

Finance, Markets and Valuation

Red neuronal fuzzy híbrida versus red neuronal backpropagation: Aplicación a la predicción del índice bursátil Ibex-35

Hybrid fuzzy neural network versus backpropagation neural network: An application to predict the Ibex-35 index stock

Javier Oliver Muncharaz ¹

¹Departamento de Economía y Ciencias Sociales, Universidad Politécnica de Valencia. Valencia, España. Email: jaolmun@ade.upv.es

JEL: G17; C45

Resumen

El uso de las redes neuronales se ha extendido en todas las áreas de conocimiento por los buenos resultados que se están obteniendo en la resolución de los diferentes problemas planteados. La predicción sobre los precios en general, y los precios bursátiles en particular, representa uno de los principales objetivos del uso de las redes neuronales en finanzas. En este trabajo se presenta el análisis de la eficiencia de la hybrid fuzzy neural network frente a una red neuronal de tipo backpropagation en la predicción del precio del índice bursátil Español (IBEX-35). El trabajo se divide en dos partes. En la primera se expone las principales características de las redes neuronales como la hybrid fuzzy y la Backpropagation, sus estructuras y sus reglas de aprendizaje. En la segunda parte se analiza la predicción del índice bursátil IBEX-35 con estas redes midiendo la eficiencia de ambas en función de los errores de predicción cometidos. Para ello se han construido ambas redes con los mismos inputs y para el mismo periodo muestral. Los resultados obtenidos sugieren que la Hybrid fuzzy neuronal network es mucho más eficiente que la, tan extendida, red neuronal backpropagation para la muestra analizada.

Keywords: Híbrida difusa; Propagación hacia atrás; Red neuronal; Predicción de índices bursátiles

Abstract

The use of neural networks has been extended in all areas of knowledge due to the good results being obtained in the resolution of the different problems posed. The prediction of prices in general, and stock market prices in particular, represents one of the main objectives of the use of neural networks in finance. This paper presents the analysis of the efficiency of the hybrid fuzzy neural network against a backpropagation type neural network in the price prediction of the Spanish stock exchange index (IBEX-35). The paper is divided into two parts. In the first part, the main characteristics of neural networks such as hybrid fuzzy and backpropagation, their structures and learning rules are presented. In the second part, the prediction of the IBEX-35 stock exchange index with these networks is analyzed, measuring the

DOI:
10.46503/ALEP9985

Corresponding author
Javier Oliver Muncharaz

Recibido: 2 May 2020
Revisado: 23 May 2020
Aceptado: 1 Jun 2020

Finance, Markets and
Valuation
ISSN 2530-3163.

Cómo citar este artículo: Oliver Muncharaz, J. (2020) Red neuronal fuzzy híbrida versus red neuronal backpropagation: Aplicación a la predicción del índice bursátil Ibex-35. *Finance, Markets and Valuation* 6(1), pp. 85–98.

efficiency of both as a function of the prediction errors committed. For this purpose, both networks have been constructed with the same inputs and for the same sample period. The results obtained suggest that the Hybrid fuzzy neural network is much more efficient than the widespread backpropagation neuronal network for the sample analysed.

Keywords: Hybrid fuzzy; Backpropagation; Neural network; Predict stock index

1 Introducción

El estudio mediante diferentes modelos para la predicción de índices bursátiles está muy extendido en la literatura financiera, dada la importancia que presenta para los inversores y gestores de fondos conocer la tendencia de los precios (García, González-Bueno, y Oliver, 2015; García, Guijarro, Oliver, y Tamošiūnienė, 2018). Una mayoría de estudios se centran en la predicción de la volatilidad por su importancia de riesgo de inversión. Una gran parte de ellos se centran en los modelos de volatilidad de la familia GARCH, como en Ahmad, Ahmed, Vveinhardt, y Streimikiene (2016) en el que analizan la volatilidad para los mercados asiáticos aplicando uno de los modelos más extendidos GARCH(1,1) en el que confirman la existencia de componentes heterocedásticos en la volatilidad del índice. Otros estudios incorporan otro tipo de modelos para el estudio de la volatilidad. Así, por ejemplo, en Roh (2007) analiza la volatilidad del índice KSE KOSPI 200 mediante diferentes tipos de redes neuronales. En Dixit, Roy, y Uppal (2013) analizan el uso de la red neuronal backpropagation para la predicción del VIX del Indian Stock Index. En Hamid y Iqbal (2004) utiliza el mismo tipo de estructura de red neuronal para predecir la volatilidad del S&P 500 del contrato de futuro. También el uso combinado de una red neuronal backpropagation con un modelo de volatilidad de la familia GARCH es habitual, como en Kristjanpoller y Minutolo (2015) para el precio del oro que utilizan una ANN-GARCH.

Por otro lado, los modelos fuzzy systems se ha utilizado no solo para la clasificación sino también para los modelos de predicción. Algunos ejemplos pueden verse en Kuo y Xue (1999) que realiza una predicción de los precios de la leche mediante un modelo fuzzy neural networks, Jang (1993) modelizan la predicción del Shanghai Stock Index.

Los modelos fuzzy logic systems han sido muy utilizados inicialmente en machining process, pero como se ha visto, rápidamente se ha extendido su uso a problemas de predicción, selección, entre otros. No obstante, la combinación de la inteligencia artificial junto a la estructura fuzzy logic han llevado a la construcción de modelos híbridos que se han desarrollado en múltiples tipos de problemas rápidamente (Adnan, Sarkheyli, Zain, y Haron, 2013).

Los modelos híbridos suponen una mejora respecto a otros modelos individuales como las redes neuronales backpropagation u otros modelos fuzzy. En Mirbagheri y Tagiev (2011) realizan una comparativa de diferentes modelos para la predicción del crecimiento económico de Irán para el periodo 2002-2006 analizando la muestra del 1959-2001 siendo el modelo hybrid fuzzy neural network el de menor error de predicción frente a un modelo fuzzy-logic.

2 Modelos de redes neuronales

A continuación, se describen las características principales de los modelos empleados en este trabajo. Por un lado, la red neuronal backpropagation, mediante su algoritmo de aprendizaje, es capaz de obtener respuestas adecuadas a los problemas planteados, gracias a su capacidad de generalización ante patrones de datos no entrenado durante el proceso de aprendizaje.

Por otro lado, la hybrid fuzzy neural network aporta la ventaja, de la aplicación de sistemas fuzzy y de la aplicación de una red neuronal en el proceso de aprendizaje de las fuzzy rules del sistema. El uso de la red neuronal facilita el proceso de aprendizaje de las fuzzy rules a través de los propios datos, no siendo necesaria el conocimiento de los expertos para crearlas.

2.1 Backpropagation neural network

La red neuronal Backpropagation es una de las redes neuronales más extendidas en su aplicación en múltiples áreas de conocimiento (Rumelhart, Hinton, y Williams, 1986). Se trata de una red con un nivel de neuronas y de capas superior a las utilizadas con la red Perceptrón (Rosenblatt, 1958). Su proceso de aprendizaje es de tipo supervisado, es decir, a la red se le suministra los inputs seleccionados y los outputs que se desean obtener. Como su nombre indica, la red neuronal Backpropagation, utiliza un mecanismo de propagación del error hacia atrás (regla delta).

En una primera etapa, la información se transmite por toda la red a través de las neuronas y, generalmente conectadas mediante pesos aleatorios. Una vez se alcanza la capa de salida, la red compara el resultado obtenido con el deseado obteniendo un error. En una segunda etapa, se modifican los pesos de cada neurona empezando desde la capa de salida en función del error cometido por cada una de ellas. Una vez modificado los pesos, la información se transmite de nuevo a la red. Este proceso iterativo finaliza bien cuando se alcanza cierto grado de error bien por tiempo de computación, siempre a decisión del supervisor del entrenamiento.

El proceso de autoadaptar los pesos de las neuronas incrementa el potencial de aprendizaje de la red. Una de las características de la red backpropagation reside en su capacidad de generalización, es decir, la capacidad de ofrecer una solución favorable ante nuevas informaciones presentadas y no entrenadas previamente. Estos ajustes o variaciones en los pesos se basan en el algoritmo de la delta (Widrow y Hoff, 1960) utilizado en la red Perceptrón y Adaline.

$$\Delta w_{ij}(t + 1) = \alpha \delta_{pj} y_{pi} \quad (1)$$

donde:

$w_{ij}(t + 1)$: variación del peso de la neurona ij .

α : factor de aprendizaje o velocidad de aprendizaje. Valor constante entre [0,1].

δ_{pj} : valor de la delta o diferencia entre el output obtenido por la red y el output real.

y_{pi} : valor de salida de la neurona i para el patrón de aprendizaje p .

El proceso iterativo del paso de la información a través de las neuronas mediante este algoritmo puede necesitar un alto número de iteraciones. Para reducirlas, se añade un parámetro β o momentum que tiene en cuenta el efecto de una variación de los pesos de las neuronas en el instante anterior (Rumelhart y cols., 1986).

La constante β consigue una mayor convergencia en el proceso de aprendizaje reduciendo considerablemente el número de iteraciones necesarias. Por ejemplo, si en un momento t el incremento de un peso w es positivo y en $t + 1$ también lo es, entonces la reducción de error se produce de forma más rápida siendo necesarias menos iteraciones. En el caso en el que el incremento del peso en un momento t ha sido positivo y en el momento $t + 1$ es negativo, implica que se ha pasado por un mínimo local de error y, por tanto, se necesitarán menos iteraciones para alcanzarlo y/o mejorarlo.

La red neuronal backpropagation presenta una estructura multicapa, aunque puede funcionar con una única capa oculta. Todas las conexiones entre las neuronas son de tipo feedforward

y las funciones de activación más habituales son continuas, generalmente lineal o sigmoideal.

Cada capa cuenta con un número determinado de neuronas. La primera capa o capa de entrada recoge la información de los inputs. Las neuronas de la última de la capa recogen los outputs obtenidos por la red durante el proceso de aprendizaje. En las capas intermedias se produce el proceso de aprendizaje. El proceso de aprendizaje se produce mediante el cálculo de los errores cometidos por cada una de las neuronas. De esta forma, cuando la información ha pasado por toda la red y ha llegado a las neuronas de la capa de salida, el valor de la delta será:

$$\delta_{pk}^o = (d_{pk} - y_{pk}) f_k^o \left(net_{pk}^o \right) \quad (2)$$

donde $f_k^o \left(net_{pk}^o \right)$ es la función de activación de las neuronas de la capa de salida que, generalmente será lineal o sigmoideal ya que requiere que sea derivable. A partir de aquí se calculan los errores de las neuronas de las capas anteriores. Para el caso de una única capa oculta los errores de las neuronas dependerán de los errores de la capa de salida.

Una vez se han calculado todos los errores de todas las neuronas de toda la red se procede a la modificación de los pesos de cada conexión, comenzando desde la capa de inicio hasta las capas de salida, para reducir el valor de la delta, y así la reducción del error de la red.

De esta forma la red neuronal backpropagation tiene la capacidad de generalización a partir del aprendizaje adaptativo. Es capaz de obtener buenos resultados, incluso con muestras reducidas.

2.2 Hybrid fuzzy neural network

Los modelos hybrid fuzzy neural network se basan en las reglas de clasificación fuzzy. Siendo A un conjunto difuso (fuzzy set) es caracterizado por una función $f_A(x)$ característica que asocia cada punto en un plano real X con un número entre [0,1]. El valor de $f_A(x)$ en x representa el grado de pertenencia de x en A (Zadeh, 1965). Un set C puede componerse como un conjunto de sets A_1, \dots, A_n mediante las conexiones \cup y \cap y puede expresarse como una red de combinaciones $\alpha_1, \dots, \alpha_n$, con $A_i \cap A_j$ y $A_i \cup A_j$ siendo, respectivamente, series y combinaciones paralelas de α_i y α_j . En Zadeh (1965) puede verse todas las operaciones algebraicas con fuzzy sets. The fuzzy concepts son un método muy eficaz para enfrentarse a problemas bajo incertidumbre y problemas no lineales. Los sistemas fuzzy se utilizan frecuentemente para tareas de clasificación, identificación y regresión. Los sistemas fuzzy se basan en el conocimiento de los expertos que se resumen en un conjunto de reglas fuzzy IF-THEN. Inicialmente se basaron en el conocimiento de los expertos en ingeniería y posteriormente en otras disciplinas. No obstante, cuando el problema es suficientemente complejo o no se tiene disponibilidad del conocimiento de los expertos puede generarse las fuzzy rule systems a partir de los datos disponibles utilizando métodos de aprendizaje. En Liu (2016) puede verse de forma exhaustiva la evolución de la teoría de la incertidumbre y los intuitionistic fuzzy sets.

Inicialmente las Fuzzy neural network surgieron para tareas de control. Nauck y Kruse (s.f.) proponen una estructura de red neuronal diseñada para este tipo de tareas. En este caso utilizan fuzzy sets como pesos de las diferentes neuronas. Para conseguir los valores de las fuzzy sets utiliza un algoritmo de aprendizaje basado en el proceso de aprendizaje de una red neuronal backpropagation. Al aplicar el aprendizaje de reglas IF-THEN, mejora la estructura de la red eliminando nodos de las capas ocultas que no contribuyen en la mejora de la resolución del problema.

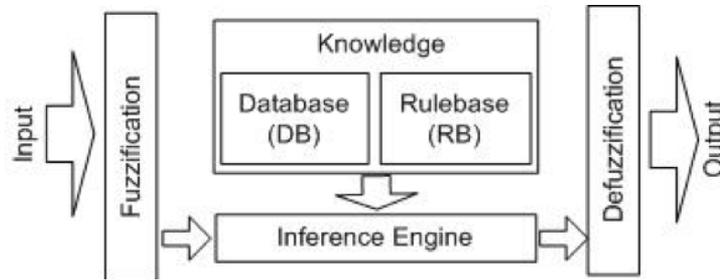


Figura 1. Fases del modelo de Mamdani y Assilian

Existen, por tanto, diferentes propuestas para realizar el aprendizaje de las fuzzy rules, entre otros, el gradient descent learning method (Ichihashi, 1990), space partition based methods (Wang y Mendel, 1992), heuristics procedures (Ishibuchi, Nozaki, Tanaka, Hosaka, y Matsuda, 1994), neural-fuzzy techniques (Jang, 1993; Kim y Kasabov, 1999), clustering methods (Chiu, s.f.; Kasabov y Song, 2002), genetic algorithms (Cordon, Herrera, Hoffmann, y Magdalena, 2001).

Aparte de la elección de un método u otro para el proceso de aprendizaje de las reglas fuzzy, debe tenerse en cuenta la estructura de la misma. Existen dos modelos básicos de estructura de las reglas fuzzy. El primero, modelo de Mamdani y Assilian (1993) construye las reglas teniendo en cuenta cuatro fases: fuzzificación, conocimiento base, motor de inferencia y defuzzificación (Figura ??). La fuzzificación transforma los inputs en variables lingüísticas. El conocimiento base incluye, por un lado, un conjunto de definiciones fuzzy, y por otro, el conjunto de reglas fuzzy del tipo IF-THEN. El motor de inferencia realiza el proceso de aprendizaje entre las reglas fuzzy y los inputs. La defuzzificación decodifica las variables lingüísticas en una variable de salida. La ventaja del modelo de Mamdani frente a otros sistemas fuzzy es su flexibilidad para construir el conocimiento. El segundo modelo, el modelo TSK (Sugeno y Kang, 1988; Takagi y Sugeno, 1985) estructura las reglas fuzzy donde la parte consecuente de la regla, se define mediante una función que es combinación lineal de los inputs. De esta forma, X_i e Y son las variables inputs y output respectivamente. Generalmente, suele utilizarse una función polinómica. Dado que en este caso se tiene una función sobre la parte consecuente, el output obtenido será un valor real, y por tanto, no es necesario la fase de defuzzificación en el modelo TSK. La ventaja de este modelo reside en que se forma un sistema de ecuaciones en la parte consecuente de las reglas fuzzy que son fáciles de estimar utilizando métodos de optimización.

Puede resumirse los modelos de Mamdani y TSK como:

$$R_k : \text{IF } x_1 \text{ is } A_{k,1} \text{ AND } \dots \text{ AND } x_{n1} \text{ is } A_{k,n1} \text{ THEN } y \text{ is } B_k \quad (3)$$

$$R_k : \text{IF } x_1 \text{ is } A_{k,1} \text{ AND } \dots \text{ AND } x_{n1} \text{ is } A_{k,n1} \text{ THEN } y_k = a_k x + b_k \text{ is } B_k \quad (4)$$

$$y = \sum_{k=1}^n w_k(x) y_k \quad (5)$$

Donde x , y son input vector $x = [x_1, \dots, x_{n1}]$ y output vector respectivamente, $A_{k,i}$, B_k son las etiquetas que conforman las variables lingüísticas con los conjuntos fuzzy, siendo $n1$ el número de inputs y n el número de reglas.

En la ecuación 3 puede verse el modelo fuzzy de Mamdani que se centra clasificación. En las ecuaciones 4 y 5 se presenta el modelo fuzzy Takagi-Sugeno-Kang (TSK) que se centra en

predicción.

Los llamados sistemas neuro-fuzzy o redes neuronales fuzzy combinan el uso de redes neuronales y sistemas fuzzy (Buckley y Hayashi, 1994). El sistema fuzzy se inserta en la estructura de la red neuronal y el algoritmo de aprendizaje de ésta trata de fijar los parámetros de la función de pertenencia. Existen varios modelos basados en los sistemas combinados neuro-fuzzy, entre los que destacan dos: Adaptive network fuzzy inference systems y Hybrid neural fuzzy inference system.

El primero, Adaptive network fuzzy inference systems o ANFIS, propuesto por Jang (1993), utiliza un modelo TSK model para las reglas fuzzy que se construye mediante una red neuronal con cinco capas. En la primera capa se produce la fase de fuzzificación donde se transforman los valores en valores lingüísticos usando una función gaussiana como la función de pertenencia. En la segunda capa, se produce el proceso de inferencia donde se determina las reglas fuzzy mediante el operador triangular, es decir, el operador intersección AND. En este caso en el output de cada nodo se determina la fortaleza de la regla.

En la tercera capa cada nodo procede a calcular la ratio de las fortalezas de las reglas fuzzy.

En la cuarta capa se calcula los parámetros para la parte consecuente, es decir, cada nodo calcula los parámetros del conjunto de reglas obtenidas en la tercera capa.

Por último, en la última capa se calcula el output final a partir de una combinación de todos los inputs.

El algoritmo de aprendizaje en el modelo ANFIS tiene dos pasos, forward and backward. El primero sigue por orden por las capas anteriormente indicadas. El Segundo paso, el backward estima la base de datos que consiste en los parámetros de la parte antecedente (IF) y los coeficientes de las ecuaciones lineales de la parte consecuente (THEN). En este caso se optimiza los parámetros de aprendizaje: media y varianza utilizando los mínimos cuadrados. Para la fase de predicción se utiliza del mismo modo visto en el modelo TSK.

El segundo modelo, Hybrid neural fuzzy inference system o HYFIS, fue propuesto por Kim y Kasabov (1999). El modelo HYFIS es una red neuronal fuzzy multicapa. En este caso, utiliza el modelo de Mamdani como estructura de las reglas fuzzy. Al igual que para el caso del modelo ANFIS, éste también tiene cinco capas.

Los nodos de la capa 1 son los nodos que reciben los datos (inputs) que representan las variables lingüísticas. Cada nodo conecta con nodos de la capa 2 que representan los valores lingüísticos correspondientes con variables lingüísticas.

En la capa 2, los nodos actúan como funciones de pertenencia para representar los términos de cada variable lingüística. En esta capa se determina el grado de pertenencia de cada input. Para ello se utiliza una función gaussiana con dos parámetros media o centro c y varianza σ . Inicialmente los pesos de estos nodos comienzan con un valor de 1 y las funciones de pertenencia se encuentran separadas a la misma distancia que el valor de los pesos de los nodos, aunque si existe conocimiento de los expertos se puede utilizar éste en vez del valor unidad. El output de estos nodos es el grado de pertenencia de los inputs de la función de pertenencia utilizando diferentes formas de funciones para las diferentes etiquetas lingüísticas. Los parámetros obtenidos en esta capa se consideran parámetros previos o iniciales a aplicar en las reglas fuzzy (Kim y Kasabov, 1999).

En la capa 3 los nodos representan la parte antecedente de la regla fuzzy (IF). Los pesos de las conexiones de las neuronas se establecen en la unidad. Además, los nodos también añaden el operador AND, en su caso. De esta forma, todos los nodos conforman la base de reglas fuzzy del sistema.

En la capa 4 los nodos representan la parte consecuyente de la fregla fuzzy (THEN). Además, se establece adicionalmente el operador OR para agrupar aquellas reglas que tengan el mismo output de las variables lingüísticas. La activación de los nodos en esta capa se interpreta como el grado que la función de pertenencia es soportada por todas las reglas fuzzy conjuntamente. Las ponderaciones de las conexiones entre los nodos de las capas 3 y 4 son seleccionados de forma aleatoria entre $[-1,+1]$. Cada una de las reglas se activan con cierto grado que se presenta como el cuadrado de los pesos.

En la capa 5 se obtiene el output del modelo. En este caso, los nodos se encargan de defuzificar los datos, es decir, obtienen un valor crisp. Para ello, se utiliza el método del centro de gravedad. Los pesos de las conexiones entre los nodos de las capas 4 y 5 son unitarios.

El proceso de aprendizaje del modelo HYFIS se produce en dos fases. La primera, denominada módulo de adquisición del conocimiento que divide los inputs y outputs en diferentes regiones fuzzy siguiendo la metodología de [Wang y Mendel \(1992\)](#). También se encarga en esta fase de la búsqueda de las reglas fuzzy, en la que se aplica las técnicas fuzzy para determinar las reglas del sistema [Kim y Kasabov \(1999\)](#). Por tanto, se trata de generar un conjunto de reglas fuzzy de los pares de inputs-outputs deseados. A continuación, se utilizan estas reglas para encontrar la estructura adecuada de la parte neuro-fuzzy del sistema HYFIS. En primer lugar, se dividen los inputs y outputs en diferentes regiones fuzzy. Después, se determinan las funciones de pertenencia para cada par input-output y quedan fijados los intervalos de las variables lingüísticas para cada uno de ellos. El siguiente paso es la generación de las reglas fuzzy a partir de los pares de datos, determinando los grados de pertenencia en cada región. A continuación, se le asigna el grado de pertenencia de cada regla, eliminando aquellas redundantes y eligiendo aquellas con mayor grado en caso de conflicto con otras reglas. Con todo esto, quedan fijadas las reglas fuzzy definitivas para el sistema.

En la segunda fase, se determina la estructura neuronal y los parámetros de aprendizaje, una vez fijadas las reglas fuzzy. Se establece la estructura de la red donde comienza el proceso de aprendizaje para ajustar y optimizar los parámetros de las funciones de pertenencia. Para este proceso de aprendizaje se utiliza un algoritmo basado en el gradiente que minimice el error.

El algoritmo de aprendizaje utiliza el gradiente descendente teniendo en cuenta que los pesos de la capa 4 que pueden ser variables y deben optimizarse los parámetros c y σ de la función de pertenencia. Pude verse para más detalle ([Kim y Kasabov, 1999](#); [Wang y Mendel, 1992](#)).

3 Caso de estudio: El índice bursátil IBEX-35

Para analizar la eficiencia de la hybrid fuzzy neural network se ha comparado, la capacidad predictiva de ésta frente a la extendida red neuronal backpropagation. Se trata, en este caso, de construir un modelo de predicción del precio de cierre del índice bursátil español IBEX-35. La muestra contiene las cotizaciones de apertura, máximo, mínimo y cierre desde 09/01/1991 hasta 20/04/2020. Se han escogido como inputs un retardo del índice IBEX-35, índice DAX, índice DJIA, índice FTSE, índice NASDAQ, índice S&P, tipo de cambio EURUSD y futuros del Brent.

Se han construido dieciocho redes neuronales de tipo backpropagation con diferentes configuraciones en sus parámetros. Ha de tenerse en cuenta que un alto número de neuronas puede provocar overfitting en el entrenamiento, aunque dependerá de la complejidad del problema. Por otro lado, los parámetros momentum y beta son también configurables y puede provocar también overfitting, mayor o menor rapidez en el entrenamiento, Dada la complejidad

vv.small	v.small	small	medium	large	v.large	vv.large
----------	---------	-------	--------	-------	---------	----------

Tabla 1. Los nombres de los términos lingüísticos de los inputs

1	medium and X50 is small and X51 is small and X52 is small and X53 is small and X54 is medium THEN X55 is medium
2	medium and X50 is small and X51 is medium and X52 is small and X53 is medium and X54 is large THEN X55 is large
3	medium and X50 is small and X51 is small and X52 is small and X53 is medium and X54 is large THEN X55 is large

Figura 2. Fuzzy IF-THEN rules

de posibilidades de configuración de la red neuronal se ha optado por construir varias redes cubriendo diferentes regiones de parámetros. Así pues, el parámetro alfa ha tenido los valores (0.1,0.5,1), el momentum (0.1,0.5,1) y se han realizado modelos con diferentes iteraciones (100, 1000).

Con estas configuraciones se pretende estimar qué combinación entre en el algoritmo de aprendizaje y el momentum o beta, para diferentes iteraciones, es la que mejor se adapta a los datos muestrales y mejor capacidad de predicción tienen. En todos los casos los pesos iniciales de todos los modelos se han generado de forma aleatoria.

Para el caso del modelo Hyfis se han construido también dieciocho modelos con diferentes configuraciones en sus parámetros. Por una lado, se determina diferentes números de variables lingüísticas (1,10). Por otro lado, se analiza el cómo de grandes son los saltos en cada iteración (0.01,0.5,1). Por último, se configuran diferentes números de iteraciones para estudiar si un mayor número mejora o no los resultados (1,100,1000). En todos los casos se ha utilizado la estructura del modelo Mamdani para establecer las reglas fuzzy.

En la tabla 1 puede verse un ejemplo de los nombres de los términos lingüísticos de los inputs utilizados en los modelos HYFIS. En este caso se han utilizado siete variables lingüísticas para definir las reglas de conocimiento entre las variables inputs y outputs. Inicialmente en el modelo HYFIS los conjuntos fuzzy definidos por las variables (inputs y outputs) tienen asignadas como variables lingüísticas las palabras “large”, “medium” y “small”. No obstante, si el problema es complejo puede necesitarse mayor precisión y estas variables lingüísticas se subdividen en “large positive”, “small positive”, “zero”, “large negative” y “small negative” (Kim y Kasabov, 1999). En nuestro caso, el software R donde se ha computado el modelo describe la variables lingüísticas como “small”, “medium” y “large” con subdivisiones “v.small”, “vv.small”, “v.large”, etc. Es decir, se utiliza otra nomenclatura, pero basado en el mismo principio de subdividir las regiones fuzzy.

Cada modelo Hyfi establece, en función de sus parámetros de configuración sus fuzzy IF-THEN rules que relaciona las variables lingüísticas con los diferentes inputs del modelo (figura 2). Puede observarse como en la parte antecedente (IF) se realizan de forma aditiva diferentes condiciones utilizando el operador de conjunción, es decir, AND. Mientras en la parte consecuente solo presenta un único elemento. Con ello se consiguen una estructura de ecuaciones lineales.

Para medir la eficiencia de la red neuronal fuzzy frente a la red neuronal backpropagation se ha realizado la predicción del precio de cierre del índice bursátil IBEX-35 y se han medido los errores de predicción de todos los modelos mediante el error medio (ME), error cuadrático

	ME	RMSE	MAE
Model 1	0,07811	0,07816	0,07811
Model 2	0,08132	0,08137	0,08132
Model 3	0,06507	0,06513	0,06507
Model 4	0,05543	0,05550	0,05543
Model 5	0,01534	0,01557	0,01534
Model 6	0,02284	0,02301	0,02284
Model 7	0,29011	0,29012	0,29011
Model 8	0,32825	0,32826	0,32825
Model 9	0,38219	0,38220	0,38219
Model 10	0,48045	0,48046	0,48045
Model 11	0,22602	0,22604	0,22602
Model 12	0,02292	0,02309	0,02292
Model 13	0,17832	0,17834	0,17832
Model 14	0,01679	0,01704	0,01679
Model 15	0,42000	0,42001	0,42000
Model 16	0,54894	0,54895	0,54894
Model 17	0,02196	0,02213	0,02196
Model 18	0,15493	0,15496	0,15493

Tabla 2. Error de predicción red neuronal Backpropagation

medio (RSME), error absoluto medio (MAE), error porcentual medio (MPE) y error porcentual absoluto medio (MAPE).

En la tabla 2 se presentan los errores de predicción de las redes neuronales estimadas para las dieciocho configuraciones. En primer lugar, destacan claramente los seis primeros modelos frente al resto con errores significativamente inferiores. Estos modelos tienen todos en común que el parámetro beta o momentum de la red neuronal es de 0.1. Esto nos lleva a pensar que la red neuronal backpropagation es conveniente realizar su entrenamiento con betas muy bajas. Los modelos 3 y 4 son los que presentan un menor error de predicción para las diferentes medidas del error seleccionadas. Ambos modelos, presentan el mismo factor de aprendizaje 0.5 y beta de 0.1, diferenciándose únicamente por el número superior de iteraciones del modelo 4 frente al modelo 3. Concretamente tiene 10 veces más de iteraciones. No obstante, este incremento no mejora sustancialmente los resultados siendo bastante similares, a diferencia del tiempo de computación que es superior.

En la tabla 3 se presentan las medidas de los errores de predicción para los modelos de redes neuronales híbridas. Al igual que ocurre para el caso de la red neuronal backpropagation, los primeros seis modelos presentan una clara reducción del error respecto del resto de modelos estimados. En este caso, estos modelos tienen en común que el parámetro step size es de 0.01, que es el mínimo posible para el gradiente descendente el cual es un número real entre 0 y 1. De entre estos seis modelos, los modelos 3 y 4 son los que presentan menor error, no habiendo cambios significativos entre ellos teniendo como diferencia el número de términos lingüísticos. Esto implica que la red neuronal híbrida puede ofrecer buenos resultados con pocos términos lingüísticos, en este caso con uno solo (modelo 4). Por otro lado, el modelo 3 presenta unos errores casi idénticos teniendo 10 veces menos iteraciones que el modelo 4, y

	ME	RMSE	MAE
Model 1	0,00306	0,00410	0,00314
Model 2	0,00306	0,00410	0,00314
Model 3	0,00215	0,00348	0,00252
Model 4	0,00215	0,00348	0,00252
Model 5	0,00275	0,00388	0,00290
Model 6	0,00275	0,00388	0,00290
Model 7	0,00275	0,00388	0,00290
Model 8	0,00275	0,00388	0,00290
Model 9	0,00627	0,00687	0,00627
Model 10	0,00627	0,00687	0,00627
Model 11	0,00627	0,00687	0,00627
Model 12	0,00627	0,00687	0,00627

Tabla 3. Error de predicción red neuronal Backpropagation

por tanto, preferible, dado que el tiempo de computación es muy superior en el modelo 5.

En la figura 3 puede verse gráficamente una comparativa entre las redes neuronales backpropagation y la red neuronal híbrida para las diferentes configuraciones. Como ya se ha indicado se aprecia que para las primeras 6 configuraciones en ambos casos presentan errores más reducidos que para el resto. En este caso, se destaca cómo la HYFI obtiene resultados más estables aún cambiando algunas configuraciones que en el caso de la red neuronal backpropagation que se aprecia mayor variabilidad.

Los resultados obtenidos son consistentes frente a otros estudios, entre otros, Bekiros (2011) que analiza y compara diferentes estrategias y modelos como una red neuronal backpropagation y la red neuronal fuzzy híbrida para la predicción de la dirección del Mercado para el índice FTSE-100. En Nair, Dharini, y Mohandas (2010) analizan la predicción de la tendencia para los índices FTSE-100, NASDAQ y NIKKEI mediante un modelo ANFIS y un modelo fuzzy híbrido siendo este último, de nuevo más eficiente. Esfahanipour y Mardani (2011) analizan el precio de cierre del índice bursátil de Teherán a partir de datos retardados del mismo, en este caso de un modelo híbrido utilizando la estructura TSK en vez de la de Mamadani comparando con un modelo de perceptrón multicapa. Llegan a la conclusión que el modelo híbrido es más eficiente en tanto en cuanto el número de inputs es mayor.

4 Conclusiones

En este trabajo se ha analizado la eficiencia del modelo híbrido frente a la red neuronal backpropagation en la predicción del índice bursátil IBEX-35. Al igual que en otros estudios realizados para la predicción de la tendencia de otros índices bursátiles, se evidencia la ventaja del modelo neuronal híbrido fuzzy frente, en este caso, a la red neuronal backpropagation. Se han analizado un conjunto de configuraciones que nos muestran diferentes rangos dentro del espectro de posibilidades de configuración para poder determinar aquellas más favorables. Se ha observado que los mejores resultados obtenidos para la red neuronal backpropagation se presentan con un factor de aprendizaje equilibrado y con una beta pequeña, no siendo necesario un alto número de iteraciones. Por otro lado, la red neuronal híbrida ha obtenido sus mejores resultados con un único término lingüístico y con un salto mínimo de 0.1. Para ambos

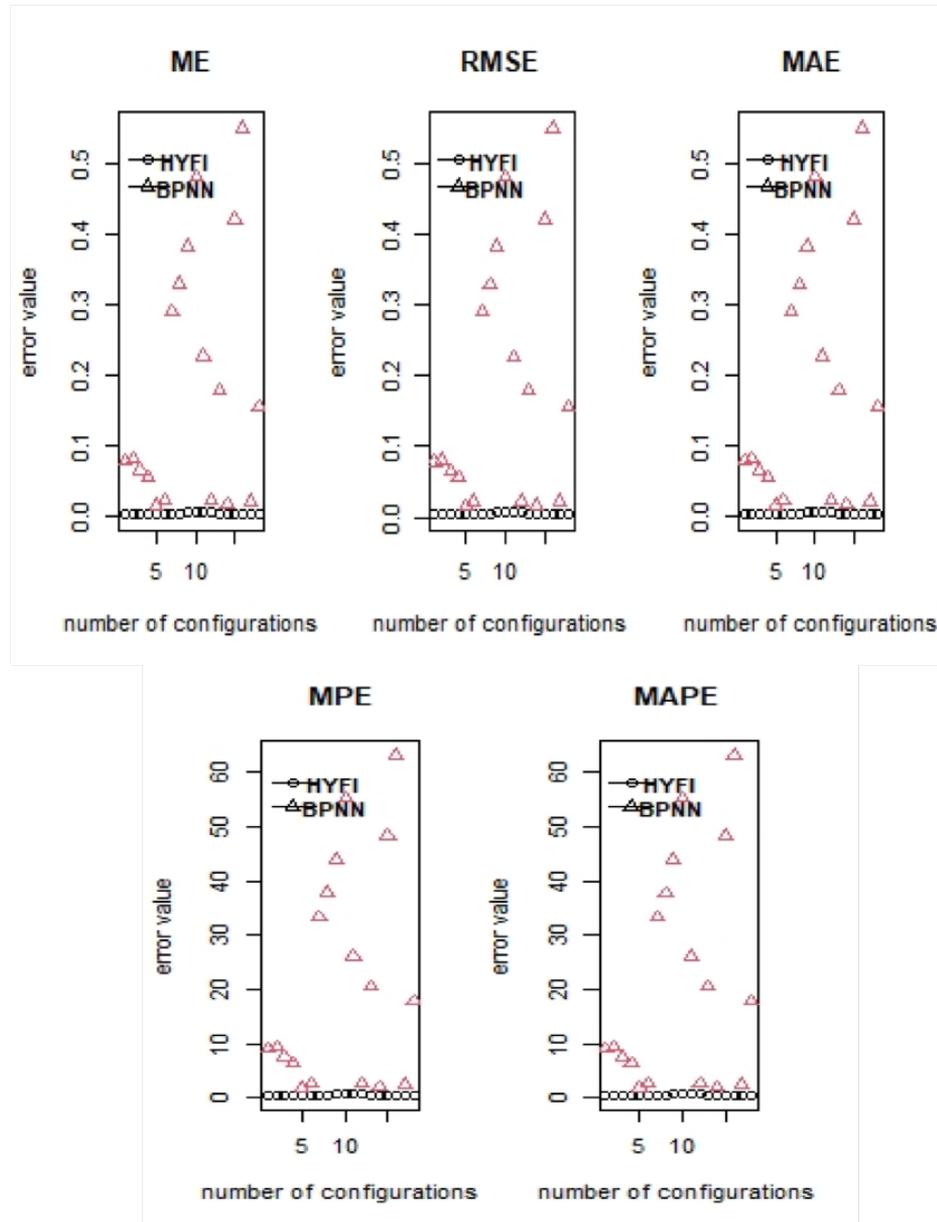


Figura 3. Fuzzy IF-THEN rules

modelos se constata que un incremento sustancial de iteraciones no mejora sensiblemente los resultados. Así pues, un número sobre 100 iteraciones parece ser también suficientes en este caso.

A pesar de las ventajas del modelo HYFIS planteadas en este trabajo, es necesario confirmar y comparar con otro tipo de modelos como las Support Vector Machines (SVM) que han dado buenos resultados en la resolución de muchos problemas frente a las redes neuronales. Además, debe analizarse en otros índices bursátiles y otros activos financieros.

Otro de los aspectos a analizar es si existe grandes diferencias empíricas en cuanto a los resultados para los problemas de predicción entre los modelos Mamdani y TSK para la estructura de las reglas fuzzy. El modelo TSK no necesita la fase de defuzzificación como el modelo de Mamdani. A pesar que el modelo Mamdani se orienta a problemas de clasificación y el modelo TSK a problemas de predicción, existen múltiples ejemplos en los que se han utilizado de forma indistinta por lo que es necesario un estudio comparativo de ambas estructuras para los dos tipos de problemas para el que, en principio, se construyeron.

Referencias

- Adnan, M. R. H. M., Sarkheyli, A., Zain, A. M., y Haron, H. (2013). Fuzzy logic for modeling machining process: a review. *Artificial Intelligence Review*, 43(3), 345–379. doi: <https://doi.org/10.1007/s10462-012-9381-8>
- Ahmad, N., Ahmed, R. R., Vveinhardt, J., y Streimikiene, D. (2016). Empirical analysis of stock returns and volatility: evidence from asian stock markets. *Technological and Economic Development of Economy*, 22(6), 808–829. doi: <https://doi.org/10.3846/20294913.2016.1213204>
- Bekiros, S. D. (2011). Sign prediction and volatility dynamics with hybrid neuro-fuzzy approaches. *IEEE Transactions on Neural Networks*, 22(12), 2353–2362. doi: <https://doi.org/10.1109/tnn.2011.2169497>
- Buckley, J. J., y Hayashi, Y. (1994). Fuzzy neural networks: A survey. *Fuzzy Sets and Systems*, 66(1), 1–13. doi: [https://doi.org/10.1016/0165-0114\(94\)90297-6](https://doi.org/10.1016/0165-0114(94)90297-6)
- Chiu, S. (s.f.). Method and software for extracting fuzzy classification rules by subtractive clustering. En *Proceedings of north american fuzzy information processing*. IEEE. doi: <https://doi.org/10.1109/nafigs.1996.534778>
- Cordon, O., Herrera, F., Hoffmann, F., y Magdalena, L. (2001). *Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases* (Vol. 19). World Scientific.
- Dixit, G., Roy, D., y Uppal, N. (2013). Predicting india volatility index: an application of artificial neural network. *International Journal of Computer Applications*, 70(4).
- Esfahanipour, A., y Mardani, P. (2011). An ANFIS model for stock price prediction: The case of tehran stock exchange. En *2011 international symposium on innovations in intelligent systems and applications*. IEEE. doi: <https://doi.org/10.1109/inista.2011.5946124>
- García, F., González-Bueno, J. A., y Oliver, J. (2015). Mean-variance investment strategy applied in emerging financial markets: Evidence from the colombian stock market. *Intellectual Economics*, 9(1), 22–29. doi: <https://doi.org/10.1016/j.intele.2015.09.003>
- García, F., Guijarro, F., Oliver, J., y Tamošiūnienė, R. (2018). Hybrid fuzzy neural network to predict price direction in the german dax-30 index. *Technological and Economic Development of Economy*, 24(6), 2161–2178. doi: <https://doi.org/10.3846/tede.2018.6394>
- Hamid, S. A., y Iqbal, Z. (2004). Using neural networks for forecasting volatility of s&p 500 index futures prices. *Journal of Business Research*, 57(10), 1116–1125. doi: [https://doi.org/10.1016/s0148-2963\(03\)00043-2](https://doi.org/10.1016/s0148-2963(03)00043-2)

- Ichihashi, H. (1990). Learning control system by a simplified fuzzy reasoning model. *Proceedings of IPMU'90*, 417–419.
- Ishibuchi, H., Nozaki, K., Tanaka, H., Hosaka, Y., y Matsuda, M. (1994). Empirical study on learning in fuzzy systems by rice taste analysis. *Fuzzy Sets and Systems*, 64(2), 129–144. doi: [https://doi.org/10.1016/0165-0114\(94\)90329-8](https://doi.org/10.1016/0165-0114(94)90329-8)
- Jang, J.-S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665–685. doi: <https://doi.org/10.1109/21.256541>
- Kasabov, N., y Song, Q. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2), 144–154. doi: <https://doi.org/10.1109/91.995117>
- Kim, J., y Kasabov, N. (1999). HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. *Neural Networks*, 12(9), 1301–1319. doi: [https://doi.org/10.1016/s0893-6080\(99\)00067-2](https://doi.org/10.1016/s0893-6080(99)00067-2)
- Kristjanpoller, W., y Minutolo, M. C. (2015). Gold price volatility: A forecasting approach using the artificial neural network–GARCH model. *Expert Systems with Applications*, 42(20), 7245–7251. doi: <https://doi.org/10.1016/j.eswa.2015.04.058>
- Kuo, R., y Xue, K. (1999). Fuzzy neural networks with application to sales forecasting. *Fuzzy Sets and Systems*, 108(2), 123–143. doi: [https://doi.org/10.1016/s0165-0114\(97\)00326-6](https://doi.org/10.1016/s0165-0114(97)00326-6)
- Liu, P. (2016). Special issue “intuitionistic fuzzy theory and its application in economy, technology and management”. *Technological and Economic Development of Economy*, 22(3), 327–335. doi: <https://doi.org/10.3846/20294913.2016.1185047>
- Mamdani, E. H., y Assilian, S. (1993). An experiment in linguistic synthesis with a fuzzy logic controller. En *Readings in fuzzy sets for intelligent systems* (pp. 283–289). Elsevier.
- Mirbagheri, M., y Tagiev, N. (2011). Analyzing economic structure and comparing the results of the predicted economic growth based on solow, fuzzy-logic and neural-fuzzy models. *Technological and Economic Development of Economy*, 17(1), 101–115. doi: <https://doi.org/10.3846/13928619.2011.554201>
- Nair, B. B., Dharini, N. M., y Mohandas, V. (2010). A stock market trend prediction system using a hybrid decision tree-neuro-fuzzy system. En *2010 international conference on advances in recent technologies in communication and computing*. IEEE. doi: <https://doi.org/10.1109/artcom.2010.75>
- Nauck, D., y Kruse, R. (s.f.). A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation. En *IEEE international conference on neural networks*. IEEE. doi: <https://doi.org/10.1109/icnn.1993.298698>
- Roh, T. H. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications*, 33(4), 916–922. doi: <https://doi.org/10.1016/j.eswa.2006.08.001>
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. doi: <https://doi.org/10.1038/323533a0>
- Sugeno, M., y Kang, G. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1), 15–33. doi: [https://doi.org/10.1016/0165-0114\(88\)90113-3](https://doi.org/10.1016/0165-0114(88)90113-3)
- Takagi, T., y Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1), 116–132. doi: <https://doi.org/10.1109/tsmc.1985.6313399>
- Wang, L.-X., y Mendel, J. (1992). Generating fuzzy rules by learning from exam-

ples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6), 1414–1427. doi:
<https://doi.org/10.1109/21.199466>

Widrow, B., y Hoff, M. E. (1960). *Adaptive switching circuits* (Inf. Téc.). Stanford Univ Ca Stanford Electronics Labs.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. doi:
[https://doi.org/10.1016/s0019-9958\(65\)90241-x](https://doi.org/10.1016/s0019-9958(65)90241-x)