componente. El primero de ellos, C4GUMBEL con arquitectura

¹² Hubo modelos de tipo II como B6LOGGUMBEL con diez y doce neuronas en capa oculta y el B6LOGGUMBEL con dos neuronas en capa oculta que tuvieron rentabilidades anualizadas superiores a 4%; no obstante, no se tuvieron en cuenta porque estas redes no "aprendieron" con el entrenamiento, prueba de ello son sus pésimos ajustes en términos del error cuadrado medio.

¹³ Si la predicción del modelo fuera acertada en todo el conjunto de datos de prueba se lograría una rentabilidad máxima posible del 17% anual.

(8,19,2) produce una rentabilidad del 4,05% anual,¹² es un modelo de negociación aceptable si se tiene en cuenta que dobla la tasa de interés del 2% anual que mantenía la Reserva Federal de Estados Unidos en ese momento.¹³

La Tabla 5 describe cuáles son las señales de negociación que produce la red neuronal seleccionada. El modelo ofrece señales cada seis horas. El valor -1 en la penúltima columna significa vender euros, y el valor 1 significa comprarlos. Estas transacciones se hacen en términos de dólares y la negociación inicia el primer día del periodo de prueba (31 de diciembre de 2004 a las 12 horas de Londres) con una inversión de un dólar. En la última columna se tiene el registro de cómo evoluciona el capital del inversionista con cada señal de negociación.

Tabla 5. Señales de negociación de la red C4GUMBEL (8,19,2). 2ª etapa.

Número de trades	Fecha	Hora (Londres)	Salidas deseadas		Salidas de la red		Señal de negociación	Evolución del capital
			EUR/USD (t)	EUR/USD (t+6)	EUR/USD (t)	EUR/USD (t+6)		
1	31/12/2004	12:00:01	1,3647	1,3579	1,3489	1,3476	-1	1,0054
2	31/12/2004	20:00:01	1,3566	1,3545	1,3428	1,3434	1	1,0039
3	03/01/2005	1:00:01	1,3536	1,3479	1,3446	1,3435	-1	1,0081
4	03/01/2005	7:00:01	1,3527	1,3519	1,3451	1,3486	1	1,0075
5	03/01/2005	13:00:01	1,3476	1,3473	1,3533	1,3533	1	1,0073
6	03/01/2005	19:00:01	1,3478	1,3460	1,3545	1,3542	-1	1,0086
7	04/01/2005	1:00:01	1,3467	1,3460	1,3554	1,3560	1	1,0081
8	04/01/2005	7:00:01	1,3467	1,3367	1,3576	1,3569	-1	1,0156
9	04/01/2005	13:00:01	1,3371	1,3303	1,3576	1,3582	1	1,0105
10	04/01/2005	19:00:01	1,3270	1,3276	1,3575	1,3577	1	1,0109
11	05/01/2005	1:00:01	1,3267	1,3283	1,3590	1,3587	-1	1,0097
12	05/01/2005	7:00:01	1,3255	1,3240	1,3579	1,3581	1	1,0085
13	05/01/2005	13:00:01	1,3241	1,3278	1,3609	1,3612	1	1,0114
14	05/01/2005	19:00:01	1,3262	1,3254	1,3615	1,3612	-1	1,0120
15	06/01/2005	1:00:01	1,3263	1,3241	1,3599	1,3596	-1	1,0137
16	06/01/2005	7:00:01	1,3215	1,3193	1,3537	1,3532	-1	1,0153
17	06/01/2005	13:00:01	1,3185	1,3175	1,3532	1,3506	-1	1,0161
18	06/01/2005	19:00:01	1,3167	1,3168	1,3312	1,3336	1	1,0162
19	07/01/2005	1:00:01	1,3184	1,3176	1,3350	1,3342	-1	1,0168
20	07/01/2005	7:00:01	1,3197	1,3202	1,3348	1,3334	-1	1,0164
21	07/01/2005	13:00:01	1,3209	1,3055	1,3326	1,3317	-1	1,0283
22	07/01/2005	19:00:01	1,3052	1,3073	1,3288	1,3290	1	1,0300
23	10/01/2005	0:00:01	1,3071	1,3071	1,3205	1,3213	1	1,0300
24	10/01/2005	6:00:01	1,3075	1,3105	1,3238	1,3238	1	1,0324
25	10/01/2005	12:00:01	1,3106	1,3088	1,3241	1,3240	-1	1,0338
26	10/01/2005	18:00:01	1,3085	1,3105	1,3218	1,3213	-1	1,0322
27	11/01/2005	0:00:01	1,3119	1,3130	1,3106	1,3090	-1	1,0313
28	11/01/2005	6:00:01	1,3119	1,3165	1,3085	1,3088	1	1,0349
29	11/01/2005	12:00:01	1,3145	1,3133	1,3083	1,3084	1	1,0340
30	11/01/2005	18:00:01	1,3134	1,3114	1,3082	1,3081	-1	1,0356
31	12/01/2005	0:00:01	1,3104	1,3115	1,3082	1,3083	1	1,0364
32	12/01/2005	6:00:01	1,3119	1,3123	1,3071	1,3071	-1	1,0361

Para ilustrar cómo un inversionista toma posiciones con estas señales, considere el noveno trade de la Tabla 5: el día cuatro de enero de 2005 a las 13 horas la red pronostica que el valor de la cotización en una hora será de 1,3576 y seis horas después de 1,3582; con esta información la regla de negociación señala que el inversionista debe comprar euros al precio de ese momento (a esta operación se le representa por un 1 en la penúltima columna). Pasadas siete horas se sabe que los valores que tomó efectivamente el EUR/USD una hora y seis horas después fueron 1,3371 y 13303, es decir, que la decisión acertada hubiese sido vender en lugar de comprar, por esta razón el inversionista tiene una pérdida con este trade; espera que lleguen las 19 horas del mismo día y ejecuta otra señal con la totalidad del capital acumulado que lleva hasta entonces, y así sucesivamente hasta que se acabe el periodo de negociación. Los aumentos y disminuciones del capital con cada trade describen la evolución de la inversión del cliente. En la Figura 7 se observa

dicho comportamiento en los ocho días de negociación de la Tabla 5.



Figura 7. Evolución del capital. Red C4GUMBEL (8,9,12) 2ª etapa.

Conclusiones

El mercado de la divisa EUR/USD es especialmente líquido y muestra tener las características de *eficiencia* que subraya Fama. No obstante, el modelo neuronal seleccionado en este trabajo enseña que sí es posible (usando el pronóstico de una red neuronal y siendo fiel a una estrategia de negociación) obtener rentabilidades positivas de manera sostenida en el mercado Forex. Las investigaciones recientes avanzan en este camino, dichos trabajos tienen por objetivo *automatizar* las decisiones que toma un *trader* destacando las siguientes ventajas:

1. Un *trader* está gobernado por sus emociones, y en momentos de euforia o de pánico toma decisiones apresuradas. Un sistema automatizado está libre de emociones y su

desempeño puede ser probado *a priori*.

2. Hay decisiones de control de riesgo que los *traders* tienden a pasar por alto y las comisionistas de bolsa necesitan disponer de una Dirección de medición y control de riesgos financieros encargada de poner un límite a los montos con los que los *traders* hacen negociaciones, también fijan un valor máximo de pérdida a partir del cual el *trader* se ve obligado a liquidar la negociación ejecutada. Estas decisiones de control de riesgos pueden ser simuladas por el mismo sistema automatizado de *trading*.

Un *sistema de negociación automatizado* (SNA) es un conjunto bien definido de reglas que se establecen para determinar una estrategia de operación en un mercado. Estos sistemas deben fijar como mínimo las siguientes reglas:

1. Cuándo entrar al mercado: establecer una *orden* de compra o venta, es decir, a qué nivel del precio del activo el *trader* debe comprar o vender.

2. Administración del dinero (*money management*): establecer cuánto dinero de su capital un *trader* está dispuesto a arriesgar en cada negociación.
3. Cuándo salir del mercado: establecer el momento en el que un *trader* debe liquidar una posición de compra o venta, esta decisión por lo regular va de la mano con las siguientes dos reglas.
4. Detener las pérdidas (*stop loss*): establecer cuánto dinero en términos absolutos o relativos un *trader* está dispuesto a perder en una negociación.
5. Apropiarse de las ganancias (*take profit*): establecer cuánto dinero en términos absolutos o relativos un *trader* está dispuesto a ganar en una negociación.

Como herramienta en la construcción de un SNA se privilegia a las redes neuronales por su capacidad de “aprender” relaciones complejas (no lineales) subyacentes a los patrones históricos presentados para su entrenamiento, y pueden “generalizar”, es decir, reconocer las relaciones aprendidas con patrones diferentes a los de entrenamiento. En comparación con los modelos estadísticos, las redes neuronales tienen la ventaja de que no requieren un conocimiento *a priori* de la forma funcional entre entradas y respuestas, además las redes son muy flexibles; hay bastantes parámetros (capas ocultas, neuronas en capa oculta, tasa de aprendizaje, etc.) con los cuales ensayar cuál es el mejor modelo para una tarea dada¹⁴.

Pero las redes neuronales también tienen desventajas:

1. Para quienes quieren encontrar una relación explícita entre las variables de entrada y la respuesta, el modelo neuronal produce una expresión altamente no lineal donde los pesos que se optimizan en el entrenamiento no tienen interpretación alguna, como sí la tienen los coeficientes de un modelo estadístico.
2. Las decisiones de negociación que se toman a diario en los mercados de valores son inmediatas. Encontrar una red óptima es un proceso de “ensayo y error” que consume demasiado tiempo, este se hace más grande si la red es muy compleja (tiene bastantes neuronas ocultas) y si además el tamaño de muestra para entrenamiento también es grande.
3. No existe un procedimiento respaldado teóricamente para determinar el número de neuronas en capa oculta ni las variables de entrada con mayor relevancia, y no se puede hacer inferencia estadística para medir la importancia de los pesos, aunque ya hay trabajos que estudian esta posibilidad (Sarle, 1994).
4. La superficie del error para una red compleja está formada por muchos mínimos locales, haciendo que sea muy probable que el algoritmo de propagación inversa quede atrapado en uno de ellos.

En este trabajo se han utilizado las *redes neuronales multicapa* para construir un SNA que sirva para operar en el mercado del EUR/USD. En el modelo entrenado aquí se simplifican algunas de las características que debe tener un SNA: el modelo sugerido produce señales (de compra o venta) cada seis horas, hay señal de compra si el pronóstico de la red seis horas adelante es mayor que el pronóstico del precio actual y de venta si es menor, no se puede liquidar una posición antes de las seis horas (en otras palabras, no hay cabida para que operen el *stop loss* y el *take profit*) y siempre se invierte la totalidad del capital (no hay administración del dinero). Con estas simplificaciones el objetivo era conseguir un modelo que funcionase muy bien con el pronóstico y que maximizara la rentabilidad de la estrategia de negociación. Sin embargo, en la práctica las consideraciones de administración del dinero son muy importantes. De hecho una de las principales desventajas del modelo entrenado en este trabajo es que no se consideraron los costos de transacción, más cuando se trata de un sistema de negociación *intraday*, donde estos pueden significar una parte importante del capital. Por ejemplo, las redes entrenadas por Yao y Tan consideraron un costo de 1% por transacción en un modelo de negociación semanal, y en las redes de Dunis y Willians (2002) se asumió un costo de 0,033% por *trade*, y aun así, las redes neuronales muestran un mejor desempeño financiero que otros modelos.

La selección del modelo no se dejó en manos de la subjetividad del investigador sino que con ayuda de la técnica ACP, por red neuronal entrenada se asignó una ponderación a cuatro indicadores (la rentabilidad anualizada, la razón de Sharpe, el cambio de dirección y el criterio de Schwarz), y con el puntaje final de la combinación lineal de estas variables y sus ponderadores se organizaron los modelos; los que tenían un mayor puntaje exhibían buenas características en términos de estas variables. En las ponderaciones resultantes se subraya la escasa importancia que tiene el SIC (un indicador en términos del error cuadrado medio) en comparación con la rentabilidad de los modelos ajustados. Esto comprueba la afirmación de Kuan y Liu (1995), quienes enfatizan que si el propósito es la rentabilidad de una estrategia de negociación, las redes deben ser entrenadas usando una función objetivo que dé cuenta de esto, y no el error cuadrado medio.

Chemotaxis puede convertirse en una alternativa poderosa para la construcción de un SNA, puesto que tiene unas propiedades que lo hacen más atractivo en finanzas que el algoritmo de propagación inversa. En la Tabla 6 se hace una comparación entre estos dos mecanismos de entrenamiento de una red neuronal.

Los trabajos futuros en la construcción de SNA utilizando redes neuronales deben avanzar no solo en la identifica-

¹⁴ Para la construcción de un SNA también se han considerado los sistemas de lógica borrosa, que sintetizan las decisiones de los traders en reglas lógicas, o los sistemas híbridos, que combinan tanto las redes neuronales como la lógica borrosa.

ción de señales exitosas de negociación sino también en la simulación de los porcentajes del capital con los que se negocia cada vez, y los montos de pérdida y ganancia que una estrategia está dispuesta a tolerar en cada *trade*. Antes que pronosticar los niveles de precios futuros, las redes neuronales, como lo sugieren Wong y Tan (1994), deben aprender estrategias exitosas y ser entrenadas, seleccionadas y probadas por su desempeño financiero. En este aspecto es de gran utilidad el algoritmo *chemotaxis* (pocas o ninguna vez citado en la literatura financiera), que permite tener como función objetivo a la razón de Sharpe o a la rentabilidad anualizada, sin tener la dificultad de encontrar el gradiente de estas funciones en términos de los pesos, como lo requiere el algoritmo de propagación inversa.

Tabla 6. Comparación entre *chemotaxis* y propagación inversa

<i>Chemotaxis</i>	<i>Propagación inversa</i>
No requiere del cálculo del gradiente de la función objetivo con respecto a los pesos.	Necesita el cálculo del gradiente de la función objetivo con respecto a los pesos.
Pese a que es de fácil implementación requiere más iteraciones que la propagación inversa.	Su implementación es más compleja que <i>chemotaxis</i> .
Es un modelo más adecuado para representar la activación sináptica. If nature optimizes through random walks, then, we conjecture, for the vertebrate brain, it should be Gaussian random walks, Bremermann y Anderson (1991: 127)	Es cuestionado por no comprobarse que la activación sináptica del cerebro funcione de acuerdo al gradiente que propone este algoritmo.

ANEXO. Selección de redes neuronales por ACP.

Bibliografía

Bremermann, H.J., Anderson, R.W., How the Brain Adjusts Synapses –Maybe., En: Automated Reasoning. Essays in Honor of Woody Bledsoe, Robert S. Boyer (Ed.), Boston, Kluwer Academic Publishers, 1991.

Delgado, A., Inteligencia Artificial y MiniRobots., Bogotá, Ecoe Ediciones, 1998.

Delgado, A., Control of Nonlinear Systems Using a Self-Organising Neural Network., Neural Computing & Applications, No. 9, 2000, pp. 113-123.

Dunis, C., Williams, M., Modelling and Trading the EUR/USD Exchange Rate: Do Neural Network Models Perform Better?., Liverpool Business School, Working Paper, 2002. Disponible en: www.cibef.com

Fama. E., The Behaviour of Stock Market Prices., Journal of Business, No. 38, 1969, pp. 34-105.

Fama. E., Efficient Capital Markets: A Review of Theory and Empirical Work., Journal of Finance, N° 25, 1970, pp. 383-417.

Funahashi, K.Y., On the Approximate Realization of Continuous Mapping by Three Neural Networks., Electronics and Communications in Japan, Part 3, No.73, 1989, pp. 61-68.

MODELO	Arquitectura	RECM. Entrenamiento. (1)	RECM. Prueba (2)	RECM Total (1) + (2)	Índice de Theil	Número de Parámetros	AIC	SIC	CD	C	RA (%)	Razón de Sharpe	Tiempo de Entrenamiento	Puntaje
C4GUMBEL (2a Etapa)	(8, 19, 2)	0,1679	0,1041	0,2720	0,1500	211	-2,2681	1,3777	62,5%	0,2058	4,0480	11,8829	1 hr 12 min 19 seg	3,17037
C4LOGGUMBEL (2a Etapa)	(8, 31, 2)	0,3817	0,5202	0,9019	0,5423	343	2,3613	8,2879	75,0%	0,2505	2,1565	9,9620	1 hr 16 min 49 seg	2,96563
B6LOGGUMBEL (Tipo II)	(12, 6, 2)	0,3277	0,2441	0,5717	0,3655	92	-1,8368	-0,2472	62,5%	0,1812	3,6618	11,2256	0 hr 57 min 14 seg	2,93065
C4GUMBEL (2a Etapa)	(8, 21, 2)	0,1835	0,0983	0,2818	0,1399	233	-2,1469	1,8790	65,6%	0,2344	3,3265	10,6144	1 hr 14 min 9 seg	2,92123
C4LOGGUMBEL (2a Etapa)	(8, 21, 2)	0,1573	0,0955	0,2528	0,1359	233	-2,2048	1,8211	65,6%	0,2344	3,3265	10,6144	1 hr 7 min 53 seg	2,91929
C4LOGGUMBEL (2a Etapa)	(8, 23, 2)	0,1655	0,1049	0,2704	0,1488	255	-1,7826	2,6235	62,5%	0,1812	3,3251	10,6117	1 hr 9 min 23 seg	2,92674
C4LOGGUMBEL (2a Etapa)	(8, 11, 2)	0,1389	0,0919	0,2308	0,1304	123	-3,4586	-1,3334	65,6%	0,2344	3,3265	10,6144	1 hr 0 min 19 seg	2,81374
A4LOGGUMBEL (Tipo II)	(8, 18, 2)	0,3156	0,2410	0,5566	0,3561	200	-0,7072	2,7485	59,4%	0,1288	3,5256	10,9832	1 hr 5 min 38 seg	2,80625
C4LOGGUMBEL (2a Etapa)	(8, 27, 2)	0,1739	0,1041	0,2779	0,1477	299	-1,3277	3,8386	62,5%	0,2328	3,2122	10,3989	1 hr 12 min 41 seg	2,80624
C4GUMBEL (2a Etapa)	(8, 27, 2)	0,1925	0,1095	0,3020	0,1557	299	-1,2267	3,9396	62,5%	0,1812	3,1794	10,3363	1 hr 19 min 57 seg	2,79181
C4LOGGUMBEL (2a Etapa)	(8, 19, 2)	0,1558	0,0973	0,2531	0,1396	211	-2,4029	1,2429	62,5%	0,2058	3,2109	10,3964	1 hr 6 min 27 seg	2,71868
C4GUMBEL (2a Etapa)	(8, 17, 2)	0,1546	0,0983	0,2529	0,1403	189	-2,6177	0,6479	62,5%	0,1812	3,2395	10,4508	1 hr 10 min 24 seg	2,71431
C4LOGGUMBEL (2a Etapa)	(8, 17, 2)	0,1420	0,0963	0,2383	0,1368	189	-2,6590	0,6087	62,5%	0,1812	3,2395	10,4508	1 hr 4 min 50 seg	2,71293
C4LOGGUMBEL	(10, 6, 2)	0,1581	0,1509	0,3090	0,2124	80	-2,9261	-1,5438	59,4%	0,1562	3,5919	11,1026	0 hr 59 min 0 seg	2,71241
C4LOGGUMBEL (2a Etapa)	(8, 9, 2)	0,1443	0,0889	0,2332	0,1240	101	-3,7612	-2,0160	62,5%	0,1812	3,2395	10,4508	1 hr 2 min 44 seg	2,62517
A4LOG	(5, 18, 2)	0,3109	0,2489	0,5597	0,3640	146	-1,2200	1,3027	81,3%	0,0286	1,4439	8,3269	1 hr 4 min 33 seg	2,56275
B6LOGGUMBEL (Tipo II)	(12, 8, 2)	0,3282	0,2484	0,5766	0,3700	122	-1,4810	0,6270	71,9%	0,3700	2,2996	8,4874	0 hr 58 min 44 seg	2,55163
B4LOGGUMBEL (Tipo II)	(8, 18, 2)	0,3229	0,2411	0,5639	0,3593	200	-0,7064	2,7493	68,8%	0,3248	2,3741	8,6510	1 hr 5 min 18 seg	2,55074
C4LOG (2a Etapa)	(8, 27, 2)	0,1586	0,0893	0,2479	0,1254	299	-1,6331	3,5333	59,4%	0,1288	2,9005	9,7876	1 hr 12 min 24 seg	2,50687
C4GUMBEL (2a Etapa)	(8, 31, 2)	0,2090	0,1361	0,3451	0,1942	343	-0,3204	5,6082	59,4%	0,1288	2,6868	9,3458	1 hr 23 min 53 seg	2,45774

Gibb, J., Back Propagation Family Album., Technical Report C/TR96-05, Macquarie University, August, 1996. Disponible en: <http://citeseer.ist.psu.edu/gibb96back.html>

Haykin, S., Neuronal Networks: A Comprehensive Foundation, New York, Mcmillan College Publishing Company, 1994.

Kröse, B., van der Smagt, P., An Introduction to Neural Networks, The University of Amsterdam. November, 1996.

Disponible en: <http://citeseer.ist.psu.edu/ose96introduction.html>

Kuan, C-M., Liu, T., Forecasting Exchange Rates using Feedforward and Recurrent Neural Networks., Journal of Applied Econometrics, Vol 10, 1995, pp. 347-364.

McCulloch, W.S., Pitts, W., A Logical Calculus of the Ideas Immanent in Nervous Activity., Bulletin of Mathematical Biophysics, N°5, 1943, pp. 115-133.

Minsky, M., Papert, S., Perceptrons: An Introduction to Computational Geometry., Cambridge, MA, MIT Press,

1969.

Peña, D., Estadística, Modelos y Métodos. Fundamentos, Alianza Editorial, 1998.

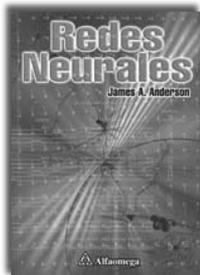
Pesaran, M., Timmermann, A., A Simple Nonparametric Test of Predictive Performance., Journal of Business and Economic Statistics, Vol 10, No 4: 1992, pp. 461-65.

Sarle, W.S., Neural Networks Implementation in SAS., Proceedings of the Nineteenth Annual SAS Users Group International Conference, April, 1994. Disponible en: <http://citeseer.ist.psu.edu/36580.html>

Wong, F., Tan, C., Hybrid Neural, Genetic and Fuzzy Systems., En: Trading on the Edge. Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets., Guido J. Deboek (Ed.), John Wiley and Sons. Inc., 1994.

Yao, J., Tan, L., A Case of Study on using Neural Networks to perform Technical Forecasting of FOREX., Neurocomputing, No. 34, 2000, pp. 79 – 98. Disponible en: <http://citeseer.ist.psu.edu/yao00case.html>

novedades **Alfaomega Colombiana S.A.**



REDES NEURONALES

ANDERSON, A. James
616 págs ilustr.
Rústica 17 X 23 cm
ISBN 978-970-15-1265-4
Alfaomega

Este libro familiariza al lector con los conceptos de la teoría de redes neuronales, reconociendo la naturaleza multidisciplinaria de esta técnica.

Resumen de contenido:

Propiedades de las neuronas individuales, Integración sináptica y modelos de neuronas, Operaciones vectoriales esenciales, Inhibición lateral y procesamiento sensorial, Operaciones sencillas de matrices, El asociador lineal. Antecedentes y fundamentos, El asociador lineal: simulaciones, Primeros modelos de redes: el perceptrón, Algoritmos de descenso en gradiente, Representación de la información, ¿Neuronas o qué, Aplicaciones de los asociadores sencillos: formación de conceptos y movimientos de objetos, Energía y redes neuronales: redes de Hopfield y máquinas de Boltzmann, Clasificadores del vecino más cercano, Mapas Adaptativos, Mapas de Kohonen, El Modelo ECC: una simple red neural autoasociativa no lineal, Cómputos asociativos, Enseñándole aritmética a una red

Adquiera nuestros textos en el punto de venta **Alfaomega Carrera 15 No 64a - 29** o en las principales librerías del país.

Afiliarse a nuestro **CLUB DEL CONOCIMIENTO** a través de nuestra página web, y reciba descuentos en nuestro punto de venta, contenidos actualizados vía Internet, información de novedades, prioridad en productos promocionales y entregas a domicilio sin costo adicional.

Visite nuestra página Web:
www.alfaomega.com.co